IBM® Information Management Software

Delivering Continuity and Extreme Capacity with the IBM DB2 pureScale Feature

- Describes how to use the IBM DB2 pureScale Feature
- Includes information about tuning the DB2 pureScale environment
- Provides detailed installation and setup instructions

Vlad Barshai
Yvonne Chan
Hua Lu
Satpal Sohal

Redbooks

ibm.com/redbooks

International Technical Support Organization

Delivering Continuity and Extreme Capacity with the IBM DB2 pureScale Feature

September 2012

SG24-8018-00

Note: Before using this information and the product it supports, read the information in "Notices" on page vii.

First Edition (September 2012)

This edition applies to Version 10.1 of IBM DB2 Database software.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

Preface

The IBM® DB2® pureScale® feature offers clustering technology that helps deliver high availability and exceptional scalability transparent to applications. The DB2 pureScale Feature helps you to meet your business needs around availability and scalability, and is also easy to configure and administer.

This IBM Redbooks® publication addresses the DB2 pureScale Feature that is available in IBM DB2 10.1 for Linux, UNIX, and Windows operating systems. It can help you build skills and deploy the DB2 pureScale Feature. This book bundles all the information necessary for a in-depth analysis into the functions of the DB2 pureScale Feature, including the actual hardware requirements. It includes validated step-by-step hardware and software installation instructions. In addition, this book provides detailed examples about how to work effectively with a DB2 pureScale cluster and how to plan and run an upgrade for all DB2 related components to DB2 10.1.

This book is intended for database administrators (DBAs) who use IBM DB2 10.1 for Linux, UNIX, and Windows operating systems who want to explore and get started with the DB2 pureScale Feature.

The team who wrote this book

This book was produced by a team of DB2 enablement specialists working at the IBM Toronto Laboratory in Canada. The authors have participated in several DB2 pureScale implementations and share their experience and preferred practices from the field in this book.

Vlad Barshai has worked with DB2 for Linux, UNIX, and Windows for the past four years. Vlad has extensive experience working with Linux and KIWI, particularly in developing software to automate installations and configurations for creating meaningful virtual appliances. His expertise in DB2, the DB2 pureScale Feature, and knowledge of hardware components (IBM System x®, IBM POWER®, storage subsystems, and networking switches) has allowed him to assist, teach, and set up various DB2 environments and cluster for customers and POCs.

Yvonne Chan is the DB2 pureScale Feature Principal on the Information Management Technology Ecosystem team helping customers implement DB2 solutions with the DB2 pureScale Feature. Before her role in this team, she spent several years on the DB2 pureScale Feature kernel team working on the Recovery Orchestration and Cluster Management parts of the DB2 pureScale Feature. Before that, she led the DB2 Linux team responsible for porting DB2 to several Linux platforms, including x86_64, IBM System z®, and Itanium. Yvonne has been with IBM for over 13 years. She earned a Bachelor of Applied Science degree in Computer Engineering from the University of Toronto.

Hua Lu is a software developer in the Information Management Technology Ecosystem team, where he provides support for IBM Information Management products on various Linux distributions and virtualization platforms. Since 2010, he has worked with IBM DB2 pureScale Feature implementations. Recently, he has been involved in activities related to IBM PureApplication™ System.

Satpal Sohal is a DB2 Solutions Migration Specialist working at the IBM Toronto Lab. In addition to developing and managing critical software infrastructure for the team, he helped create a DB2 pureScale Feature Workshop, which provides IBM clients and IBM Business Partners a course on the DB2 pureScale Feature. Satpal is a certified DB2 database administrator and has delivered many DB2 training courses to audiences worldwide.

Thanks to the following people for their contributions to this project:

Pandu Mutyala
DB2 System Verification, IBM Information Management

Steve Rees
Senior Manager, DB2 Performance QA, IBM Information Management

Antonio Maranhao
Manager, Technology Assets, IBM Information Management

Jason Chan
Manager, Appliances and Platforms Ecosystems, IBM Information Management

Tom Chong
Automation and Integration Lead, Appliances and Platforms Ecosystem, IBM Information Management

Piotr Pruski
Technical Enablement Specialist, Appliances and Platforms Ecosystem, IBM Information Management

Melody Ng
Senior Manager, Information Management Ecosystem Partnerships, IBM Information Management

Cathy Guo
Graphic Designer, Information Management Partnership Technologies, IBM Information Management

Andre Albuquerque
DB2 Specialist, Technology Assets, IBM Information Management

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks publications

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

An overview of the DB2 pureScale Feature

As the amount of data produced daily increases, instant and continuous availability of that data is a critical component to business success. Companies need to access, process, or store transactions even during peak times or maintenance periods. Additionally, companies cannot afford to lose data because an application is seeking data from servers that are unavailable. To meet these high expectations, businesses need an infrastructure that provides high speed, continuous availability, and scalability.

You can use the IBM DB2 pureScale Feature to scale a database (DB) for a set of servers in an *active-active* approach. Traffic intended for a failed node is either passed on to an existing node or load balanced for the remaining nodes. This technology is similar to the proven data-sharing architecture found in DB2 for IBM z/OS®. This feature provides the benefits of transparency, continuous availability, and scalability at a lower operational cost.

The DB2 pureScale system runs on up to 128 multiple hosts that are accessing shared data simultaneously, without needing to explicitly modify the application. You can use this transparency to perform maintenance operations on hosts, add additional hosts, or remove unnecessary hosts, without impacting an application. With this method, you can control the number of active hosts to handle the workload and to ensure that you remain at the wanted transaction rate.

This chapter covers a basic overview of the DB2 pureScale Feature and its functions. This feature includes various dependent components to ensure that the system runs smoothly. Thus, the description begins with an overview of each of these components and continues to the differences between these components. Additionally, this chapter provides an explanation between the various DB2 versions and editions and their respective differences. This explanation can help you in tying this description together to understand how an application interacts with the DB2 pureScale Feature.

This chapter includes the following topics:

- Technology overview
- The DB2 pureScale Feature technical components
- The DB2 pureScale Feature and your application
- The DB2 pureScale Feature versions and editions

1.1 Technology overview

The DB2 pureScale technology provides an active-active configuration that transparently delivers high throughput, continuous availability, and scalability.

Figure 1-1 depicts the main components of a DB2 pureScale cluster.

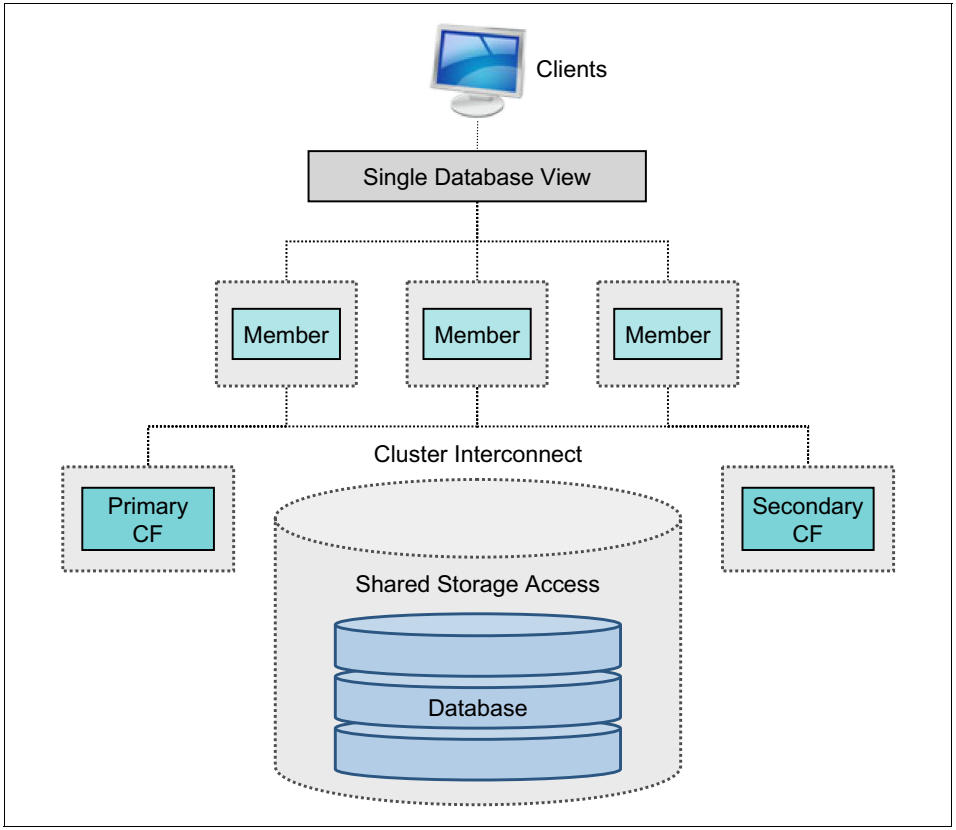

Figure 1-1 DB2 pureScale cluster

As illustrated in Figure 1-1, a DB2 pureScale cluster is composed of the following components:

- A *DB2 member* (also referred to as a *member*) is the core component that handles and processes database transactions (application requests) within the clustered system. DB2 members are the database engine processes that manage and retrieve the data from the database. Adding a DB2 member refers to adding a processing node within the cluster for data sharing.
- The *cluster caching facility* (CF) is responsible for instance-wide lock management and global caching. This component relieves the members from having to handle locking, caching, and communication with each other to maintain vital locking and data consistency information.

- The *shared storage* is a storage unit that is connected to all DB2 members and CFs in the cluster to allow for all the data to be shared and used simultaneously. The DB2 pureScale Feature supports all storage area networks (SAN), which work as a storage switch, and fibre attached shared block storage, referenced as a logical unit number (LUN).
- The *cluster interconnect* refers to the high speed, low latency networking that allows the DB2 members and CFs to communicate among each other and transfer data or logs to process transactions. The cluster interconnect uses either a 10 Gigabit Ethernet (GbE) or InfiniBand connection to communicate. These features are used to take advantage of Remote Direct Memory Access (RDMA), which helps use centralized locking and buffer caching (explained further in 1.2, “The DB2 pureScale Feature technical components” on page 9).
- The *clients* (application) refer to clients and the applications that are running transactions against the DB2 pureScale Feature and associated databases.

Important: These names are referenced throughout the book and are critical to the overall understanding of the DB2 pureScale Feature. Take a moment to ensure that you are familiar with each of them.

Figure 1-1 on page 3 illustrates that the application is not apparent to the overall DB2 pureScale cluster because it requires a single connection to the cluster to run requests. It also illustrates that each physical machine has the following characteristics:

- It exists on a public network that allows for client connectivity.
- It has a 10 GbE or InfiniBand card that allows for high speed, low latency communication between members and CFs. This communication is achieved using Remote Direct memory access, referenced as *uDAPL* on InfiniBand networks, and *RDMA over Converged Ethernet* on 10 Gigabit Ethernet.
- It has shared connectivity to a common set of disks.

The remainder of this section touches on the various advantages of the DB2 pureScale Feature, including transparency, availability, and scalability.

1.1.1 Application transparency

The word *transparency* for the DB2 pureScale Feature is key. Ideally, you expect that any increase in workload or failures in a database cluster do not impact the operation of an application. Thus, from the application standpoint, your only concern is to retrieve user requests and respond back with appropriate information. Any additional layers that are required to make the application more cluster aware complicates the application, might result in a workload change, and can cost additional development time in testing, deployment, verification, and certification.

The DB2 pureScale Feature provides the following advantages without changing an application:

- Scale as you grow: Ability to buy only enough to satisfy current capacity needs and add additional DB2 members to scale as your workload grows.
- Maintain availability: Maximize hardware utilization rates and keep response times consistent because incoming requests are automatically load balanced, that is, spread the workload among the multiple processing servers.
- Add or remove resources easily: Add resources during peak hour or seasonal times to avoid slow response times, or when there is required downtime.

1.1.2 Continuous availability

Availability ensures that the cluster remains operational both during planned or unplanned outages. During maintenance operations, hardware failures, or network failures, the DB2 pureScale instance can still process database requests without interruptions. Moreover, availability also ensures that you receive the expected query response to service requests in a timely manner, no matter the circumstances.

Continuous availability is dependant upon unplanned and planned events, as described in the next sections.

Unplanned events

Hardware failures can be highly disruptive, even in systems with redundant components. By design, the DB2 pureScale Feature contains functions, such as heartbeats, fencing mechanisms, automatic restarts, and role shifting, to keep an instance running and to minimize the effects of failures on the remainder of the database system in such scenarios.

Figure 1-2 illustrates a failure scenario.

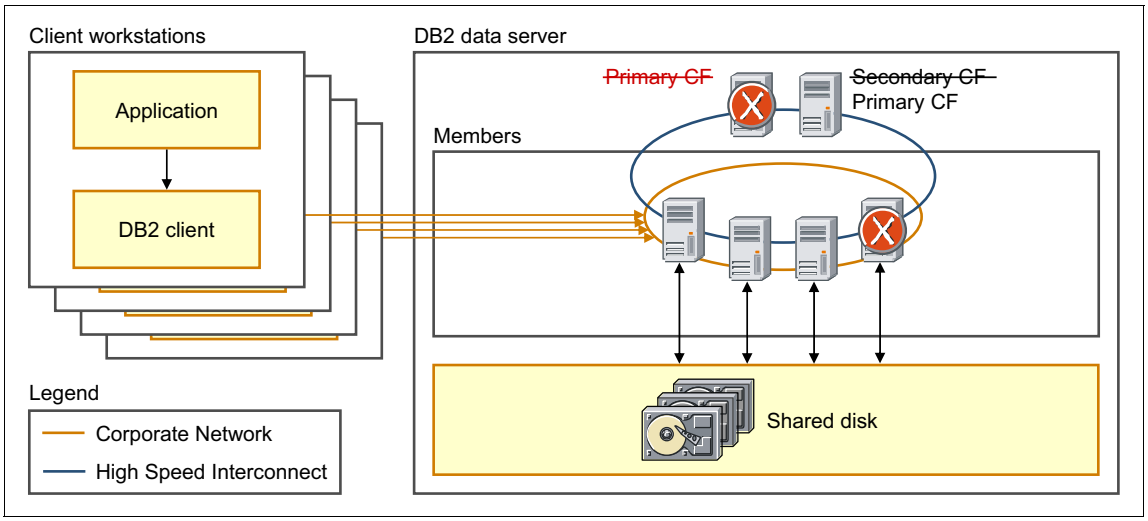

Figure 1-2 Failure scenario for a DB2 member or DB2 cluster CF

The following DB2 pureScale Features provide continuous availability:

- Heartbeats: A *heartbeat* sent between hosts ensures that a host failure is detected and isolated immediately.
- Fencing mechanisms: When a member fails during processing of database requests, its failure needs to be detected immediately and any communication with it must be terminated as soon as possible. This feature allows communication with the member to be *fenced* in addition to its access to disk (if this failure was also a host failure). Meanwhile, data is still accessible to the remaining active members processing database requests.
- Automatic restart of member: During a software failure of a member, the member attempts to recover automatically on the same home host to resume transactions. If the member cannot restart on its home host, a restart occurs on another host so that any locks held by the affected member are released immediately. Any transactions that were not completed at the time of the failure are also rolled back, and the application is notified.
- Automatic restart of cluster CF: On a software or hardware failure of the primary cluster CF, the second cluster CF containing the duplexed information takes over the primary role. Take over is mostly transparent to the applications and the instance remains available.

With the DB2 pureScale Feature, availability characteristics are integrated directly into the architecture. All necessary resources are monitored automatically by DB2 pureScale cluster services. These resources are restarted and recovered as needed. Member failures are not detected by the client applications because the clients are rerouted automatically to the remaining active members. However, one of the distinguishing factors that makes the DB2 pureScale Feature competitive is that there is no cluster-wide freeze that occurs when a member fails. In fact, data continues to be processed, and all members are updated to inform other members that this member is no longer available until recovery is complete. Applications on active members that are trying to access locked data on failing member are briefly in lock-wait state, but do not automatically
receive errors.

In all cases of failure, after a host is available in the network, the DB2 pureScale Feature automates the recovery and attempts to bring the restarted host back into the cluster without intervention. The DB2 pureScale Feature restarts the DB2 member that should be running there.

Planned events

There are several circumstances where you might want to take down a system temporarily, such as with system maintenance or hardware upgrades. The DB2 pureScale Feature is designed to perform these types of maintenance tasks with as little impact to the entire DB2 pureScale cluster as possible.

You can use DB2 to place a member into a *quiesce* state, which allows the connections to drain as transactions (and connections). When in a quiesce state, subsequent transactions and connections are redirected to other available members, and the current member eventually stops processing transactions. The client applications do not even notice that a member was taken offline (also known as *stealth maintenance*).

Because the client applications are now connected to other members, you can take the member offline and perform maintenance, causing as little disruption as possible to the client applications. You can use this method to roll out system upgrades without stopping the DB2 pureScale instance or affecting database availability. After the maintenance is complete, you can reintegrate the member into the DB2 pureScale instance.

1.1.3 Scalability and extreme capacity

You can use the DB2 pureScale Feature to increase the number of nodes up to 128 while still delivering near-linear scalability and maximum throughput. You can use this transparent approach to scalability to add resources as they are needed during peak times. You can adjust for seasonal variations or to meet changing application requirements.

Figure 1-3 illustrates the behavior of this scalability when you add additional members to the cluster. The cluster can process more workload as soon as new members join the cluster operations. Overall throughput almost doubles as the number of members doubles.

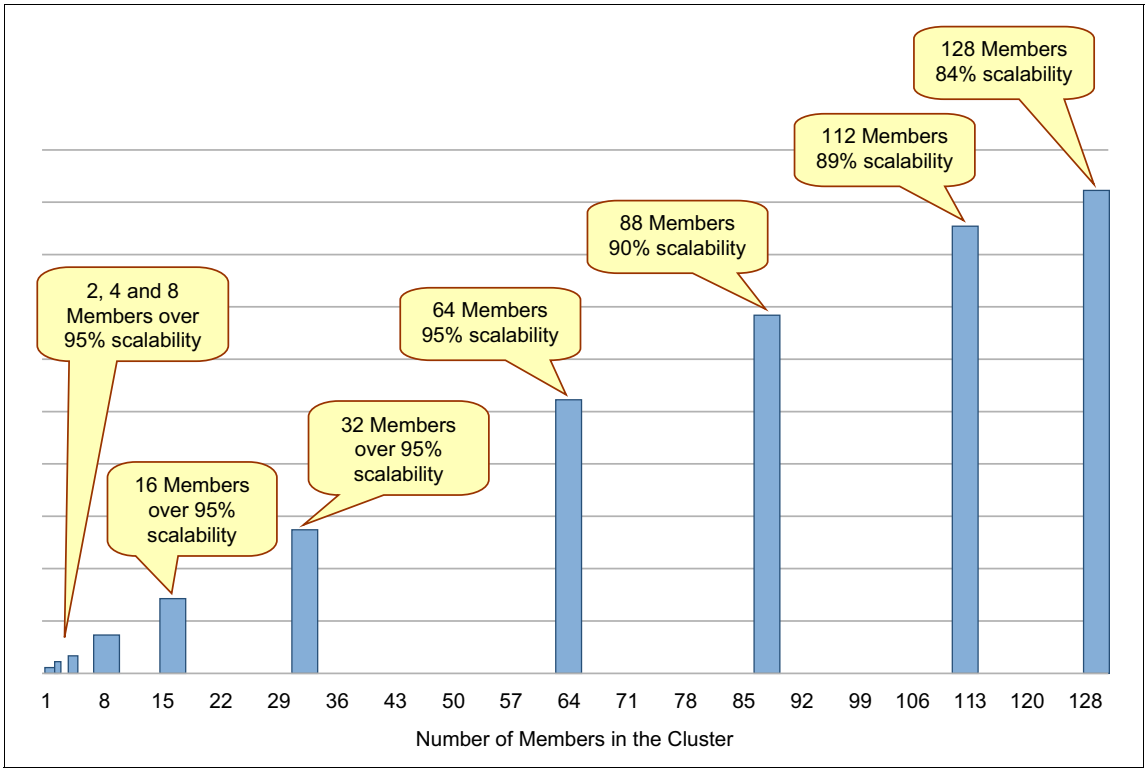

Figure 1-3 Throughput increases with the addition of DB2 members

1.2 The DB2 pureScale Feature technical components

To understand how the DB2 pureScale Feature environment works, you must understand the various components that make up a DB2 pureScale instance and how they work together. This section covers some of the DB2 cluster services and processes that make up the DB2 pureScale Feature architecture and that allow for automatic recovery after a member failure.

1.2.1 DB2 cluster services overview

DB2 cluster services are used to group the integrated cluster services components that are used to monitor and automate the failure recovery in the system. These services have daemons that run on each of the hosts; however, decisions are made in a group manner. You must have enough hosts communicating in these groups to decide what actions must be taken.

The following components make up the DB2 cluster services:

- *IBM Reliable Scalable Cluster Technology* (RSCT) is a set of software components that help provide cluster monitoring of the environment. These processes are responsible for maintaining the *heartbeat* between each of the hosts and provide a layer that is used for secure communication between the hosts. In addition, Reliable Scalable Cluster Technology provides the fencing mechanisms that are called by DB2 when a host fails, and is used to notify DB2 processes when specific conditions occur.
- *IBM Tivoli® System Automation for Multi-platforms* provides fast detection of DB2 outages and monitors for availability within the cluster. Tivoli System Automation for Multi-platforms is the layer DB2 uses to define what actions should be taken after specific failures. By integrating an automated detection and recovery system, the component reduces the manual monitoring of the DB2 pureScale cluster to ensure that everything is online and continues to function as it should. At the sam time, it provides the time you need to focus on your business rather on the database itself.
- *IBM General Parallel File System* (GPFS™) provides a cache coherent shared file system. In short, it provides the management of the file systems that DB2 uses to store your data and logs. With the DB2 pureScale Feature, all data and logs must be stored on the General Parallel File System, so that they can be shared between the members for recovery purposes. Underneath General Parallel File System is the storage subsystems, which can be either SAN-attached storage, or fibre attached storage.
- *Remote Direct Memory Access* (RDMA) is the modern hardware interconnect that provides the ability to use the centralized locking and buffer caching algorithms used by the cluster CF.

RDMA allows each member in the cluster to directly access memory in the CF and for the CF to directly access the memory of each member. This configuration allows for faster access to lock verification and data page retrieval, increasing the performance of queries. This technology is based on one of two different types of networking interfaces:

- InfiniBand, which is still the fastest, lowest latency network that supports RDMA through the uDAPL application programming interface
- 10 GbE, which supports RDMA over Converged Ethernet protocols (also known as *RoCE*)

1.2.2 Member crash recovery services overview

Each member has several processes that are used for the sole purpose of *member crash recovery*. These processes ensure that memory is pre-allocated and reduce the recovery time that is required should a member fail. Normally, these processes are *idle* in that they do not consume any processor resources. After a member fails over to one of these idle processes from another host, the member commences the crash recovery process, by rolling back in-flight data. During this brief period, only in-flight data is locked, and all other data is fully available for clients who are accessing the database. Also, while a failed member is on a host other than its home host, it cannot accept any new connections or transactions. After the member's home host becomes available, the member fails back and regains its status as an active member in the cluster.

By understanding the process of how these services interact with one another to provide automatic recovery, it is possible to get a general overview about how DB2 reacts upon a member failure and how the services are involved. This process can play a significant role in diagnosing a problem or any general issues after a member fails and is now trying to recover back into the cluster.

1.3 The DB2 pureScale Feature and your application

As explained in previous sections, the DB2 pureScale Feature is transparent to an application. Thus, an application does not need to be aware of which DB2 members are handling requests, allowing you to focus on customers and users instead of worrying about how data is handled by the database. This section briefly outlines some preliminary considerations about how an application works with the DB2 pureScale Feature.

DB2 handles access to applications that are developed using the IBM data server client, which includes embedded SQL or calls to the command-line interface (CLI), Open Database Connectivity (ODBC) applications, Java applications using the Java Database Connectivity (JDBC) or SQLJ interfaces, PHP applications, Ruby or Ruby on Rails applications, Perl applications, and Python applications.

For client connections, the DB2 pureScale Feature supports the following capabilities:

- Client affinity: The ability to connect to a particular server to process requests
- Workload balancing: The ability to dynamically distribute a workload for all the active members

1.3.1 Client affinity

Client affinity is a client-only method for providing automatic client rerouting. It is available for applications that use CLI, .NET, or Java. All rerouting for client affinity is controlled by the driver and is intended for situations in which you need to connect to a particular primary server. If an outage occurs during the connection to the primary server, a specific order for failover to alternative servers is used that reroutes that particular client to another processing host.

As part of the configuration for client affinities, you must specify a list of alternative servers and the order in which connections to the alternative servers are tried. When client affinities are in use, connections are established based on the list of alternative servers instead of the host name and port number that are specified by the application. For example, if an application specifies that a connection is made to server1 but the configuration process specifies that servers are tried in a specific order (server2, server3, and then server1), the initial connection is always made to server2 instead of server1.

Failover with client affinities is seamless if the following conditions are true:

- The failure is either of the connection to the server or of the first SQL statement of a connection.
- There are no global temporary tables in use on the server.
- There are no open held cursors.

When using client affinities, you can specify that if the primary server returns to operation after an outage, connections return from an alternative server to the primary server on a transaction boundary.

1.3.2 Workload balancing

Workload balancing is based on the utilization characteristics of the different machines. Workload balancing (also called *transaction-level workload balancing*) for connections to DB2 for Linux, UNIX, and Windows operating systems contributes to high availability by balancing work among servers in a DB2 pureScale instance at the start of a transaction.

When clients connect to a DB2 database and transaction-level workload balancing is enabled, the following process occurs:

1. Upon establishing a connection to the DB2 pureScale instance, the member to which the client connects returns a server list with the connection details (such as IP address, port, and weight) for the members of the DB2 pureScale instance.

   This server list that was retrieved is cached by the client. The default lifespan of the cached server list is 30 seconds.
2. At the start of a new transaction, the client reads the cached server list to identify a server that has unused capacity and looks for an idle transport that is tied to the underutilized server. (An *idle transport* is a transport that has no associated connection object.)
3. When there is an idle transport available, the client associates the connection object with the transport.
4. When the transaction runs, it accesses the server that is tied to the transport.
5. After the transaction ends, the client verifies with the server that transport reuse is still allowed for the connection object.
6. The client copy of the server list is refreshed when a new connection is made, every 30 seconds, or is refreshed based on the user-configured interval.

When transaction-level workload balancing is required for a new transaction, the client uses this process to associate the connection object with a transport.

1.4 The DB2 pureScale Feature versions and editions

The DB2 pureScale Feature is an added feature of DB2. The scope of this book covers only the following DB2 editions that support the DB2 pureScale Feature:

- IBM DB2 Enterprise Server Edition
- IBM DB2 Advanced Enterprise Server Edition
- IBM DB2 Workgroup Server Edition
- IBM Database Enterprise Developer Edition

For an extensive list of all available DB2 editions and their respective features, visit the DB2 10 Information Center at:

http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.licensing.doc/doc/r0053238.html

After describing the DB2 editions, this section also describes the feature differences and covers the differences between DB2 9.8 and the latest release of DB2 10.1.

1.4.1 DB2 versions and differences

The DB2 pureScale Feature was initially available on its own, as a separate release called *DB2 9.8*. Starting in DB2 10.1, this feature is incorporated into the baseline DB2 code and is now a feature of the normal DB2 releases.

> Resource note: This book covers only the two DB2 versions that offer the DB2 pureScale Feature. For all previous versions of DB2, visit the Information Center.

This section introduces both DB2 9.8 and DB2 10.1 and explains some of their key differences. For more information about how to upgrade to DB2 10.1, see Chapter 5, “Upgrading to DB2 10.1” on page 251.

DB2 9.8

This version of DB2 was generally available (GA) in December 2009 and is a database cluster solution for non-mainframe platforms that is suitable for online transaction processing (OLTP) workloads. This release is similar in functionality to DB2 9.7, with the added capabilities of the DB2 pureScale Feature to provide a more fault-tolerant architecture.

DB2 10.1

This version of DB2 was generally available (GA) in April 2012 and merges the two code streams of DB2 9.7 and DB2 9.8 into a single product stream. The DB2 10.1 pureScale Feature provides the same architecture as DB2 9.8, with additional features, enhancements, and support.

Feature enhancements in DB2 10.1 for the DB2 pureScale Feature

The basic architecture of the DB2 pureScale Feature remains the same. Differences between DB2 9.8 and DB2 10.1 are primarily related to performance optimizations and feature expansions to ease the ability to perform certain tasks. Note the following differences between the two versions:

- DB2 10.1 now supports the DB2 **`SET WRITE`** command.

  The **`SET WRITE`** command is typically used for splitting a mirrored database because it allows a user to suspend I/O writes or to resume I/O writes for a database. This mirroring is achieved through a disk storage system.

- DB2 10.1 supports multiple CF interconnects.

  With multiple cluster interconnects on the CFs, you can connect each CF to more than one switch to improve fault tolerance in your DB2 pureScale environment. A one-switch multiple cluster interconnect configuration increases the throughput of request to CFs, whereas a two-switch configuration helps increase throughput and provide a higher availability characteristic. DB2 pureScale environments do not require multiple cluster interconnects or multiple switches; however, the redundancy helps improve availability in the case of a failure.

- DB2 10.1 provides increased control over processor resources when using workload management (WLM).

  By allowing the ability to set processor shares and limits, you are able to control the percentage of processor resources that different workloads can consume.

- DB2 10.1 has adaptive compression.

  Adaptive compression is a new type of compression that further improves your table compression through an advanced row compression technique that uses two levels of compression dictionaries (table-level and page-level) to improve compression ratios, particularly as data changes. Page-level compression dictionaries are smaller than table-level dictionaries. So, as data changes on a page, it is easy to update automatically and quickly, and removes the need to perform table reorganizations to maintain compression ratio over time. In DB2 10.1, all new tables use adaptive compression by default.

  Adaptive compression provides the following benefits:

  - Helps achieve high compression ratios without taking data offline (such as when performing a table reorganization)
  - Continues improvements in query performance

  - Increase in storage savings and system availability
  - Reduces costs through decreased storage requirements

- DB2 10.1 has data security enhancements.

  DB2 10.1 includes critical enhancements to security by introducing row and column access control (RCAC), which is sometimes referred to as *fine-grained access control*. RCAC helps further secure data by allowing you to create varying security rules at the data level. These security rules ensure that users who are members of the approved roles or groups see only the data that they are allowed to see. Security rules also remove security constraints and performance headaches that result from complex views and predicates. These enhancements provide a centralized and auditable process that controls data at a lower cost and with reduced time to value for business process and applications.

- DB2 10.1 provides multi-temperature data management.

  Data is assigned a priority (hot, warm, or cold) and moved to different classes of storage. For example, you can store transaction records for the current quarter, which are accessed more frequently, in a high-performance storage location. After the quarter is finished, you can then move these records to cheaper storage (warm or cold) because they do not need to be accessed as frequently. Providing multi-temperature data management can reduce the total cost of ownership (TCO) and provide for efficient use of storage hardware.

- DB2 10.1 has new performance enhancements.

  Performance enhancement focus on reducing the processing time without causing significant administration or application changes. Upgrading to DB2 10.1 provides the following improvements:

  - Improvements in the **RUNSTATS** command
  - Improved query optimizer techniques
  - Star schema query optimization
  - Improved performance for queries on tables with composite indexes
  - Improved multi-core parallelism
  - Data and index prefetching
  - Improved use of statistical views

- DB2 10.1 includes SQL compatibility improvements.

  When working with a relational database product other than DB2 products, there are additional interfaces and compatibility features that can help make DB2 products more familiar to you. These improvements can reduce the amount of time that it takes to enable applications written for other relational database products to run quickly in a DB2 environment.

- DB2 10.1 includes a new time travel query feature.

  This enhancement allows the database to be *time-aware* by keeping a history of data changes and management of business time. You can use this feature to query past data as it appears at different points in time, which provides the following benefits:

  – Provides a cost-effective means to resolve auditing and compliance issues
  – Lowers application development time by using efficient and standardized time-based SQL syntax
  – Allows for creation of a time-based warehouse at low costs without additional application logic required

- DB2 10.1 includes new features to workload management.

  You can now use DB2 workload manager (DB2 WLM) in a DB2 pureScale environment to manage workloads to maximize system efficiency and throughput to achieve your business performance objectives.

For further information and a more detailed description of the feature enhancements for DB2 10.1, visit the DB2 10.1 Information Center at:

http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/index.jsp

1.4.2 DB2 editions and differences

This section provides information about the following DB2 editions, which are suitable for businesses of various sizes:

- IBM DB2 Workgroup Server Edition
- IBM DB2 Enterprise Server Edition
- IBM DB2 Advanced Enterprise Server Edition
- IBM Database Enterprise Developer Edition

IBM DB2 Workgroup Server Edition

The DB2 Workgroup Server Edition solution is ideal for deployment in a departmental, workgroup, or medium-sized business environments. It is offered in per authorized user, value unit, or limited use socket pricing models to provide an attractive price point for medium-size installations, and provides a full-function data server.

DB2 Workgroup Server Edition includes high availability disaster recovery (HADR), online reorganization, IBM pureXML®, web services federation support, DB2 Homogeneous Federation, Homogeneous SQL Replication, backup compression, and Tivoli System Automation for Multi-platforms.

DB2 Workgroup Server Edition uses up to 16 cores and 64 GB of memory with the ability to be deployed on both Linux and UNIX server environments. Under authorized user licensing, you must acquire a separate user license for each authorized user of this product, with a minimum purchase of five users per installation.

IBM DB2 Enterprise Server Edition

The DB2 Enterprise Server Edition is designed to meet the needs of medium to large sized businesses for high-performing, robust, on-demand enterprise solutions. It can be deployed on any number of processors and on both physical and virtual servers. DB2 Enterprise Server Edition is an ideal foundation for building enterprise-wide solutions, such as high-performing 24x7, high-volume transaction processing business solutions or web-based solutions. It is the data server back-end system of choice for industry-leading independent software vendors (ISVs) building enterprise solutions.

Additionally, DB2 Enterprise Server Edition offers connectivity, compatibility, and integration with other enterprise DB2 and IBM Informix® data sources. DB2 Enterprise Server Edition is available on either a Processor Value Unit or per Authorized User pricing model. You must acquire a separate user license for each Authorized User of this product with a minimum purchase of 25 users per 100 Processor Value Units.

IBM DB2 Advanced Enterprise Server Edition

The Advanced Enterprise Server Edition contains enhanced features to meet your more demanding business needs. The sections that follow cover these features and explain what they are in the context of DB2.

IBM Database Enterprise Developer Edition

Important: This edition cannot be used for production systems, because it is intended solely for application development.

This edition offers a package to be used for a single application developer to design and build prototype applications for deployment on any of the IBM Information Management client or server platforms. This comprehensive developer offering includes DB2 Workgroup Server Edition, DB2 Enterprise Server Edition, IBM Database Enterprise Developer Edition, IBM DB2 Connect™ Unlimited Edition for System z, and all the DB2 Version 10.1 features, so you can build solutions that use the latest data server technologies.

1.4.3 Feature availability and differences between editions

Table 1-1 summarizes the differences between the following product editions:

- IBM DB2 Enterprise Server Edition
- IBM DB2 Workgroup Server Edition
- IBM DB2 Advanced Enterprise Server Edition

Table 1-1 DB2 features and functions by edition

Features and functions	Description	DB2 Workgroup Edition	DB2 Enterprise Server Edition	DB2 Advanced Enterprise Server Edition
Adaptive Compression (*new in DB2 10.1*) and Classic Row Compression	Adaptive compression uses two compression approaches: A table-level dictionary used in classic row compression to compress data based on repetition, and a page-level dictionary-based compression algorithm to compress data based on data repetition within each page of data.	No	DB2 Storage Optimization Feature, which is a separately priced feature	Yes
Continuous Data Ingest (*new in DB2 10.1*)	A high-speed client-side DB2 utility that streams data from files and pipes the data in to DB2 target tables. The ingest utility can move large amounts of real-time data without locking the target table.	No	No	Yes
DB2 Governor	The DB2 Governor extracts records of a specified type from the governor log files and monitors or changes the behavior of applications that run against a database.	No	Yes	Yes
DB2 pureScale functionality	This feature provides an active/active clustered DB2 instance.	Up to 16 cores and 64 GB total cluster size	DB2 pureScale Feature	DB2 pureScale Feature

Features and functions	Description	DB2 Workgroup Edition	DB2 Enterprise Server Edition	DB2 Advanced Enterprise Server Edition
Materialized query tables (MQTs)	MQTs are a powerful way to improve response time for complex queries that aggregate data over one or more dimensions or joins, and aggregate data over a group of tables (data from a commonly accessed subset of data (hot data)).	No	Yes	Yes
Multidimensional clustering (MDC)	MDC provides an elegant method for clustering data in tables along multiple dimensions in a flexible, continuous, and automatic way. MDC can significantly improve query performance.	No	Yes	Yes
Multi-Temperature Storage (*new in DB2 10.1*)	You can manage your IT budget more efficiently by configuring your database so that only frequently accessed data (hot data) is stored on expensive fast storage and infrequently accessed data (cold data) is stored on slower, less-expensive storage.	No	Yes	Yes
Q Replication with two other DB2 for Linux, UNIX, and Windows servers	A replication solution that uses IBM WebSphere® MQ message queues for high-volume and low-latency replication.	No	No	Yes
Query parallelism	Components can be run in parallel, and factors are interrelated to dramatically enhance performance. The following types of parallelism are supported: ► I/O ► Query ► Utility	No	Yes	Yes

Features and functions	Description	DB2 Workgroup Edition	DB2 Enterprise Server Edition	DB2 Advanced Enterprise Server Edition
Table partitioning	Table partitioning is a data organization scheme in which table data is divided across multiple data partitions according to values in one or more partitioning columns of the table.	No	Yes	Yes
Workload Management	The ability to manage your workloads to maximize system efficiency and throughput, while helping you achieve your business performance objectives.	No	No	Yes

For this book, the table does not include information about IBM Database Enterprise Developer Edition. For information about IBM Database Enterprise Developer Edition, go to the Information Center at:

http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/index.jsp?topic=%2Fcom.ibm.db2.luw.licensing.doc%2Fdoc%2Fr0053238.html

These editions include the following features:

- Advanced Copy Services (ACS): This application programming interface (API) defines a set of functions that the database manager (DBM) uses to communicate with storage hardware to perform snapshot backup operations.
- High availability disaster recovery (HADR): A disaster recovery (DR) solution that uses log shipping and provides data to standby systems when partial or complete site failure occurs on a primary system.
- IBM Data Studio: IBM Data Studio is a software that can be used for database administration tasks, analyzing and tuning queries, and creating, deploying, and debugging database applications.
- Label-based access control (LBAC): LBAC greatly increases the control you have over who can access your data. You can use LBAC to decide exactly who has write access and who has read access to individual rows and individual columns.

- Online reorganization: Over time, data in your tables can become fragmented, which increases the size of tables and index. Reorganization of tables and indexes compacts your data, reclaiming wasted space and improving data access.
- Oracle compatibility: Adds support to Oracle SQL dialect, the PL/SQL language, and other semantic capabilities to quickly enable applications initially developed for Oracle to DB2.
- pureXML: You can use the pureXML feature to store well-formed XML documents in database table columns that have the XML data type. By storing XML data in XML columns, the data is kept in its native hierarchical form, rather than stored as text or mapped to a different data model.
- Replication tools: The replication tools consist of the ASNCLP command-line program, the Replication Center, and the Replication Alert Monitor tool.
- Spatial Extender: Support and use of DB2 Spatial Extender involves two main activities: setting up DB2 Spatial Extender and working on projects that use spatial data.
- SQL Replication: A type of replication that uses staging tables.
- (*New in DB2 10.1*) Row and column access control (RCAC): RCAC is an additional layer of data security for row and column access control. RCAC controls access to a table at the row level, column level, or both, and can be used to complement the table privileges model.
- (*New in DB2 10.1*) Time travel query: There are many business needs requiring the storage and maintenance of time-based data. Without this capability in a database, it is expensive and complex to maintain a time-focused data support infrastructure. With temporal tables, the database can store and retrieve time-based data without additional application logic.

1.4.4 Selecting the best DB2 version and edition

If you are using any version older than DB2 10.1, it is best to upgrade to DB2 10.1 to ensure that you get the performance and feature benefits that it provides.

If you are using DB2 9.8, Chapter 5, “Upgrading to DB2 10.1” on page 251 describes the steps to upgrade to DB2 10.1.

If you are assessing the editions of DB2, refer to the information provided in this chapter and determine the features that can best help you achieve the results that you need.

2

Planning and architecture

Chapter 1, “An overview of the DB2 pureScale Feature” on page 1 describes the overall architecture and provided a general overview of the DB2 pureScale technology. This chapter takes a deeper look into the actual hardware requirements so that you can plan and design your own DB2 pureScale cluster.

This chapter includes the following topics:

- Before you begin
- Hardware requirements and considerations
- Software requirements and considerations
- Topology requirements and considerations
- High availability and disaster recovery
- Client requirements and considerations

Our goal is to prompt you in to thinking about how your configuration fits your business needs, and how it can relate to some of the examples provided. Having a good understanding of the technology ensures that your workload works well with the various configuration options that you decide to use.

2.1 Before you begin

A DB2 pureScale cluster consists of various hardware components, including the servers, networking, and storage. To get the best results, planning the type of hardware to be purchased and understanding the needs of the overall architecture is critical.

Whether you are currently using another database server or a different version of DB2, it is important to understand and plan the transition to the DB2 pureScale Feature. Although you can modify decisions along the way, some decisions are far too costly to make as you move further into the project. Also, understanding the fundamental hardware requirements for the cluster, understanding what your business needs are, and matching those business needs to the overall solution, can help finalize decisions that provide the basis for your solution. The DB2 pureScale Feature provides flexibility in the amount of redundancy that is required and different failure recovery times to accommodate different budgets and business needs.

Before continuing, here is a review of the basic concepts that were covered in Chapter 1, “An overview of the DB2 pureScale Feature” on page 1:

- The DB2 *members* are those members that are responsible for handling the overall database workloads.
- The DB2 *cluster caching facilities* (CFs) are responsible for lock management and global caching.
- The *shared storage* lets you access data from various servers simultaneously.

The minimal recommended configuration in a DB2 pureScale environment is two DB2 members, two cluster CFs, and a single shared storage subsystem. Understanding the basic configuration can help you understand the configuration options that are available and the hardware considerations to ensure a successful cluster deployment that matches your workload.

Figure 2-1 shows a sample DB2 pureScale Feature configuration composed of three DB2 members and two CFs.

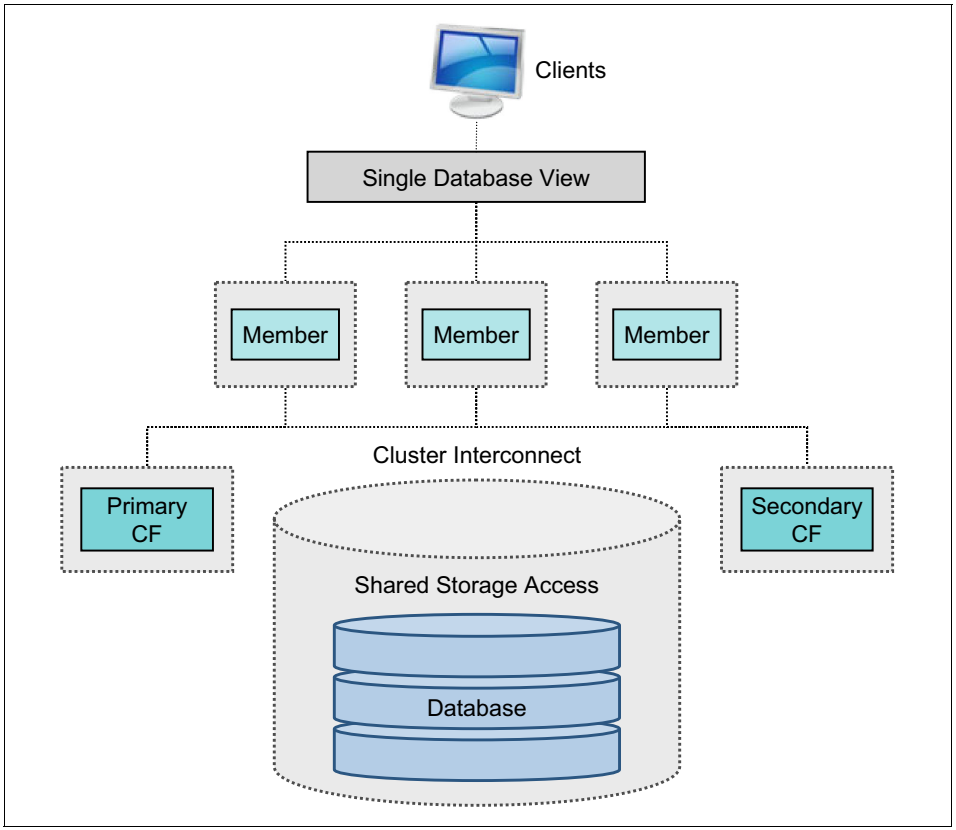

Figure 2-1 Sample DB2 pureScale cluster - overview and components

Understanding the hardware and software requirements is critical to successfully deploying a DB2 pureScale cluster. There are various components in the DB2 pureScale Feature that interact on a low level with the operating system to provide functions, such as monitoring and automated failure recovery. Therefore, understanding business requirements is a crucial part of the planning process to ensure that you make the best decisions early on and that you avoid making costly changes in the future.

It is important to read through all the requirements before putting a plan together to cover any preliminary prerequisites to ensure the optimal deployment of the DB2 pureScale Feature.

2.2 Hardware requirements and considerations

Building the DB2 pureScale cluster involves different hardware components, including the servers, storage subsystem, and switches. The next sections go through the supported hardware for each of these components and then explain the considerations to take into account.

2.2.1 Supported hardware

At the core of the DB2 pureScale cluster are the following hardware components:

- The *servers* run the DB2 pureScale software.
- The *storage subsystem* holds the database and all of its related components.
- The *networks* route all of the communication between the clients, servers, CFs, and members.

Supported servers

Support for servers and operating systems (OS) is linked. For example, servers that run on Linux operating systems are IBM System x servers, whereas IBM AIX® systems are all POWER based servers. Thus, in some organizations, which server you use is determined by your operating system (AIX or Linux).

After deciding upon the architecture of the server, there are a limited number of choices available that have been thoroughly tested with the DB2 pureScale Feature. For System x, four possible models are available. POWER systems offer almost the entire lineup of machines. Each model of server has different processor and memory capacities, along with differences in the number of interconnect adapters that can be used.

Table 2-1 provides information about the supported machine hardware types.

Table 2-1 Supported machine hardware types for the DB2 pureScale Feature

System type	System model	Operating system	Website to find more information
System x	x3650 M3	Linux	http://www-03.ibm.com/systems/x/hardware/rack/x3650m3/specs.html
System x	x3690 X5	Linux	http://www-03.ibm.com/systems/x/hardware/enterprise/x3690x5/specs.html
System x	x3850 X5	Linux	http://www-03.ibm.com/systems/x/hardware/enterprise/x3850x5/specs.html

System type	System model	Operating system	Website to find more information
IBM BladeCenter®	BladeCenter HS22	Linux	http://www-03.ibm.com/systems/bladecenter/hardware/servers/hs22/specs.html
IBM POWER6®	P6 550, 595	AIX	P6 550: http://www-03.ibm.com/systems/power/hardware/550/index.html P6 595: http://www-03.ibm.com/systems/power/hardware/595/specs.html
IBM POWER7®	710, 720, 730, 740, 750, 770, 780, 795	AIX	P710: http://www-03.ibm.com/systems/power/hardware/710/specs.html P720: http://www-03.ibm.com/systems/power/hardware/720/specs.html P730: http://www-03.ibm.com/systems/power/hardware/730/specs.html P740: http://www-03.ibm.com/systems/power/hardware/740/specs.html P750: http://www-03.ibm.com/systems/power/hardware/750/specs.html P770: http://www-03.ibm.com/systems/power/hardware/770/specs.html P780: http://www-03.ibm.com/systems/power/hardware/780/specs.html P795: http://www-03.ibm.com/systems/power/hardware/795/specs.html

Based purely on your specifications, POWER systems provide more memory, cores, and networking bandwidth. Therefore, for larger workloads that require more memory and processing power, build the architecture using POWER systems.

Important: You must use the same platform for all servers. For example, you cannot have System x system members and POWER system CFs. In addition, it is a preferred practice to keep the processor types the same between members and the same for CFs. If you use different processor types, you must account for processing capability differences between the processors while determining how many processor cores are needed.

Supported storage subsystems

The DB2 pureScale Feature supports three categories for storage subsystems. These categories are grouped by capabilities and are based on whether the storage subsystem and associated driver support fast I/O fencing and whether the DB2 cluster services tiebreaker disk can be created and used appropriately:

- Fast I/O fencing: Before recovering a failed member in the DB2 pureScale instance, DB2 cluster services ensure that it is not possible for the failed member to modify shared data on disk.
- DB2 cluster services tiebreaker: The tiebreaker disk is used to determine quorum for Reliable Scalable Cluster Technology and protect the cluster when half of the cluster fails.

To ensure that you achieve shorter recovery times, have faster I/O fencing capabilities and use storage units with SCSI3-PR support. In this case, it is important to ensure that all required drivers are installed on the OS so that the communication between the storage unit works as expected.

Category 1: Storage device and multipath I/O driver combinations that support the fastest recovery times in the DB2 pureScale Feature

This category results in the highest resiliency and fastest recovery times. These storage subsystems and drivers are validated by the DB2 pureScale Feature and support both the fast I/O fencing and DB2 cluster services tiebreaker that are described previously. Table 2-2 summarizes the set of supported storage devices, the drivers that are required for AIX or Linux, and the protocol that is used to communicate between the servers and the storage.

Table 2-2 Storage device and multipath I/O driver combinations to achieve the highest recovery times

Storage devices	Multipath I/O (MPIO) drivers for AIX systems	Multipath I/O drivers for Linux systems	Protocol
IBM System Storage® DS8000® series	Subsystem Device Driver Path Control Module (SDDPCM)	DM-MP	Fibre Channel (FC)
IBM System Storage DS5000 series	MPIO	DM-MP or RDAC	Fibre Channel

Storage devices	Multipath I/O (MPIO) drivers for AIX systems	Multipath I/O drivers for Linux systems	Protocol
IBM System Storage DS4000® series	MPIO	DM-MP or RDAC	Fibre Channel
IBM System Storage DS3200 series	MPIO	DM-MP or RDAC	Fibre Channel
EMC VMAX or Symmetrix family	MPIO driver provided by EMC (driver file `EMC.symmetrix.fcp.MPIO.rte`)	DM-MP	Fibre Channel
NetApp FAS filers		DM-MP	Fibre Channel
NetApp FAS filers	MPIO driver provided by NetApp	DM-MP	Internet SCSI (ISCSI)
Virtual I/O Server (VIOS)	MPIO or SDDPCM	DM-MP	Fibre Channel

Category 2: Storage device and multipath I/O driver combinations that the DB2 pureScale Feature supports

The storage devices in this category work with the DB2 pureScale Feature. These combinations support the tiebreaker disk but do not provide fast I/O fencing. Thus, the recovery times for host or hardware failures might be longer than those times listed in Table 2-2 on page 28, assuming that all other variables are the same. These failure times are dependent on a lease recovery wait time that the shared file system must abide by to ensure that a failed host cannot modify data in the shared storage. Table 2-3 describes these storage components in detail.

Table 2-3 Storage device and multipath I/O driver combinations in the DB2 pureScale Feature

Storage devices	Multipath I/O drivers for AIX systems	Multipath I/O drivers for Linux systems	Protocol
IBM Storwize® V7000	MPIO or SDDPCM	DM-MP	Fibre Channel
IBM SAN Volume Controller	SDDPCM	DM-MP	Fibre Channel
EMC VMAX or Symmetrix family	EMC Power Path		Fibre Channel
Hitachi Universal Storage Platform V (USP V)	MPIO		Fibre Channel

Category 3: Storage device and multipath I/O driver combinations

These storage devices are all other storage units that are not validated to work with the DB2 pureScale Feature. So, any device here is still a supported storage unit; however, any issues encountered might result in longer than expected diagnosis time and might require consultation with the hardware vendor. These storage subsystems have not been tested officially and might or might not support fast I/O fencing or the DB2 cluster services tiebreaker disk. The general minimum storage considerations (requirements) still apply for these devices as well and can be found in 2.2.2, “Hardware considerations” on page 31.

Supported networks

With the DB2 pureScale Feature, a minimum of two networks are required:

- The first network is the regular Ethernet network that you might see on any DB2 instance. This network is the network that clients tend to use to access the database server.
- The second network is the fast interconnect network that is used for communications between the members and the cluster CFs. This interconnect network must be a high speed, low latency network to allow for quick transfer of data between members and the CF.

The interconnect must be a Remote Direct Memory Access (RDMA) capable fabric. RDMA provides the ability for one host to make direct updates in memory on another host without needing interrupt the processor on either host. Two technologies can provide RDMA interconnect:

- The InfiniBand fabric, which provides RDMA through the uDAPL API
- RDMA over Converged Ethernet (*RoCE*), which works on top of 10 Gigabit Ethernet (GbE)

Although RoCE is not as low latency as uDAPL is on InfiniBand, it can be sufficient for most workloads.

Table 2-4 provides a list of supported network cards for the DB2 pureScale Feature.

Table 2-4 List of supported network cards for the DB2 pureScale Feature

Network part	InfiniBand network	10 GbE network
Network card	InfiniBand adapters (Mellanox ConnectX-2 VPI card)	10 GbE adapters (capable of Remote Direct Memory Access over Converged Ethernet)

Network part	InfiniBand network	10 GbE network
Network switch	InfiniBand switch (Mellanox InfiniScale IV QDR InfiniBand Switch)	Any 10 GbE switch that supports priority-based flow control, as specified by IEEE 802.1Qbb

Because a DB2 pureScale cluster requires more memory and processors for more intensive workloads, the need for more adapters at the cluster CF increases. As of DB2 9.8 FP4, support for multiple cluster interconnect adapters on the CFs is added. This addition improves the redundancy in the system and allows for fast interconnect adapters or ports to fail without affecting the DB2 pureScale Feature. In addition, this addition allows for increased bandwidth at the cluster CF, so that more data is passed between members and cluster CFs.

In addition to the adapters and switches, you need cables to connect the components together. The cables that are required come in various lengths to suit the needs of your data center. Table 2-5 describes the IBM part numbers (and associated feature codes) for the cables.

Table 2-5 Supported cables for the DB2 pureScale Feature

	1 m	3 m	5 m
IBM Part Number	46K6182	46K6183	46K6184
Feature Code Number	EN01	EN02	EN03

It is important to understand that DB2 enables CF to use 1 - 4 ports within servers that can handle 1 - 4 adapters. Any single adapter can have either one or two ports; however, using both ports on a single adapter does not improve bandwidth and cannot add tolerance, for example, for an adapter failure. Using two ports on a single adapter does provide for redundancy, for example, if a cable or port on either the adapter or switch fails. If two ports on two adapters are used, you can improve bandwidth and add tolerance for adapter or main system board slot failures.

2.2.2 Hardware considerations

To ensure expected availability (workload response times), keep in mind the following considerations for processor, memory, and storage to ensure that the workload performs as expected and provides for optimal results.

Processing power considerations

When considering the number of processor cores, several factors can affect the number of cores per server (or logical partitions (LPARs) on AIX). Consider the topology for your DB2 pureScale instance. In other words, do you know how many members you need?

> **Tip:** This description makes an assumption that you have two cluster CFs. In some instances, you can have one CF. However, for any production-ready cluster, two CFs are recommended to ensure fault tolerance.

In some cases, the number of cores needed to run an application influences the topology as well. The first step to determining the topology and number of cores that you need for the DB2 pureScale Feature is to determine how many cores you need in a non DB2 pureScale instance. Treat the workload as though it is running on a “normal” single host DB2 instance and use the tools that you are familiar with to determine this first core count.

Then, after you have that initial core count, you can start to look at whether you need multiple systems or if you need to increase from two to more members. Other factors can increase the number of members, including whether workload affinities are used or if there needs to be dedicated batch processing members in addition to the regular transactional processing.

When you know how many initial cores there are and how many members are needed, you can apply some general rules of thumb to determine the initial base core counts per member and how many cores you need for the cluster CF.

Apply the following general rules:

- Peak workload requirements

  Consider the number of peaks your application or applications that are connecting to the DB2 pureScale cluster might have. These peaks might include seasonal peaks (the peak sales season for your product), daily peaks (users who log in at specific times of the day), and reporting peaks (batch jobs that run and whether these jobs require more processing power than daily work).

  These questions are similar to how you size any DB2 instance. Capacity on demand might be an option if the peaks are of a short duration and can be predicted. If so, consider sizing for daily usage and have enough capacity on hand to apply it only for the peaks. If the demands are more day to day, you likely need to size for that type of demand plus some extra capacity for growth. The peak workload requirements affect the initial core count.

- Number of members

  When there are more members in the cluster, you need fewer cores per members because this workload is spread across the members. However, you might need to increase the cores on the CF to handle the extra memory requirements and page transfers.

- Failure tolerances

  As described previously, there are several reasons why a member might fail. Adding more members into the cluster can improve failure tolerances, but you also need to ensure that there is enough capacity on the remaining members to continue processing transactions in a timely manner. You also need to consider how many member failures you are willing to tolerate.

  For example, if you have four members in the DB2 pureScale cluster, it is assumed that you can tolerate one member failing. But what happens if two members fail? This situation is particularly important if you are on an AIX system and have partitioned physical machines in to LPARs that are used by the DB2 pureScale instance. The possibility that you need to shut down one physical machine results in all the LPARs on that system being shut down. Then, the question becomes, if you had to shut down that system, are you vulnerable to more failures? You might need to increase the number of cores per member if you must tolerate more members failing. Do not reduce the number of cores per member because you think nothing will fail.

You have determined the total number of cores needed for the DB2 pureScale cluster. These numbers are a rough estimate and do not take the specifics of your case in to account. So, you need to test and verify the number.

You can divide the total number of cores between members and cluster CFs in a ratio that is dependent on the workload's read/write ratio. The more reads there are in the workload, the less CF cores that are required. For a workload that is 95% read, you need only one CF core for every 10 member cores. When the workload is more write driven and less read driven, for example only 70% reads, the number of cores that one CF core can deal with is six member cores:

- Write heavy application: For every six member cores, you need one core per CF.
- Read heavy application: For every 10 member cores, you need one core per CF.

For example, if your application is a write heavy application and you calculated a total core count of 24 cores for the DB2 pureScale Feature, you need a minimum of one CF core for every six cores that a member has. Allocate 18 cores for the DB2 members, and the remaining cores are used as three on the primary CF and three on the secondary CF. Thus, the total core count is still 24 (18 + 3 + 3).

If you look at the number of members that are in the instance, the total cores can still be adjusted up or down. For example, if you have two members, you can split the 18 cores between the two members and end up with nine cores each. Now, that is fewer cores than the initial baseline sizing for a non DB2 pureScale instance. However, when creating that baseline, you also add in extra idle space (25% - 50%, depending on the extra capacity that you want). This idle space is used in cases where one of the two members fail. If the idle space was planned for near 0, or less than 20%, you need to increase the idle space (increase the total number of cores by two) so that you can have 10 cores per member and two members total for 20 cores. You must round the number of cores up for the CF. So, you need four cores in this case.

If you look at the same example with three members, you need six cores per member. However, if you want to be able to survive two member failures, six cores might not be enough to handle the workload. Most customers look only at a single member failure, and the 12 remaining cores are more than sufficient to handle the full workload in that case.

In cases where there might be workload affinity or batch processing, where there is perhaps a need to consider more than a single member failure, consider using shared processor pools on AIX systems to manage the failure cases. In these setups, you can have LPARs defined to have a base entitlement and a maximum that covers the number of cores that are needed if multiple members fail. Those cores can then have a higher priority to provide the processing power when needed. Otherwise, the cores are used for other purposes on the same physical system. On Linux systems, this affinity can be achieved using processor binding methods so that DB2 runs only on a specific number of processors.

Memory considerations

When considering the amount of memory that is needed, the goal is to have as much memory as needed to prevent swapping at the operating system level. DB2 grabs available memory on the system and leaves only a small fraction free on the system. DB2 uses a large chunk of the memory for the buffer pool.

In the DB2 pureScale Feature, the member's buffer pool is now referred to as the *local buffer pool* (LBP). There is also a *group buffer pool* (GBP) at the cluster CF that is used to hold dirty pages. This two-tier buffer pool system is the bulk of the memory that is required for the DB2 pureScale Feature.

Generally, members need anywhere from 4 GB to 8 GB of memory per core that is being used. This memory is the memory that is needed on the host for just DB2. You also need to add memory for the operating system and any other processing that might occur on the host.

In most cases, DB2 memory settings are set to AUTOMATIC. So, DB2 uses as much memory as possible on the system and still prevents swapping from occurring. Alternatively, the cluster CF must have a certain percentage of memory based on the amount of memory that all members have. In general, for a workload that is 70% reads, at least 35% - 40% of the LBP size also exists in the GBP of the CF per database. In cases where the write percentage is small for the workload, you still need a minimum of 25% of all LBPs per database. In all cases where there are two members, consider using 40% - 50% for the GBP, because more memory is required when any component in the DB2 pureScale cluster fails.

Storage considerations

Disk space is required for storing data, logs, and the DB2 installation components. This section provides information about the minimum storage requirements locally on the host and the shared storage requirements.

Each host requires the following amount of minimum storage before installing the DB2 pureScale Feature:

- 3 GB to extract the installation
- 3.5 GB for the installation path
- 5 GB for the `/tmp` directory
- 1 GB for the instance home directory
- 5 GB for the `/var` directory

In addition to these requirements, Table 2-6 lists other requirements to ensure a successful setup of the DB2 pureScale environment.

Table 2-6 Disk descriptions and minimum requirements for the DB2 pureScale Feature

Disk name	Description	Minimum recommended space
DB2 Shared Disk (SQLLIB)	This disk contains all DB2 instance's configuration and metadata files.	10 GB
Shared Database Data	The amount of space allocated to this disk varies according to the amount of data stored within the database.	Depends on application needs
TieBreaker	This disk acts as the quorum device, helping the cluster manager make cluster management decisions when the cluster manager's normal decision process does not produce a clear choice.	25 MB

Disk name	Description	Minimum recommended space
Shared Log	This disk is used to store all the logs, and its size depends on the expected number of transactions and the applications logging requirement.	Depends on application needs

Here are some tips to ensure the best I/O throughput and quickest recovery times:

- It is possible to have the shared disk (SQLLIB) and the data on the same file system. This setup, however, is not preferred because it can lead to I/O contention.
- Support for SCSI-3 PR is a preferred practiced because it provides faster I/O fencing capabilities that can result in shorter recovery times.

You must consider redundant array of independent disks (RAIDs) when taking your storage system into account. RAID is what is used to define the overall way data is written in to the disks. Selecting the RAID configuration can depend on the performance that you are looking to achieve, because some RAIDs are meant for faster performance than others.

Table 2-7 lists some possible configurations.

Table 2-7 RAID levels for required DB2 pureScale disks

Disk name	RAID Level	Reason
Shared Logs	RAID 1+0	RAID 1+0 requires a minimum of four disks, but provides both redundancy and speed because a mirror of logs is available for access.
Shared Database Data	RAID 5	It is possible to do this level as RAID 1+0, but RAID 5 is ideal because you get single disk failure redundancy and have more disks available to hold a larger data set.
Tiebreaker	RAID 0	This level is the quorum disk and does not require redundancy. Thus, you can use RAID 0.
DB2 Shared Disk (SQLLIB)	RAID 5	It is possible to do this level as RAID 1+0, but RAID 5 is ideal because you get single disk failure redundancy and do not require faster throughput for this portion.

Disk name	RAID Level	Reason
Host Disk	RAID 1	Adding up the totals shows that the operating system with the base DB2 installation needs no more than 120 GB of space. Therefore, you can mirror two disks to produce redundancy and, at the same time, save on space.

Altering the RAID level: Altering the RAID level after installing the DB2 pureScale Feature can result in complications. Take time to understand the RAID level that is most suitable for your application and purposes. For most general environments, following the guidelines listed in Table 2-7 can help ensure that you set up the storage correctly and achieve optimal results.

2.3 Software requirements and considerations

Because the DB2 pureScale Feature is a software product that runs on top of an operating system, it is critical to take software requirements and considerations into account. Doing so ensures that the DB2 pureScale Feature is compatible with the operating system level, the drivers that are used to access storage and the network, and your systems are up to date from a software perspective.

This section provides information about the supported operating systems and considerations about those systems.

2.3.1 Supported operating systems

Table 2-8 lists the operating systems that the DB2 pureScale Feature supports.

Table 2-8 Supported OS versions for the DB2 pureScale Feature

Distribution	Supported versions	Remarks
AIX	AIX V6.1 TL 3, 4, 5	Support removed in DB2 10.1. Minimum SP level 5 for all TLs.
AIX	AIX V6.1 TL 6+	Minimum SP level 1.
AIX	AIX V7.1 TL 0+	Minimum SP level 1.
Linux Novell OS	SUSE® Linux Enterprise Server 10 SP3, SP4	SUSE Linux Enterprise Server 10 SP3 removed in DB2 V10.1.
Linux Novell OS	SUSE Linux Enterprise Server 11 SP1	

Distribution	Supported versions	Remarks
Linux Red Hat OS	Red Hat Enterprise Linux 5.5 +	Red Hat Enterprise Linux 5.5 was removed in DB2 V10.1.
Linux Red Hat OS	Red Hat Enterprise Linux 6.1+	Support available from DB2 V10.1.

Using a supported operating system is critical to successfully deploying a DB2 pureScale cluster. Various components that make up the recovery automation and the communication between the hosts are closely tied at a low level with operating system modules. Do *not* use the DB2 pureScale Feature on any other operating system than the ones listed in Table 2-8 on page 37.

2.3.2 Operating system considerations

After ensuring that the base operating system requirements are met, consider the following operating considerations:

- When service pack (SP) upgrades are available for new releases (for example, when SUSE Linux Enterprise Server 11 comes out with a new SP such as SP2 or SP3), you might want to upgrade.

  Before you upgrade: You need to verify that DB2 supports the latest SP release before you start the upgrade.

- Regularly install OS updates to ensure that the latest OS security compliances are met.
- Install only OS patches that are provided from your specific distribution unless otherwise directed by the DB2 support team or the Information Center.
- DB2 pureScale instances require specific users and groups, including fenced users. You can create the users before starting the DB2 Setup wizard, or have the wizard create them for you as you progress through the panels. If you are not creating or modifying instances, you can create the required users after completing the installation but before creating the instance.
- If using an AIX system, ensure that you are using the correct firmware on the hardware, as described in 2.4, “Hardware firmware requirements” on page 41.
- If using an AIX system, you must ensure that the correct version of User Direct Access Programming Language (uDAPL) is installed and configured at the level specified in the software prerequisites.

- Ensure that OpenSSH is installed and configured properly between all of the hosts to allow access without a password between the hosts using both the long and short host names of the system.
- Ensure that the shared storage is accessible by all of the hosts in the cluster. All hosts need to see the same physical volume identifier (PVID) for the same device on AIX systems.
- Confirm that Ethernet and all additional communication ports are properly configured and connected to their respective switches. You can use the **ifconfig** command to check for **ethX** and **ibX** network configurations on Linux systems or **enX** and **ibX** network devices on AIX systems.
- Ensure that different packages that are required and kernel versions that are required are installed as per Table 2-9 or consult the DB2 Information Center for the latest information.

Table 2-9 OS packages and kernel versions required

Linux distribution	Required packages	Remarks
Red Hat Enterprise Linux 5.6	libstdc++ (both 32-bit and 64-bit libraries) glibc++ (both 32-bit and 64-bit libraries) cpp gcc gcc-c++ kernel-headers kernel-devel binutilsOpenSSH sg3_utils ntp-4.2.2p1-15.el5	Kernel version level: 2.6.18-194.26.1.el5. Red Hat Enterprise Linux is supported in DB2 pureScale environments with an InfiniBand cluster interconnect.
Red Hat Enterprise Linux 6.1	dapl (64-bit libraries only) ibsim (64-bit libraries only) ibutils (64-bit libraries only) libibverbs-rocee (10 GbE) libibverbs (InfiniBand network) librdmacm libcxgb3 libibmad libibumad libipathverbs (64-bit libraries only) libmlx4-rocee (10GbE) libmlx4 (InfiniBand network) libmthca libnes (64-bit libraries only) rdma (no architecture) ntp-4.2.4p8-2.el6.x86_64/ntpdate-4.2.4p8-2.el6.x86_64	Supported since DB2 10.1.

Linux distribution	Required packages	Remarks

SUSE Linux Enterprise Server 10 SP 3 or 4	`libstdc++` (both 32-bit and 64-bit libraries)
	`glibc++` (both 32-bit and 64-bit libraries)
	`cpp`
	`gcc`
	`gcc-c++`
	`kernel-source`
	`binutils`
	`OpenSSH`
	`sg3_utils`
	`scsi*.rpm`
	`ntp-4.2.4p8-1.3.28`
		Kernel version level: 2.6.16.60-0.69.1-smp 3.
		
		Support for SUSE Linux Enterprise Server 10 SP3 removed in DB2 10.1.

SUSE Linux Enterprise Server 11 SP 1	`libstdc++` (both 32-bit and 64-bit libraries)
	`glibc++` (both 32-bit and 64-bit libraries)
	`cpp`
	`gcc`
	`gcc-c++`
	`kernel-default`
	`kernel-default-devel`
	`kernel-default-base`
	`kernel-source`
	`kernel-syms`
	`binutils`
	`OpenSSH`
	`sg3_utils`
	`ntp-4.2.4p8-1.3.28`
		On SUSE Linux Enterprise Server 11 SP1, the default kernel must be upgraded to version 2.6.32.36-0.5, requiring the following packages:
		`kernel-default-2.6.32.36-0.5.2`
		`kernel-default-devel-2.6.32.36-0.5.2`
		`kernel-default-base-2.6.32.36-0.5.2`
		`kernel-source-2.6.32.36-0.5.2`
		`kernel-syms-2.6.32.36-0.5.2`

AIX 6 and 7	OpenSSH level 4.5.0.5302 or higher. XL C/C++ runtime library level is 9.0.0.12 or higher.	

In general, when updating operating systems, consult the DB2 Information Center for any additional information about the service packs for the operating systems before applying them. In particular, check the prerequisites for the SP level and ensure that those prerequisites are applied.

By following these OS considerations, you can ensure that your DB2 pureScale Feature installation succeeds and is maintained with the latest standards in terms of both OS updates and packages to install.

2.4 Hardware firmware requirements

To ensure that you are working with the latest firmware, validate your current firmware version with the ones that are supported, as listed in the following sections. Not keeping up with the latest firmware can cause various complications, because the DB2 pureScale Feature interacts closely with the hardware components to provide optimal results.

2.4.1 Linux on System x systems firmware requirements

The minimum firmware required for 10 GbE or Host Channel Adapter (HCA) adapters for System x servers is Version 2.9.1000.

Install the latest supported firmware for your System x server from the following website:

http://www.ibm.com/support/us/en/

2.4.2 AIX on POWER systems firmware requirements

Table 2-10 shows the firmware requirements when working with POWER servers for an AIX installation of the DB2 pureScale Feature.

Table 2-10 AIX firmware levels for the DB2 pureScale Feature

Server	Required platform firmware level
IBM POWER7 795 (9119-FHB)	AH720_102 or higher
IBM POWER7 780 (9179-MHB)	AM720_102 or higher
IBM POWER7 770 (9117-MMB)	AM720_102 or higher
IBM POWER7 750 (8233-E8B)	AL730_035 or higher
IBM POWER7 740 (8205-E6C)	AL720_102 or higher
IBM POWER7 740 (8205-E6B)	AL720_102 or higher
IBM POWER7 730 (8231-E2B)	AL720_102 or higher
IBM POWER7 720 (8202-E4C)	AL720_102 or higher
IBM POWER7 720 (8202-E4B)	AL720_102 or higher
IBM POWER7 710 (8231-E2B)	AL720_102 or higher

Server	Required platform firmware level
IBM POWER6 595 (9119-FHA)	EH350_071 or higher
IBM POWER6 550 Express (8204-E8A)	EH350_071 or higher

After upgrading to the latest firmware, you can continue with the installation of the DB2 pureScale Feature, because all of the components that require updates from a software perspective have been handled. If you require firmware updates after the DB2 pureScale instance is created and is running, perform the firmware updates in a rolling manner (unless the DB2 Information Center says otherwise) and ensure that you update each host in the cluster separately.

2.5 Topology requirements and considerations

Topology is the configuration of a DB2 pureScale cluster with respect to the number of members and cluster CFs, the number of physical machines, and also the layout of the members and CFs over those machines. Because the DB2 pureScale Feature uses several hosts for availability, scalability, and transparency, certain requirements must be met and considerations must be understood to have a cluster that reacts as expected in failure scenarios. The next sections describe these considerations to ensure that you plan your topology correctly.

2.5.1 Topology requirements and considerations

There are several requirements that are necessary for a successful deployment of a DB2 pureScale cluster in a production environment. For example, a preferred configuration has a minimum of two members and two CFs for a production environment because you need at least two of each for redundancy purposes.

Your test environment should always have the same number of members as your production environment, but if you cannot have as many, the minimum recommended number of members is still two. You can have one member in your test environment, but you cannot test how the application reacts to failures of the member to which it is connected. For testing environments, a single CF is sufficient, unless you need high availability in the test environment itself. CF failures are internally processed in DB2 and rarely lead to any specific application reaction.

With that said, you can achieve these configurations in several ways because a DB2 member and a CF can share a single host on both AIX and Linux systems. Having the CF and member share a single host is not preferable from a high availability point of view, because when a host fails, both a member and a CF are down at the same time. This event leaves up only a single CF and the remainder of the members, which produces a vulnerable state because subsequent failures can lead to a full outage.

Ask yourself the following questions to understand how many DB2 members you actually need for your particular cluster:

- Is there any workload affinity?
- Will workload balancing be used or can a single host take most of the work?
- Is there batch work that occurs on a regular basis?

The answers to these questions can help you with the topology that is best suited to your needs.

Ask yourself the following questions regarding the number of instances that you might want or need for testing purposes:

- Will this cluster be used for testing of functionality or system verification?
- Will this cluster be used for production scenarios?

Knowing whether you are putting a cluster together for production versus for testing can influence whether certain options are valid and if those options might introduce a less resilient system. Consider the time line of events that will happen in the future. Is it possible for some systems to have a dual duty in the organization so that you can reduce the hardware purchased?

Systems such as the following systems might not be required for production level testing:

- A system to verify that database backups from a production system are correctly created.
- A system to verify tuning parameters before they are used on a production system.
- Dry-runs of upgrade or migration work on the system. These dry-runs might or might not be DB2 related, such as firmware updates or updates to OS drivers.

Answering these questions can help you better consider the topology that is best for your environment and decide how many members your cluster should have.

2.5.2 Common topology examples

Any topology needs at least two CFs and two members because this topology is the bare minimum that enables the failover capabilities of the DB2 pureScale Feature. Various possible hardware configurations can achieve this topology. Take continuous availability in to account when selecting the topology that works for your particular scenario.

Figure 2-2 shows the common scenario of having two members and two CFs as common topology.

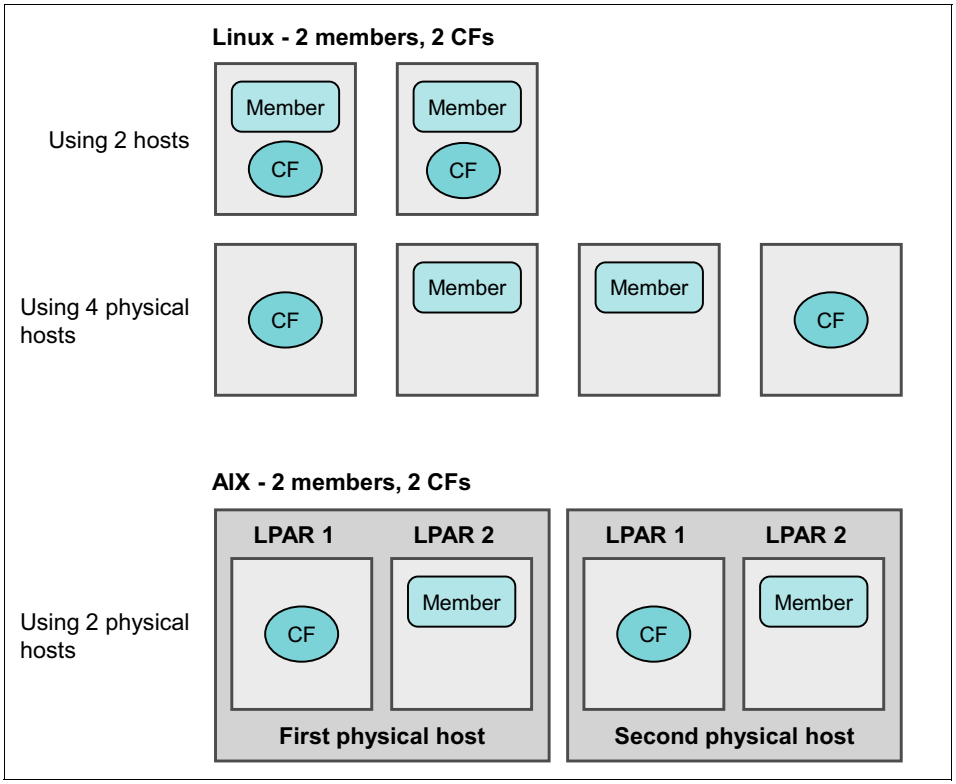

Figure 2-2 Sample configurations with two members and two CFs

Figure 2-2 illustrates the differences in availability characteristics between the two physical machines configuration and the four physical machines configuration. In the two physical machines case, if you must shut down one of the two physical machines, you automatically lose both the member and the CF because they are using a single host. Whereas, in the four physical machines case, you lose only one of the four components.

Another common scenario is to have multiple members (for example, four) and two CFs. Figure 2-3 depicts different possibilities for having multiple members and two CFs.

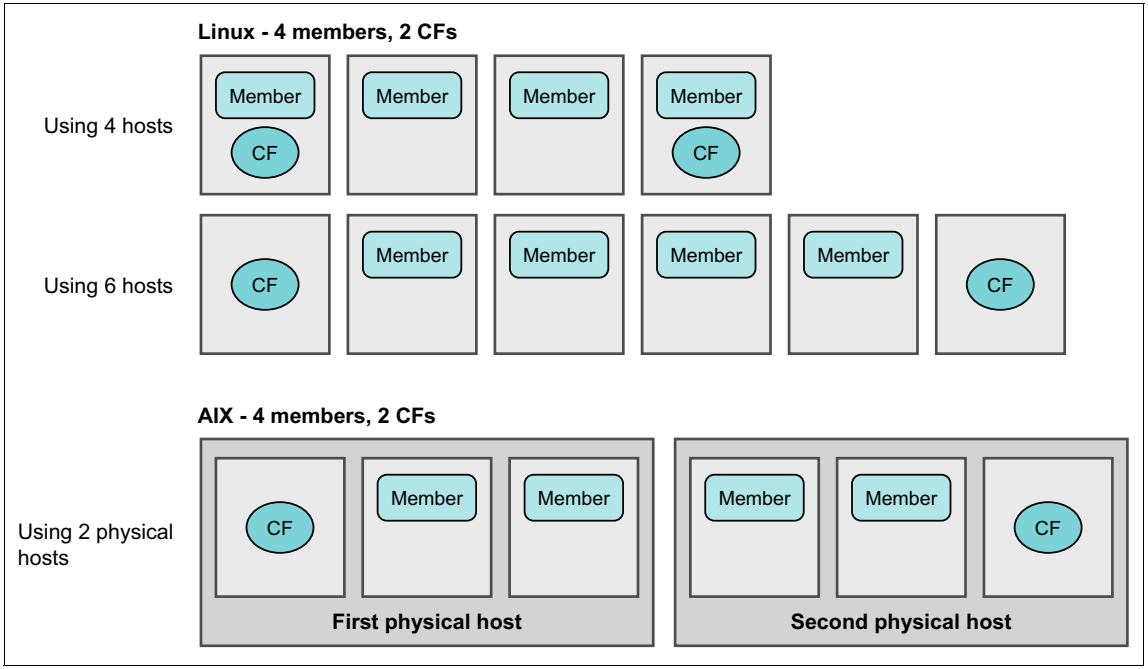

Figure 2-3 Possibilities for four members and two CFs for a DB2 pureScale cluster

The general idea is to be able to expand to additional members and have two CFs (hopefully on separate hosts) to achieve maximum tolerance against failures or uptime.

2.6 High availability and disaster recovery

Although the DB2 pureScale Feature inherently brings a local high availability solution, many customers also require a disaster recovery solution to meet business continuity requirements. When considering purchase and system design choices regarding availability, it is tempting to review long lists of high availability features and technologies. However, there are also basic principles with respect to keeping a system highly available that are as much about making good configuration choices as they are about practicing sound administrative procedures and emergency plans.

High availability is the ability of a system to continue working regardless of failures that occur with the system, up to a certain point. In all highly available systems, there is a point when enough failures render the system useless, and a *disaster recovery plan* must be in place and run at that time.

Disaster recovery is the contingency plan when the primary cluster fails to the point where it cannot be brought back online. This disaster can be as simple as a site failure because of a building fire or natural disaster. The contingency plan can be another database at a different site, either a few kilometers away or many kilometers away in another province or country.

This section covers the basic principles that provide a highly available cluster and then follows up on the different disaster recovery options that are available.

2.6.1 Basic principles of high availability

To achieve the most comprehensive availability for your investment, you must identify high availability strategies and basic principles to help meet your business demands.

When designing or configuring your database solution for high availability, consider avoiding outages, minimizing the impact of outages, and recovering quickly from unplanned outages. This plan maximizes your overall decisions when it comes to the DB2 pureScale Feature disaster recovery by trying to avoid common issues that can happen.

Avoiding outages

Try to avoid outages when possible. Think about where a failure can occur, either hardware in the components, such as hosts, adapters, cables, switches, or the infrastructure being used, such as power circuits, air conditioning units, or fans. If there are any single points of failure, you might want to consider adding redundancy at those points. For example, when a single power circuit is used to power a server, what happens if that circuit fails? Having a secondary power circuit in the building powered by another power grid might be a possible solution.

Minimizing the impact of outages

You can configure a database solution to minimize the impact of planned and unplanned outages by distributing the database solution components so that they are spread out. This solution allows some user applications to continue processing even if there is a single component offline, helping you better anticipate for both planned and unplanned outages.

For more information about how to spread the components of a cluster on to physical machines to reduce the impact of a failure, see 2.5.1, “Topology requirements and considerations” on page 42.

Recovering quickly from unplanned outages

Maximize your ability to recover quickly in the event of an unplanned outage by having a set of documented procedures that administrators can follow easily in the event of an unplanned outage. Create clear architectural documents that describe the components of the system and have all the contact information and service agreements organized so that issues can be resolved more promptly.

In addition to the documents, ensure that practice or dry-runs of recovery from unplanned outages are tried and continue to be tried over time. You can catch both errors in documentation, and changes to procedures when software is updated before it becomes a concern when an unplanned outage actually occurs. When people are faced with an unplanned outage, having these documents and procedures can alleviate uncertainty and improve the time it takes to recover, because the people involved are confident that they know what must be done to recover from the failure.

In addition to recovering from the immediate problem, understanding how to collect diagnostic information to provide support personnel with the necessary data to uncover the reason behind the failure also helps reduce the likelihood that the same failure occurs again. Although removing this step can speed up recovery, it can also be detrimental if the problem cannot be diagnosed, and continues to recur.

2.6.2 Disaster recovery techniques

As mentioned earlier, disaster recovery is intended to be the contingency plan for when the primary DB2 pureScale cluster fails beyond an immediately repairable situation. There are several approaches for disaster recovery with the DB2 pureScale Feature. The sections that follow cover the most common approaches of DB2 pureScale disaster recovery (DR) using IBM InfoSphere® Replication Server and IBM InfoSphere Change Data Capture (CDC).

Benefits of disaster recovery for the DB2 pureScale Feature

Ensuring that a solution is disaster recovery ready and can be continuously available even during primary site outages provides the following advantages:

- Minimizes business downtime and interruption
- Minimizes the periods of low productivity and continuously provides a great customer experience to avoid losing business

- Provides vital record and information assets preservation and availability
- Fulfills expectations of IBM Business Partners, shareholders, stakeholders, strategic alliances, and regulators
- Increases your ability to avoid business interruptions and key processes

The general idea is that by understanding how you can add disaster recovery capabilities in to the equation, you can further ensure that your database remains continuously available, processing transactions and handling application queries.

While looking at the options in the sections that follow, consider the following items when you determine which of these options are appropriate for your situation:

- How much time can a disaster recovery site lag behind a primary site? This question produces a *recovery point objective* (RPO), which is the amount of data that might be lost between the time the primary site failed and the disaster recover site took over.
- How much time can it take to recover the business at the disaster recovery site? This question produces a number referred to as the *recovery time objective* (RTO), which is the amount of time that it can take to get the disaster recovery site into operational mode.

Each of the methods described in the sections that follow has different RPO and RTO capabilities. RTOs can differ between solutions based on the setup and the capabilities of the people switching the primary site over to the disaster recovery site. This RTO value is not something that can be explicitly defined with the solution. Alternatively, the RPO value has its number tied directly into the solution. Some solutions can offer a 0 second RPO. Other solutions require the RPO to be more than 0 seconds, and it can be up to minutes if needed. You must understand your business need and choose a strategy that fits it.

Disaster recovery using IBM InfoSphere Replication Server

Q replication is part of IBM InfoSphere Replication Server for DB2 z/OS and for Linux, UNIX, and Windows operating systems. Q replication can replicate transactions between databases with low latency (subsecond) at high throughputs (millions of changes per minute) by capturing changes to source tables and converting committed transactional data to messages.

In Q replication, the data is not staged in tables. As soon as the data is committed at the source and read by the Q replication process, the data is sent and read at the target location, where it is converted from messages back into transactional data. The transactions are then applied to target tables with a highly parallelized method that preserves the integrity of the data.

The distances for this replication can happen over practically unlimited distances for thousands of DB2 tables, while preserving transactional integrity and tolerating system and network outages. Each database is active, allowing for workload distribution. Databases can be configured differently and can be on different types of hardware and different software versions, allowing for immediate failover during outages and maintenance, upgrades, and migrations without any downtime.

Figure 2-4 depicts how Q replication works across a source server and target server.

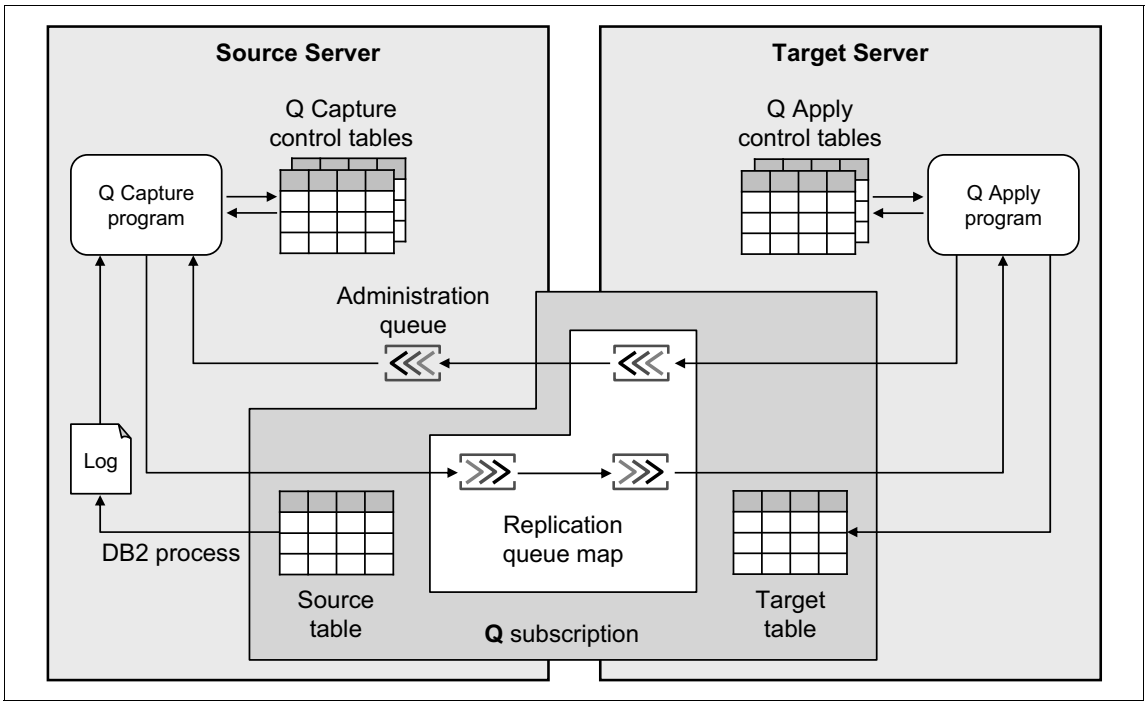

Figure 2-4 Q replication for the DB2 pureScale Feature

This solution has an RPO of greater than 0 seconds.

For more information about implementing Q replication as a disaster recovery solution for the DB2 pureScale Feature, see the following website:

http://www.ibm.com/developerworks/data/bestpractices/purescaleqreplication/index.html

Disaster recovery using IBM InfoSphere Change Data Capture

IBM InfoSphere Change Data Capture integrates information for heterogeneous data stores in real time. Change Data Capture uses real-time data replication to support data migrations, application consolidation, data synchronization, dynamic warehousing, MDM, service-oriented architecture (SOA), Business Analytics, and Extract, Transform, and Load (ETL) or data quality processes. It works through a log-based CDC technology that transfers from publisher to subscriber systems and replicates mission-critical data events in real time without impacting system performance. Rather than performing queries directly against the database, it is possible to capture changed data directly from the database logs.

Flexible implementation enables unidirectional, bidirectional, many-to-one, and one-to-many delivery of data across the enterprise.

Change Data Capture supports both AIX and Linux, along with the DB2 pureScale Feature as either a source or a target. There are various monitoring capabilities and a graphical user interface (GUI) available to provide increased understanding of the replication environment to accelerate troubleshooting.

This solution has an RPO of greater than 0 seconds.

Disaster recovery using storage replication

Storage replication is the ability of the storage subsystem to replicate the disk writes over fibre to another storage subsystem that might be either local or remote. For disaster recovery, this process provides an exact copy of the database at a remote site. If there is a primary site failure, the disaster recovery site breaks the storage replication link with the primary site, and the DB2 pureScale cluster is started on the disaster recovery site. The database undergoes crash recovery, and transactions continue as though nothing happened to the database.

At the time of the writing of this book, this solution is the only available solution with the DB2 pureScale Feature that offers 0 second RPO.

2.7 Client requirements and considerations

DB2 handles access to applications that are developed using the IBM data server client. These applications include embedded SQL, calls to the CLI, ODBC applications, Java applications using the JDBC or SQLJ interfaces, PHP applications, Ruby or Ruby on Rails applications, Perl applications, and Python applications. For client connections, the DB2 pureScale Feature supports both workload balancing and client affinity.

The following sections describe considerations for the application and important configuration parameters to ensure that the optimal parameters are used for communication between the DB2 pureScale Feature and the application.

2.7.1 Application considerations

It is important to understand how an application behaves with the DB2 pureScale database and whether additional options are required for your needs. Running an application against a DB2 pureScale database requires the following setup:

1. Ensure that the server is configured and running.
2. On the DB2 server, ensure that the database manager is started on the database server to which the application program is connected.

   If the database manager is not started, run **db2start** at the server before starting the application.
3. Ensure that you can connect to the database that the application uses.
4. Bind the necessary files to support the database application driver being used.
5. Run the application program.

Before running the application and connecting to DB2, consider the following additional items that might be relevant according to the type of application:

- Java based applications (JDBC, JNI, JCC, and so on)

  Considerations regarding the property files, properties on the connection object, and URLs, as described in 2.7.2, “DB2 client configurations for an application” on page 52
- Non Java based applications (CLI, perl, php, ruby, C/C++, ODBC, and so on)

  Considerations that need to happen regarding configuration files for DB2 on the client side to ensure that the application is properly communicating with the application

The next sections cover some of the configuration parameters that can help ensure that clients are communicating with DB2 correctly and that all parameters are set to the optimal values.

2.7.2 DB2 client configurations for an application

Several client configuration parameters are required to ensure that a DB2 pureScale cluster is configured properly for connecting with an application. Understanding these configuration parameters can elevate issues when ensuring that the database and the application can connect to one another and transfer data quickly and correctly.

DB2 server configurations

Most configurations depend on the type of application that you have. (For more information, see “Additional configurations for Java based applications” on page 57 and “Additional configurations for non Java based applications” on page 58.) However, you can still modify various configuration parameters on the DB2 pureScale Feature level. Setting these configurations can help ensure better optimized application communication with the DB2 pureScale Feature.

The `db2dsdriver.cfg` configuration file is an XML document that contains database directory information and client configuration parameters in a human-readable format. Using various keywords, the `db2dsdriver.cfg` configuration file sets values to enable various features for ODBC, CLI, .NET, OLE DB, PHP, or Ruby applications. The keywords can be associated globally for all database connections.

You do not have to create or populate the `db2dsdriver.cfg` file, but by doing so, you can avoid the need to specify information about the database name, hosts, ports, aliases, and other configuration parameters in the client applications.

Configuration file note: The `db2dsdriver.cfg` configuration file supports a consistent set of XML tags that are in lowercase and that do not include underscore characters (_).

The `db2dsdriver.cfg` configuration file includes the following structures that you can use to define a certain keyword to be either global or only for a specific set of configurations:

- `<dsncollection>`

  The data source name section is contained within the `<dsncollection>` and `</dsncollection>` tags. Parameters in this section apply only to a given data source name.

- `<databases>`

  The database information section is contained within the `<databases>` and `</databases>` tags. Parameters in this section apply only to a given database connection.

These two subsections can be defined under the database information section to enable the following high availability features:

- `<wlb>`

  The workload balancing subsection is contained within the `<wlb>` and `</wlb>` tags. Workload-balance related parameters are placed in this section, and they apply to a specific database connection.

- `<acr>`

  The automatic client reroute (ACR) subsection is contained within the `<acr>` and `</acr>` tags. Automatic client reroute related parameters are placed in this section, and they apply to a specific database connection.

- `<parameters>`

  The global attributes section is contained within the `<parameters>` and `</parameters>` tags. Parameters in this section apply to all databases and aliases.

- `<ldapserver>`

  For Lightweight Directory Access Protocol (LDAP) support in CLP Plus, the LDAP section is contained within the `<ldapserver>` and `</ldapserver>` tags and can be used to specify LDAP server information.

For example, the `db2dsdriver.cfg` file can look as shown in Figure 2-5.

```
<configuration>
        <dsncollection>
               <dsn alias="alias1" name="name1" host="server1.net1.com" port="50001"/>
               <!-- Long aliases are supported -->
               <dsn alias="longaliasname2" name="name2" host="server2.net1.com" port="55551"/>
                      <parameter name="Authentication" value="Client"/>
               </dsn>
        >/dsncollection>
        <databases>
               <database name="name1" host=server1.net1.com" port="50001">
                      <parameter name="CurrentSchema" value="OWNER1"/>
                      <wlb>
                              <parameter name="enableWLB" value="true"/>
                              <parameter name="maxTransports" value="50"/>
                      </wlb>
                      <acr>
                              <parameter name"enableACR" value="true"/>
                      </acr>
               </database>
               <!-- Local IPC connection -->
               <database name="name3" host="localhost" port="0">
                      <parameter name="IPCInstance" value="DB2"/>
                      <parameter name="CommProtocol" value="IPC"/>
               </database>
          </databases>
          <parameters>
               <parameter name="GlobalParam" value="Value"/>
          </parameters>
</configuration>
```

Figure 2-5 db2dsdriver.cfg file example

The `db2dsdriver.cfg` file specifies that the `<dnscollection>` tags and the databases and list of servers can handle the requests.

Additional Linux system configurations

Table 2-11 lists specific communication parameters that must be set at the OS level for Linux systems on the client side. These parameters can help improve the overall time that it takes for a member to recover after a failure.

Table 2-11 Additional client configurations for a Linux system

Kernel variable name	Description	Recommended value
`tcp_keepalive_probes`	This kernel variable controls the number of probes that are sent and unacknowledged before the client considers the connection broken and notifies the application layer. Adjusting this value can help with the overall recovery of failure.	10
`tcp_keepalive_time`	This kernel variable controls the interval between the last data packet sent and the first keepalive probe. Adjusting this value can help with the amount of time a connection stays alive to minimize the amount of time that it takes for recovery.	6
`tcp_keepalive_intvl`	This kernel variable controls the interval between subsequent keepalive probes. Adjusting this value can help the amount of time that a connection is alive and can aid during recovery.	1
`tcp_retries2`	This kernel variable controls the maximum number of times a packet is retransmitted before giving up	3

The parameters described in Table 2-11 are all at the kernel level of the OS. To set these parameters, run the following commands:

- **`sysctl -w net.ipv4.tcp_keepalive_probes=10; echo “net.ipv4.tcp_keepalive_probes=10” >> /etc/sysctl.conf;`**
- **`sysctl -w net.ipv4.tcp_keepalive_time=6; echo “net.ipv4.tcp_keepalive_time=6” >> /etc/sysctl.conf;`**
- **`sysctl -w net.ipv4.tcp_keepalive_intvl=1; echo “net.ipv4.tcp_keepalive_intvl=1” >> /etc/sysctl.conf`**
- **`sysctl -w net.ipv4.tcp_retries2=3; echo “tcp_retries2” >> /etc/sysctl.conf`**

You can verify each of these variables using a simple **`cat`** of the OS file that corresponds with these variables. Notice that each of these variables is associated with the same directory structure, allowing for the following verification:

```
cat /proc/sys/net/ipv4/tcp_retries2
```

If you notice that the value corresponds with these values, you can be sure that the parameters are set.

> **Overwriting these communication values**: These communication values are updated by the OS during each boot through the `sysctl.conf` file. Overwriting the contents of the file does not make the values stay there permanently. You must set the values using the **`sysctl`** command.

Updating these configurations results in a quicker response time for TCP/IP response at the client level, allowing for a faster reconnect.

Additional AIX system configurations

Table 2-12 lists communication parameters that must be set at the OS level for AIX systems on the client side that can help improve the overall time that it takes for a member to recover after a failure.

Table 2-12 Additional client configurations for an AIX system

Kernel variable name	Description	Recommended value
`tcp_keepidle`	This kernel variable controls the length of time to keep an idle TCP connection active. Adjusting this value can help reduce the amount of time active connections are held, resulting in a faster overall recovery after a member failure.	12
`tcp_keepintvl`	This kernel variable controls the interval between packets sent to validate the TCP connection. Adjusting this value reduces the interval that the packets are sent, thus reducing the overall amount of time until detecting that a member failed.	2
`tcp_keepcnt`	This kernel variable controls the number of keepalive probes to be sent before terminating the connection. Adjusting this value can help the amount of time a connection stays alive because the number of probes are reduced until it is deemed dead, aiding in overall member recovery.	10

These parameters are all at the kernel level of the OS. To set these parameters, run the following commands:

- **`no -o tcp_keepidle=12;`**
- **`no -o tcp_keepintvl=2;`**
- **`no -o tcp_keepcnt=10;`**

The `tcp_keepidle` time and `tcp_keepintl` time are both measured in half-seconds.

Additional configurations for Java based applications

Table 2-13 lists additional configuration settings that should be set when using Java applications, to get the fastest possible recovery times. Because communication of availability is the most critical aspect to a DB2 pureScale Feature setup, set these values to achieve the best results.

Table 2-13 Java configurations for the client side

IBM Data Server Driver for JDBC and SQLJ setting	Value
`enableClientAffinitiesList`	`DB2BaseDataSource.YES (1)`
`clientRerouteAlternateServerName`	A comma-separated list of the primary server and alternative servers.
`clientRerouteAlternatePortNumber`	A comma-separated list of the port numbers for the primary server and alternative servers.
`enableSeamlessFailover`	`DB2BaseDataSource.YES (1)` for seamless failover; `DB2BaseDataSource.NO (2)` or `enableSeamlessFailover` not specified for no seamless failover.
`maxRetriesForClientReroute`	The number of times to try the connection to each server, including the primary server, after a connection to the primary server fails. The default is `3`.
`retryIntervalForClientReroute`	The number of seconds to wait between tries. The default is `no wait`.
`affinityFailbackInterval`	The number of seconds to wait after the first transaction boundary to fail back to the primary server. Set this value if you want to fail back to the primary server.

For multithreaded Java applications, you can use `enableSysplexWLB=true` to take advantage of transaction level workload balancing. Also, it is good to avoid any blocks on reads if a member fails. Keeping `blockingReadConnectionTimeout` at the default value of 0 ensures that there are no timeouts.

Client affinity is optional and can be use to connect to certain servers, but it is also possible with Java applications. There are several configuration parameters that must be modified to ensure that this task happens properly. They are described in Table 2-13 and are defined within the property settings of the Java driver.

When the first SQL statement in a transaction runs, if the IBM Data Server Driver for JDBC and SQLJ receives a communication failure because the data server drops the connection or the `blockingReadConnectionTimeout` value is exceeded, the driver tries the SQL statement a number of times before reporting an error. On every try, the driver closes the existing transport, obtains a new transport, and then runs the transaction. During these tries, if the `maxRetriesForClientReroute` and `retryIntervalForClientReroute` properties are set, their values apply only to the process of obtaining a new transport during each try.

You can further optimize a Java application using query optimizations to limit the amount of time it takes for queries to complete. However, this description is beyond the scope of this book.

Additional configurations for non Java based applications

To allow for a quicker response time, you can modify the `DB2TCP_CLIENT_KEEPALIVE_TIMEOUT` configuration parameter at the DB2 level of the client that is connecting to the DB2 database. Doing so can improve recovery of a failed member. By default, the value of this parameter is 0 (that is, it is not set). You can set the value to any value 0 - 32,767 seconds. Keep the value at around 10 for quicker response time.

Client affinity, which is the ability to connect to certain servers, is also possible with non Java applications (CLI and .NET applications). You can modify several configuration parameters to ensure that client affinity happens properly, as listed in Table 2-14. These parameters are defined within the property settings of the `db2dsdriver.cfg` file.

Table 2-14 Non Java configurations for the client side

Element in acr section of the db2dsdriver.cfg file	Value
`enableAcr` parameter	`true`.
`maxAcrRetries` parameter	The number of times that a connection to each server in the list of alternative servers is tried during automatic client reroute. The valid range is 0 - MAX_INT. If the value is 0, the number of tries is 1. The default is 3.
`acrRetryInterval` parameter	The number of seconds to wait between tries. The valid range is 0 - MAX_INT. The default is `no wait (0)`.
affinityFailbackInterval parameter	The number of seconds to wait after the first transaction boundary to fail back to the primary server. The default is `0`, which means that no attempt is made to fail back to the primary server.

Element in acr section of the db2dsdriver.cfg file	Value
alternateserverlist	<server> elements that identify the host name and port number for each server that is used for automatic client reroute through client affinities. One of the elements must identify the primary server. The presence of these elements does not activate automatic client rerouting.
affinitylist	<list> elements with serverorder attributes. The serverorder attribute value specifies a list of servers, in the order that they should be tried during automatic client rerouting with client affinities. The servers in <list> elements must also be defined in <server> elements in <alternateserverlist>. You can specify multiple <list> elements, each of which has different server orders. The presence of the <affinitylist> element does not activate automatic client rerouting.
client_affinity	A <clientaffinitydefined> element or a <clientaffinityroundrobin> element that defines the order in which to try server connections for each client. When you include a <clientaffinitydefined> element, you define the server order by defining <client> elements, each of which specifies a <list> element that defines the server order. When you include a <clientaffinityroundrobin> element, you also specify <client> elements, but those <client> elements do not specify a <list> element. Instead, the order of the <client> elements defines the server order. All clients that connect to a database must be specified within a <clientaffinitydefined> or a <clientaffinityroundrobin> element.
clientaffinitydefined	<client> elements that define the server order for automatic client rerouting for each client. Each <client> element contains a listname attribute that associates a client with a <list> element from the <affinitylist> element.
clientaffinityroundrobin	<client> elements whose order in the <clientaffinityroundrobin> element defines the first server that is chosen for automatic client reroute. Each <client> element has an index. The first <client> element in the <clientaffinityroundrobin> element has index 0, the second <client> element has index 1, and so on. Suppose that the number of servers in the <alternateserverlist> element is *n* and the index in the <clientaffinityroundrobin> element of a <client> element is *i*. The first server to be tried is the server whose index in the <alternateserverlist> element is $i \bmod n$. The next server to be tried is the server whose index in the <alternateserverlist> element is $(i + 1) \bmod n$), and so on.

There are various other parameters and configurations that can help optimize non Java applications, such as setting DB2 to use query optimization to further enhance and optimize queries and to limit the amount of time that queries take. This description, however, is beyond the scope of this book.

3

Implementation and setup

This chapter provides validated, step-by-step hardware and software installation instructions for setting up and configuring DB2 10.1 with the DB2 pureScale Feature on both Linux and AIX environments. Instructions include how to set up the hardware, install the required software on the operating system, and install the DB2 pureScale Feature.

The instructions guide you through a setup of a DB2 pureScale cluster environment that contains two members and two cluster caching facilities (CFs). Additionally, this chapter includes recommended post-installation tasks for a production environment, such as how to separate the log and data files for better I/O throughput and how to enable SCSI-3 PR.

Images in this chapter: The images in this chapter are for setup and configuration on a SUSE Linux Enterprise Server 11 system and Red Hat Enterprise Linux systems. The exact steps and images for your system might vary if you use different hardware or software versions.

For more details about hardware and software requirements, visit the IBM DB2 10.1 Information Center at:

http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/index.jsp

This chapter is divided into sections for each operating system and provides a full flow from start to finish for each operating system. If you are interested in the AIX installation on POWER machines, go to 3.3, “Configuring an AIX system to install the DB2 pureScale Feature” on page 150. Otherwise, begin with the Linux system hardware setup on IBM System x.

This chapter includes the following topics:

- Hardware setup on a Linux operating systems
- Configuring a Linux system to install the DB2 pureScale Feature
- Configuring an AIX system to install the DB2 pureScale Feature
- Installing the DB2 pureScale Feature
- Post-installation tasks

3.1 Hardware setup on a Linux operating systems

This section describes how to set up the hardware to install DB2 10.1 with the DB2 pureScale Feature in a Linux environment. The instructions include how to add PCI Express cards (the Fibre Channel adapter, Mellanox InfiniBand HCA adapter, and 10 Gigabit Ethernet (GbE) adapter) in the server and how to cable the servers for storage, for the cluster interconnect, and for Ethernet. On the Linux environment, the steps set up two servers with one DB2 member and one cluster caching facility (CF) on each server.

This setup uses a Fibre Channel connection between servers and external storage, so a Fibre Channel card is required on each server. The DB2 pureScale Feature requires a high speed, low-latency interconnect network. If you use SUSE Linux Enterprise Server or Red Hat Enterprise Linux 6.1, you can choose between an InfiniBand network or a 10 GbE network; however, if you use Red Hat Enterprise Linux 5, you can use only an InfiniBand network.

Follow the instructions in this section to install the Fibre Channel card and either an InfiniBand HCA adapter or a 10 GbE network adapter in each server.

Figure 3-1 shows the overall Linux environment architecture of the DB2 pureScale cluster used in this section.

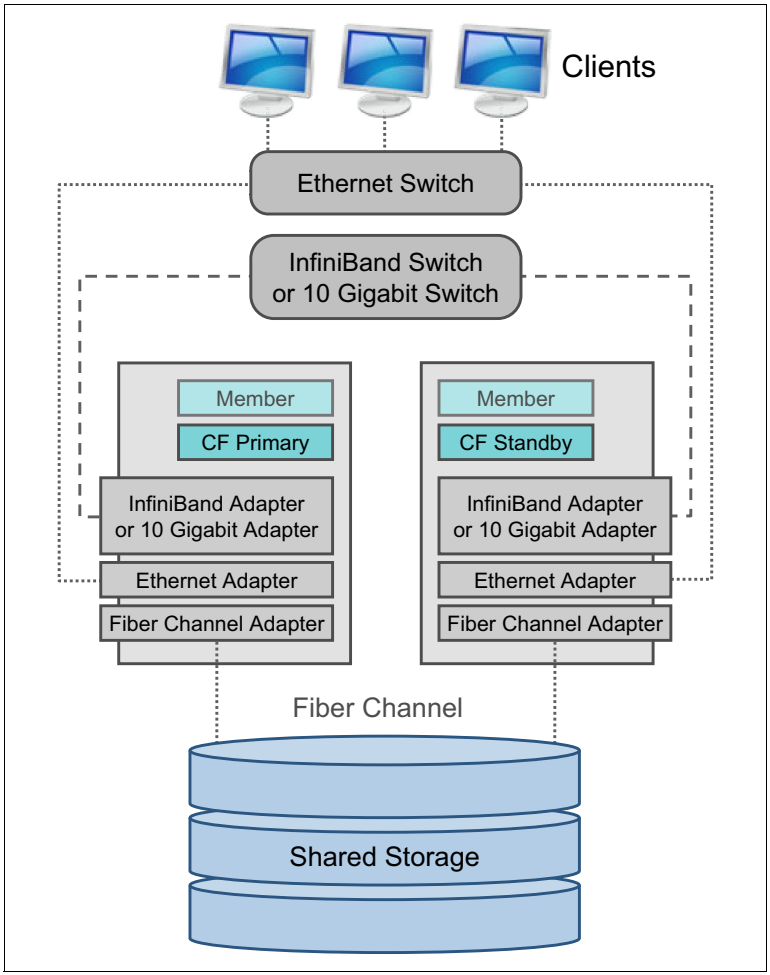

Figure 3-1 DB2 pureScale cluster hardware architecture in a Linux environment

3.1.1 Adding a PCI Express card

To connect IBM System x servers with other devices (storage, InfiniBand, or a 10 GbE switch), you need to add certain adapters, such as a Fibre Channel adapter, an InfiniBand HCA adapter, or a 10 GbE adapter.

Attention: Follow any safety instructions that might come with your server and rack before continuing.

To add any of these PCI Express cards in the server, complete the following steps:

1. Power off the server, and ensure that the power cable is unplugged.

   Attention: If the server is installed in a rack, slide the server out of the rack enclosure first!

2. Open the server chassis by pressing in on the blue tab on the cover-release latch, and slide the cover towards rear to remove the server cover (Figure 3-2).

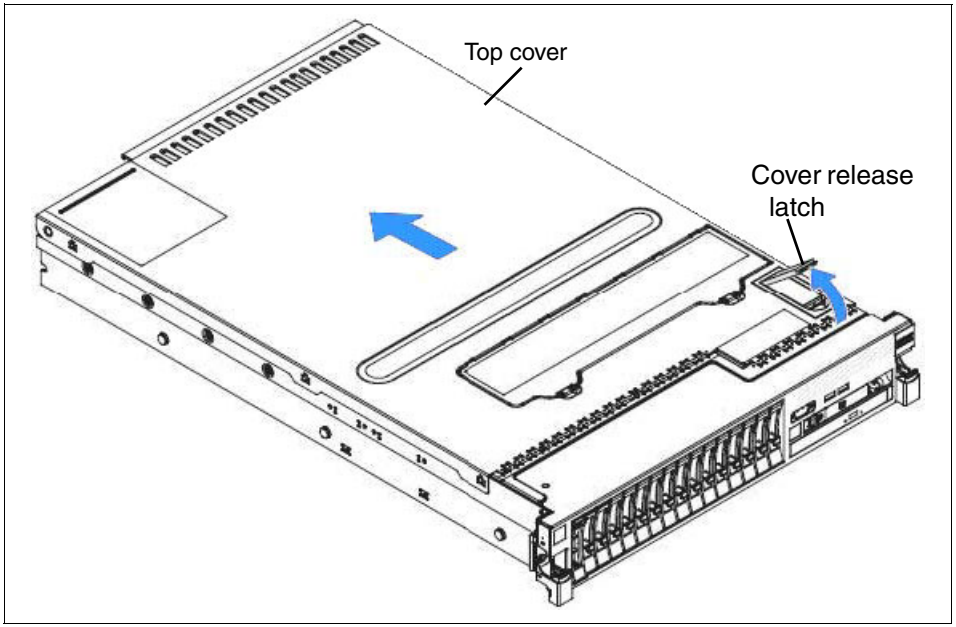

Figure 3-2 Remove the cover from the server

3. Grasp the front and rear of the PCI riser-card assembly and lift it out of the PCI riser-card assembly slot on the system board (Figure 3-3).

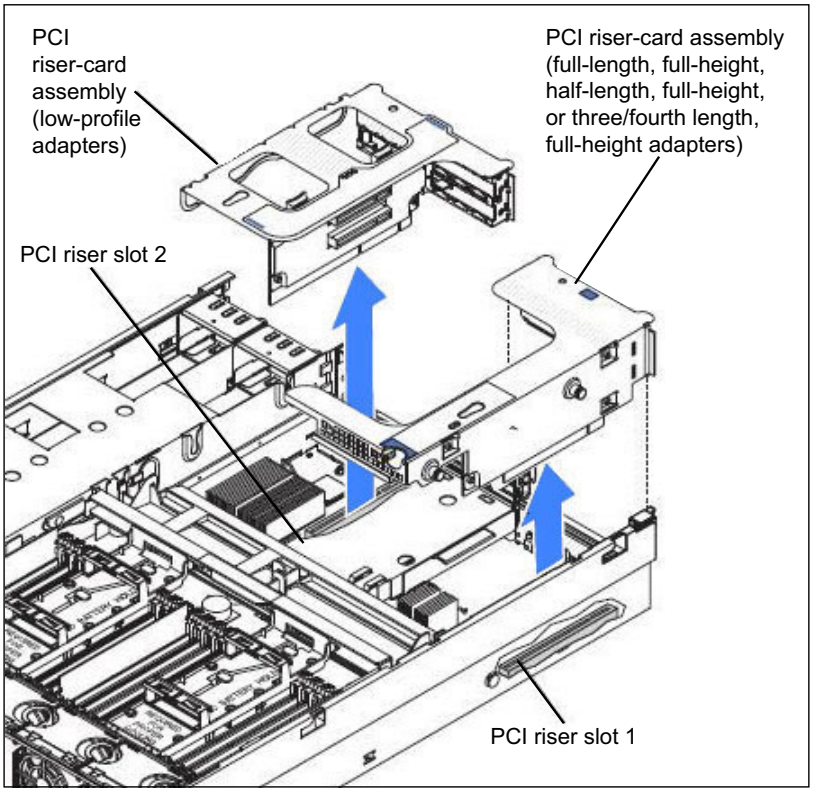

Figure 3-3 Remove the PCI riser card from the server

4. Insert the PCI Express card into the riser-card assembly, aligning the edge connector on the adapter with the connector on the riser-card assembly. Press the edge of the connector firmly into the riser-card assembly (Figure 3-4).

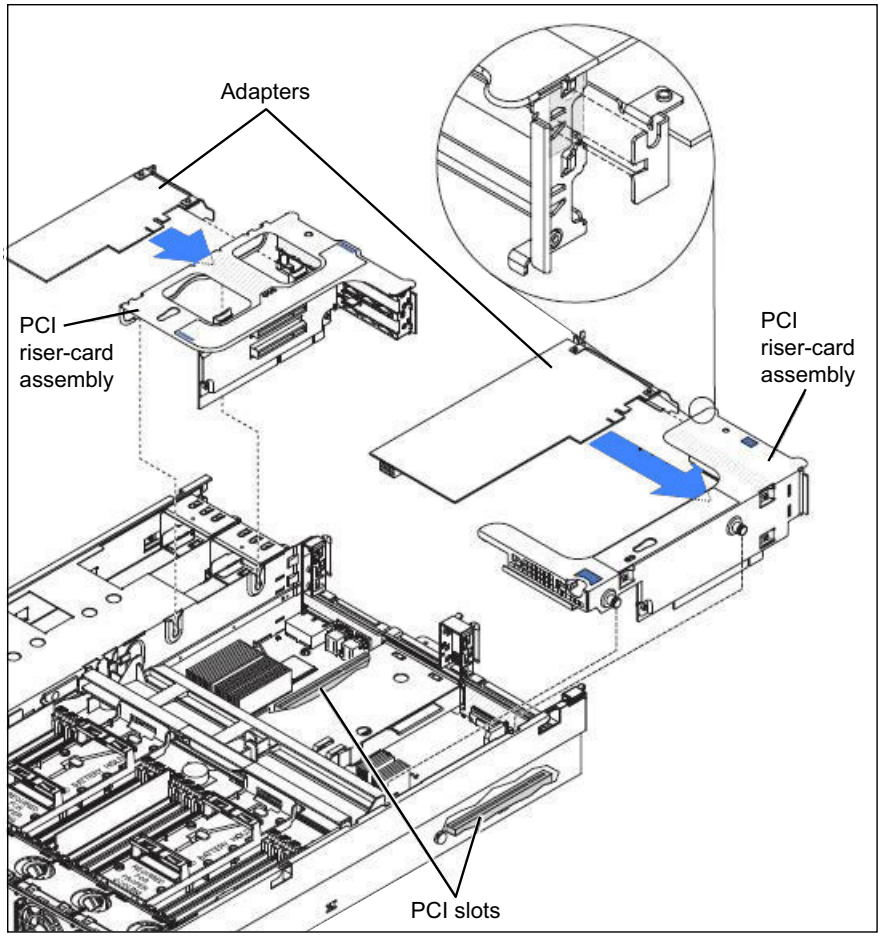

Figure 3-4 Insert the PCI Express cards into the riser card

Important: Ensure that the PCI Express card is correctly inserted into the riser-card assembly.

5. Align the PCI riser-card assembly with the PCI slot connector on the system board by aligning the nailheads with the slots on the chassis. Then press down firmly until the PCI riser-card assembly is seated correctly in the connector on the system board (Figure 3-5).

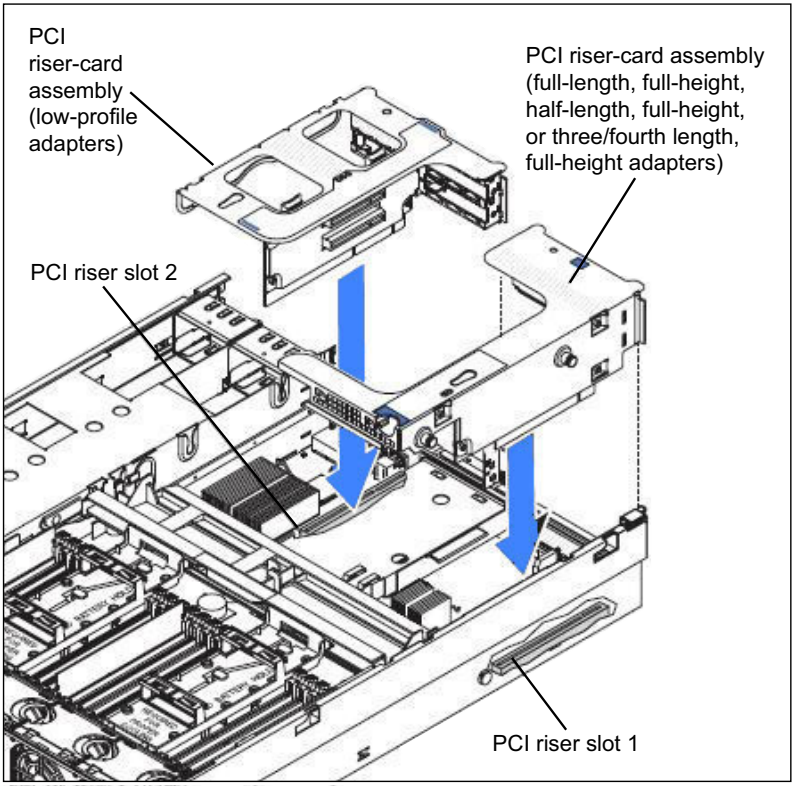

Figure 3-5 Insert the PCI riser-card back to the server

Important: Ensure that the riser-card assembly is securely inserted in the riser-card connector on the system board before powering on the server.

6. Ensure that the PCI Express card and its assembly are seated correctly and that there are no loose parts or tools inside the server.

7. Align the cover over the server until the cover edges slip into position over the chassis, and press down on the cover-release latch until it clicks into place (Figure 3-6).

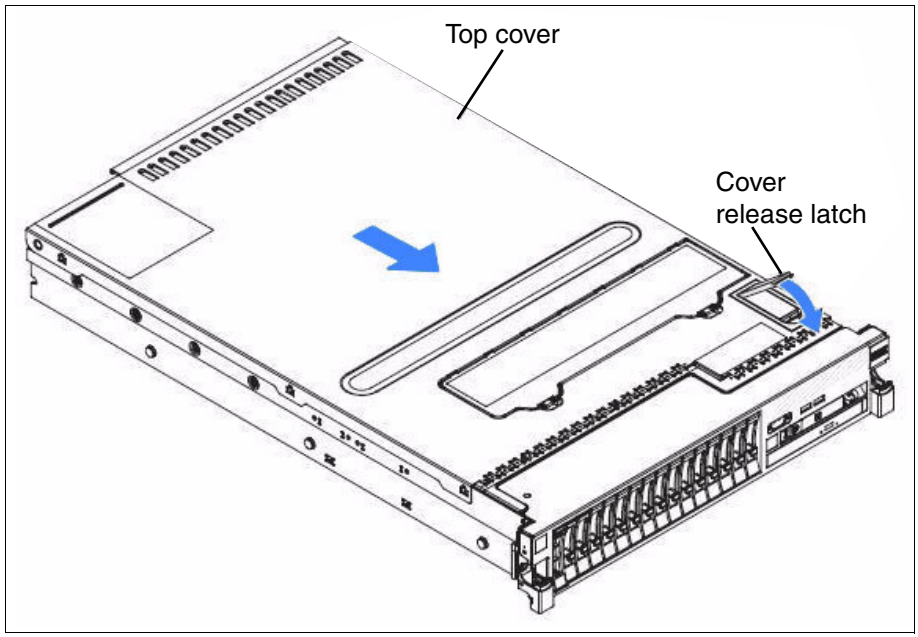

Figure 3-6 Close the cover

3.1.2 Connecting the external storage

Both servers need to connect to the IBM System Storage DS3524 dual controller storage through a Fibre Channel connection. From the rear view of the storage, you can find two controllers (controller A and controller B) with four fiber ports each. Connect both Fibre Channel ports from the adapter on the server to both controllers on the storage. Connect port 1 of the adapters to controller A and port 2 of the adapters to controller B.

Figure 3-7 shows the cabling structure.

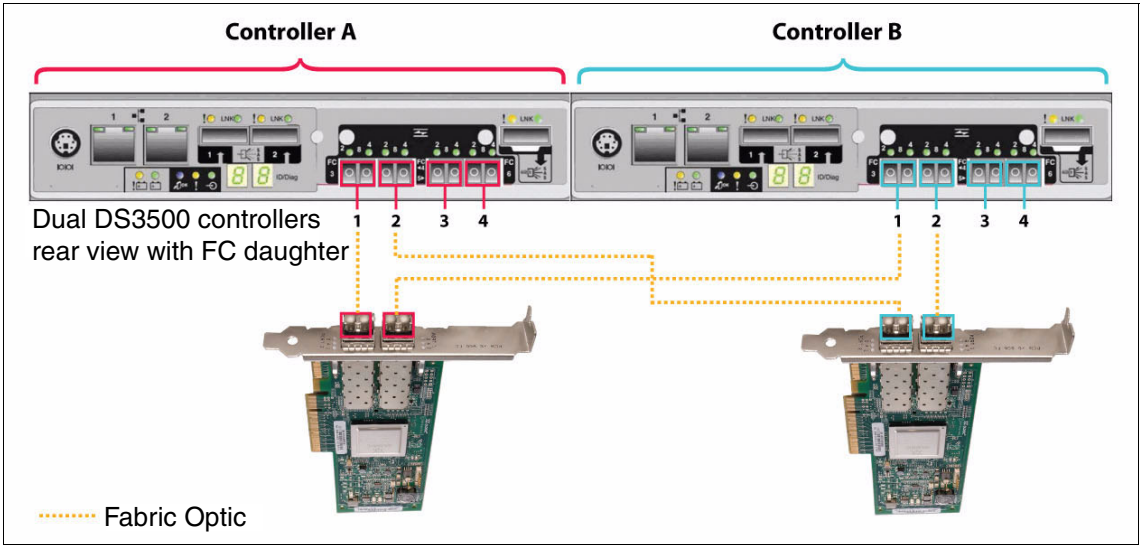

Figure 3-7 Connect a server with storage

3.1.3 Connecting the Ethernet network

The DB2 pureScale system needs an Ethernet network to communicate between servers. To connect servers with the Ethernet network, find an empty Ethernet port on the back of the server, and connect it to an empty port on the Ethernet switch with a regular RJ-45 cable.

3.1.4 Connecting an interconnect network

The DB2 pureScale system requires one high speed communication network, which must be an InfiniBand network or a 10 GbE network. This section explains how to set up the high speed communication network hardware.

InfiniBand network

Find an empty port on the InfiniBand switch and connect it to an InfiniBand port on the server with a QSPF cable. Figure 3-8 shows the InfiniBand HCA adapter that was used in our example.

> **Important:** If you use dual-port QDR InfiniBand HCA, ensure that you use *port 1* to connect to the InfiniBand switch.

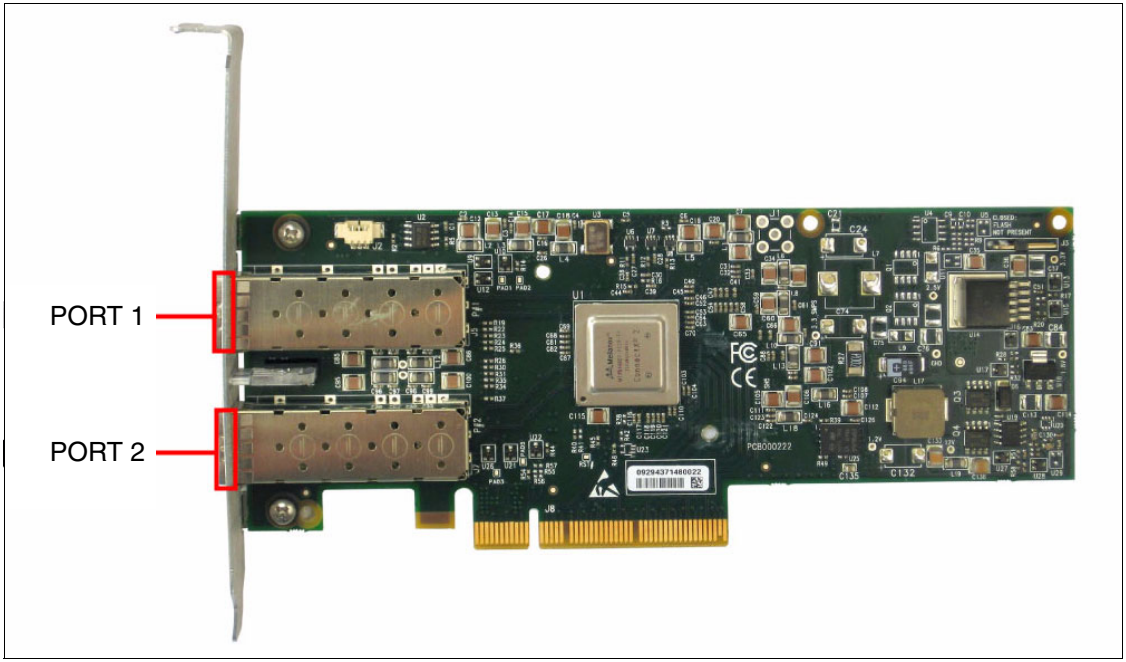

Figure 3-8 InfiniBand HCA adapter

10 GbE network

Find an empty port on the 10 GbE switch and connect it to a 10 Gb adapter on the server with an SPF+ cable.

> **Important:** If you use dual-port adapter, make sure that you use port 1 to connect to the switch.

3.1.5 Configuring the internal disks

The internal disks are used to install the operating system and software, including parts of the DB2 software. If your server does not have the internal disks already set up, you must set up the disks. You must set up the RAID level and create a driver group before installing any operating system.

This section explains how to configure the internal disks on an IBM System x3690 X5 server. The server comes with four 300 GB SAS hard disk drives and will be configured with RAID 5. The resulting virtual drive is set as the boot drive.

> **Important:** This procedure erases all data on the internal disks of the server. Ensure that you back up any necessary data to another server before starting this task.

To configure internal disks, use EFI WebBIOS, which comes with the IBM System x server, to complete the following steps:

1. Boot the server and press F1 to enter the BIOS (Figure 3-9).

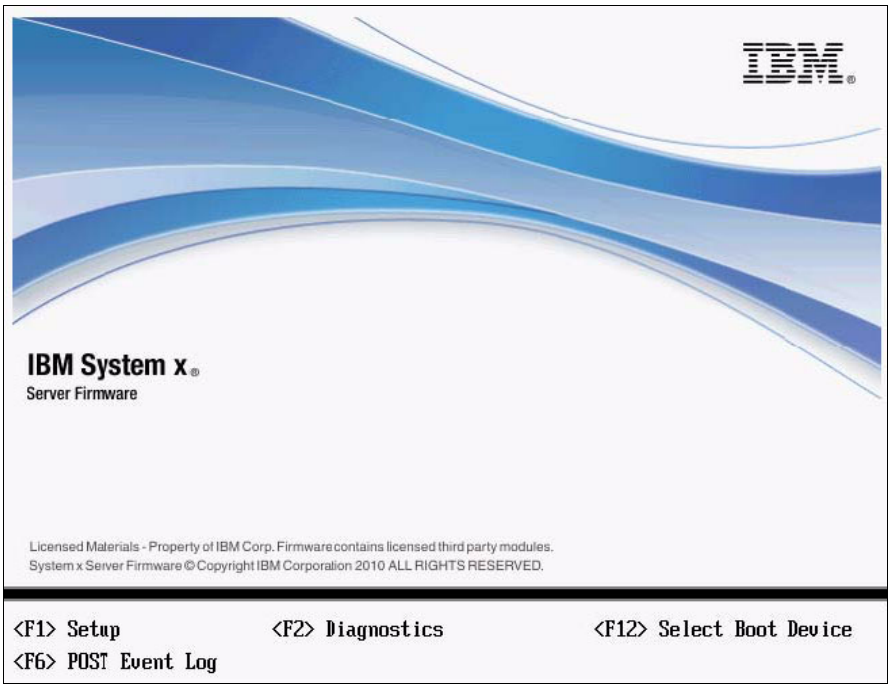

Figure 3-9 IBM System x server boot

2. Select System Settings, and then choose Adapters and UEFI Drivers.

3. On the Adapters and UEFI Drivers panel (Figure 3-10), press Enter to compile the list of drivers and navigate to the first entry under section LSI EFI SAS Driver. Press Enter to continue.

```
                          Adapters and UEFI Drivers
------------------------------------------------------------------------------

                                                      Select this option
Available Adapters and Drivers                        to configure the
                                                      specific device
  LSI EFI SAS Driver                                  managed by a driver.
  |-PciRoot (0x0)/Pci (0x1,0x0)/Pci (0x0,0x0)
  |-|-PciRoot (0x0)/Pci (0x1,0x0)/Pci (0x0,0x0/Ctrl
  (0 .   x0)
  |-|-PciRoot (0x0)/Pci (0x1,0x0)/Pci (0x0,0x0)/Ctrl
  (0 .   x1)

  QLogic Fibre Channel Driver
  |-PciRoot (0x0)/Pci (0x3,0x0)/Pci (0x0,0x0)

  Broadcom Gigabit Ethernet Driver
  |-PciRoot (0x0)/Pci (0x1C,0x0)/Pci (0x0,0x0)

------------------------------------------------------------------------------
  ↑↓=Move Highlight       <Enter>=Select Entry       Esc=Exit
------------------------------------------------------------------------------
  [!] Severe Error(s) Detected. Check POST Event Viewer.
```

Figure 3-10 Adapter and UEFI drivers

4. When prompted for EFI WebBIOS, press 1 to start it.
5. Select the correct adapter and click Start to continue.

6. Figure 3-11 is the home page for EFI WebBIOS. Click **Configuration Wizard**, select **New Configuration**, and click **Next** to start.

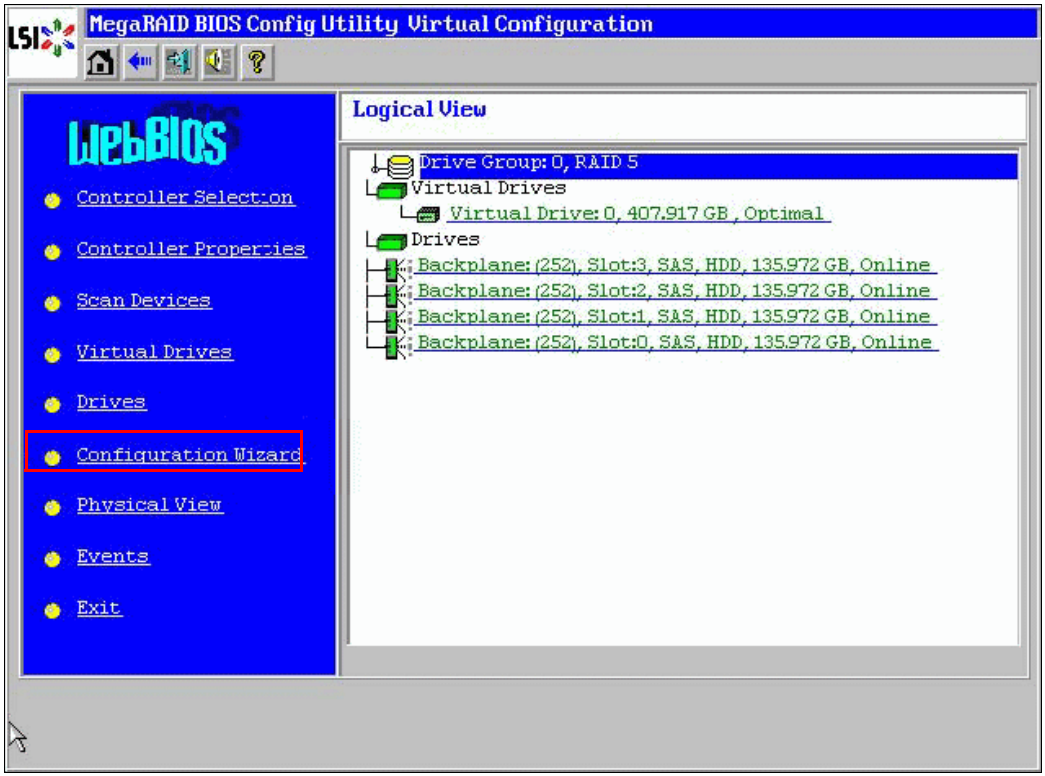

Figure 3-11 Choose the configuration wizard on WebBIOS

7. Click **Manual Configuration** (Figure 3-12) and click **Next** to continue.

> **Important:** Reconfiguring the internal disks deletes all the data on the disks. Ensure that you back up the important data before you click **Yes** to continue.

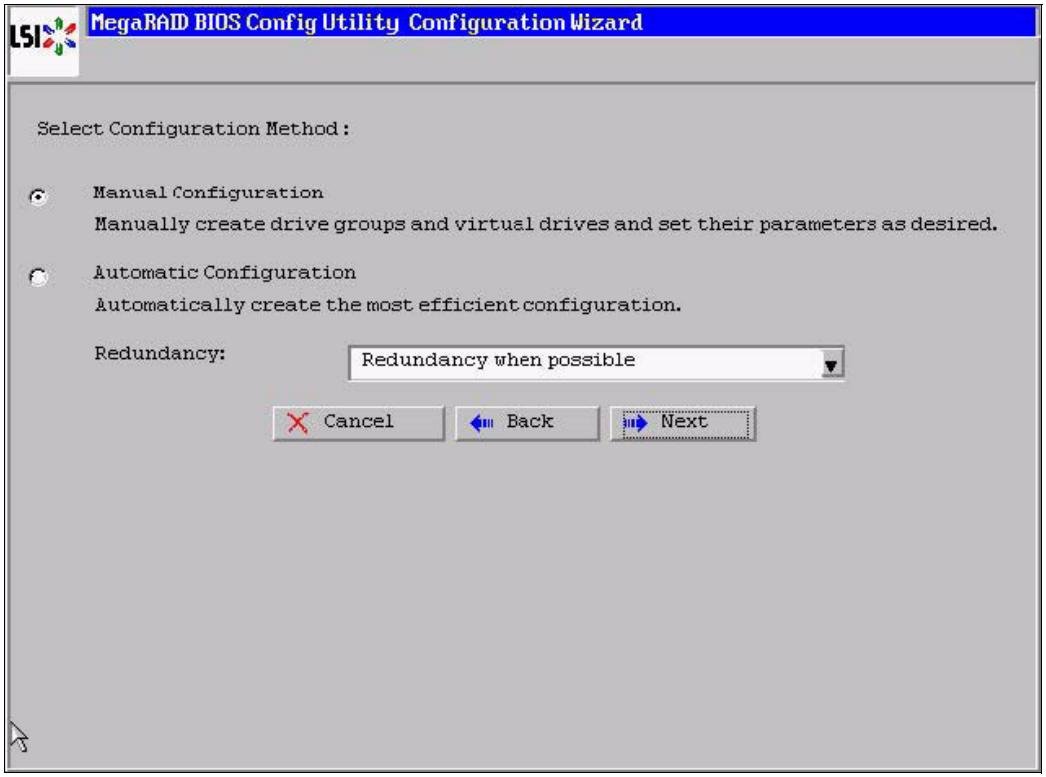

Figure 3-12 Select configuration methods

8. Next, create one drive group (DG) that contains all four internal disks. Select each of the four drives in the left panel, and click **Add to Array** to add it to the Drive Groups pane (Figure 3-13). Click **Accept DG** and then click **Next**.

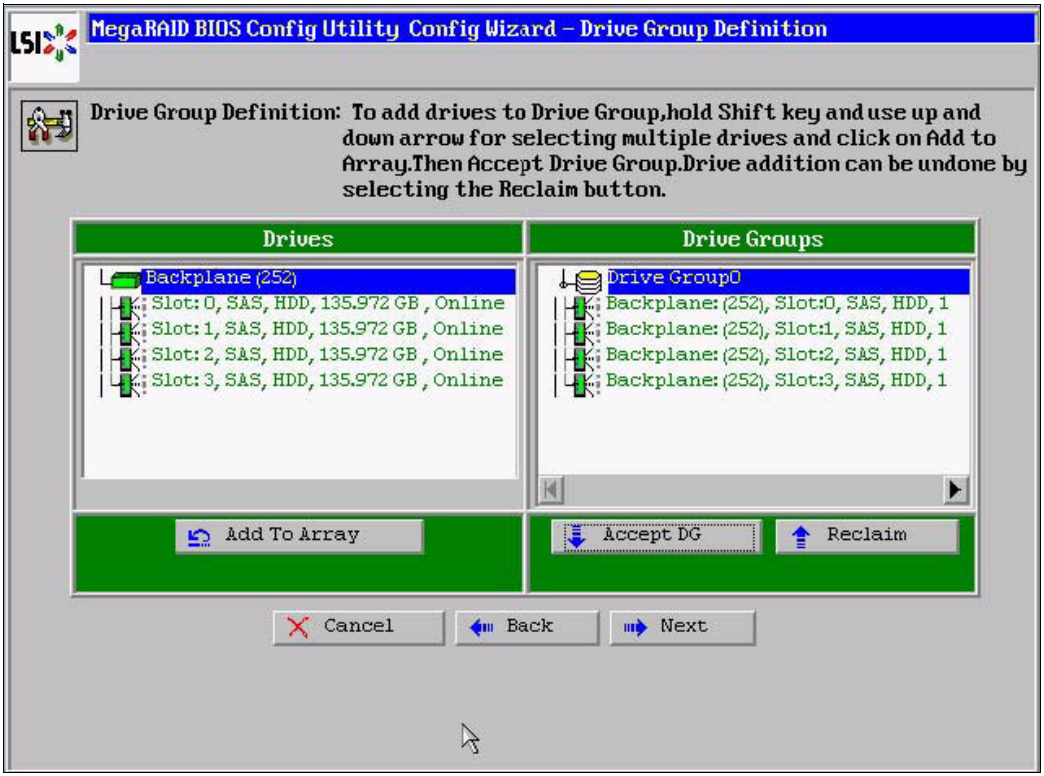

Figure 3-13 Drive Group configuration

9. Click **Add to SPAN** to add Drive Group:0 to the span (Figure 3-14), and then click **Next**.

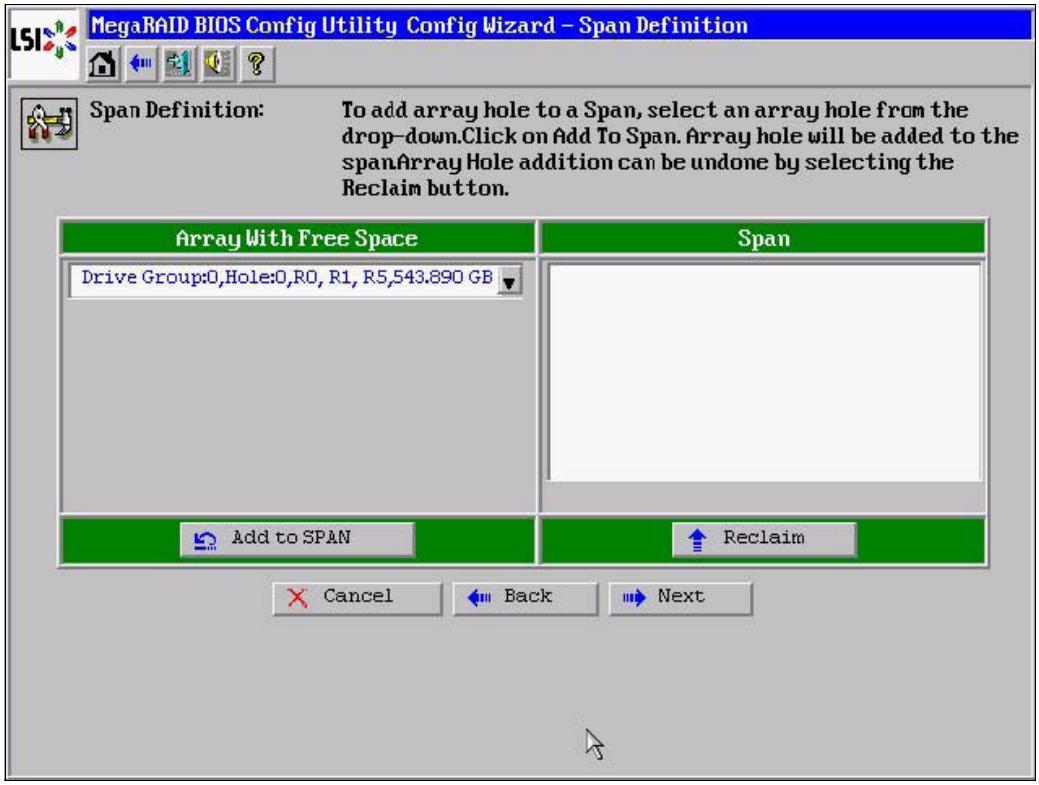

Figure 3-14 Span definition

10. Select **RAID 5** from the RAID level menu, and enter the size in the input box (Figure 3-15). The size for different RAID levels displays in the right pane. Click **Accept** and then click **Next**.

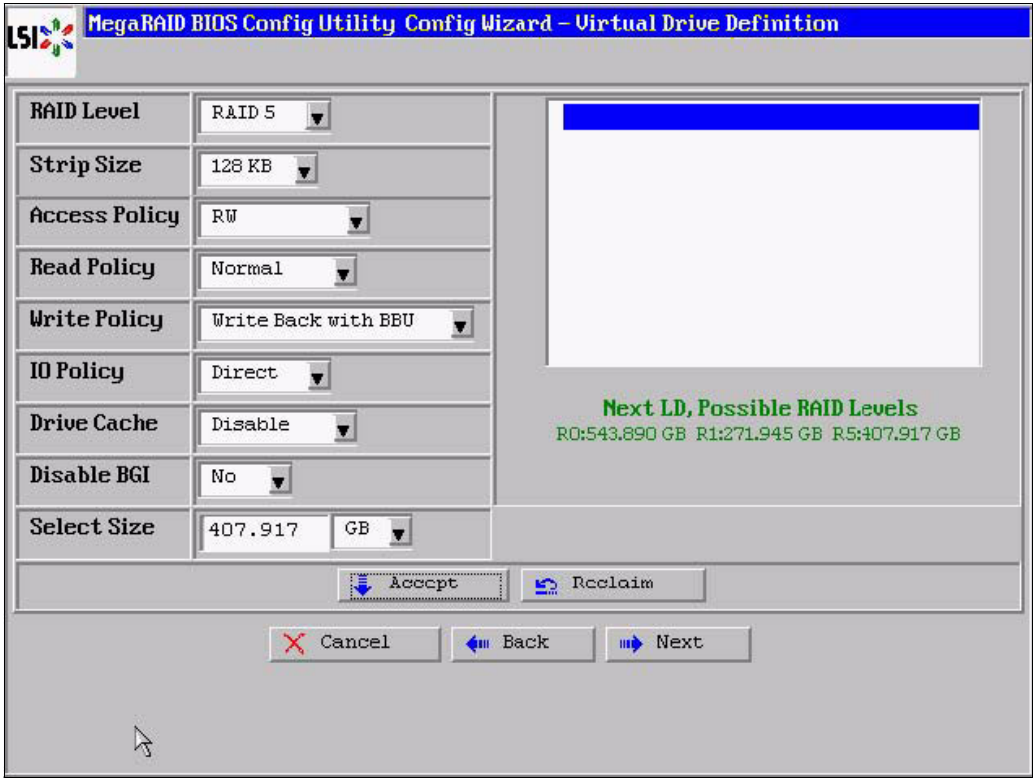

Figure 3-15 Virtual driver definition

11. Click **Accept** and **Yes** to save the configuration.
12. Click **Yes** again to start initializing the virtual drive you created.

13. After the virtual drive is initialized, select **Set Boot Drive** and click **Go** to set this virtual drive as the boot device (Figure 3-16). After the setup, the current boot device value is changed to 0, which means that the server now boots from disk 0.

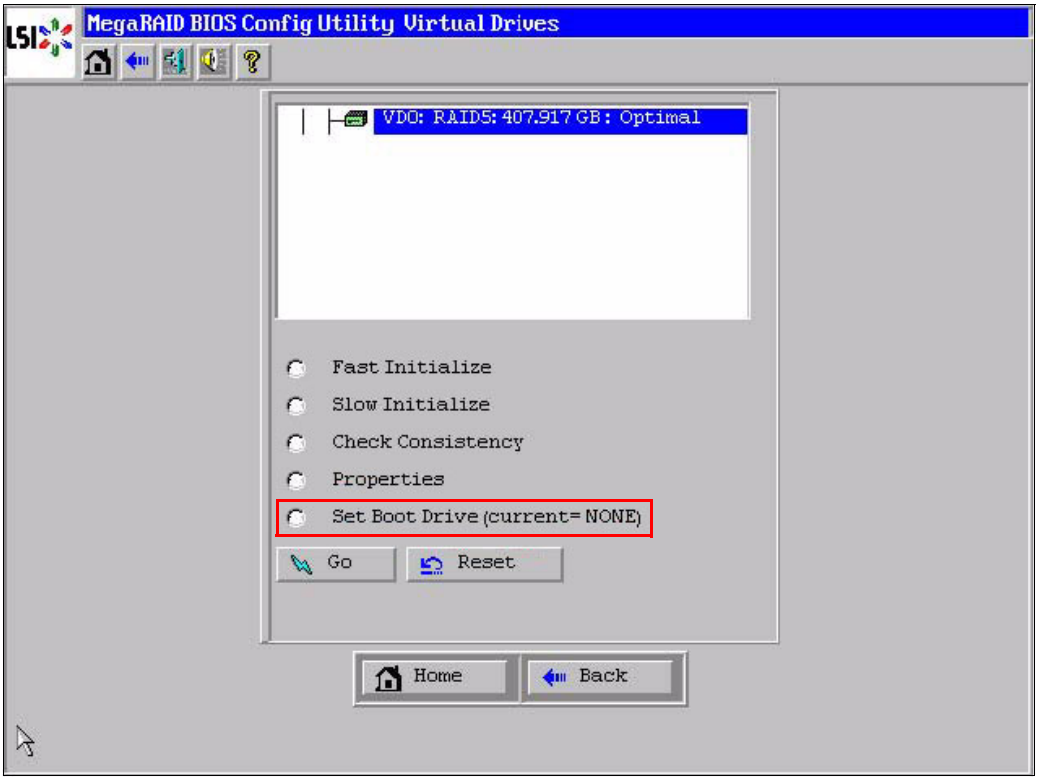

Figure 3-16 Virtual drives

14. Click **Home** to navigate back to the main menu, and you should see that there is one drive group 0 and four disks under it (Figure 3-17). Click **Exit** to exit WebBIOS. If the server requires a reboot, press Enter to do so. To exit the BIOS, press ESC and Y to confirm.

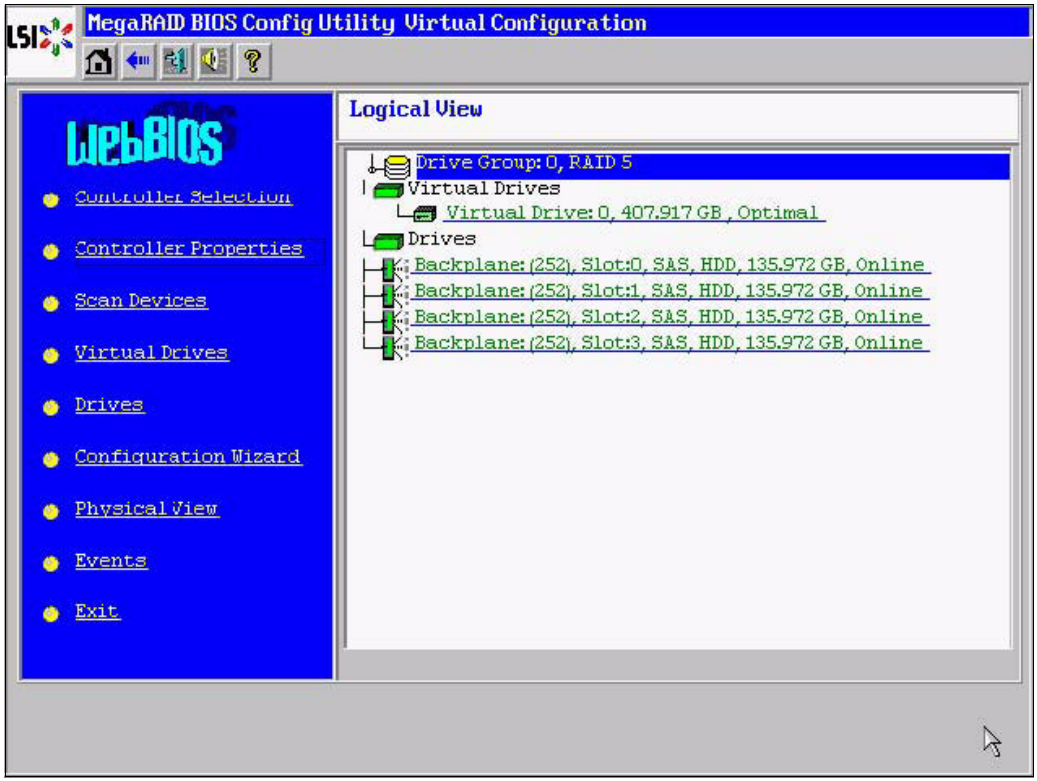

Figure 3-17 WebBIOS home page

3.2 Configuring a Linux system to install the DB2 pureScale Feature

This section contains the steps that you must complete before installing the DB2 pureScale Feature in a Linux system. These steps install the operating system, update the kernel, configure both the Ethernet and interconnect network, configure the necessary parameters, and complete other required tasks.

Each step includes instructions for SUSE Linux Enterprise Server 11, Red Hat Enterprise Linux 5, and Red Hat Enterprise Linux 6. In each step, you can choose the appropriate set of instructions for your operating system and hardware.

Before you begin: Ensure that you complete the hardware configuration steps in 3.1, “Hardware setup on a Linux operating systems” on page 62 and gather all the required software listed in 2.3, “Software requirements and considerations” on page 37.

The basic steps for this process are as follows:

- Step 1: Installing the operating system
- Step 2: Updating the kernel
- Step 3: Installing the multipath device driver
- Step 4: Configuring the external storage
- Step 5: Installing the required software packages
- Step 6: Setting up the Ethernet network
- Step 7: Setting up the interconnect network
- Step 8: Editing the hosts file
- Step 9: Installing and setting up OpenSSH
- Step 10: Blacklisting the system watchdog modules
- Step 11: Modifying the mlx4_core parameter

3.2.1 Step 1: Installing the operating system

DB2 10.1 with the DB2 pureScale Feature supports various Linux distributions. These instructions describe how to install the feature on the SUSE Linux Enterprise Server 11, Red Hat Enterprise Linux 5, and Red Hat Enterprise Linux 6 operating systems. The OpenFabrics Enterprise Distribution (OFED) and *C++ library* packages should be included in the operating system deployment as well. Choose the appropriate operating system that matches your needs.

SUSE Linux Enterprise Server 11 SP1

To install the operating system on SUSE Linux Enterprise Server 11 SP1:

1. Power on the IBM System x server and insert the SUSE Linux Enterprise Server 11 SP1 installation DVD. Pay attention to the options that display only momentarily. Press F12 to select the boot order menu (Figure 3-18).

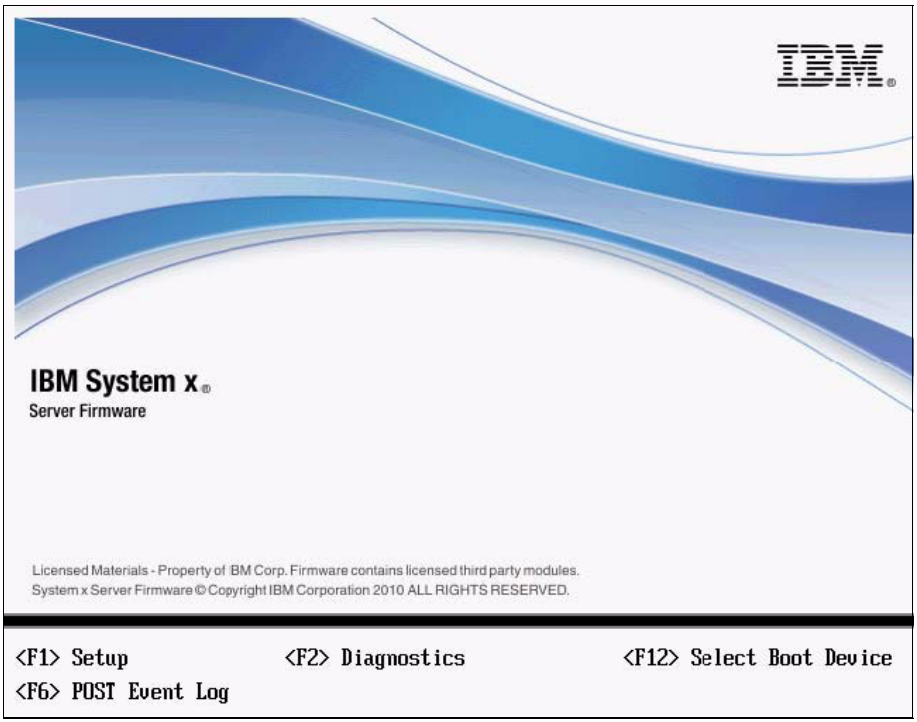

Figure 3-18 IBM System x boot

2. From the boot order menu, select CD/DVD-ROM to boot from the installation DVD.
3. Wait until the installation presents you with a welcome window. Read and then accept the license, and click **Next**.
4. You can check the installation DVD by clicking **Start Check**, or skip this section by clicking **Next**.
5. Choose **New Installation mode** and click **Next** to start.
6. Ensure that the clock and time zone are correct, and click **Next**.
7. Because you are using the real cluster, accept the default setting of **Physical Machines**, and click **Next**.

8. On the Installation Settings window (Figure 3-19), click the **Software** link to add the required packages for the DB2 pureScale Feature.

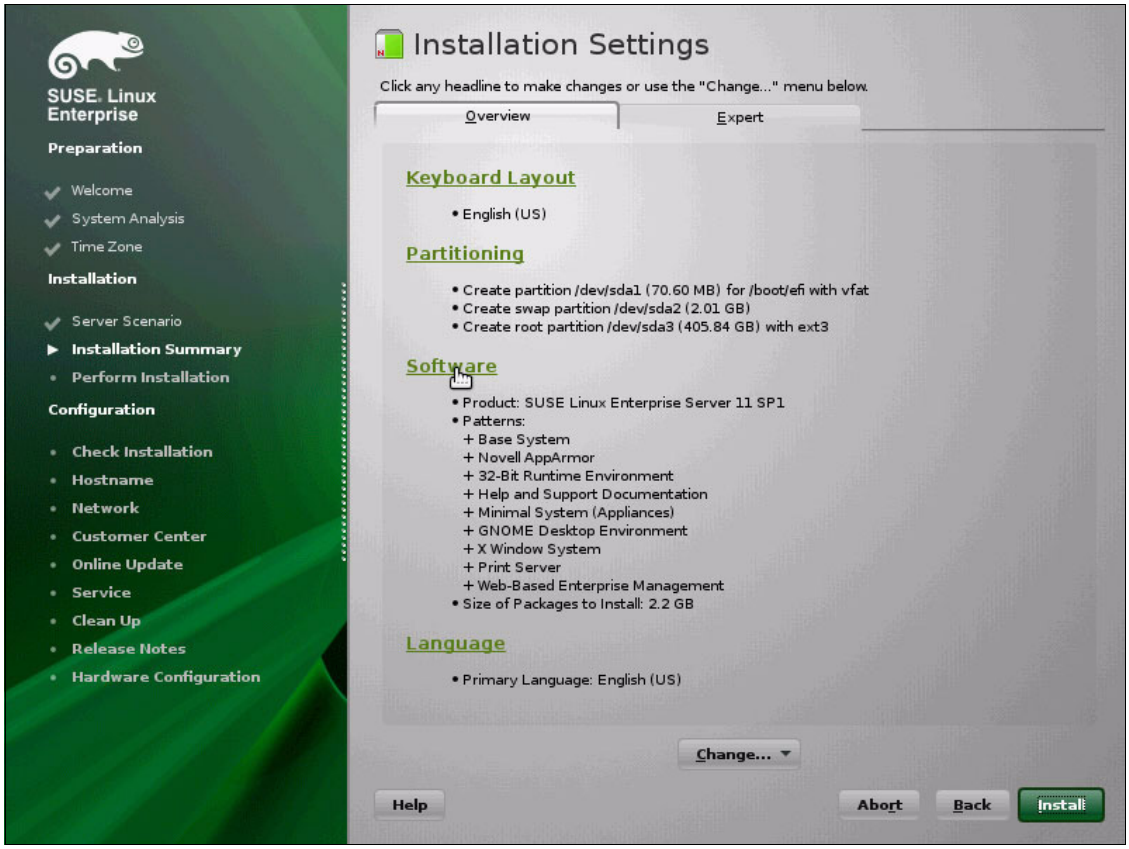

Figure 3-19 Installation Settings

9. On the Software Selection and System Tasks window (Figure 3-20), add the InfiniBand (OFED) and C/C++ Compiler and Tools packages, and then click **OK** to continue. If prompted with a license agreement, read it and then click **Accept** to continue.

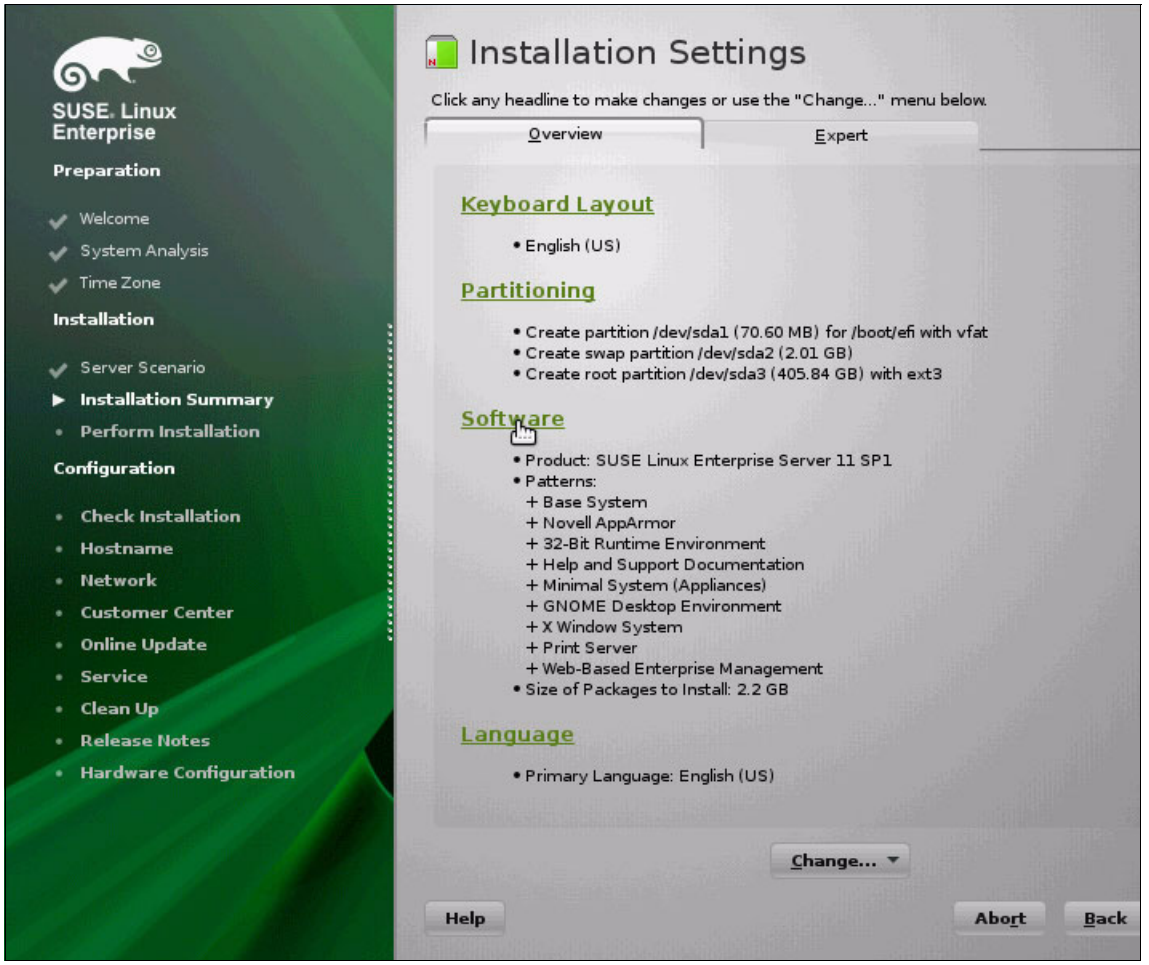

Figure 3-20 Software selection and system tasks

10. After the installation is complete, you must reboot the server. After the reboot, enter the root password and click **Next**.

11. Enter the host name and domain name in the window (Figure 3-21). Use node101 as the host name and purescale.demo as the domain name for the first host. On the second node, use node102 as the host name and purescale.demo as the domain name. Click **Next**.

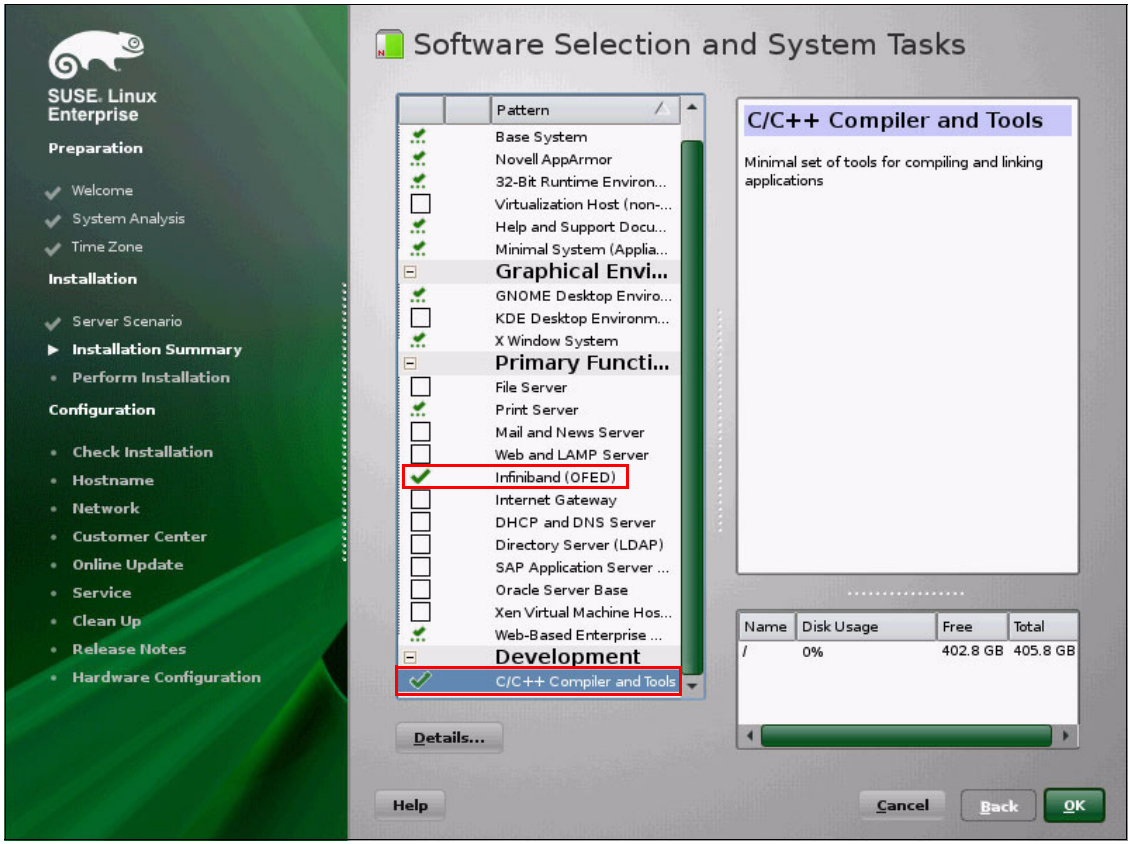

Figure 3-21 Host name and domain name

12. On the Network Configuration window (Figure 3-22), click **Network Interfaces** to set up the Ethernet network.

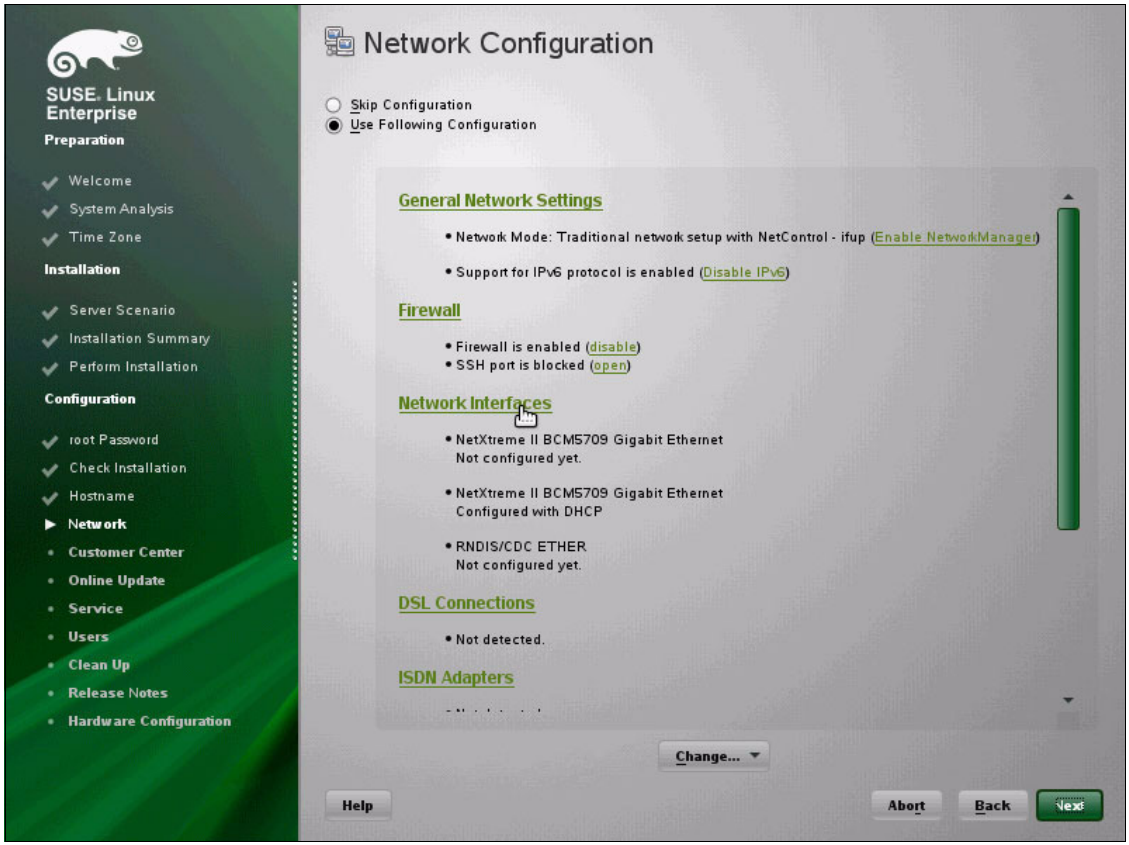

Figure 3-22 Network Configuration

13. Click the Ethernet port that shows DHCP in the IP Address section, and click **Edit**.

> **DHCP addresses not supported:** DHCP addresses are not supported by the DB2 pureScale Feature. Changes to the DHCP address affect the client as well.

14. Select **Statically assigned IP Address** and enter the correct host name, IP address, and subnet mask (Figure 3-23). For example, node101.purescale.demo, 192.168.1.101 and 255.255.255.0 for the first node. Click **Next** to continue.

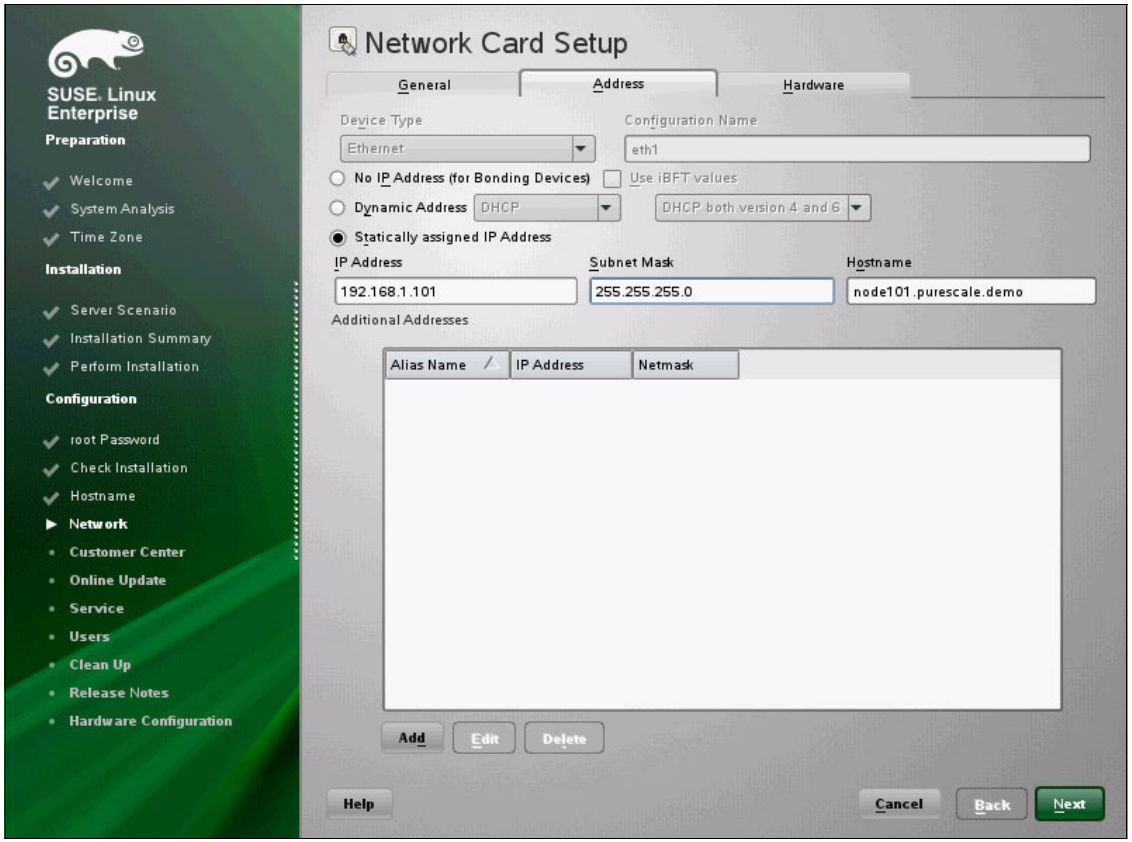

Figure 3-23 Network Card Setup

> **Important:** Change the host name and IP address to node102.purescale.demo and 192.168.1.102 on the second server. You can also follow the instructions in 3.2.6, “Step 6: Setting up the Ethernet network” on page 124.

15. On the network setting page (Figure 3-24), notice that the IP address section is changed from DHCP to the IP address that you assigned to this network adapter. Click **OK** and then click **Next**.

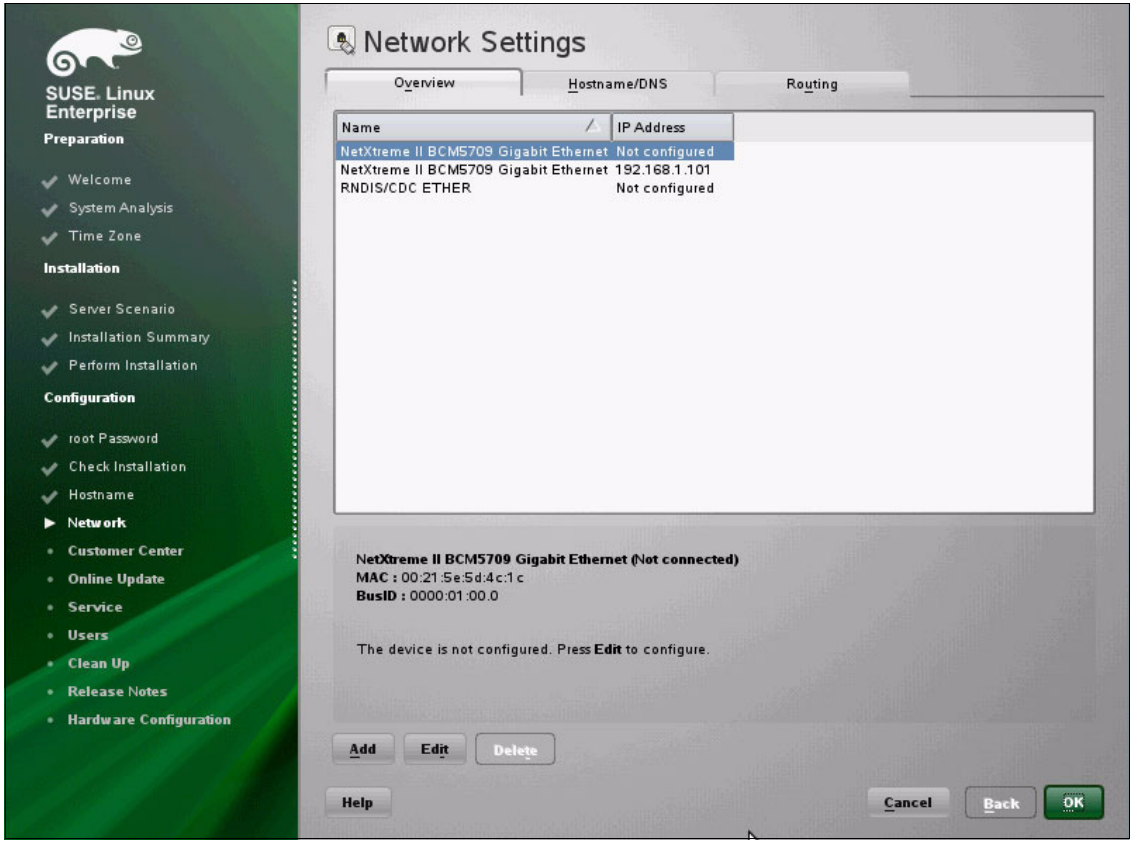

Figure 3-24 Network Settings

16. You can test your Internet connection or skip this test. Click **Next** to continue.
17. You can set up the network services on this page, or you can set up this information later after the installation. Click **Next**.
18. Select the correct user authentication method, and click **Next**.
19. Create a user with a password if needed. Click **Next** to continue.
20. Review the release notes, and click **Next**.
21. The installation requires that you perform hardware configuration, click **OK** to detect graphics card on your system. After the detection completes, click **Next**.
22. Click **Finish** to finish the SUSE Linux Enterprise Server 11 SP1 installation.

Important: You have finished the installation of SUSE Linux Enterprise Server 11 SP1 operating system. Ensure that you performed these steps on all servers in the cluster. After the operating system finishes installing on all hosts, you can proceed to 3.2.2, “Step 2: Updating the kernel” on page 95.

Red Hat Enterprise Linux 5

To install the operating system on Red Hat Enterprise Linux 5, complete the following steps:

1. Power on the IBM System x server, and insert the Red Hat Enterprise Linux installation DVD. Press F12 to open the boot order menu (Figure 3-25).

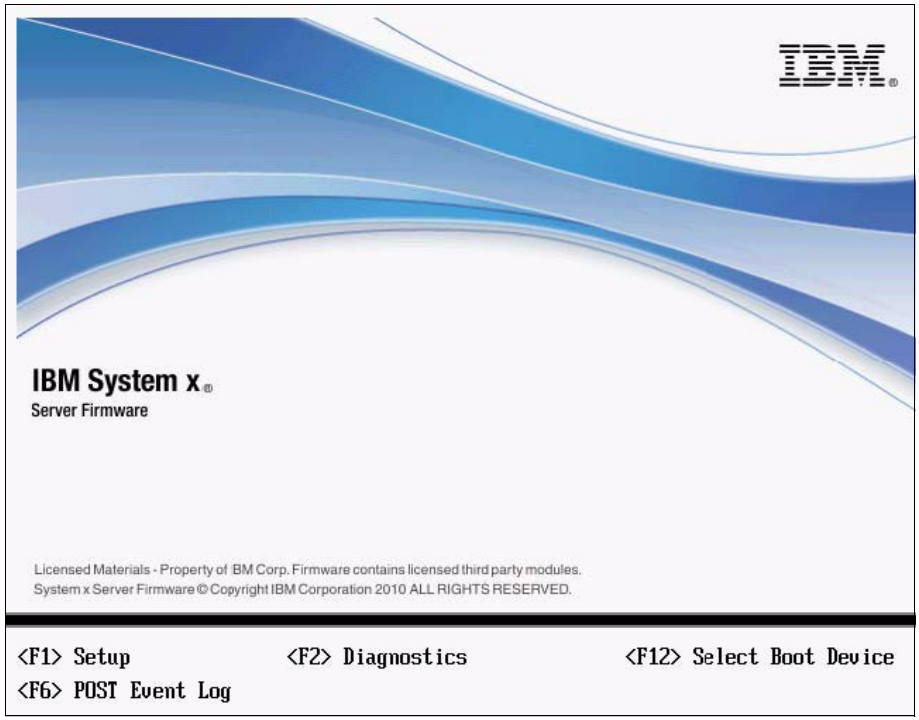

Figure 3-25 IBM System x boot

2. From the boot order menu, select CD/DVD-ROM to boot from the installation DVD.
3. When the Red Hat Enterprise Linux 5.6 boot panel opens, press Enter to start the installation in graphical mode. Click either **OK** to test the image or **Skip** to skip the installation media testing.
4. Review the Release Notes and then click **Next** to continue.

5. Select the appropriate language and keyboard layout for your system. Click **Next**.
6. Retrieve the installation number from your system administrator or skip this retrieval for now. Click **OK** to continue.
7. Red Hat Enterprise Linux finds the virtual drive device (407 GB) that you created as sda. The setup wizard prompts you to initialize this device to continue.

> **Important:** If the server is connected to a preconfigured external storage, Red Hat Enterprise Linux might pick up the external drives as sdb, sdc, and so on, and it also asks your permission to initialize all of the external drives. You can skip initializing external drives and initialize only the disk that you want to install the OS on to speed up the installation.

8. Ensure that the disk you selected in the window is correct before clicking **Next**.

9. Switch the host name from automatic to manual, and enter the host name that you use. This example uses node101.purescale.demo. Select the correct Ethernet device in the list, and click **Edit** to input the IPv4 address and subnet mask, 192.168.1.101/255.255.255.0 (Figure 3-26).

> **Important:** Change the host name and IP address to node102.purescale.demo/192.168.1.102 on the second server. If you are not sure which Ethernet device to choose, follow the instructions in 3.2.6, “Step 6: Setting up the Ethernet network” on page 124.

Figure 3-26 Network configuration

10. Select **Software Development** in the menu (Figure 3-27). It contains the required C++ library packages that are installed during the deployment. Choose **Customize now**, and click **Next** to continue.

Figure 3-27 Package selection

11. Click **Base System**, and select **OpenFabrics Enterprise Distribution** to install the OFED driver during the deployment (Figure 3-28). Click **Next** to continue.

Figure 3-28 Package selection

12. Click **Next** to start the installation.
13. After the installation completes, remove the Red Hat Enterprise Linux installation DVD from the DVD drive, and click **Reboot** to reboot the server. Click **Forward** to continue.
14. Review the license agreement on this page, and click **Forward** after accepting the license.

15. To install DB2 10.1 with the DB2 pureScale Feature on Red Hat Enterprise Linux 5.6, you must disable the firewall and SELinux feature (Figure 3-29 and Figure 3-30 on page 94). Click **Forward** to continue.

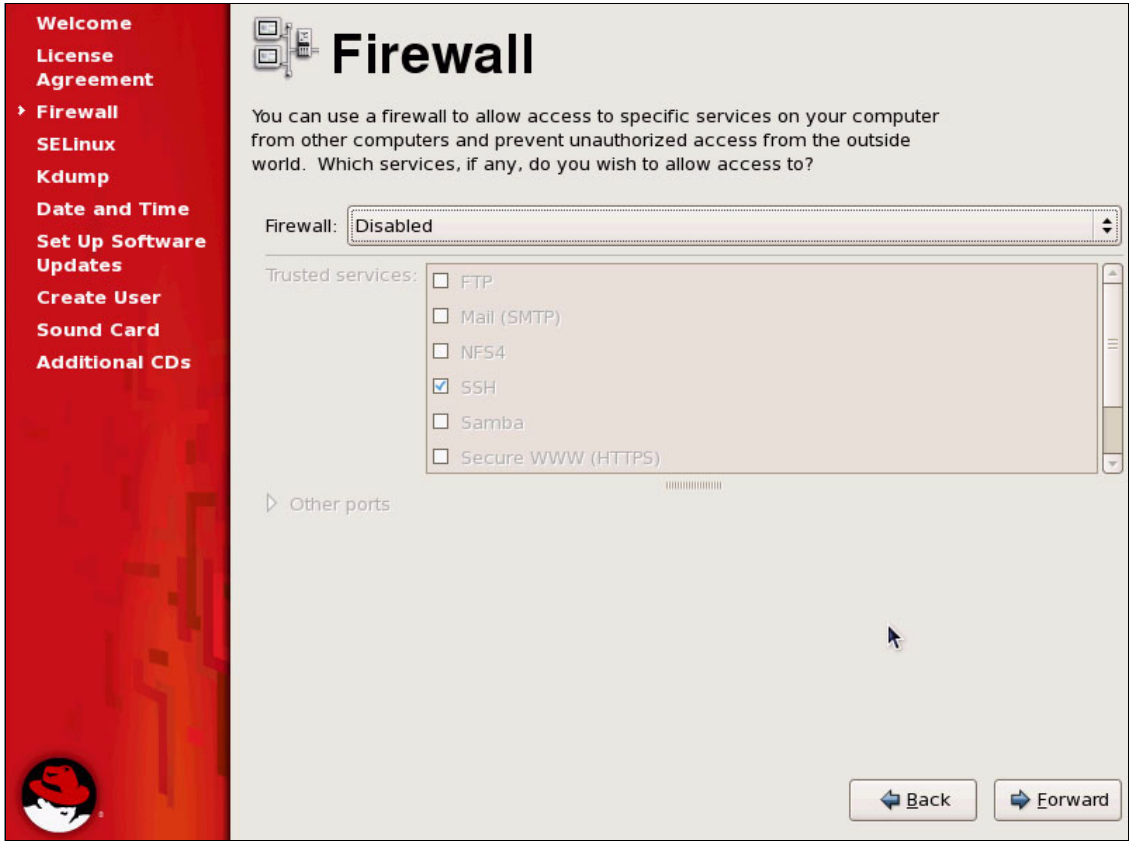

Figure 3-29 Disable firewall

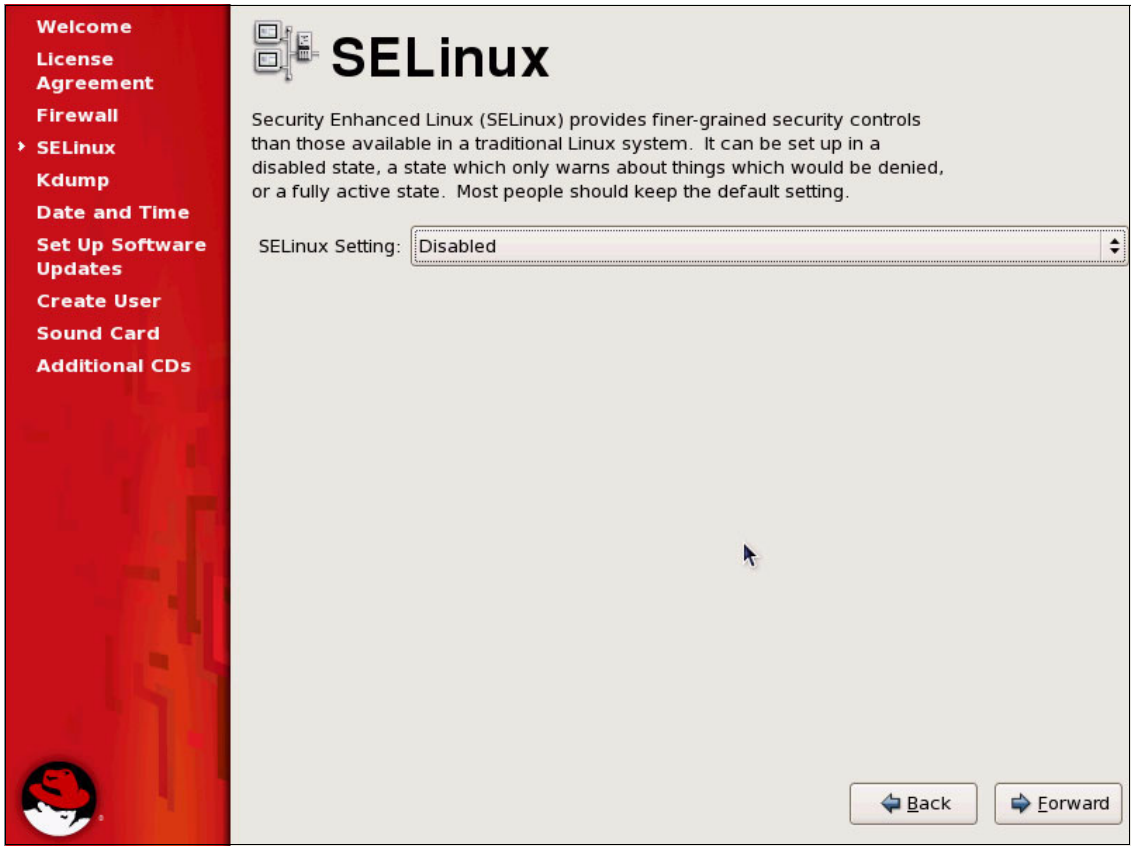

Figure 3-30 Disable SELinux

16. You do not need to enable the kdump feature. Click **Forward** to continue.
17. Ensure that the date and time settings are correct and click **Forward**.
18. Set up the Red Hat Network, and then click **Forward** to continue.
19. Create a user if needed. Click **Forward**.
20. Click **Forward** to accept the sound card configuration.
21. Click **Finish** to finish the installation. The server requires reboot afterward.

Important: Ensure that you installed Red Hat Enterprise Linux 5.6 on all servers in the cluster. After the installation is complete, proceed to 3.2.2, “Step 2: Updating the kernel” on page 95 to continue.

Red Hat Enterprise Linux 6.1

The installation of Red Hat Enterprise Linux 6.1 is similar to the installation on Red Hat Enterprise Linux 5.6. The major difference is to choose various software packages during the installation configuration wizard. The following steps are the steps that are distinct from the Red Hat Enterprise Linux 5.6 installation:

1. At the software package pane, choose **Customize Now** and add the following package groups:
   a. Under the Base System category, choose the following packages:
      - Compatibility libraries
      - Existing UNIX compatibility with extra packages, ksh, rsh, rsh-server, telnet, and telnet-server
   b. Under the Server category, choose the following packages:
      - FTP server
      - NFS file server
      - System administrator tools with extra package
   c. Under the Desktop category, choose the following packages:
      - Desktop
      - Existing Windows operating system compatibility with extra packages, libxp, and openmotif
   d. Under the Development category, choose the following packages:
      - Additional development with extra packages, libXpm-devel, libaio-devel, libcap-devel, openmotif-devel, and tcl-devel
      - Development tools with extra packages, compat-gcc (three of them), expect, and libstdc++
2. Do *not* set up the Ethernet network during the OS installation. You configure this network later.
3. Ensure that you install and enable NTP, and the time is synchronized among all the servers.
4. Ensure that kdump is disabled during the installation and finish the installation.

3.2.2 Step 2: Updating the kernel

The DB2 pureScale Feature requires specific kernel versions on Linux distributions. This section explains how to update the kernel to the required version on both SUSE Linux and Red Hat Linux distributions. Choose the appropriate set of instructions that matches your operating system.

SUSE Linux Enterprise Server 11

The minimum kernel version level on SUSE Linux Enterprise Server 11 SP1 for the DB2 pureScale Feature is 2.6.32.36-0.5. You must update the kernel on SUSE Linux Enterprise Server 11 SP1 to perform the installation.

> **Attention:** The kernel version level on SUSE Linux Enterprise Server 11 SP2 satisfies the DB2 pureScale Feature requirement. A kernel update is not required.

To update the kernel, complete the following steps:

1. Open a terminal and update the kernel rpms by running the **`rpm -i`** command:

   `rpm -i <rpm package>`

   Install the required kernel rpm files in the following order:

   a. `kernel-default-base-2.6.32.36-0.5.2`
   b. `kernel-default-2.6.32.36-0.5.2`
   c. `kernel-source-2.6.32.36-0.5.2`
   d. `kernel-default-devel-2.6.32.36-0.5.2`
   e. `kernel-syms-2.6.32.36-0.5.2`

   > **Attention:** You can download the kernel `rpm` files from the SUSE Updates repository. Contact your system administrator to get a copy of the kernel files.

2. Reboot the server to apply the kernel changes.

   > **Important:** You must reboot the server after updating the kernel. After all the hosts are updated, you can continue to 3.2.3, “Step 3: Installing the multipath device driver” on page 97.

Red Hat Enterprise Linux 5.6

The required kernel version level for the DB2 pureScale Feature on Red Hat Enterprise Linux 5.6 is 2.6.18.194.26.1.el5. If your operating system does not meet this requirement, download the correct kernel rpm packages from Red Hat Network. Then, open a terminal and update the kernel by running **`rpm -i`**. Use the following required kernel packages:

- `kernel-devel-2.6.18-194.26.1.el5`
- `kernel-2.6.18-194.26.1.el5`

After updating the kernel, reboot the server to pick up the new kernel.

Important: You must reboot the server after updating the kernel. After you update all the hosts, continue to 3.2.3, “Step 3: Installing the multipath device driver” on page 97.

Red Hat Enterprise Linux 6.1

The kernel version level on Red Hat Enterprise Linux 6.1 satisfies the minimum requirement of the DB2 pureScale Feature. You do not need to update the kernel on Red Hat Enterprise Linux 6.1.

3.2.3 Step 3: Installing the multipath device driver

The multipath driver provides the capability of configuring multiple paths to a single device. This section describes how to set up and configure the multipath daemon service.

SUSE Linux Enterprise Server 11

Attention: This task is the same on both SUSE Linux Enterprise Server 11 SP1 and SP2. Complete these tasks on all servers in the cluster.

To set up and configure the multipath daemon service on SUSE Linux Enterprise Server 11 SP1 or SP2, first check whether the multipath daemon is already started by running the following command:

```
chkconfig --list | grep multipathd
```

Example 3-1 shows the command and its output.

Example 3-1 Check multipath daemon status

```
node101:~ # chkconfig --list | grep multipathd
multipathd 0:off  1:off  2:off  3:off  4:off  5:off  6:off
```

If the output shows that the multipath daemon service is not started, run the following command to start it:

```
chkconfig multipathd on
```

Example 3-2 shows the command and its output.

Example 3-2 Start multipath daemon service

```
node101:~/ # chkconfig multipathd on
node101:~/ # chkconfig --list | grep multipathd
multipathd 0:off 1:off 2:off 3:on 4:off 5:on 6:off
```

The multipath daemon service starts on the next boot.

> **Important:** Ensure that you perform these steps on all servers in the cluster. Then proceed to 3.2.4, “Step 4: Configuring the external storage” on page 99 to continue.

Red Hat Enterprise Linux 5.6

To set up and configure the multipath daemon service on Red Hat Enterprise Linux 5.6, complete the following steps:

1. Open a terminal and run the following commands to ensure that the status of the multipath daemon service is turned off:
   - **`chkconfig --list | grep multipathd`**
   - **`chkconfig multipathd off`**
2. Edit the `/etc/multipath.conf` file. Find the lines shown in Example 3-3.

   Example 3-3 Original lines in multipath configuration file

   ```
   blacklist {
   devnode “*”
   }
   ```

3. Comment out the text, as shown in Example 3-4.

   Example 3-4 Edited lines in multipath configuration file

   ```
   # blacklist {
   # devnode “*”
   # }
   ```

4. Save the changes and exit.
5. Turn on the service by running the following command:

   ```
   chkconfig multipathd on
   ```

6. Run the following command to check the multipath daemon’s status:

   ```
   chkconfig --list | grep multipathd
   ```

Example 3-5 shows the output of this command.

Example 3-5 Check the multipath daemon status

```
[root@node101 ~]# chkconfig --list | grep multipathd
multipathd      0:off   1:off   2:on    3:on    4:on    5:on
6:off
```

Red Hat Enterprise Linux 6.1

You must register the server with Red Hat Network before you complete the following steps. Contact your system administrator if you are not sure how to register your server.

To set up and configure the multipath daemon service on Red Hat Enterprise Linux 6.1, run the following commands:

1. Install the Storage Availability Tools group package by running the following command. This package contains the multipath daemon.

   `yum groupinstall "Storage Availability Tools"`

2. Run the **`mpathconf --enable`** command to generate the multipath `/etc/multipath.conf` configuration file.
3. Edit the configuration file to add the following line:

   `find_multipaths yes`

4. Run the **`mpathconf`** command again to see the output shown in Example 3-6.

Example 3-6 Output of mpathconf command

```
multipath is enabled
find_multipaths is enabled
user_friendly_names is enabled
dm_multipath module is loaded
multipathd is chkconfiged on
```

Important: Ensure that you complete these steps on all servers before you continue to the next step.

3.2.4 Step 4: Configuring the external storage

This section explains how to configure the external storage. This section also shows how to use Storage Manager to create the required arrays, logical drives, and setup mappings so that the operating system can use the external disks.

Attention: You need Storage Manager installed only on one server. When installed, you can manage the external storage for all other servers. The examples in this book install Storage Manager on node101. The process is the same on SUSE Linux Enterprise 11, Red Hat Enterprise Linux 5, and Red Hat Enterprise Linux 6 systems.

Installing Storage Manager

Storage Manager version: At time of writing, the latest version of Storage Manager is Version 10.77.x5.28. You can download this software from the IBM Support site at:

http://www.ibm.com/support

Check the appropriate user guide if you are using a different Storage Manager version.

To install Storage Manager, complete the following steps:

1. Open a terminal session and navigate to the directory where the Storage Manager `tar` file is located. Extract the contents of the file by running the following command:

   ```
   tar -zxvf <storage manager tar file path>
   ```

2. Navigate to the `Linux_x64_10p77/Linux_x86-64/` folder. Change the file permissions of the BIN file so that the file is executable:

   ```
   chmod a+x <SMIA bin file path>
   ```

3. Run the BIN file to start the installation wizard:

   ```
   ./SMIA-LINUXX64-10.77.A5.16.bin
   ```

4. Complete the installation by following the wizard.

Creating arrays and logical drives

This section describes how to create arrays and logical drives in Storage Manager. An array is a set of drives that is logically grouped and assigned a RAID level. An array provides the overall capacity needed to create one or more logical drives. A logical drive is a logical component created for the host to access storage. Although a logical drive might consist of more than one drive, it appears as one logical component to the host.

After you have the Storage Manager software installed, continue to configure the external storage. Create three arrays and four logical drives on the storage (Table 3-1).

Table 3-1 External storage logical drives consideration

Array name	RAID level	Logical drive name and size
TBSQLLIB	RAID 1	TieBreaker (100 MB) SQLLIB (10 GB)
SharedData	RAID 5	pureScaleData (500 GB)
SharedLog	RAID 5	pureScaleLog (250 GB)

To create arrays and logical drives, complete the following steps:

1. Open Storage Manager by double-clicking the shortcut on the desktop, or run the following command in a terminal session:

   ```
   /opt/IBM_DS/client/SMclient
   ```

2. If managed storage is not configured, Storage Manager prompts you to perform a discovery operation, which can be either automatic or manual. Choose **Automatic** and click **OK** to complete the automatic discovery.

3. After the automatic discovery is complete, click the **Device** tab. The external storage is listed in the window and shows an in-band management connection. Right-click the storage entry, and select **Manage Storage Subsystem** to start the subsystem management window (Figure 3-31).

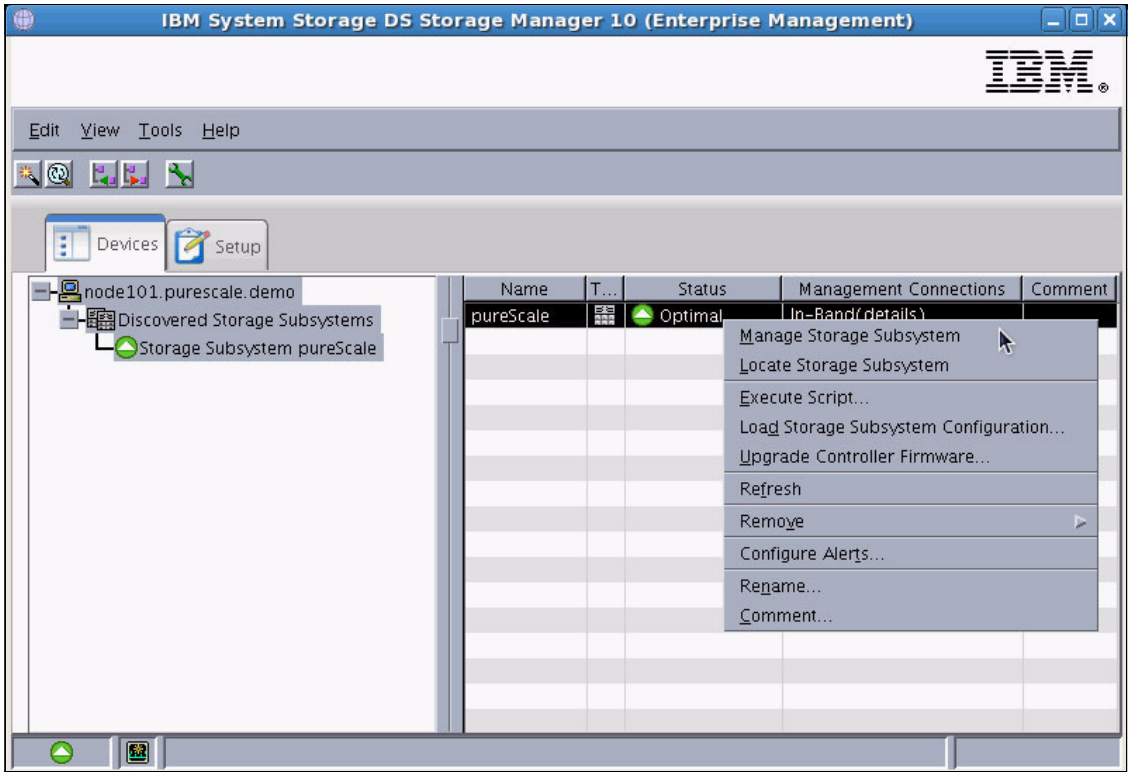

Figure 3-31 Storage Manager 10 devices tab

4. When prompted, enter the storage subsystem password to continue. If there is no password set up, you can enter a password to use as the storage subsystem password.

5. Click the **Logical** tab. The total unconfigured capacity displays in the view shown in Figure 3-32.

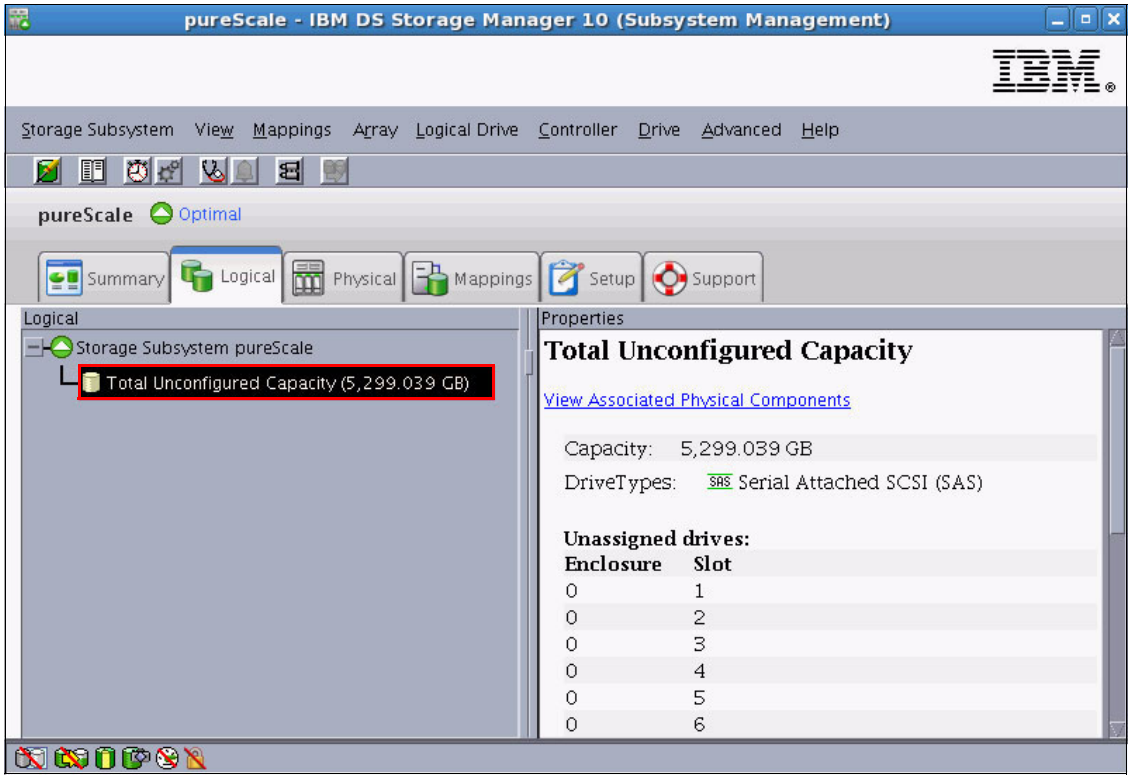

Figure 3-32 Subsystem Management Logical tab

6. Right-click **Total Unconfigured Capacity**, and select **Create Array**.

7. Enter a name for the first array, for example, TBSQLLIB (Figure 3-33). Select **Manual**, and click **Next**.

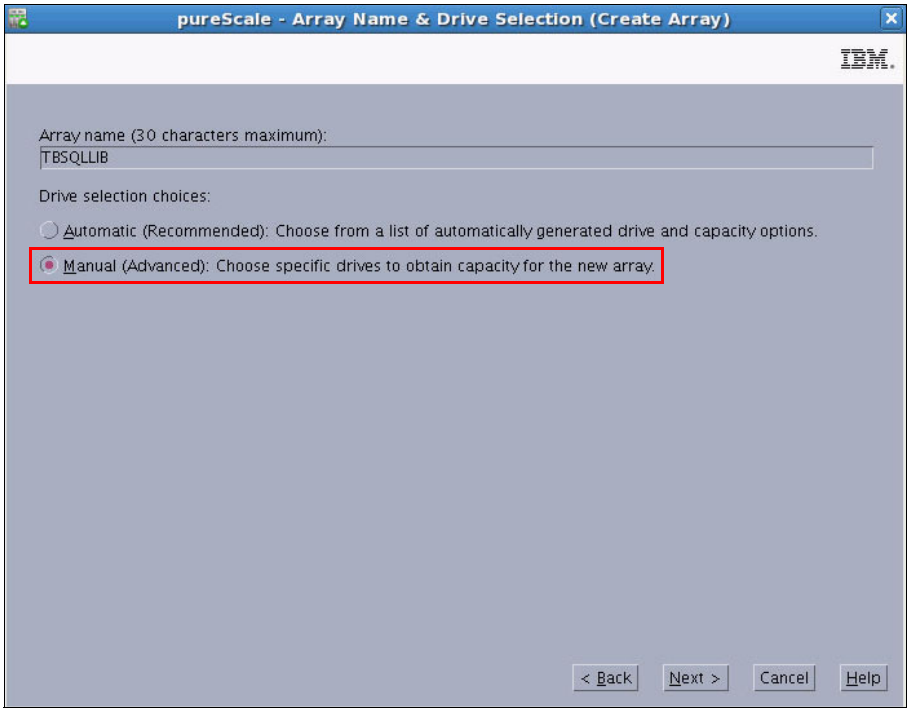

Figure 3-33 Array name and drive selection

8. Use RAID 1 for this array, TBSQLLIB, which requires a minimum of two disks. Select two disks from the left pane, and add them to the right pane. Select **RAID 1** from the RAID level list (Figure 3-34).

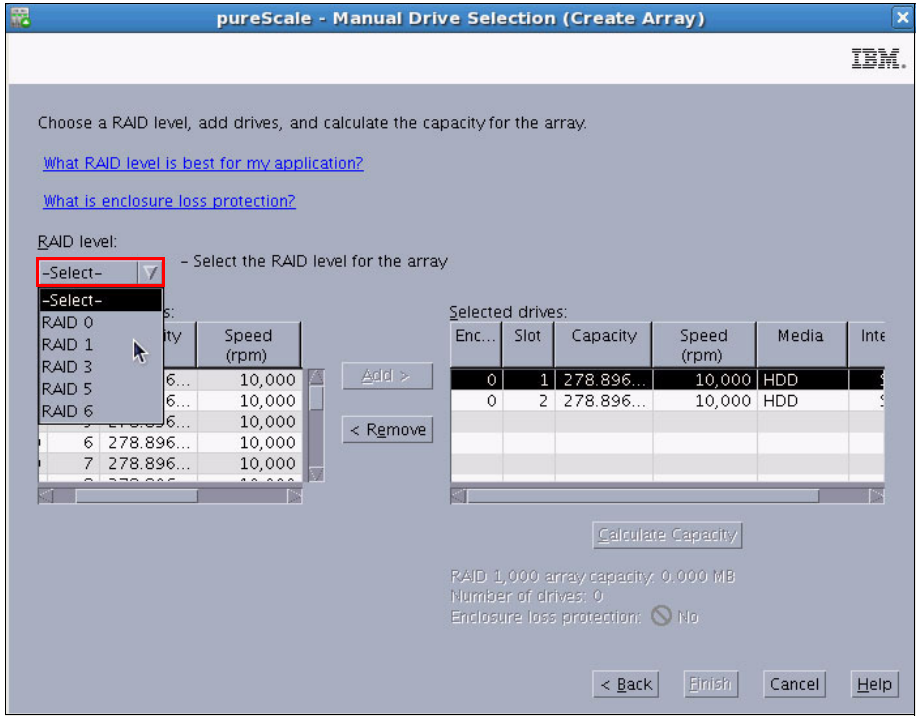

Figure 3-34 Manual drive selection

9. On the Manual Drive Selection window (Figure 3-35), click **Calculate Capacity** to check the total size of the array. Click **Finish**.

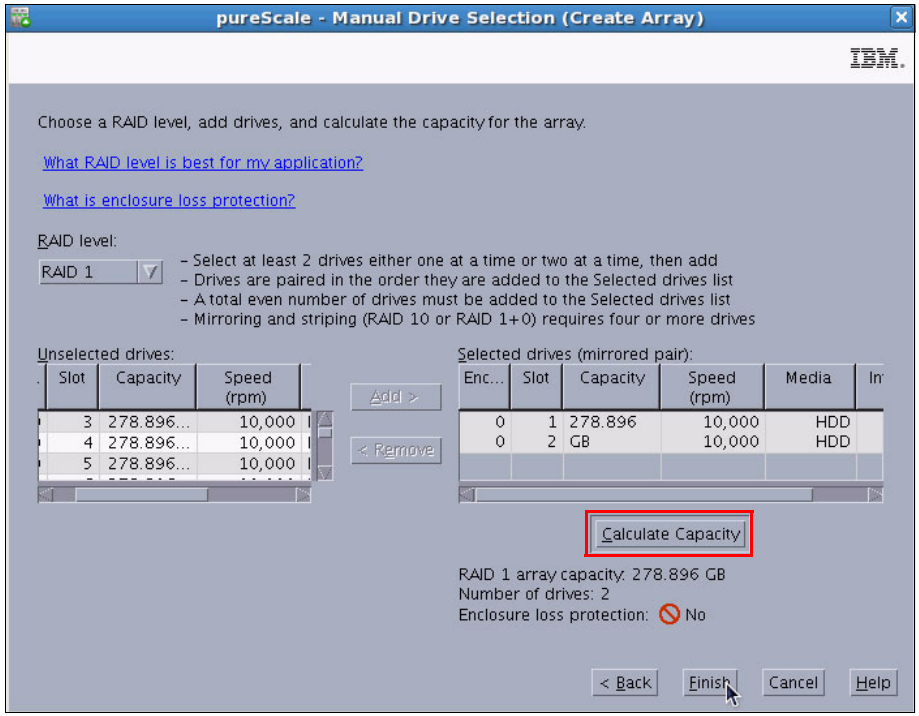

Figure 3-35 Manual drive selection

10. When prompted to create the logical drives, click **Yes**.

11. Create the first logical drive that is used as the tie breaker. Name the logical drive whatever you want, for example, TieBreaker (Figure 3-36), and make its size 100 MB. Click **Next** to continue.

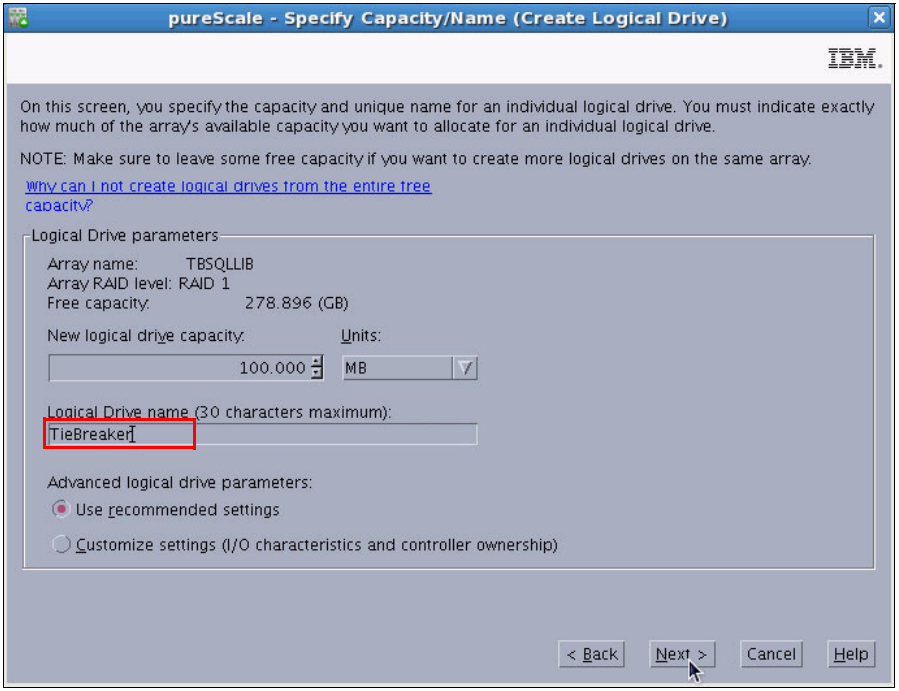

Figure 3-36 Create the logical drive and specify the capacity and name

12. Select **Map later using the Mapping View** (Figure 3-37) and click **Finish** to exit. You configure the mappings after all the disks are created.

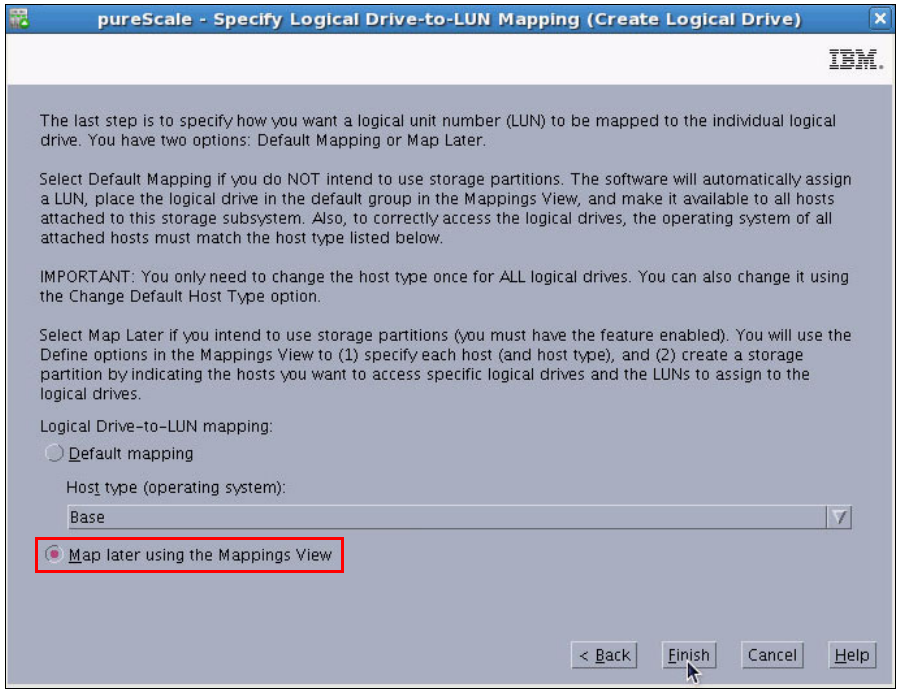

Figure 3-37 Specify logical drive-to-LUN mapping

13. When prompted to create another logical drive, click **Yes**.

14. Create the next logical drive on the same disk array. Name the logical drive SQLLIB and make its size 10 GB (Figure 3-38). Click **Next** to continue.

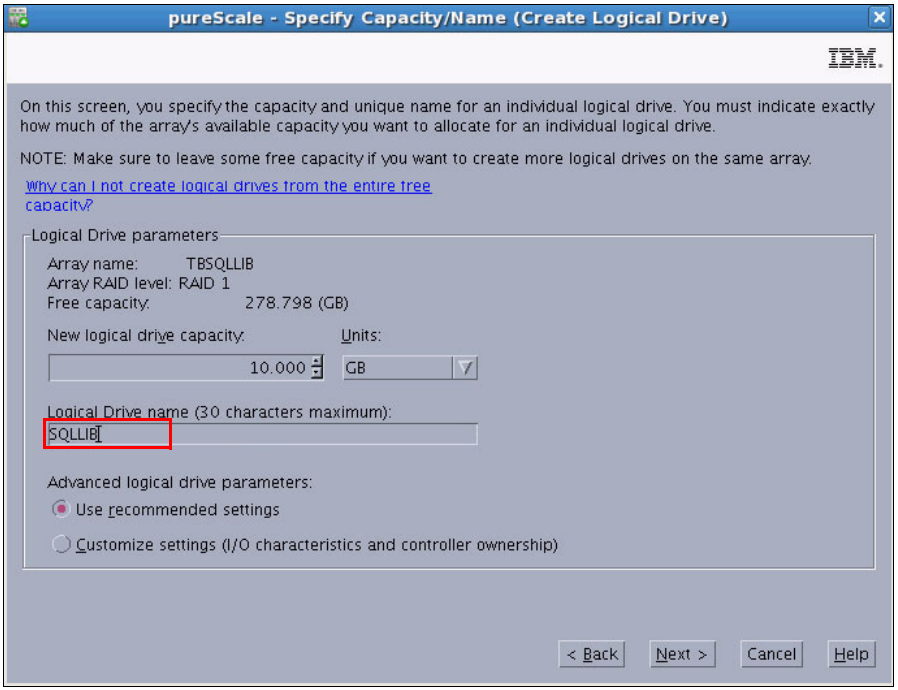

Figure 3-38 Create the logical drive and specify the capacity and name

15. Select **Map Later using the Mapping View** and click **Finish**. When prompted to create another logical drive, click **No** to exit (Figure 3-39).

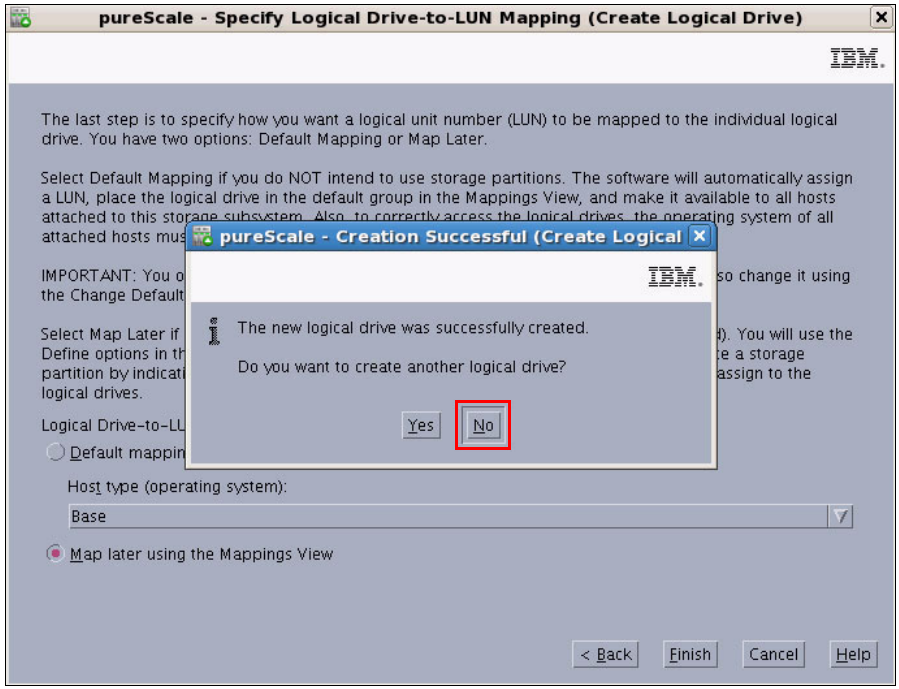

Figure 3-39 Specify logical drive-to-LUN mapping

16. Repeat steps 6 - 15 to create the disk arrays and logical drives shown in Table 3-2.

Table 3-2 External storage array and logical drives

Array name	RAID level	Logical drive name	Size
SharedData	RAID 5	pureScaleData	500 GB
SharedLog	RAID 5	pureScaleLog	250 GB

17. After you create all of the required disk arrays and logical drives, check the array and logical drives on the Logical tab. The tab resembles the window shown in Figure 3-40.

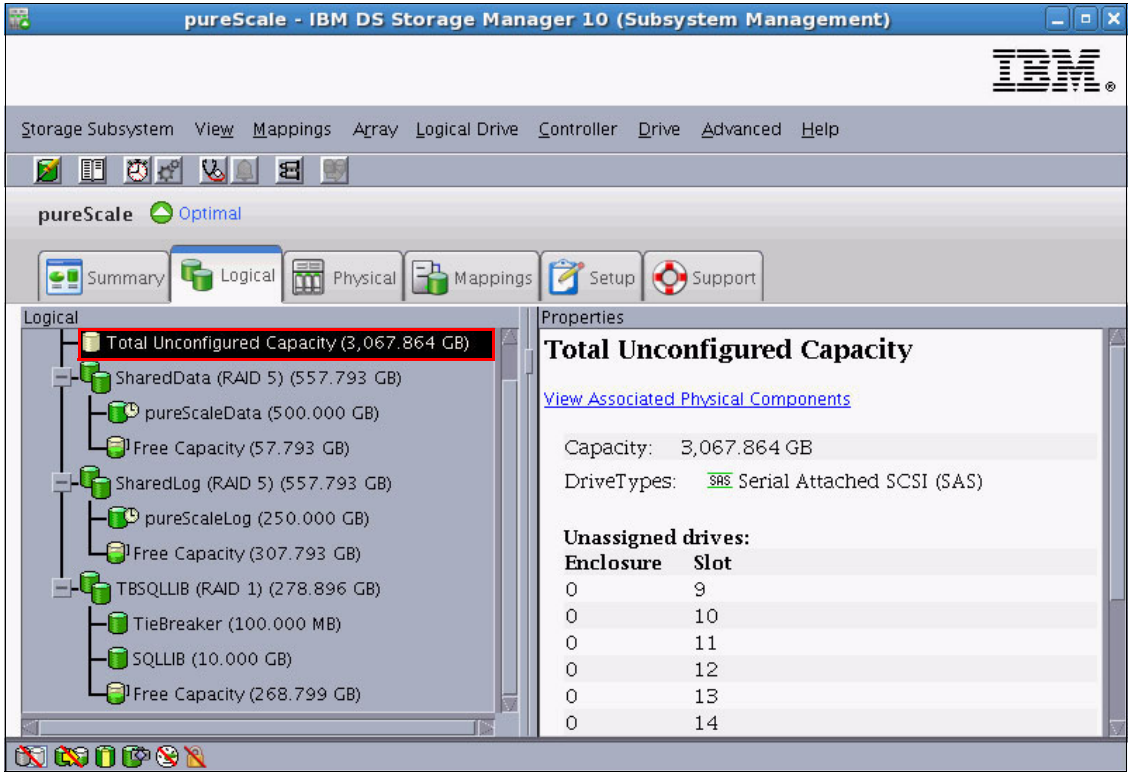

Figure 3-40 Storage Manager 10 Logical tab

> **Attention:** Do not exit Storage Manager, because you use it in the next section.

Creating a host, host group, and setup mappings

A *host* is a computer that is attached to a storage subsystem. It accesses a logical drive that is assigned to it through the HBA host ports or through the iSCSI host ports. A *host group* is a logical entity that identifies a collection of hosts that share access to the same logical drives. *Mapping* is an association of a logical drive with a single logical unit number (LUN). Mapping specifies which host group or host has access to the logical drive.

To create a host, a host group, and setup mappings to enable access from hosts to logical drives, complete the following steps:

1. Open a terminal on both servers, and run the following command. The command returns the fabric port identifier number on the server.

   ```
   systool -c fc_host –v | grep fabric_name
   ```

   The setup for our example uses a dual-port HBA with two port identifiers on each server (Example 3-7).

   Example 3-7 Determine the fabric port identifier information

   ```
   [root@node101 ~]# systool -c fc_host –v | grep fabric_name
         fabric_name = "0x20000024ff295710"
         fabric_name = "0x20000024ff295711"
   ```

   Attention: Make note of the port identifiers for each server, because you need the port identifiers later.

2. If Storage Manager is not running, open Storage Manager by double-clicking the shortcut on your desktop or running the following command in a terminal session:

   ```
   /opt/IBM_DS/client/SMclient
   ```

3. Switch back to Storage Manager, and click the **Mappings** tab. Right-click **Default Group** and click **Define** → **Host Group**.

4. Enter the name for the new host group. Enter “pureScale” and click **OK**.

5. Click the **Mappings** tab. Right-click **Host Group pureScale** and click **Define** → **Host** (Figure 3-41).

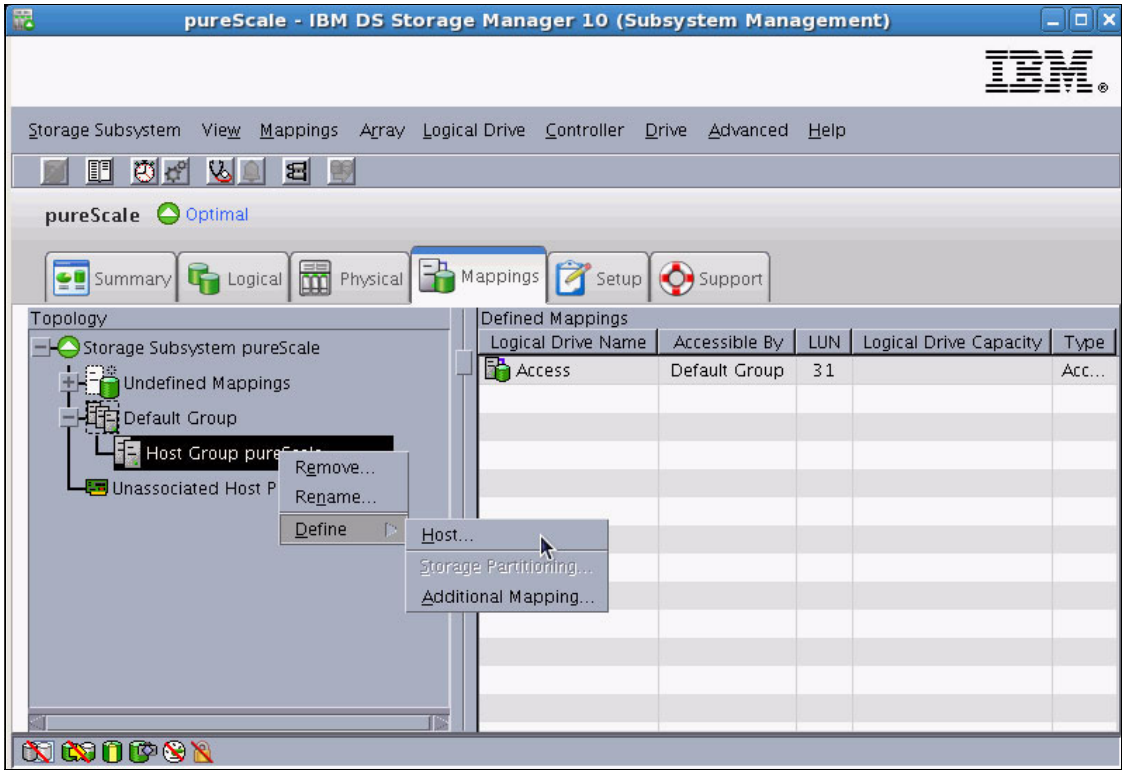

Figure 3-41 Storage Manager 10 Mappings tab

6. Define the first host by completing the following steps:
   a. Enter a name for the host. This example uses node101 for the first node and node102 for the second node. Click **Next** to continue.
   b. Choose a host interface type from the list. For this example, select **FC**.
   c. Select **Add by selecting a known unassociated host port identifier**.

d. From the “Known unassociated host port identifier” list (Figure 3-42), select an identifier that belongs to the first node, for example **21:00:00:24:ff:29:57:10**. The identifier number must match the output of step 1 on page 112.

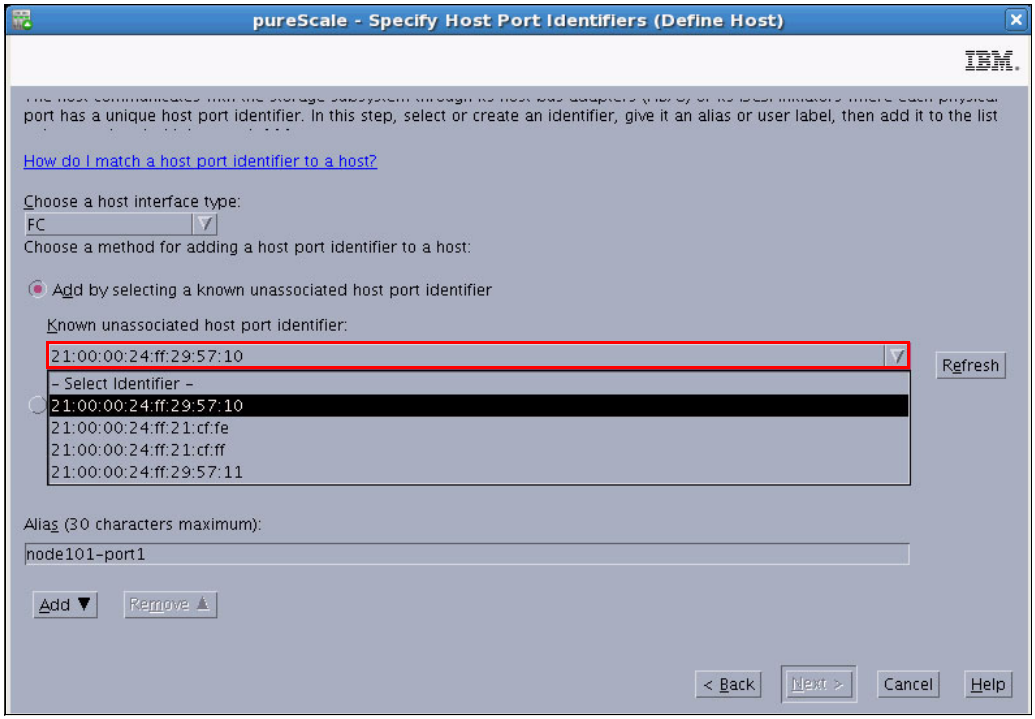

Figure 3-42 Specify the host port identifiers

e. In the Alias field, enter a name for the identifier, and click **Add**. This example uses node101-port1 for the first port.

f. From the “Known unassociated host port identifier” list, choose another identifier that belongs to the first node (Figure 3-43). The identifier number should match the output of step 1 on page 112.

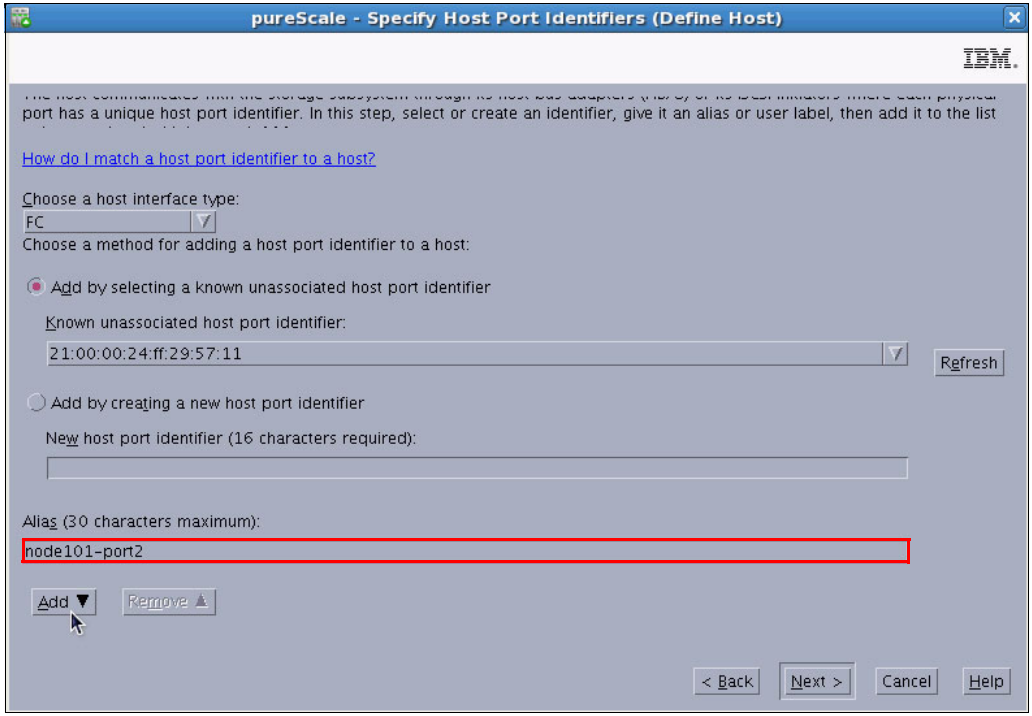

Figure 3-43 Specify the host port identifiers

g. In the Alias field, enter a name for the identifier, and click **Add**. This example uses node101-port2.

h. Click **Next** to continue. From the Host type (operating system) list, select **Linux** and click **Next**.

i. Review the summary and verify that both port identifiers belong to the correct host. Click **Finish** to continue. The summary window resembles the window shown in Figure 3-44.

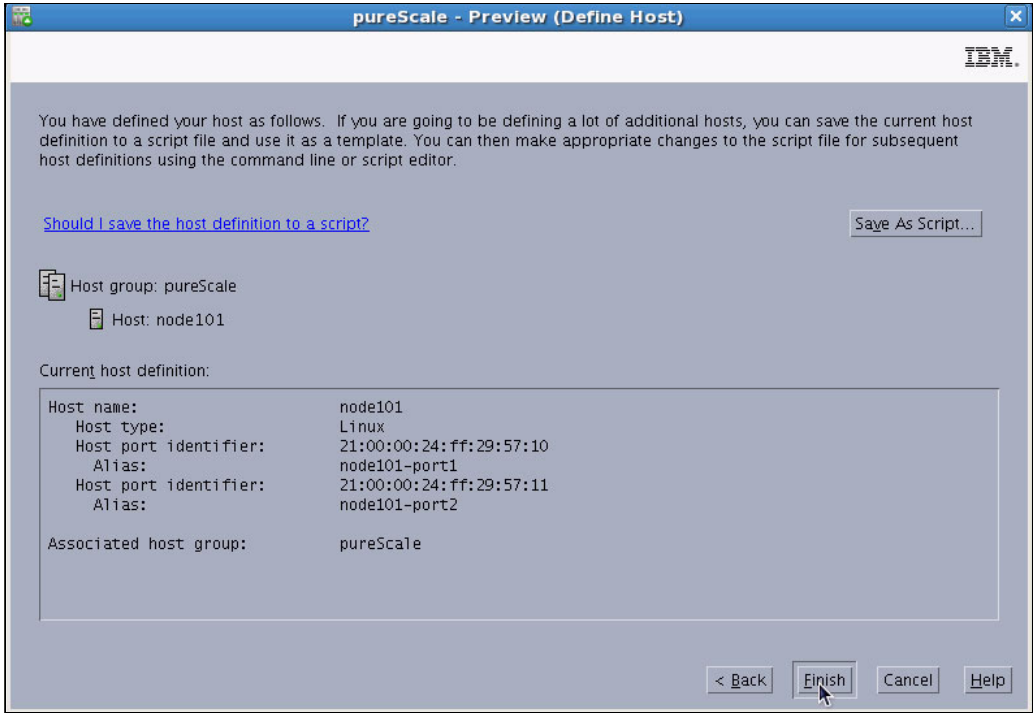

Figure 3-44 Define the host preview

7. When prompted to create another host, click **Yes** and repeat step 6 on page 113 to add a second host with the correct identifiers.

8. After you create both hosts, confirm that they are shown in Storage Manager on the Mapping tab under Host Group pureScale (Figure 3-45).

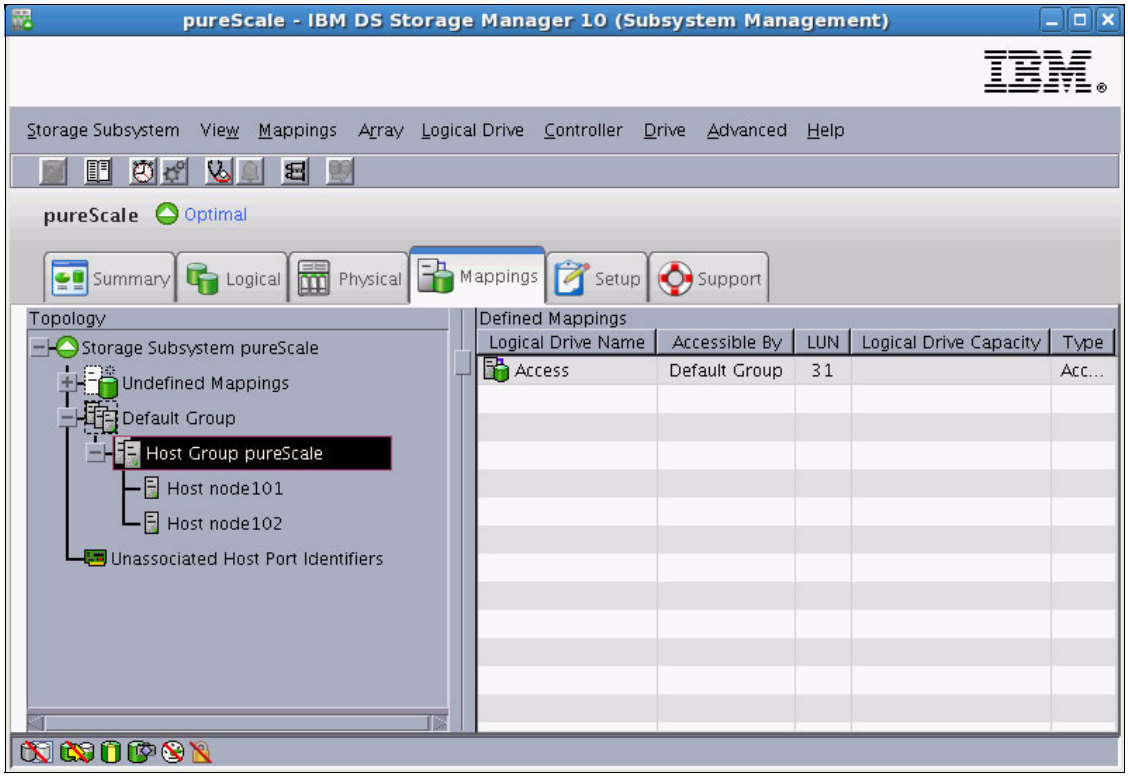

Figure 3-45 Storage Manager 10 Mappings tab

9. Expand **Undefined Mappings** to see the four logical drives that you just created (Figure 3-46):
   - TieBreaker
   - SQLLIB
   - pureScaleData
   - pureScaleLog

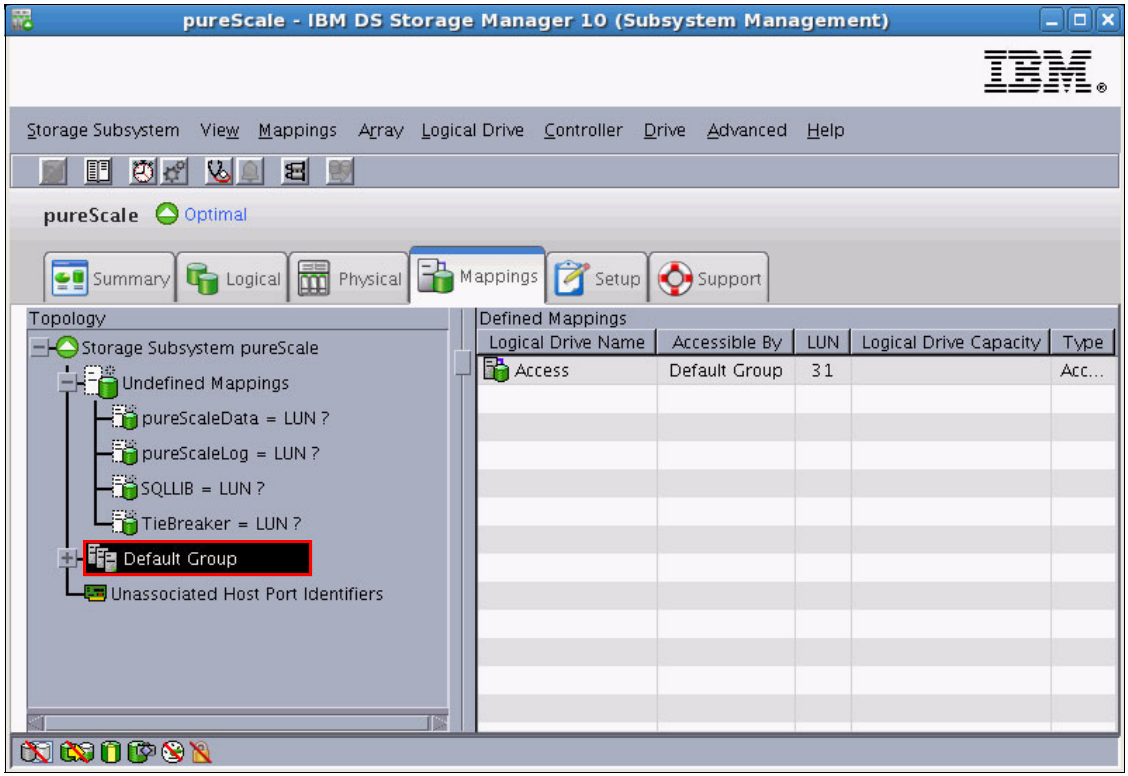

Figure 3-46 Storage Manager 10 Mappings tab

10. Right-click **TieBreaker** and select **Define Additional Mapping**.
11. Select **Host Group pureScale** from the list. You find the LUN number drop-down list and the four logical drives created earlier. The examples in this book use the following assignments:

LUN 0	TieBreaker
LUN 1	SQLLIB
LUN 2	pureScaleData
LUN 3	pureScaleLog

Important: The LUN affects the device mapping order on the host. For example, if LUN 0 is assigned to TieBreaker, LUN 1 to SQLLIB, LUN 2 to pureScaleData, and LUN 3 to pureScaleLog, the device paths for these four logical drives are `/dev/dm-0`, `/dev/dm-1`, `/dev/dm-2`, and `/dev/dm-3`.

12. After you add all four logical drives to the Host Group pureScale, verify the status on the Mappings tab (Figure 3-47). Then, exit Storage Manager.

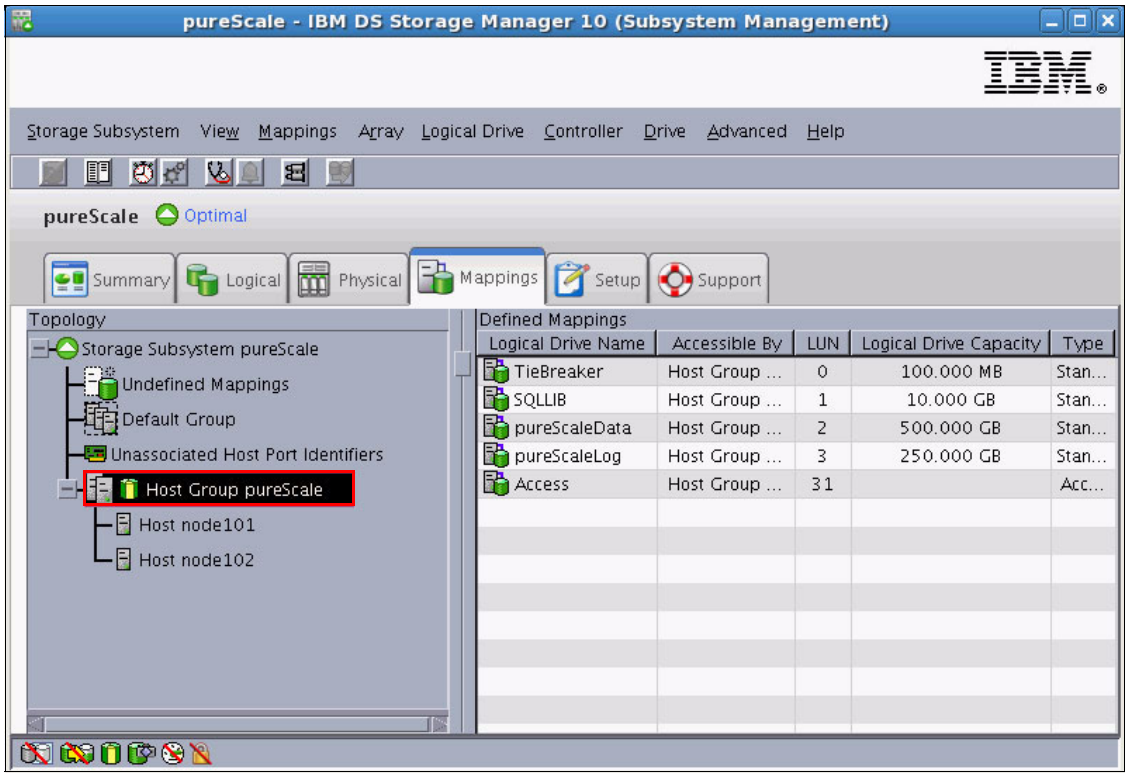

Figure 3-47 Storage Manager 10 Mappings tab

13. Reboot both servers in the cluster to pick up the external logical drivers. Then, run the following command to list the external logical drives:

```
multipath -l
```

The output should look like Example 3-8.

Example 3-8 The output of the multipath command

```
360080e50001b757e000093b74f3a3e15 dm-0 IBM,1746      FAStT
size=100M features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 rdac' wp=rw
|-+- policy='round-robin 0' prio=-1 status=active
| `- 4:0:0:0 sdf 8:80  active undef running
`-+- policy='round-robin 0' prio=-1 status=enabled
`- 3:0:0:0 sdb 8:16  active undef running
360080e50001b70e800001b0b4f3a3e7d dm-1 IBM,1746      FAStT
size=10G features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 rdac' wp=rw
|-+- policy='round-robin 0' prio=-1 status=active
| `- 3:0:0:1 sdc 8:32  active undef running
`-+- policy='round-robin 0' prio=-1 status=enabled
`- 4:0:0:1 sdg 8:96  active undef running
360080e50001b70e800001b634f5f123c dm-3 IBM,1746      FAStT
size=250G features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 rdac' wp=rw
|-+- policy='round-robin 0' prio=-1 status=active
| `- 3:0:0:3 sde 8:64  active undef running
`-+- policy='round-robin 0' prio=-1 status=enabled
`- 4:0:0:3 sdi 8:128 active undef running
360080e50001b757e000093bb4f3a3e7c dm-2 IBM,1746      FAStT
size=500G features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 rdac' wp=rw
|-+- policy='round-robin 0' prio=-1 status=active
| `- 4:0:0:2 sdh 8:112 active undef running
`-+- policy='round-robin 0' prio=-1 status=enabled
`- 3:0:0:2 sdd 8:48  active undef running
```

The examples in this book use the following disk paths:

- The TieBreaker logical disk path is `/dev/dm-0`.
- The shared instance disk path is `/dev/dm-1`.
- The SharedData logical disk path is `/dev/dm-2`.
- The SharedLog logical disk path is `/dev/dm-3`.

3.2.5 Step 5: Installing the required software packages

To install the DB2 pureScale Feature, you need to install additional software packages on your system.

The `sg3_utils` package is required for tiebreaker functionality, and it usually comes with the operating system installation DVD.

The system clock of all members in a DB2 pureScale environment must be synchronized to synchronize operations and facilitate time-sensitive operations. Thus, the Network Time Protocol (`ntp`) package must be installed and the system clock must be synchronized with an NTP server. This package contains utilities and daemons that synchronize the servers' time using NTP protocol and NTP server. This package is included in the installation DVD.

Attention: Complete this task on all servers in the cluster.

SUSE Linux Enterprise Server 11

To install the required additional packages on SUSE Linux Enterprise Server 11, complete the following steps:

1. The `sg3_utils` and `ntp` packages should have been installed during the SUSE Linux Enterprise Server 11 deployment. You can do a quick check to ensure that they are installed properly by running the following commands as root:

   - **`rpm -qa | grep sg3_utils`**
   - **`rpm -qa | grep ntp`**

   The output should be similar to Example 3-9.

   Example 3-9 Check the sg3_utils and ntp packages

   ```
   node101:~ # rpm -qa | grep sg3_utils
   sg3_utils-1.31-1.11.14
   node101:~ # rpm -qa | grep ntp
   ntp-4.2.4p8-1.3.28
   ```

2. If the `sg3_utils` and `ntp` packages are not installed by default on your system, use the Software Management menu to install them from the SUSE Linux Enterprise Server DVD.

Red Hat Enterprise Linux 5.6

Use the Red Hat package manager to add the `sg3_utils` package from the Red Hat Enterprise Linux 5.6 installation DVD. The systems for this book used the `sg3_utils-1.25-4.el5.x86_64` package.

To install the required additional packages on Red Hat Enterprise Linux 5.6, complete the following steps:

1. Insert the Red Hat Enterprise Linux 5.6 installation DVD in to the CD tray.
2. Click **Add/Remove Software** from the Applications drop-down menu.
3. Open the Repositories in the package manager and click **Add** to add a repository.
4. Ensure that the location is in the correct format (Figure 3-48) and click **OK** to close the window. The path to the location is:

   `file:///<path of the RHEL installation DVD>/Server`

Figure 3-48 Add a repository

5. Navigate to the **Search** tab (Figure 3-49) and search for `sg3_utils` and `ntp`. Choose the `sg3_utils-1.25-4.el5.x86_64` package, and click **Apply** to install it.

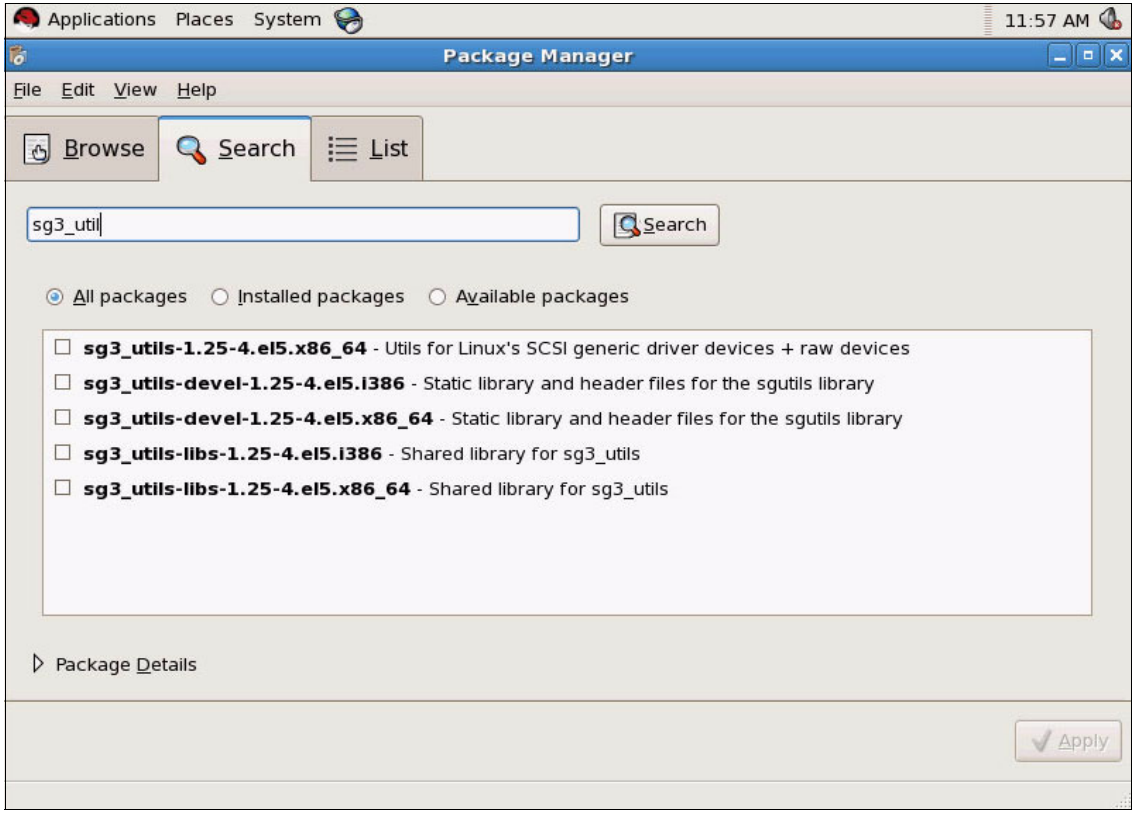

Figure 3-49 Package manager

Red Hat Enterprise Linux 6.1

To install and update additional packages on Red Hat Enterprise Linux 6.1, complete the following steps:

1. Ensure that you have set up the Red Hat Network for your system. Then update the `ksh` rpm package by running the **`yum update ksh`** command. As of this writing, the latest `ksh` version is ksh-20100621-16.el6.
2. Install the `sg3_utils` and `ntp` packages. The Red Hat Enterprise Linux 6.1 installation DVD contains these packages.
3. Reboot the server to pick up these changes.

3.2.6 Step 6: Setting up the Ethernet network

This section explains how to set up the Ethernet network for the DB2 pureScale Feature cluster. The examples in this book use the values listed in Table 3-3 to configure the network.

Table 3-3 Ethernet network configuration

Host	First node	Second node
Host name	node101	node102
Domain	purescale.demo	purescale.demo
IP address	192.168.1.101	192.168.1.102
Netmask	255.255.255.0	255.255.255.0

Important: On Linux and AIX, ensure that you configure a default gateway on all hosts that are part of the DB2 pureScale cluster.

SUSE Linux Enterprise Server 11

Attention: This task is the same on both SUSE Linux Enterprise Server 11 SP1 and SUSE Linux Enterprise Server 11 SP2. Complete these tasks on all servers in the cluster.

To set up the Ethernet network on SUSE Linux Enterprise Server 11, complete the following steps:

1. If you have multiple Ethernet ports on your system, detect which Ethernet port is used by the system. Ensure that you connected the server with an Ethernet switch from one of the ports.
2. Open a terminal and run the following command. If there are multiple Ethernet ports on the server, run the command for each Ethernet port.

   ```
   ethtool eth0
   ethtool eth1
   ....
   ```

   Example 3-10 shows the output of the **ethtool** command.

   Example 3-10 Output of ethtool command

   ```
   Settings for eth0 :
         Supported ports :[ TP ]
         Supported link modes :10baseT/Half  10baseT/Full
   ```

```
            100baseT/Half  100baseT/Full
            1000baseT/Full
Supports auto-negotiation :  Yes
Advertised link modes :10baseT/Half  10baseT/Full
            100baseT/Half  100baseT/Full
            1000baseT/Full
Advertised auto-negotiation: Yes
Speed: 1000Mb/s
Duplex: Full
Port: Twisted Pair
PHYAD: 1
Transceiver: internal
Auto-negotiation: on
Supports Wake-on: g
Wake-on: g
Link detected: yes
```

If part of the command output reads `Link detected`, the port is connected to the switch by Ethernet connections. Take note of the port number so that you can assign a static IP address to the Ethernet port.

3. Open **Network Settings** in Yast2 to change the network settings or the port connected to the Ethernet. Select the **Overview** tab (Figure 3-50).

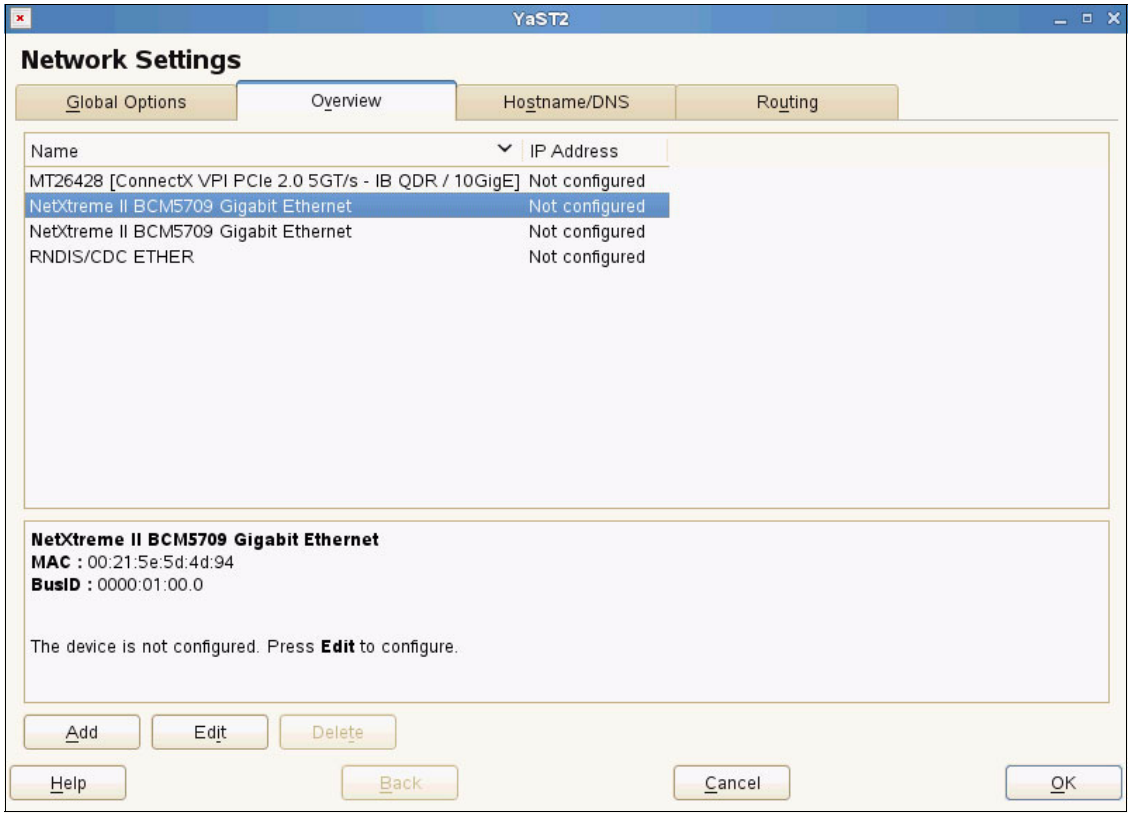

Figure 3-50 YaST2 network configuration

4. On the Address tab of the Network Card Setup window of the Network Settings tool, ensure that the value in the Configuration Name field matches the Ethernet port you noted in step 2 on page 124 (Figure 3-51).

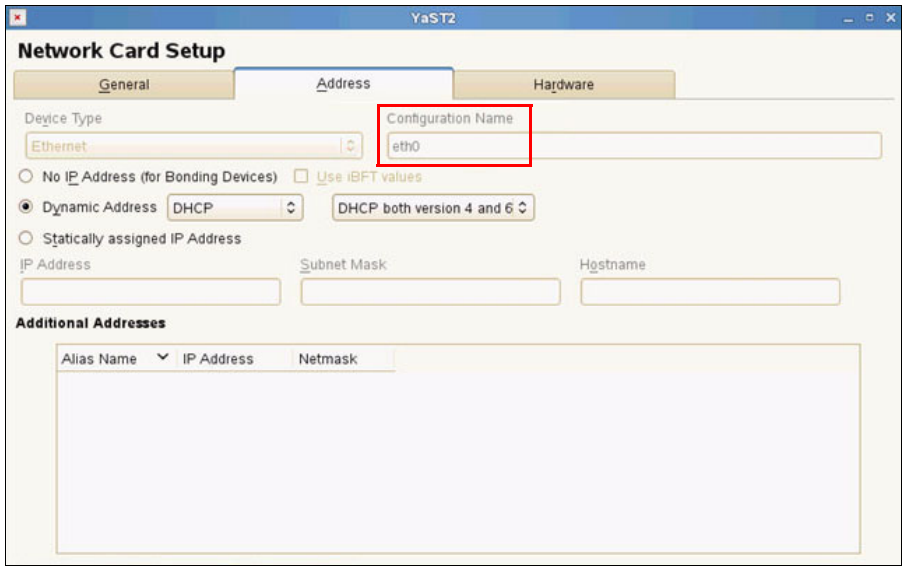

Figure 3-51 Network card setup

5. Assign a static IP address, subnet mask, and host name for the server (Figure 3-52). Click **Next**.

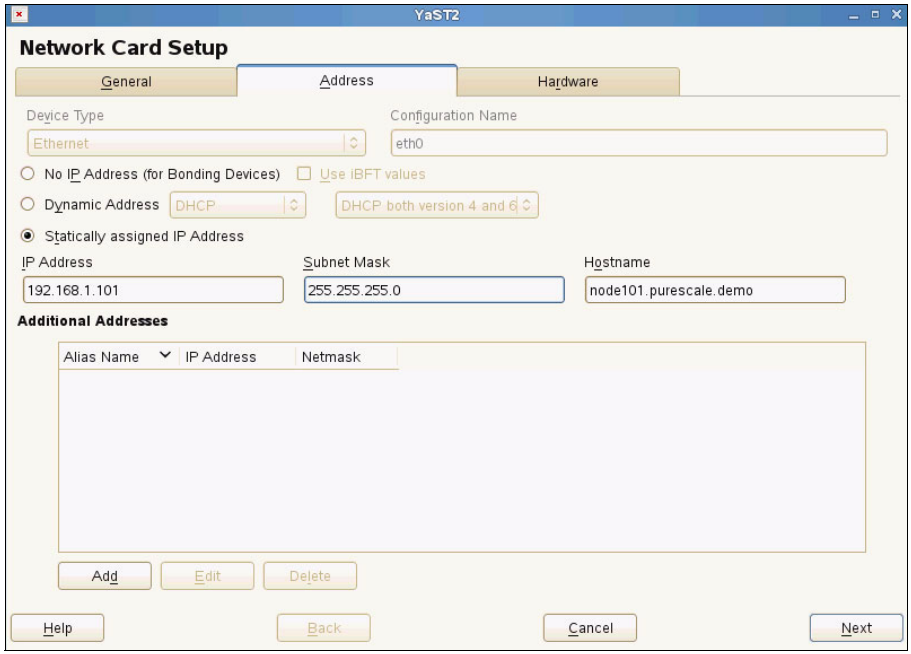

Figure 3-52 Network card setup

6. On the Hostname/DNS tab, ensure that the host name and domain name are correct. Click **OK** to continue.
7. Open a terminal session and restart the network by running the following command:

```
/etc/init.d/network restart
ifconfig
```

Important: You have finished Ethernet network setup on SUSE Linux Enterprise 11. Proceed to 3.2.7, “Step 7: Setting up the interconnect network” on page 132 to continue.

Red Hat Enterprise Linux 5.6

Attention: Complete these tasks on all servers in the cluster.

To set up the Ethernet network on Red Hat Enterprise Linux 5.6, complete the following steps:

1. If you have multiple Ethernet ports on your system, you need to detect which Ethernet port is used by the system. Ensure that you connected the server with Ethernet switch from one of the Ethernet ports on the server.
2. Open a terminal and run the following command. If there are multiple Ethernet ports on the server, run the command for each Ethernet port.

   ```
   ethtool eth1
   ethtool eth2
   ...
   ```

   If part of the command output reads `Link detected`, the port is connected to the switch by Ethernet connections. Take note of the port number so that you can assign a static IP address to the Ethernet port.

3. Open the network configuration tool (Figure 3-53) by clicking **System** → **Administration**.

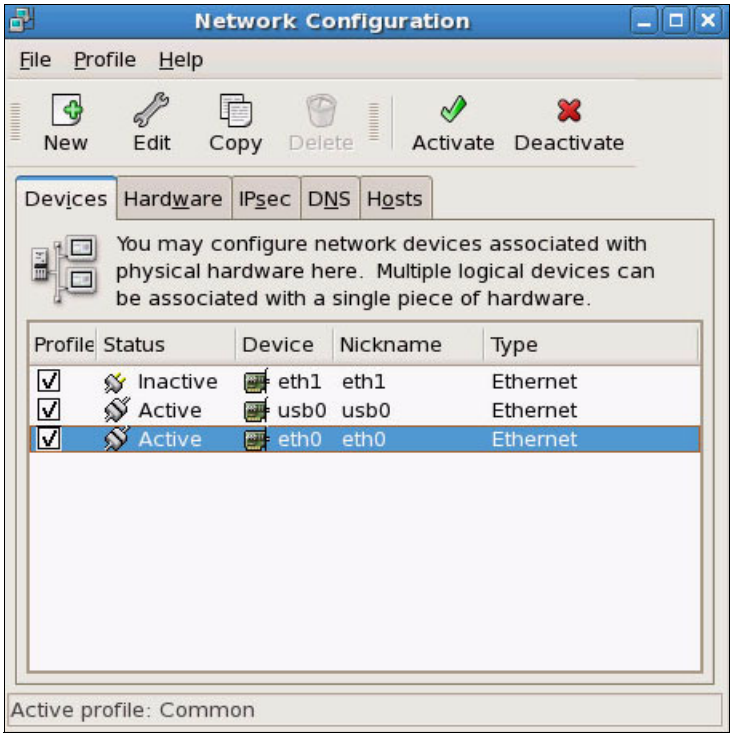

Figure 3-53 Network configuration

4. Choose the correct eth port and click **Edit** (Figure 3-54). Enter the correct IP address and subnet mask.

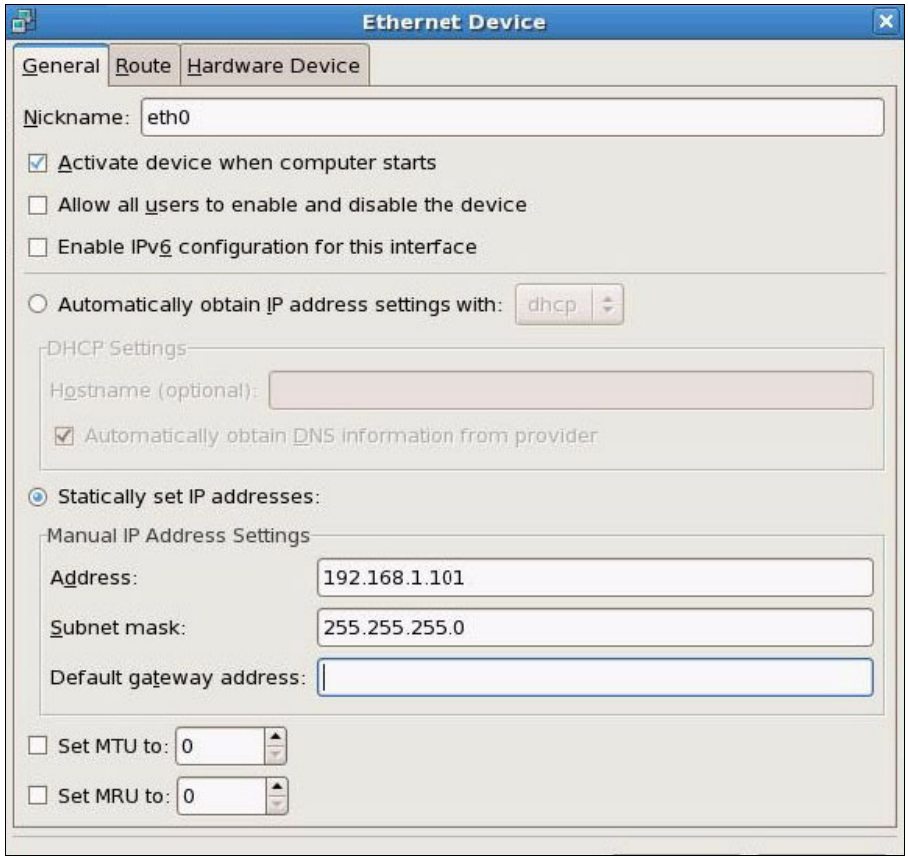

Figure 3-54 Ethernet device

5. When complete, you can exit the Network Configuration window. When prompted, you must restart the network to pick up the changes. To do so, run the following command:

```
/etc/init.d/network restart
```

The output is similar to the following output:

```
[root@node101 ~]# /etc/init.d/network restart
Shutting down interface eth0:
Shutting down interface usb0:
Shutting down loopback interface:
Bringing up loopback interface:
Bringing up interface eth0:
```

```
Bringing up interface usb0: [ OK ]
Determining IP information for usb0... done. [ OK ]
```

6. After you set up both servers, run the **ifconfig** command to check the network interface information.

Important: You have finished the Ethernet network setup on Red Hat Enterprise Linux 5.6. Proceed to 3.2.7, “Step 7: Setting up the interconnect network” on page 132 to continue.

Red Hat Enterprise Linux 6.1

Attention: Perform this task on all servers in the cluster.

To set up the Ethernet network on Red Hat Enterprise Linux 6.1, complete the following steps:

1. As root, open a terminal and run the following commands:
   - **chkconfig NetworkManager off**
   - **service NetworkManager stop**
2. Open a terminal and run **system-config-network**, and follow the wizard to configure the Ethernet network.
3. Run the following command in a terminal to restart network service and check the Ethernet interface information:

```
service network restart
ifconfig
```

Important: You have finished setting up the Ethernet network on the Red Hat Enterprise Linux 6.1 system and can continue to the next section.

3.2.7 Step 7: Setting up the interconnect network

An interconnect network is required to install the DB2 pureScale Feature on a Linux distribution. Cluster CFs support multiple cluster interconnects to help DB2 pureScale environments with high availability and scalability. This section explains how to set up the interconnect network with one cluster interconnect adapter port for each CF.

The DB2 pureScale Feature supports multiple high speed interconnect networks in one cluster. If you want to set up multiple high speed networks in one cluster, visit the DB2 Information Center for more information.

Important: On Linux and AIX, ensure that you configure a default gateway on all hosts that are part of the DB2 pureScale cluster.

This document uses the IP address and net mask listed in Table 3-4.

Table 3-4 Interconnect network IP address and net mask

Host	First node	Second node
Interconnect IP address	192.168.100.151	192.168.100.152
Interconnect netmask	255.255.255.0	255.255.255.0

The instructions here are for different combinations of operating systems and types of interconnect network. Find the correct instructions for your environment in the next sections:

- If you are using SUSE Linux Enterprise Server 11 with and InfiniBand network, see “Setting up an InfiniBand network on SUSE Linux Enterprise Server 11” on page 133.
- If you are using SUSE Linux Enterprise Server 11 with a 10 GbE network, see “Setting up the 10 GbE network on SUSE Linux Enterprise Server 11” on page 138.
- If you are using Red Hat Enterprise Linux 5.6 with an InfiniBand network, see “Setting up the InfiniBand network on Red Hat Enterprise Linux 5.6” on page 140.
- If you are using Red Hat Enterprise Linux 6.1 with a 10 GbE network, see “Setting up the 10 GbE network on Red Hat Enterprise Linux 6.1” on page 143.

Important: Complete these steps on all servers in the cluster.

Setting up an InfiniBand network on SUSE Linux Enterprise Server 11

To install the InfiniBand (OFED) driver on SUSE Linux Enterprise Server 11, you must install several `rpm` packages. You also must configure the SUSE updates repository for your version of SUSE Linux Enterprise Server.

To set up the InfiniBand network on SUSE Linux Enterprise Server 11, complete the following steps:

1. If you are using SUSE Linux Enterprise Server 11 SP1, install the following packages from the updates repository. Contact your system administrator to get a copy of these packages and use the **rpm** command to install the packages. If you are using SUSE Linux Enterprise Server 11 SP2, these packages are on the installation DVD.

   - compat-dapl-1.2.19-0.5.1
   - compat-dapl-32bit-1.2.19-0.5.1
   - dapl-32bit-2.0.30-0.5.1
   - dapl-doc-2.0.30-0.5.1
   - dapl-2.0.30-0.5.1
   - ibutils-32bit-1.5.4-0.3.3
   - Ibutils-1.5.4-0.3.3
   - infiniband-diags-1.5.7-0.3.2
   - libcxgb3-rdmav2-32bit-1.2.5-0.3.1
   - libcxgb3-rdmav2-1.2.5-0.3.1
   - libibcm-1.0.5-0.3.1
   - libibcm-32bit-1.0.5-0.3.1
   - libibcommon1-1.1.2_20090314-0.1.1
   - libibcommon1-32bit-1.1.2_20090314-0.1.1
   - libibmad5-1.3.6-0.3.1
   - libibmad5-32bit-1.3.6-0.3.1
   - libibumad3-1.3.6-0.3.1
   - libibumad3-32bit-1.3.6-0.3.1
   - libibverbs-1.1.4-0.3.1
   - libibverbs-32bit-1.1.4-0.3.1
   - libipathverbs-1.2-0.3.1
   - libipathverbs-32bit-1.2-0.3.1
   - libmlx4-rdmav2-1.0-5.21.1
   - libmlx4-rdmav2-32bit-1.0-5.21.1
   - libmthca-rdmav2-1.0.5-5.18.1
   - libmthca-rdmav2-32bit-1.0.5-5.18.1
   - libnes-rdmav2-1.1.0-0.3.1
   - librdmacm-1.0.13-0.3.1
   - librdmacm-32bit-1.0.13-0.3.1
   - libsdp-32bit-1.1.103-0.3.1
   - libsdp-1.1.103-0.3.1
   - mpi-selector-1.0.3-0.3.1
   - mstflint-1.4-2.25.1
   - ofed-doc-1.5.2-0.7.1
   - ofed-kmp-default-1.5.2_2.6.32.29_0.3-0.7.1
   - ofed-1.5.2-0.7.1
   - ofed-doc-1.5.2-0.7.1

- ofed-kmp-default-1.5.2_2.6.32.29_0.3-0.7.1
- opensm-32bit-3.3.7-0.5.1
- opensm-3.3.7-0.5.1
- ibvexdmtools-0.0.1-75.16.1
- qlvnictools-0.0.1-75.16.1
- sdpnetstat-1.60-5.22.1
- srptools-0.0.4-6.8.2
- ksh-93u-0.8.1.x86_64.rpm[1]

Attention: If you are using SUSE Linux Enterprise Server 11 SP1, you do not need to use the Software Management tool described in the next step. You can skip to step 3 on page 136 to continue.

[1] If you are using SUSE Linux Enterprise Server 11 SP2.

2. If you are using SUSE Linux Enterprise Server 11 SP2, open the Software Management tool (Figure 3-55) from the YaST control center, and search for the packages. If the packages are not installed, select the packages, and click **Accept** to install them.

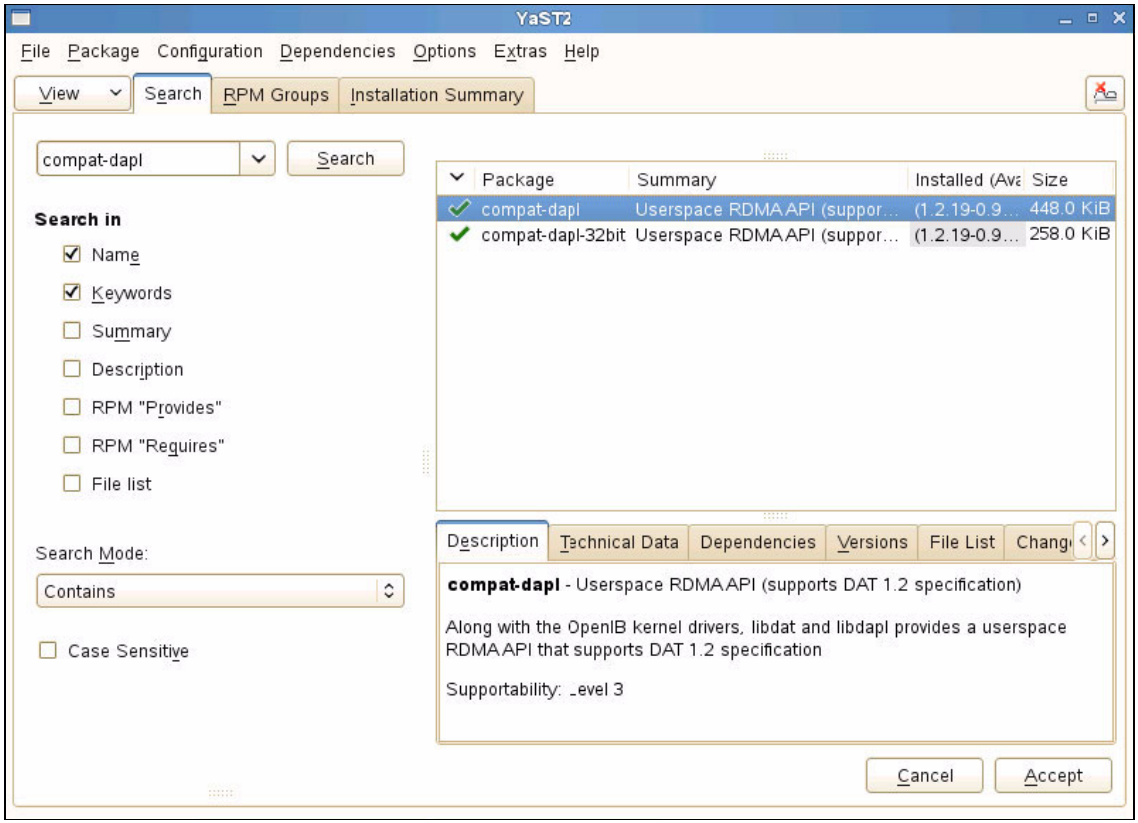

Figure 3-55 Software Management

3. Start a terminal session. Navigate to the `/etc/sysconfig/network/` folder and create a file called `ifcfg-ib0`. Add the following lines to the file:

```
DEVICE='ib0'
BOOTPROTO='static'
IPADDR= <Interconnect network IP address>
NETMASK= <Interconnect Netmask address>
STARTMODE='onboot'
WIRELESS='no'
```

For example, on node101, this file looks like as follows:

```
DEVICE='ib0'
BOOTPROTO='static'
IPADDR='192.168.100.151'
NETMASK='255.255.255.0'
STARTMODE='onboot'
WIRELESS='no'
```

4. Run the following command to start the InfiniBand adapter:

   ```
   ifup ib0
   ```

   The sample output is as follows:

   ```
   ib0 Device : Mellanox Technologies MT26428 [ConnectX VPI PCIe 2.0
   5GT/s - IB QDR / 10GigE] (rev b0)
   ```

5. Check the network status by running the following command:

   ```
   ifconfig
   ```

   Locate the ib0 entry in the output (Example 3-11).

Example 3-11 The ib0 entry

```
eth0      Link encap:Ethernet  HWaddr E4:1F:13:68:45:D4
          inet addr:192.168.1.101 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::e61f:13ff:fe68:45d4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:13640883 errors:0 dropped:128210 overruns:0 frame:0
          TX packets:5808497 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2719527853 (2593.5 Mb)  TX bytes:1648923163 (1572.5 Mb)
          Interrupt:28 Memory:96000000-96012800

ib0       Link encap:InfiniBand  HWaddr
80:00:00:48:FE:80:00:00:00:00:00:00:00:00:00:00:00:00:00:00
          inet addr:192.168.100.151 Bcast:192.168.100.255  Mask:255.255.255.0
          inet6 addr: fe80::202:c903:4a:b187/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:65520  Metric:1
          RX packets:3131461 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2998518 errors:0 dropped:13 overruns:0 carrier:0
          collisions:0 txqueuelen:256
          RX bytes:328529513 (313.3 Mb)  TX bytes:320819465 (305.9 Mb)
```

Important: You have completed the configuration of the InfiniBand network on SUSE Linux Enterprise Server 11. Continue to 3.2.8, “Step 8: Editing the hosts file” on page 146.

Setting up the 10 GbE network on SUSE Linux Enterprise Server 11

Attention: If you are using SUSE Linux Enterprise Server 11 SP1 or SP2, follow the instructions in “Setting up an InfiniBand network on SUSE Linux Enterprise Server 11” on page 133 to install the InfiniBand (OFED) packages before you continue here.

To set up the 10 GbE network on SUSE Linux Enterprise Server 11, complete the following steps:

1. To set up the 10 GbE network, you must configure the 10 GbE switch first:
   a. Find a cable to connect the micro-USB port on the switch to the serial port on your system, and connect your terminal-emulation software using the following configuration:

Default band rate	9600 bps
Character size	Eight characters
Parity	None
Stop bits	One
Data bits	Eight
Flow control	None

   b. When prompted for a password, enter the default `admin` password. If a blank pane displays, you might need to press Enter first before the password prompt appears.
   c. Enter the global configuration mode on the switch by running the following commands:
      - **`enable`**
      - **`configure terminal`**
   d. Create the interface IP address needed for the 10 GbE subnet on the CF. Pick an address that is on the same subnet as the interconnect network, and enter the information into the switch by running the following command:

      ```
      interface ip 1
      ip address 192.168.100.2
      ip netmask 255.255.255.0
      exit
      ```

   e. Next, ensure that the configuration is used each time the switch is rebooted by running the following commands:
      - **`copy running-config startup-config`**
      - **`exit`**

2. Determine the port name and the card name for the 10 GbE network adapter and card name. Open a terminal in SUSE Linux Enterprise 11 and run the following command:

   ```
   hwinfo --short --netcard | grep 10GigE
   ```

   The first section of the output is the port name for the adapter, and the second section is the card name for the adapter. In our setup, the port name is eth2 and card name is Mellanox MT26448 [ConnectX EN 10GigE, PCIe 2.9 5GT/s]. The output should be similar to Example 3-12.

   Example 3-12 Determine the 10 GbE network card name and port number

   ```
   node101 :~/ # hwinfo --short --netcard | grep 10GigE
   eth2  Mellanox MT26448 [ConnectX EN 10GigE, PCIe 2.0 5GT/s]
   ```

3. Open the `/etc/dat.conf` file in a text editor, and add the following line:

   ```
   ofa-v2-roe u2.0 nonthreadsafe default libdaplofa.so.2 dapl.2.0
   "<port name>" ""
   ```

   Ensure that you use correct value to replace <*port name*>.

4. Go to the `/etc/sysconfig/network/` folder and create a file called `ifcfg-<port name>`. Add the following lines to the file:

   ```
   BOOTPROTO='static'
   BROADCAST=''
   ETHTOOL_OPTIONS=''
   IPADDR=<Interconnect IP Address>
   MTU=''
   NAME=<Adapter card name>
   NETMASK=<Interconnect netmask address>
   NETWORK=''
   REMOTE_IPADDR=''
   STARTMODE='auto'
   USERCONTROL='no'
   ```

   On node101, create the sample `ifcfg-eth2` file as follows:

   ```
   BOOTPROTO='static'
   BROADCAST=''
   ETHTOOL_OPTIONS=''
   IPADDR='192.168.100.151'
   MTU=''
   NAME=' Mellanox MT26448 [ConnectX EN 10GigE, PCIe 2.0 5GT/s]'
   NETMASK='255.255.255.0'
   NETWORK=''
   REMOTE_IPADDR=''
   ```

```
STARTMODE='auto'
USERCONTROL='no'
```

5. Reboot the server so that your system picks up the OFED driver and connects to the interconnect network.

Important: You have completed setting up the 10 GbE network on SUSE Linux Enterprise Server 11. Continue to 3.2.8, “Step 8: Editing the hosts file” on page 146.

Setting up the InfiniBand network on Red Hat Enterprise Linux 5.6

To set up the InfiniBand network on Red Hat Enterprise Linux 5.6, complete the following steps:

1. Ensure that the OFED group packages were included when you installed the operating system installation. If they are not installed, use the Red Hat software management tools to install the package. The package is part of the base system packages.
2. Navigate to the `/etc/sysconfig/network-scripts/` directory and create a file called `ifcfg-ib0`. Add the following lines to the file:

   ```
   DEVICE=ib0
   BOOTPROTO=’static’
   IPADDR=<InfiniBand IP address>
   NETMASK=<InfiniBand Netmask>
   STARTMODE=’onboot’
   WIRELESS=’no’
   ONBOOT=’yes’
   ```

   The following example shows this file with the InfiniBand information completed:

   ```
   DEVICE=ib0
   BOOTPROTO=’static’
   IPADDR=’192.168.100.151’
   NETMASK=’255.255.255.0’
   STARTMODE=’onboot’
   WIRELESS=’no’
   ONBOOT=’yes’
   ```

3. Edit the `/etc/sysconfig/network-scripts/ifup-ib` file, find line 79, and change it from:

   ```
   Line 79: if [ -n "${MTU}" -a $MTU -gt 2044 ]; then
   ```

   To:

   ```
   Line 79: if [ -n "${MTU}" ] && [ $MTU -gt 2044 ] ; then
   ```

4. Start and enable the **openibd** service by running the following command:

   ```
   service openibd start
   chkconfig openibd on
   ```

5. Restart the network by running the following command:

   ```
   service network restart
   ```

6. Run the following command in a terminal, and you discover that the network interface ib0 is enabled:

   ```
   ifconfig
   ```

   Locate the `ib0` entry from the output (Example 3-13).

Example 3-13 The ib0 entry

```
ib0       Link encap:InfiniBand  HWaddr
80:00:00:48:FE:80:00:00:00:00:00:00:00:00:00:00:00:00:00:00
          inet addr:192.168.100.151  Bcast:192.168.100.255  Mask:255.255.255.0
          inet6 addr: fe80::202:c903:e:9d99/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:2044  Metric:1
          RX packets:1299764 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1364497 errors:0 dropped:19 overruns:0 carrier:0
          collisions:0 txqueuelen:256
          RX bytes:130990014 (124.9 MiB)  TX bytes:985719928 (940.0 MiB)
```

7. Test the connection by pinging other hosts through the InfiniBand interface.

8. Open the Network Configuration window (Figure 3-56), select device ib0, and click **Edit** to edit it.

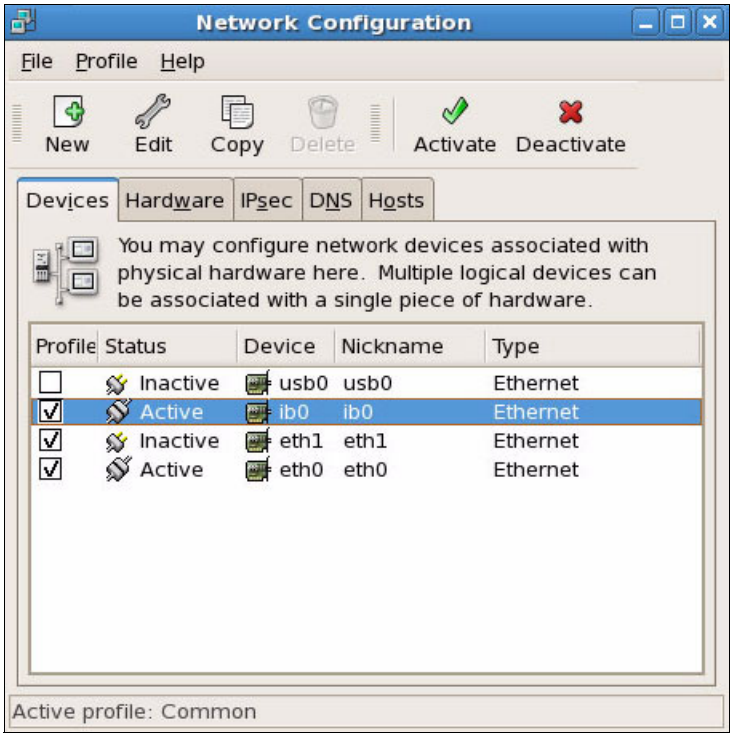

Figure 3-56 Network configuration

9. Ensure that the **Activate device when computer starts** option is selected for the InfiniBand (ib0) device (Figure 3-57). Click **OK**.

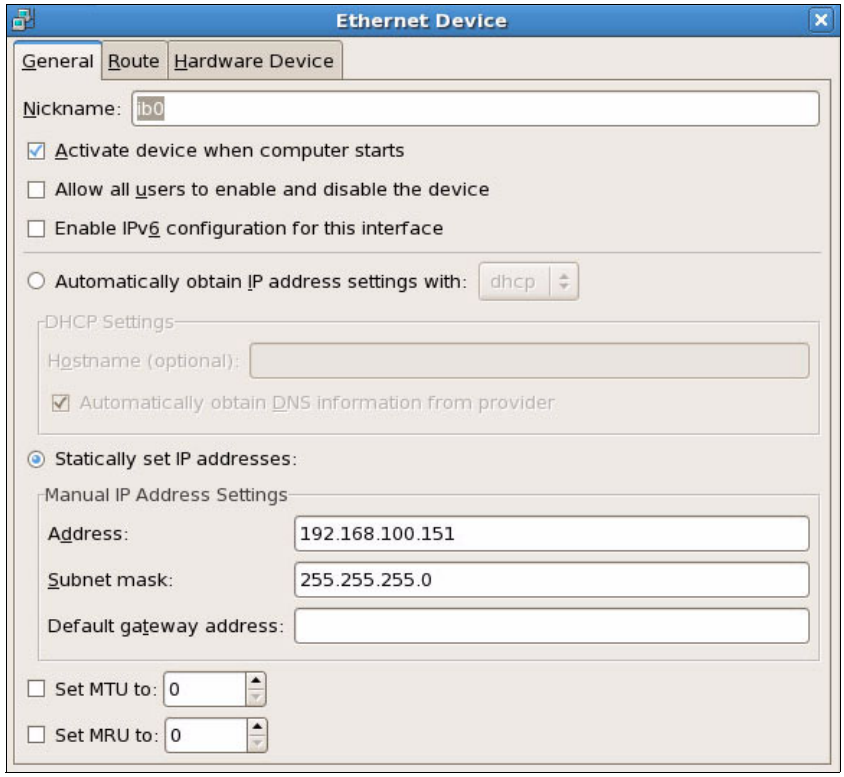

Figure 3-57 Ethernet device

Important: You have finished configuring the InfiniBand network on Red Hat Enterprise Linux 5.6. Continue to 3.2.8, “Step 8: Editing the hosts file” on page 146.

Setting up the 10 GbE network on Red Hat Enterprise Linux 6.1

To set up the 10 GbE network on Red Hat Enterprise Linux 6.1, complete the following steps:

1. Follow the instructions in step 1 on page 138 in “Setting up the 10 GbE network on SUSE Linux Enterprise Server 11” on page 138 to set up the 10 GbE network switch first.
2. Ensure that you registered your server on the Red Hat Network.

3. Log in to the Red Hat Network. Then, under the System tab, click your server and select **Alter Channel Subscriptions**.
4. Add the following channels to each of your servers:
   - Red Hat Enterprise Linux Server High Performance Networking Debug info (v.6 for x86_64)
   - Red Hat Enterprise Linux Server High Performance Networking (v.6 for x86_64)
5. Open a terminal, and run the following command to install the InfiniBand Support group package:

   ```
   yum groupinstall “InfiniBand Support”
   ```

6. Install the `infiniband-diags` package for diagnostics utilities by running the following command:

   ```
   yum install infiniband-diags
   ```

7. Verify that global pause flow control is enabled by running the following commands:
   - **`cat /sys/module/mlx4_en/parameters/pfctx`**
   - **`cat /sys/module/mlx4_en/parameters/pfcrx`**
8. If the output is not zero, run the following commands to change it:
   - **`echo "options mlx4_en pfctx=0 pfcrx=0" >> /etc/modprobe.d/modprobe.conf`**
   - **`reboot`**
9. Install the required i686 packages by running **`yum install <package name>`**. If the output states that the i686 package version does not match the x86_64 packages version, upgrade the current x86_64 package first and then install the i686 package again by running the following commands:
   - **`yum upgrade pam.x86_64`**
   - **`yum install pam.i686`**

   Install the following i686 packages:
   - `pam*.i686`
   - `libstdc++*.i686`
   - `libibcm.i686`
   - `libibverbs-rocee.i686`
   - `librdmacm.i686`
   - `libcxgb3.i686`
   - `libibmad.i686`

- libibumad.i686
- libmlx4-rocee.i686
- libmthca.i686
- libibcommon.i686
- compat-dapl
- compat-libstdc++-33.i686
- infiniband-diags
- libibcommon
- mstflint
- perftest
- qperf
- srptools

10. After installing the packages, turn on the RDMA service by running the following command:

    ```
    chkconfig rdma on
    ```

11. Find the correct Ethernet port by running the following command:

    ```
    cat /etc/udev/rules.d/*-persistent* | awk '/mlx/||/eth/'
    ```

    The output displays port1 as `eth#`. Write down the eth port number.

12. Create a `/etc/sysconfig/network-scripts/ifcfg-eth#` file. Change the number sign (#) in the file name to the eth port number that is returned from the last step. It might already be created by the OS. Then, input the following content into this file:

    ```
    DEVICE=eth#
    HWADDR=<HWADDR>
    TYPE=Ethernet
    IPADDR='192.168.100.151'
    NETMASK='255.255.255.0'
    MUT=''
    NAME='eth#'
    NETWORK=''
    REMOTE_IPADDR=''
    STARTMODE='auto'
    USERCONTROL='no'
    ```

    Make sure to change the `DEVICE`, `HWADDR`, and `IPADDR` parameters to the appropriate values for the different servers in your environment.

13. Add the following line to the `/etc/rdma/dat.conf` file, and change the eth# to the eth port number from step 11 on page 145.

    ```
    ofa-v2-roe u2.0 nonthreadsafe default libdaplofa.so.2 dapl.2.0 "<eth
    port number> 0" ""
    ```

14. Comment out all other network interfaces in the `/etc/rdma/dat.conf` file, such as the regular Ethernet port.
15. Run the following commands to restart the RDMA service and network:
    - **`service rdma restart`**
    - **`service network restart`**

> **Important:** You have configured the 10 GbE network on Red Hat Enterprise Linux 6.1. Continue to 3.2.8, “Step 8: Editing the hosts file” on page 146.

3.2.8 Step 8: Editing the hosts file

The `hosts` file is used to map host names to IP addresses in an operating system. To communicate between servers using both an Ethernet network and an interconnect network, put the full host names and IP addresses for both networks in the `/etc/hosts` file for the DB2 pureScale Feature in the Linux environment.

> **Attention:** This task is the same on SUSE Linux Enterprise Server 11 and Red Hat Enterprise Linux 5 and 6. Complete this task on all servers in the cluster.

To add the host names and IP addresses to the `/etc/hosts` file, complete the following steps:

1. Open the `/etc/hosts` file.
2. Add host names and IP addresses for all the servers in the cluster as shown in the following sample:

   ```
   192.168.1.101 node101.purescale.demo node101
   192.168.1.102 node102.purescale.demo node102

   192.168.100.151 node101-ib
   192.168.100.152 node102-ib
   ```

Important: If your system is using Red Hat Enterprise Linux 5.6 and the full host name is assigned to the lookback IP address, remove that host name from that list (Figure 3-58). Otherwise, the host name might resolve to the wrong IP address.

```
root@node101:~
File  Edit  View  Terminal  Tabs  Help
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1               node101.purescale.demo node101 localhost.localdomain loc
alhost
::1             localhost6.localdomain6 localhost6

192.168.1.101   node101.purescale.demo node101
192.168.1.102   node102.purescale.demo node102

192.168.160.151 node101-ib
192.168.160.152 node102-ib
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                          10,27         All
```

Figure 3-58 Hosts file

Important: You have finished configuring hosts file. Continue to the next step.

3.2.9 Step 9: Installing and setting up OpenSSH

Open Secure Shell (OpenSSH) provides encrypted communication between servers by using SSH protocols. For the DB2 pureScale Feature, password-less SSH access for the root user is required. The DB2 installation process sets up the password-less SSH for the instance user, if the user is not set up for password-less SSH. This section describes how to set up password-less SSH access for the root user.

Attention: This task is the same on both SUSE Linux Enterprise Server 11 and Red Hat Enterprise Linux 5 and 6. Complete these tasks on all servers in the cluster.

To install and set up OpenSSH, complete the following steps:

1. If you are using Red Hat Enterprise Linux 6.1, disable SELinux and the firewall first:
   a. Edit the `/etc/selinux/config` file and change the SELinux value to the following value:

      ```
      SELinux=disabled
      ```

   b. Run the following commands to disable the firewall:

      - **chkconfig iptables off**
      - **service iptables stop**

   c. Reboot the server to pick up the changes.
2. Generate the SSH key by running the following command. Accept the default path and do not set a passphrase.

   ```
   ssh-keygen -t rsa
   ```

3. Copy the SSH key to all the hosts, include the host you are using, by running the following commands. You might be prompted for the root password.

   - **ssh-copy-id –i <path of the ssh key> root@<full hostname>**
   - **ssh-copy-id –i <path of the ssh key> root@<hostname>**
   - **ssh-copy-id –i <path of the ssh key> root@<netname>**

   For example:

   - **ssh-copy-id -i ~/.ssh/id_rsa.pub root@node101.purescale.demo**
   - **ssh-copy-id -i ~/.ssh/id_rsa.pub root@node101**
   - **ssh-copy-id -i ~/.ssh/id_rsa.pub root@node101-ib**
   - **ssh-copy-id -i ~/.ssh/id_rsa.pub root@node102.purescale.demo**
   - **ssh-copy-id -i ~/.ssh/id_rsa.pub root@node102**
   - **ssh-copy-id -i ~/.ssh/id_rsa.pub root@node102-ib**

Now, you can log in without a password using SSH on all hosts in your cluster.

Important: You have finished setting up password-less SSH. Proceed to the next section.

3.2.10 Step 10: Blacklisting the system watchdog modules

The DB2 pureScale Feature uses its own watchdog module. Therefore, you must blacklist the system watchdog modules to avoid any conflict. This section describes how to blacklist these modules.

Important: Complete this task on all servers in the cluster.

SUSE Linux Enterprise Server 11

To blacklist the system watchdog module on SUSE Linux Enterprise Server 11, complete the following steps:

1. Add the following lines to the `/etc/modprobe.d/blacklist` file:

   ```
   alias iTCO_wdt off
   alias iTCO_vendor_support off
   ```

2. Reboot the server to pick up the changes.

Important: You added the system watchdog module to the blacklist on SUSE Linux Enterprise Server 11. Continue to 3.2.11, “Step 11: Modifying the mlx4_core parameter” on page 149.

Red Hat Enterprise Linux 5 and 6

To blacklist the system watchdog module on Red Hat Enterprise Linux 5 and 6, complete the following steps:

1. Add the following lines to the `/etc/modprobe.d/blacklist.conf` file:

   ```
   blacklist iTCO_wdt
   blacklist iTCO_vendor_support
   ```

2. Reboot the server to pick up the changes.

Important: You added the system watchdog module to the blacklist on Red Hat Enterprise Linux 5 and 6. Continue to the next section.

3.2.11 Step 11: Modifying the mlx4_core parameter

For the cluster CF to start with higher memory regions for communication buffers, increase the Mellanox HCA driver mlx4_core `mlx4_core` parameter to 7.

Attention: Complete this task on all servers in the cluster.

SUSE Linux Enterprise Server 11

To increase the `mlx4_core` parameter on SUSE Linux Enterprise Server 11, complete the following steps:

1. Run the following command and reboot your server for the changes to take effect:

   ```
   echo “options mlx4_core log_mtts_per_seg=7” >>
   /etc/modprobe.conf.local
   ```

2. After rebooting, run the following command to verify that the change took effect:

   ```
   cat /sys/module/mlx4_core/parameters/log_mtts_per_seg
   ```

Important: You have finished this task. Continue to 3.4, “Installing the DB2 pureScale Feature” on page 168.

Red Hat Enterprise Linux 5 and 6

To increase the `mlx4_core` parameter on Red Hat Enterprise Linux 5 and 6, complete the followings steps:

1. Run the following command and reboot your server for the changes to take effect:

   ```
   echo “options mlx4_core log_mtts_per_seg=7” >> /etc/modprobe.conf
   ```

2. After rebooting, run the following command to verify that the change took effect:

   ```
   cat /sys/module/mlx4_core/parameters/log_mtts_per_seg
   ```

Important: You have finished this task. Continue to 3.4, “Installing the DB2 pureScale Feature” on page 168.

3.3 Configuring an AIX system to install the DB2 pureScale Feature

This section contains detailed instructions about how to prepare an AIX environment for the DB2 pureScale Feature installation. The instructions include how to upgrade and verify the platform firmware level, how to set up and configure the InfiniBand network, how to install and set up OpenSSH, and other installation and setup tasks.

You can perform all the steps in the checklist to ensure that your system is ready for the DB2 10.1 installation. You can download all the software (firmware, technology level package, and service packs) used in this section from the IBM Fix Central website at:

http://www.ibm.com/support/fixcentral/

Figure 3-59 shows the overall AIX environment architecture of the DB2 pureScale cluster used in this section.

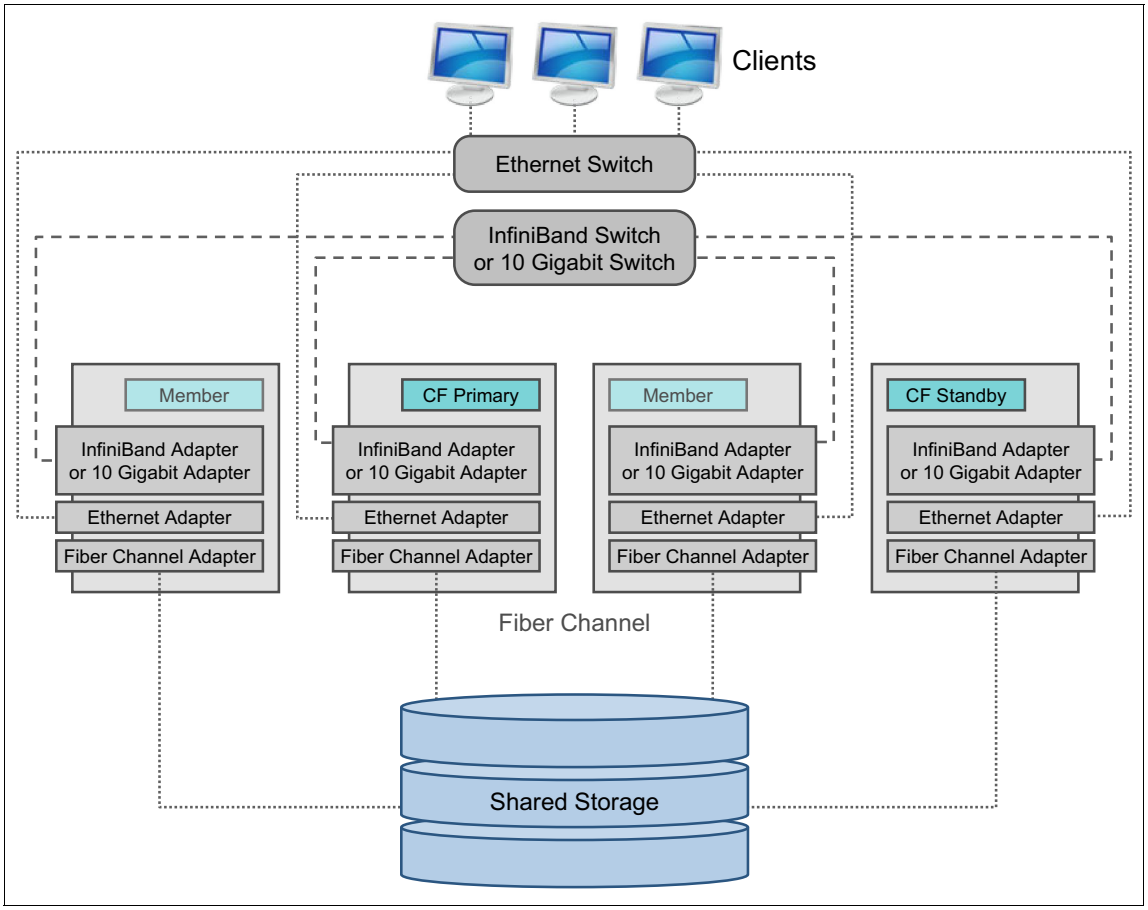

Figure 3-59 DB2 pureScale cluster hardware architecture in the AIX environment

The examples in this section use four LPARs as two DB2 members and two cluster CFs. The base operating system is AIX V7.1. The examples apply technology level 1 and SP levels 1, 2, and 3 on each system. All LPARs are connected to shared storage through Fibre Channel adapter cables. The LPARs also configure a Host Channel Adapter (HCA) for a high speed interconnect network. Contact your AIX system administrator for assistance.

3.3.1 Preinstallation checklist

Before you install the DB2 pureScale Feature on an AIX system, verify whether the AIX system is ready for installation by completing the following steps on all the hosts in the cluster:

1. DB2 pureScale instances require specific users and groups, including fenced users. You can create the users before starting the DB2 Setup wizard or have the wizard create them for you as you progress through the wizard. If you are not creating or modifying instances, you can create the required users after completing the installation.
2. Ensure that your platform firmware level satisfies the installation prerequisites. Follow the instructions in 3.3.2, “Validating the platform firmware level” on page 155 to verify that the platform firmware is installed at the correct version level.
3. Ensure that the required AIX Version, Technology Level (TL), and service pack (SP) are installed according to the software prerequisites. The **oslevel -s** command returns the operating system level followed by the technology level followed by the service pack level. The following sample output is returned for an AIX V7.1 system with Technology Level 1 and Service Pack 3:

   ```
   7100-01-03-1207
   ```

4. Ensure that both the Ethernet and InfiniBand network adapters are defined on each server. Run **ifconfig -l** to list all available network adapters.

   This sample output shows an Ethernet network adapter and an InfiniBand network adapter:

   ```
   en0 ib0 lo0
   ```

5. Ensure that User Direct Access Programming Library (uDAPL) is installed and configured at the level specified for your system in the software prerequisites. The following example illustrates the commands used to verify the uDAPL configuration, run on a system with AIX V7.1 SP3:

   ```
   lslpp -l bos.mp64 devices.chrp.IBM.lhca.rte
   devices.common.IBM.ib.rte udapl.rte
   ```

The output of **lslpp** shows the file set name, the level installed on the system, the state of the installation, and a short description of the file set. Ensure that the level shows the AIX version followed by the TL level (7.1.0 in this example). The state should be either APPLIED, COMMITTED, or EFIXLOCKED. If the state is EFIXLOCKED, an interim fix is added to the system. The output of the command should be similar to Figure 3-22 on page 85.

Example 3-14 Check the uDAPL packages

```
Fileset                      Level  State      Description
----------------------------------------------------------------------------
Path: /usr/lib/objrepos
bos.mp64                  7.1.0.15  APPLIED    Base Operating System 64-bit
Multiprocessor Runtime

devices.chrp.IBM.lhca.rte 7.1.0.15  APPLIED    Infiniband Logical HCA Runtime
Environment

devices.common.IBM.ib.rte 7.1.0.15  APPLIED    Infiniband Common Runtime Environment
                                      EFIXLOCKED
udapl.rte                 7.1.0.15  COMMITTED  uDAPL
Path: /etc/objrepos
bos.mp64                  7.1.0.15  APPLIED    Base Operating System 64-bit
Multiprocessor Runtime
devices.chrp.IBM.lhca.rte 7.1.0.0  COMMITTED  Infiniband Logical HCA Runtime
Environment
devices.common.IBM.ib.rte 7.1.0.15  APPLIED    Infiniband Common Runtime Environment
                                      EFIXLOCKED
udapl.rte                 7.1.0.15  COMMITTED  uDAPL
```

6. Check the /etc/dat.conf file and ensure that there is a line like the following one and ensure that the device path /dev/iba0, port number 1, and InfiniBand adapter name ib0 are correct.

   ```
   hca0 u2.0 nonthreadsafe default
   /usr/lib/libdapl/libdapl2.a(shr_64.o) IBM.1.1 "/dev/iba0 1 ib0" " "
   ```

7. Follow the instructions in “Step 9: Installing and setting up OpenSSH” on page 147 to set up OpenSSH on all the hosts. Complete the followings step to ensure that OpenSSH is installed and password-less access for the root user is configured on each host.

As root, validate ssh access between all hosts. From the current host, run **hostname** on all hosts in the cluster through the **ssh** command. The result of the **hostname** command matches the host name identified in the **ssh** command verifies password-less ssh access between the two hosts (Example 3-15).

Example 3-15 Validate the password-less ssh access

```
root@democf1:/> ssh democf1 hostname
democf1
root@democf1:/> ssh democf1-ib0 hostname
democf1
root@democf1:/> ssh democf2 hostname
democf2
root@democf1:/> ssh democf2-ib0 hostname
democf2
root@democf1:/> ssh demom0 hostname
demom0
root@democf1:/> ssh demom0-ib0 hostname
demom0
root@democf1:/> ssh demom1 hostname
demom1
root@democf1:/> ssh demom1-ib0 hostname
demom1
```

8. Run the following command to verify that the C++ runtime level satisfies the prerequisites:

   `lslpp -l xlC.rte`

   The `AIX xLC.rte` requirement for DB2 10.1 installation is at least 11.1.0.1. The output of the command should be similar to Example 3-16.

Example 3-16 Verify the C++ runtime level

```
Fileset                      Level  State      Description
----------------------------------------------------------------------------
Path: /usr/lib/objrepos
xlC.rte                   11.1.0.1  COMMITTED  XL C/C++ Runtime
```

9. Ensure that the shared disks have the same physical volume identifier (PVID) on all hosts. Compare these results between each host in the DB2 pureScale cluster. The minimum number of shared disks is three, based on your storage needs. You can find the detailed instructions in 3.3.7, “Configuring a physical volume identifier for an IBM DB2 pureScale instance” on page 164.

10. Ensure that the I/O completion ports (IOCPs) are installed and configured. You can find detailed instructions in 3.3.8, “Configuring the I/O completion ports” on page 166.

11. As root, ensure that the `/tmp` directory has at least 5 GB of free space. Run **`df -g /tmp`**, which shows the free space in the `/tmp` directory:

```
Filesystem    GB blocks      Free %Used    Iused %Iused Mounted on
/dev/hd3           8.50      6.26   27%     2358     1% /tmp
```

You can run **`chfs`** to extend the file system. For example, you can add 5 GB of disk space to the `/tmp` folder by running the following command:

```
chfs -a size=+5G /tmp
```

3.3.2 Validating the platform firmware level

To verify whether the platform firmware version satisfies the prerequisites, complete the following steps:

1. Run **`prtconf`** to check the system model, firmware level, and processor type (Example 3-17).

Example 3-17 Check the platform firmware version

```
System Model: IBM,9179-MHB
Machine Serial Number: 10DBBDE
Processor Type: PowerPC_POWER7
Processor Implementation Mode: POWER 7
Processor Version: PV_7_Compat
Number Of Processors: 2
Processor Clock Speed: 3864 MHz
CPU Type: 64-bit
Kernel Type: 64-bit
LPAR Info: 53 coralpib269
Memory Size: 10240 MB
Good Memory Size: 10240 MB
Platform Firmware level: AM730_035
Firmware Version: IBM,AM730_035
Console Login: enable
Auto Restart: true
Full Core: false
```

2. Compare the **`prtconf`** command output to the prerequisites required platform firmware level.

3. If your system firmware level does not meet the requirement, go the IBM Fix Central website, and download the necessary firmware and apply it through the Hardware Management Console (HMC). If you do not know how to do this task, contact your AIX system administrator.

3.3.3 Installing the technology level and service packs

This section explains how to install the technology level and SPs for the DB2 10.1 installation. Complete these instructions after installing the AIX V7.1 base operating system.

Before you begin: According to the DB2 10.1 Information Center, the minimum software requirement for InfiniBand on the AIX V7.1 operating system is Technology Level 1 and SP 1. This section explains how to install Technology Level 1 and SPs 1, 2, and 3 on the AIX V7.1 system. Ensure that you download the required software from the IBM Fix Central website.

To install the technology levels and SP levels, complete the following steps:

1. Log in to the system.
2. Run **`oslevel -s`** and check the Technology Level and service pack version. This example produces the following output:

   ```
   7100-00-01-1037
   ```

3. Navigate to the directory that contains the Technology Level 1 software. Run **`smitty install`** to view the menu shown in Example 3-18.

Example 3-18 Smitty install menu

```
                Software Installation and Maintenance

Move cursor to desired item and press Enter.

  Install and Update Software
  List Software and Related Information
  Software Maintenance and Utilities
  Software Service Management
  Relocatable Software Installation and Maintenance
  Network Installation Management
  EZ NIM (Easy NIM Tool)
  System Workload Partition Software Maintenance
  System Backup Manager
  Alternate Disk Installation
```

```
  EFIX Management
  Thin Server Maintenance
```

4. Select the Install and Update Software option and then the Update Installed Software to Latest Level (Update All) option.
5. Enter the technology level directory path in the field (Example 3-19) and press Enter to continue.

Example 3-19 Update installed software

```
Update Installed Software to Latest Level (Update All)

Type or select a value for the entry field.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* INPUT device / directory for software
[/tmp/7_1_1_0_tl/]
```

6. Press the Tab key to switch the ACCEPT new license agreements option from `no` to `yes` (Example 3-20) and press Enter twice to start the installation.

Example 3-20 Update the installed software

```
Update Installed Software to Latest Level (Update All)

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                          [Entry
Fields]
* INPUT device / directory for software                  .
* SOFTWARE to update                                     _update_all
  PREVIEW only? (update operation will NOT occur)        no
  COMMIT software updates?                               yes
  SAVE replaced files?                                   no
  AUTOMATICALLY install requisite software?              yes
  EXTEND file systems if space needed?                   yes
  VERIFY install and check file sizes?                   no
  DETAILED output?                                       no
  Process multiple volumes?                              yes
  ACCEPT new license agreements?                         yes
  Preview new LICENSE agreements?                        no

  WPAR Management
      Perform Operation in Global Environment            yes
```

```
Perform Operation on Detached WPARs              no
    Detached WPAR Names                         [_all_wpars]
Remount Installation Device in WPARs             yes
Alternate WPAR Installation Device              []
```

7. Use the same method to install SPs 1, 2, and 3.
8. Run **oslevel -s** after installation. The output is as follows:

   7100-01-03-1207

3.3.4 Configuring the InfiniBand network

A high-speed interconnect network is required for the DB2 pureScale Feature. This section uses the InfiniBand network as an example of how to configure the interconnect network. The instructions include how to install the uDAPL package, how to set up the IP address for InfiniBand, and how to set up the InfiniBand communication manager.

Ensure that you downloaded the required uDAPL level from the IBM Fix Central website before you complete the instructions in this section. For more information about required the uDAPL level, check the installation prerequisites for the DB2 pureScale Feature on AIX in the IBM DB2 10.1 Information Center.

> **Important:** On Linux and AIX, ensure that a default gateway is configured on all hosts that are part of the DB2 pureScale cluster.

To configure the InfiniBand network, complete the following steps:

1. Log in as root, and navigate to the directory that contains the uDAPL package. If the AIX system has installed software from this directory previously, remove any `.toc` file that was created from the previous installation. Then, run **smitty install** from the uDAPL directory.
2. Select Install and Update Software, and then click Install Software. Enter '.' (the current directory) as the input device. Example 3-21 shows a sample panel. Press Enter to continue.

   Example 3-21 Install Software

```
                            Install Software
Type or select a value for the entry field.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* INPUT device / directory for software              [.]
```

3. Ensure that you change the Accept new license agreement value to `Yes` (Example 3-22) and press Enter to start the installation.

Example 3-22 Software installation and maintenance

```
Install Software

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry
Fields]
* INPUT device / directory for software                 .
* SOFTWARE to install                                  [_all_latest]
  PREVIEW only? (install operation will NOT occur)      no
  COMMIT software updates?                              yes
  SAVE replaced files?                                  no
  AUTOMATICALLY install requisite software?             yes
  EXTEND file systems if space needed?                  yes
  OVERWRITE same or newer versions?                     no
  VERIFY install and check file sizes?                  no
  Include corresponding LANGUAGE filesets?              yes
  DETAILED output?                                      no
  Process multiple volumes?                             yes
  ACCEPT new license agreements?                        yes
  Preview new LICENSE agreements?                       no

  WPAR Management
      Perform Operation in Global Environment           yes
      Perform Operation on Detached WPARs               no
          Detached WPAR Names                          [_all_wpars]
      Remount Installation Device in WPARs              yes
      Alternate WPAR Installation Device               []
```

4. Verify that your system has the correct uDAPL and InfiniBand file sets by running the following command. The command output varies depending on the different AIX version, Technology Level, and service pack level. The sample output shown in Example 3-23 is based on AIX V7.1 Technology Level 1 and SP 3.

```
lslpp -l bos.mp64 devices.chrp.IBM.lhca.rte
devices.common.IBM.ib.rte udapl.rte
```

Example 3-23 Verify the uDAPL level

```
Fileset                      Level  State      Description
  ----------------------------------------------------------------------------
```

```
Path: /usr/lib/objrepos
  bos.mp64                   7.1.1.3  COMMITTED  Base Operating System 64-bit
                                                 Multiprocessor Runtime
  devices.chrp.IBM.lhca.rte  7.1.1.3  COMMITTED  Infiniband Logical HCA Runtime
                                                 Environment
  devices.common.IBM.ib.rte  7.1.1.3  COMMITTED  Infiniband Common Runtime
                                                 Environment
  udapl.rte                 7.1.1.15  COMMITTED  uDAPL

Path: /etc/objrepos
  bos.mp64                   7.1.1.3  COMMITTED  Base Operating System 64-bit
                                                 Multiprocessor Runtime
  devices.chrp.IBM.lhca.rte  7.1.0.0  COMMITTED  Infiniband Logical HCA Runtime
                                                 Environment
  devices.common.IBM.ib.rte  7.1.1.3  COMMITTED  Infiniband Common Runtime
                                                 Environment
  udapl.rte                 7.1.1.15  COMMITTED  uDAPL
```

5. Reboot the system by running **shutdown -Fr**.
6. If your AIX system administrator never set up InfiniBand, you must configure the InfiniBand subsystem. Run **smitty icm** and:
   a. Select Add an InfiniBand Communication Manager.
   b. Press Enter and wait until the command completes.
   c. Exit the smitty interface.
7. Run **smitty inet** to configure the IP addresses.
8. Select Change/Show Characteristics of a Network Interface, and then select the first InfiniBand interface, ib0.
9. Set the values for the Network Interface Name, INTERNET ADDRESS (dotted decimal), Network Mask, HCA Adapter, and Adapter's port number fields. You can use the default values for the other fields. Our example is in Example 3-24.

Example 3-24 Change/Show an InfiniBand Interface

```
Change / Show an IB Interface

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry
Fields]
  Network Interface Name                              ib0
  INTERNET ADDRESS (dotted decimal)                  [10.10.0.34]
```

```
  Network MASK (hexadecimal or dotted decimal)       [255.255.255.0]
  IPV6  ADDRESS (colon separated)                    []
  Prefixlength                                       []
  HCA Adapter                                        [iba0]
* Adapter's port number                              [1]
  Partition Key                                      [0xFFFF]
  MTU                                                [65532]
  Queue Sizes                                        [4000]
  QKey                                               [0x1E]
  Superpacket                                         off
  Interface Specific Network Options
     ('NULL' will unset the option)
    rfc1323                                          [1]
    tcp_mssdflt                                      []
    tcp_nodelay                                      [1]
    tcp_recvspace                                    [262144]
    tcp_sendspace                                    [262144]
  Current STATE                                       up
  Apply change to DATABASE only                       no
```

10. Press Enter to save the changes.
11. Reboot the system by running **shutdown -Fr**.
12. Edit the `/etc/dat.conf` file and add the line shown in Example 3-25. Modify the `/dev/iba0` device, port number 1, and InfiniBand adapter name `ib0` if you are using different values.

Example 3-25 Sample line in the dat.conf file

```
hca0 u2.0 nonthreadsafe default
/usr/lib/libdapl/libdapl2.a(shr_64.o) IBM.1.1 "/dev/iba0 1 ib0" " "
```

13. Run **lsdev -C | grep ib** to verify the InfiniBand subsystem and verify that the InfiniBand components are in the Available state (Example 3-26).

Example 3-26 Verify the InfiniBand state

```
ib0        Available       IP over InfiniBand Network Interface
iba0       Available       InfiniBand Host Channel Adapter
icm        Available       InfiniBand Communication Manager
```

14. Run the **ibstat -v** command and ensure that the InfiniBand ports are active and the links are up. The sample output is shown in Example 3-27.

Example 3-27 Verify that the InfiniBand ports and links are available

```
-------------------------------------------------------------------
 IB PORT 1 INFORMATION (iba0)
-------------------------------------------------------------------
Global ID Prefix:                       fe.80.00.00.00.00.00.00
Local ID (LID):                         003a
Local Mask Control (LMC):               0000
Logical Port State:                     Active
Physical Port State:                    Active
Physical Port Physical State:           Link Up
Physical Port Speed:                    5.0G
Physical Port Width:                    4X
Maximum Transmission Unit Capacity:     2048
Current Number of Partition Keys:       1
Partition Key List:
  P_Key[0]:                             ffff
Current Number of GUID's:               1
Globally Unique ID List:
  GUID[0]:                              00.02.55.00.70.1e.70.03
```

15. You can **ping** other hosts in the same IP subnet through InfiniBand interfaces to verify the connection.

3.3.5 Editing the hosts file

The `hosts` file is used to map host names to IP addresses in an operating system. To communicate between servers using both the Ethernet network and interconnect network, put full host names and IP addresses for both networks in the `/etc/hosts` file for the DB2 pureScale Feature in the AIX environment.

To add the host names and IP addresses to the `/etc/hosts` file, complete the following steps:

1. Open the `/etc/hosts` file.
2. Add host names and IP addresses for all the servers in the cluster (Example 3-28).

Example 3-28 Edit the hosts file

```
192.168.0.31    democf1
192.168.0.32    democf2
192.168.0.33    demom0
```

```
192.168.0.34    demom1

10.10.0.31 democf1-ib0
10.10.0.32 democf2-ib0
10.10.0.33 demom0-ib0
10.10.0.34 demom1-ib0
```

3. Save and exit the file.

3.3.6 Installing and setting up OpenSSH

Open Secure Shell provides encrypted communication between servers by using SSH protocols. For the DB2 pureScale Feature, SSH access for the root user without a password is required. The DB2 installation process sets up the password-less SSH for the instance user, if the user is not set up for password-less SSH.

To generate SSH keys and set up SSH access for the root user without a password, complete the following steps:

1. Ensure that OpenSSH is installed by running the following command:

   ```
   lslpp -la "openssh.*"
   ```

   The output should be similar as shown in Example 3-29.

Example 3-29 Check that OpenSSH is installed

```
Fileset                      Level  State      Description
----------------------------------------------------------------------------
Path: /usr/lib/objrepos
openssh.base.client     5.4.0.6100  COMMITTED  Open Secure Shell Commands

openssh.base.server     5.4.0.6100  COMMITTED  Open Secure Shell Server

openssh.license         4.5.0.5302  COMMITTED  Open Secure Shell License

openssh.man.en_US       5.4.0.6100  COMMITTED  Open Secure Shell
Documentation - U.S. English
openssh.msg.EN_US       5.4.0.6100  COMMITTED  Open Secure Shell Messages - U.S. English (UTF)
openssh.msg.en_US       5.4.0.6100  COMMITTED  Open Secure Shell Messages - U.S. English
Path: /etc/objrepos
openssh.base.client     5.4.0.6100  COMMITTED  Open Secure Shell Commands

openssh.base.server     5.4.0.6100  COMMITTED  Open Secure Shell Server
```

2. Generate the SSH key by running **ssh-keygen -t rsa**. Accept the default path and do not set a passphrase. The output should be similar to Example 3-30.

Example 3-30 Generate the SSH key

```
root@democf1:/> ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (//.ssh/id_rsa):
Created directory ''.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in //.ssh/id_rsa.
Your public key has been saved in //.ssh/id_rsa.pub.
```

3. Navigate to the /.ssh/ folder, and you should find two files. If you are using RSA encryption, they should be the `id_rsa` and `id_rsa.pub` files. Copy the `id_rsa.pub` file to other hosts in the cluster and append its content to the /.ssh/authorized_keys file. You can accomplish that task by running the following command:

   ```
   cat id_rsa.pub >> authorized_keys
   ```

4. Ensure that you import the `id_rsa.pub` file from all the hosts and append its contents to the `authorized_keys` file.
5. Copy the `authorized_keys` file under /.ssh/ to all other hosts in the cluster.
6. You can SSH to all the hosts (including itself) without a password now (Example 3-31).

Example 3-31 SSH to all hosts in the cluster without a password

```
root@democf1:/.ssh> ssh democf1 hostname
democf1
root@democf1:/.ssh> ssh democf2 hostname
democf2
root@democf1:/.ssh> ssh demom0 hostname
demom0
root@democf1:/.ssh> ssh demom1 hostname
demom1
```

3.3.7 Configuring a physical volume identifier for an IBM DB2 pureScale instance

In an AIX environment, the shared disk to be used by the DB2 pureScale instance must have the same physical volume identifier (PVID) configured on all hosts.

To assign a PVID and synchronize it on all hosts, complete the following steps:

1. Run **lspv** to list the disks and PVIDs (Example 3-32).

   Example 3-32 List disks and PVIDs

   ```
   hdisk0          00c201d5c8bf44ab                    rootvg
   active
   hdisk1          00c201d5e7d5e775                    None
   hdisk2          none                                None
   hdisk3          none                                None
   hdisk4          none                                None
   ```

2. Run the following command to assign a PVID to a disk:

   ```
   chdev -l <disk_name> -a pv=yes
   ```

   For example, to assign *hdisk2*, you should run the following command:

   ```
   chdev -l hdisk2 -a pv=yes
   ```

3. Run **lspv** again to check the output. Example 3-33 shows that `hdisk2` is assigned a PVID.

   Example 3-33 List disks and PVIDs

   ```
   hdisk0          00c201d5c8bf44ab                    rootvg
   active
   hdisk1          00c201d5e7d5e775                    None
   hdisk2          00c201d5e81c4444                    None
   hdisk3          none                                None
   hdisk4          none                                None
   ```

4. Remove the same hdisk device on all other hosts by running the following command:

   ```
   rmdev -dl <disk_name_for_the_same_shared_disk>
   ```

   For example, to remove hdisk2, run the following command:

   ```
   rmdev -dl hdisk2
   ```

   Run the **lspv** command again, and you should find the hdisk2 disk is removed from the list (Example 3-34).

   Example 3-34 List disks and PVIDs

   ```
   hdisk0          00c201d5c8bf44ab                    rootvg          active
   hdisk1          00c201d5e7d5e775                    None
   hdisk3          none                                None
   hdisk4          none                                None
   ```

5. Run **cfgmgr** on all other hosts to retrieve the PVID from the storage array and run **lspv** to return the list shown in Example 3-35.

Example 3-35 Retrieve PVID information from the storage array

```
hdisk0          00c201d5c8bf44ab                    rootvg
active
hdisk1          00c201d5e7d5e775                    None
hdisk2          00c201d5e81c4444                    None
hdisk3          none                                None
hdisk4          none                                None
```

6. Compare the hdisk2 PVID from all hosts and ensure that they are all the same.

3.3.8 Configuring the I/O completion ports

You must configure the I/O completion ports on AIX before you install the DB2 pureScale Feature. To configure these ports, complete the following steps:

1. Run **lslpp** to check whether the IOCP module is installed on your system:

   ```
   lslpp -l bos.iocp.rte
   ```

   Example 3-36 shows the output from this command.

Example 3-36 Check that the IOCP module is installed on the system

```
Fileset                      Level  State      Description
----------------------------------------------------------------------------
Path: /usr/lib/objrepos
bos.iocp.rte              7.1.0.15  APPLIED    I/O Completion Ports API
Path: /etc/objrepos
bos.iocp.rte               7.1.0.0  COMMITTED  I/O Completion Ports API
```

2. Run the following command to check whether the status of IOCP is available:

   ```
   lsdev -Cc iocp
   ```

   The output should be look like:

   ```
   iocp0 Available I/O Completion Ports
   ```

3. If the IOCP port status is `Defined`, change it to `Available`:

   a. Log in as root and run **smitty iocp**.
   b. Select the Change / Show Characteristics of I/O Completion Ports option.
   c. Switch the configured state at system restart from `Defined` to `Available`.
   d. Reboot the system or run **cfgmgr**.

3.3.9 Allocating storage

The following storage requirements must be met before you install the DB2 pureScale Feature.

- Allocate the following local disk space on each host:
  - 3 GB for the extraction directory.
  - 3.5 GB for the installation path.
  - 5 GB for the `/tmp` directory.
  - 1 GB for the instance home directory.
  - 5 GB for the `/var` directory.
  - 1 GB for the `/root` directory.
- Allocate the following shared disk space:
  - 10 GB for the instance shared files.
  - The data space depends on your application needs.
  - The logs space depends on the expectant number of transactions and the applications logging requirements.

3.3.10 Creating the required groups and users

DB2 10.1 requires a user ID for the instance user and another user ID to use fenced user-defined functions or procedures. Ensure that you follow these instructions to create required groups and users before performing the DB2 pureScale installation.

To create the required groups and users, complete the following steps:

1. Run the following commands to create the DB2 instance owner group and fenced user group.
   - **`mkgroup id=999 db2iadm1`**
   - **`mkgroup id=998 db2fadm1`**
2. Run the following commands to create DB2 instance users and fenced users that belong to each group. Change the user ID or home directory if needed.
   - **`mkuser id=1004 pgrp=db2iadm1 groups=db2iadm1 home=/db2home/db2sdin1 core=-1 data=491519 stack=32767 rss=-1 fsize=-1 db2sdin1`**
   - **`mkuser id=1003 pgrp=db2fadm1 groups=db2fadm1 home=/db2home/db2sdfe1 db2sdfe1`**

3.4 Installing the DB2 pureScale Feature

This section describes how to install the DB2 pureScale Feature on a Linux or AIX system. Instructions for the following installation methods are provided:

- A command-line installation method that uses the **db2_install** command.
- A wizard-based installation method that uses the DB2 setup wizard.

Before proceeding, ensure that the following conditions are met:

- Your system meets the installation, memory, and disk requirements.
- You completed all the preinstallation tasks.
- You are logged in to the system as the root user.
- If you plan to use the DB2 setup wizard, you have an X Server running on the X Window System client machine.
- You have the host name, net name, and disk information ready:
  - The host name must be the same as the output returned by the **hostname** command.
  - The net name represents the cluster interconnect net name. For example, node101-ib is used as the interconnect net name for node101 and node102-ib is used as the net name for node102.

The DB2 pureScale installation media is required to be placed only on one server. The installer installs and configures DB2 and instance users on other servers during the installation.

3.4.1 Checking the installation prerequisites

Before you install the DB2 pureScale Feature, run **db2prereqcheck** to verify if the system satisfies the DB2 installation prerequisites. This command checks prerequisites, such as operating system level, Linux distribution, AIX Technology Level, C library, runtime patch, Client, uDAPL package, GPL, or other DB2 pureScale Feature specific requirements.

The **db2prereqcheck** command comes with DB2. Before you can run **db2prereqcheck**, ensure that you extracted the DB2 10.1 binary file on your system. You also need read and write access to the system to run **db2prereqcheck**.

To check the installation prerequisites by running **db2prereqcheck**, complete the following steps:

1. Log in the system and navigate to the directory that contains the DB2 10.1 binary files.
2. Run **db2prereqcheck -p -v 10.1.0.0** to check whether the system meets the prerequisites for the DB2 pureScale Feature.
3. After the command is run, the sample output should contain the validation result (Example 3-37). As you can see, the command verifies several prerequisites and tells you whether the system meets the installation requirement.

Example 3-37 Sample output of the db2prereqcheck command

```
Validating kernel level ...
   Required minimum operating system kernel level : "2.6.16".
   Actual operating system kernel level: "2.6.32".
   Requirement matched.

Validating SELinux status ...
   SELinux is disabled.
   Requirement matched.

Validating C++ Library version ...
   Required minimum C++ library: libstdc++.so.6
   Standard C++ library is located in the following directory:
"/usr/lib64/libstdc++.so.6.0.13".
   Actual C++ library: CXXABI_1.3.1
   Requirement matched.

Validating 32 bit version of libstdc++.so.6 ...
   Found the 32 bit "/usr/lib/libstdc++.so.6" in the following
directory "/usr/lib".
   Requirement matched.

Validating libc.so version ...
   glibc library is located in the following directory
"/lib64/libc-2.12.so".
   Required minimum glibc library version: 2.4.0
   Actual glibc library version: 2.12.0
   Requirement matched.

Validating libaio.so version ...
   Loading the libaio.so.1 succeed.
   Requirement matched.
```

```
Validating dapl ...
   Required minimum version and release for dapl: 2.0.25-(null)
   Actual version of package: 2.0.32

   Requirement matched.

Validating sg3_utils ...
   Package (or file) found: sg3_utils
   Requirement matched.

Validating sg_persist ...
   Package (or file) found: /usr/bin/sg_persist
   Requirement matched.

DBT3533I  The db2prereqcheck utility has confirmed that all
installation prerequisites were met for DB2 database server with DB2
pureCluster feature. Version: "10.1.0.0".
```

3.4.2 Using the command line on a Linux environment

This section describes how to install a DB2 database with the DB2 pureScale Feature by running **db2_install**.

> **Attention:** The **db2_install** command is deprecated in a future Fix Pack release. Use the **db2_setup** GUI installation.

To install a DB2 database with the DB2 pureScale Feature by running **db2_install**, complete the following steps:

1. Create DB2 instance owner and group, for example, db2sdin1/db2iadm1, on all the hosts in the cluster.
2. Create DB2 fence user and group, for example, db2sdfe1/db2fadm1, on all the hosts in the cluster.
3. Run **db2_install** to install the DB2 pureScale database product on any host. This host is referenced as the installation-initiating host.
4. On the installation-initiating host, run **db2icrt** to create a member and a cluster caching facility. The **db2icrt** command copies the DB2 files to the second host and then creates a member and a cluster caching facility.

To install the DB2 pureScale Feature using the command-line interface (CLII) method, complete the following steps:

1. Log in as root on node101.
2. Go to the folder that contains the DB2 installer `tar` file. Extract the installer files by running the following command:

   ```
   tar -zxvf <DB2 10.1 pureScale feature installer tar file path>
   ```

3. Create the DB2 instance administrator group db2iadm1 and the DB2 fenced group db2fadm1 by running the following commands:

   - **groupadd –g 999 db2iadm1**
   - **groupadd –g 998 db2fadm1**

4. Create the DB2 instance user db2sdin1 and DB2 fenced user db2sdfe1 by running the following commands:

   - **useradd –u 1004 –g db2iadm1 –m –d /home/db2sdin1 db2sdin1**
   - **useradd –u 1003 –g db2fadm1 –m –d /home/db2sdfe1 db2sdfe1**

5. Go to the folder that was extracted, and run the following the command to install the DB2 pureScale Feature on the installation-initiating host:

   ```
   db2_install –t /tmp/db2_install.trc –l /tmp/db2_install.log
   ```

   This command installs the DB2 database with the DB2 pureScale Feature to the `/opt/ibm/db2/V10.1/` default directory. Add the **-b <DB2DIR>** parameter to the **db2_install** command to change the installation directory. The installation directory must be the same on all hosts in your cluster.

   In the command, the **-t** parameter specifies the full path of the installation trace file. The **-l** parameter specifies the full path to the installation log file. Although both parameters are optional, the installation trace and log files can help DB2 diagnose installation problems in the unlikely event that they occur.

6. Run the **/usr/local/bin/db2ls** command to verify the installation.

   Use the following syntax to create a DB2 instance with one DB2 member and one CF on the installation-initiating host:

   ```
   db2icrt –d –m <member_hostname> -mnet <member_netname> –cf
   <cf_hostname> -cfnet <cf_netname> –instance_shared_dev <disk name>
   -tbdev <tb disk name> -u <DB2 fence user name> <DB2 instance user
   name>
   ```

On node101, run the command shown in Example 3-38 to create the DB2 instance.

Example 3-38 Create a DB2 instance

```
db2icrt –d –m node101.purescale.demo -mnet node101-ib –cf
node101.purescale.demo -cfnet node101-ib –instance_shared_dev
/dev/dm-1 -tbdev /dev/dm-0 -u db2sdfe1 db2sdin1
```

7. To add the host node102 as the second cluster caching facility to the DB2 pureScale cluster environment, run the following command:

   ```
   db2iupdt –d –add –cf node102.puresclae.demo -cfnet node102-ib
   db2sdin1
   ```

8. To add the host node102 as the second member to the DB2 pureScale cluster environment, run the following command:

   ```
   db2iupdt –d –add –m node102.purescale.demo -mnet node102-ib db2sdin1
   ```

Important: You have finished the DB2 pureScale installation. Continue to 3.5, “Post-installation tasks” on page 187.

3.4.3 Using the DB2 setup wizard

This section describes how to install a DB2 database with the DB2 pureScale Feature using the DB2 setup wizard on Linux systems. The DB2 setup wizard provides default values for most of the fields and options.

To install a DB2 database with the DB2 pureScale Feature using the DB2 setup wizard, complete the following steps:

1. Log in as root on node101. Navigate to the folder that contains the DB2 installer `tar` file. Extract the installer files by running the following command:

   ```
   tar -zxvf <DB2 10.1 pureScale feature installer tar file path>
   ```

2. Navigate to the folder that was extracted, and run the following command to start the DB2 Setup Launchpad:

   ```
   ./db2setup -l /tmp/db2setup.log -t /tmp/db2setup.trc
   ```

   Where:

   -t Specifies the full path of the installation trace file
   -l Specifies the full path to the installation log file

   Although both parameters are optional, the installation trace and log files can help DB2 support diagnose installation problems in the unlikely event that they occur.

3. In the left menu of the DB2 Setup Launchpad, click **Install a Product**, and then click **Install New**. Follow the prompts to complete the wizard.
4. Review the DB2 pureScale Feature in the introduction. Then, review and accept the license agreement to continue.
5. Set the installation directory. The default installation directory is `/opt/ibm/db2/V10.1/` (Figure 3-60). The installation directory must be the same on all hosts in your cluster.

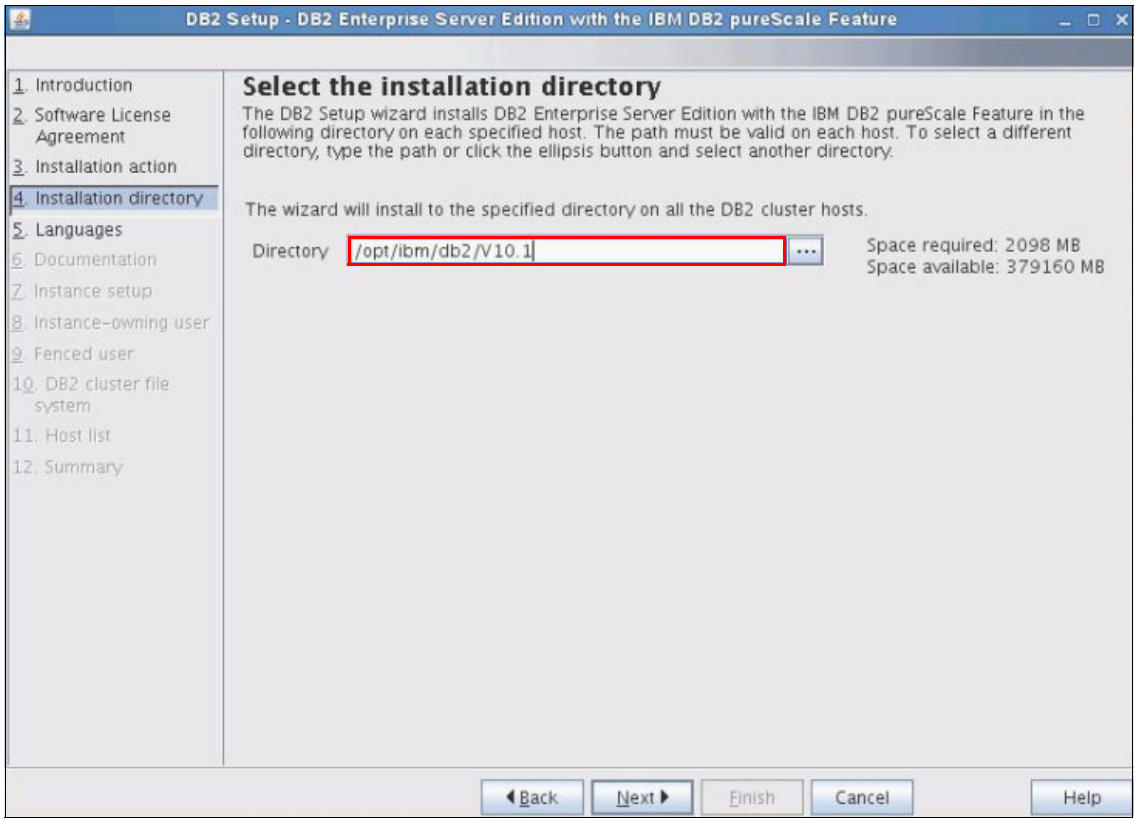

Figure 3-60 Select the installation directory

6. Set the language for the user interface and product. The default language is English (Figure 3-61), but you can add other languages if needed.

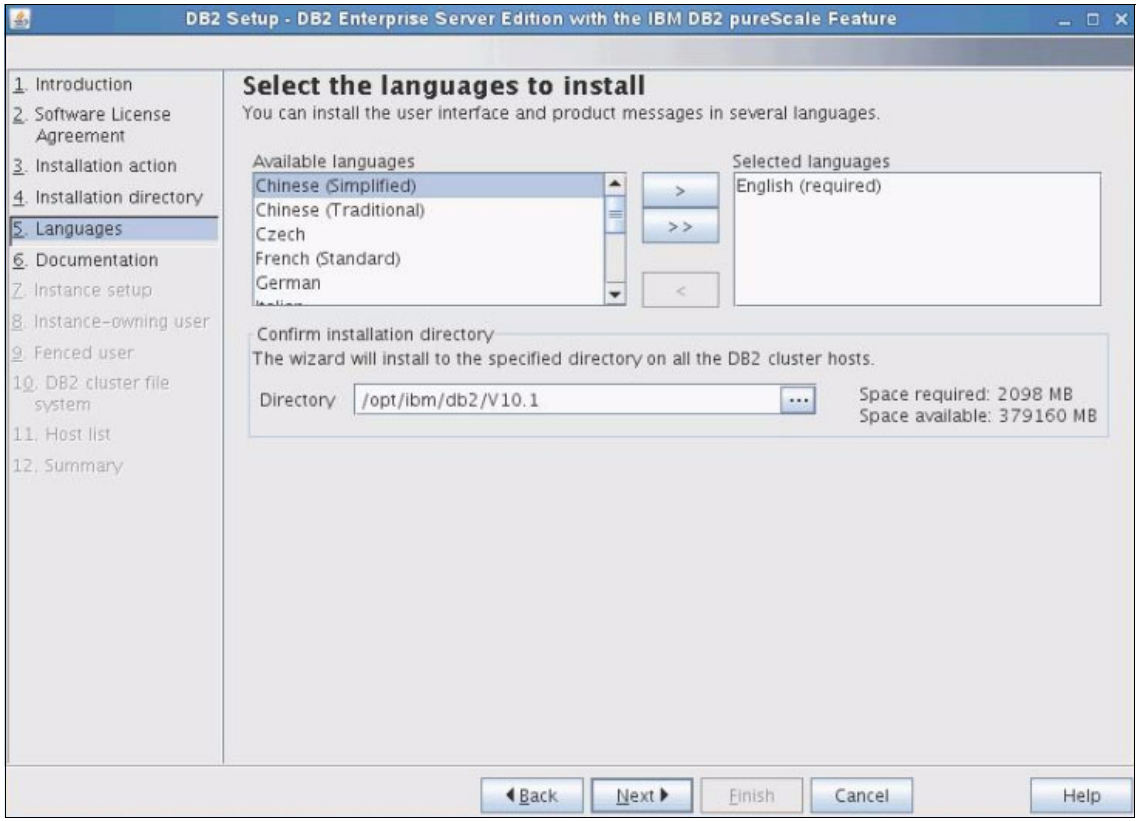

Figure 3-61 Select the languages to install

7. Specify the location of DB2 Information Center (Figure 3-62).

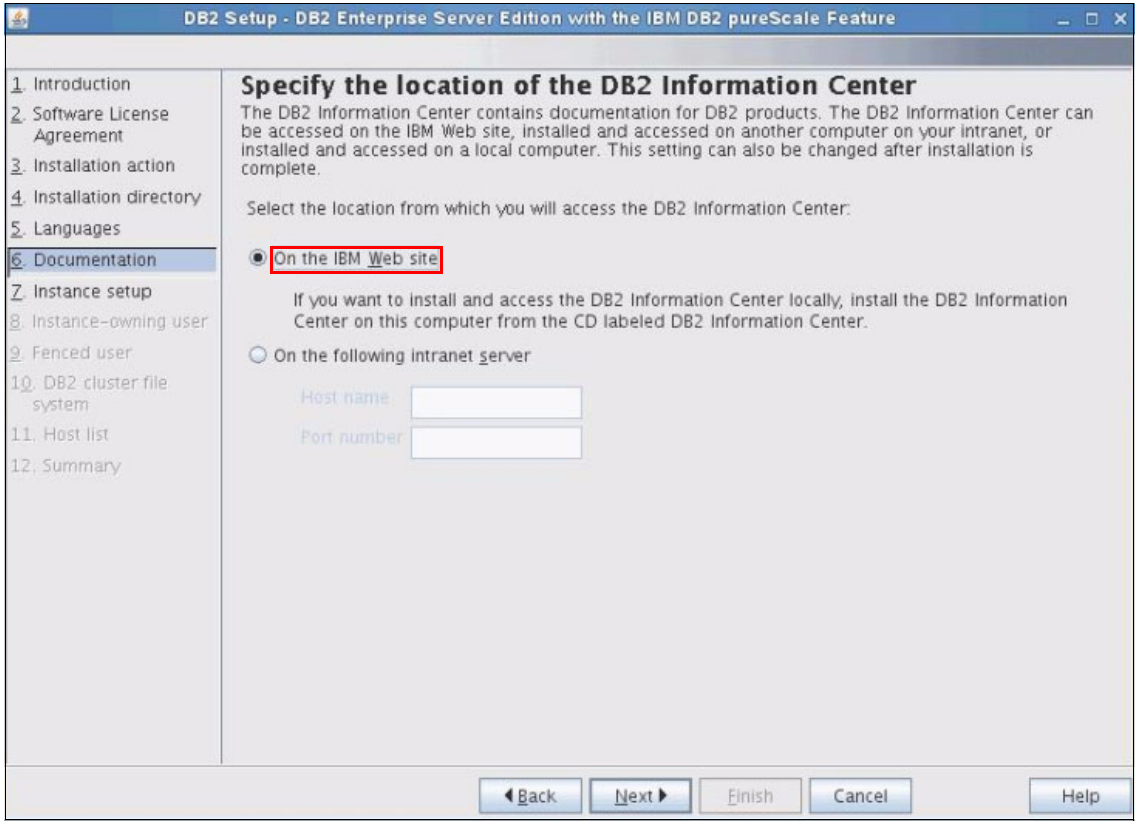

Figure 3-62 Specify the location of the DB2 Information Center

8. Set up a DB2 instance (Figure 3-63). Select **Create a DB2 instance**.

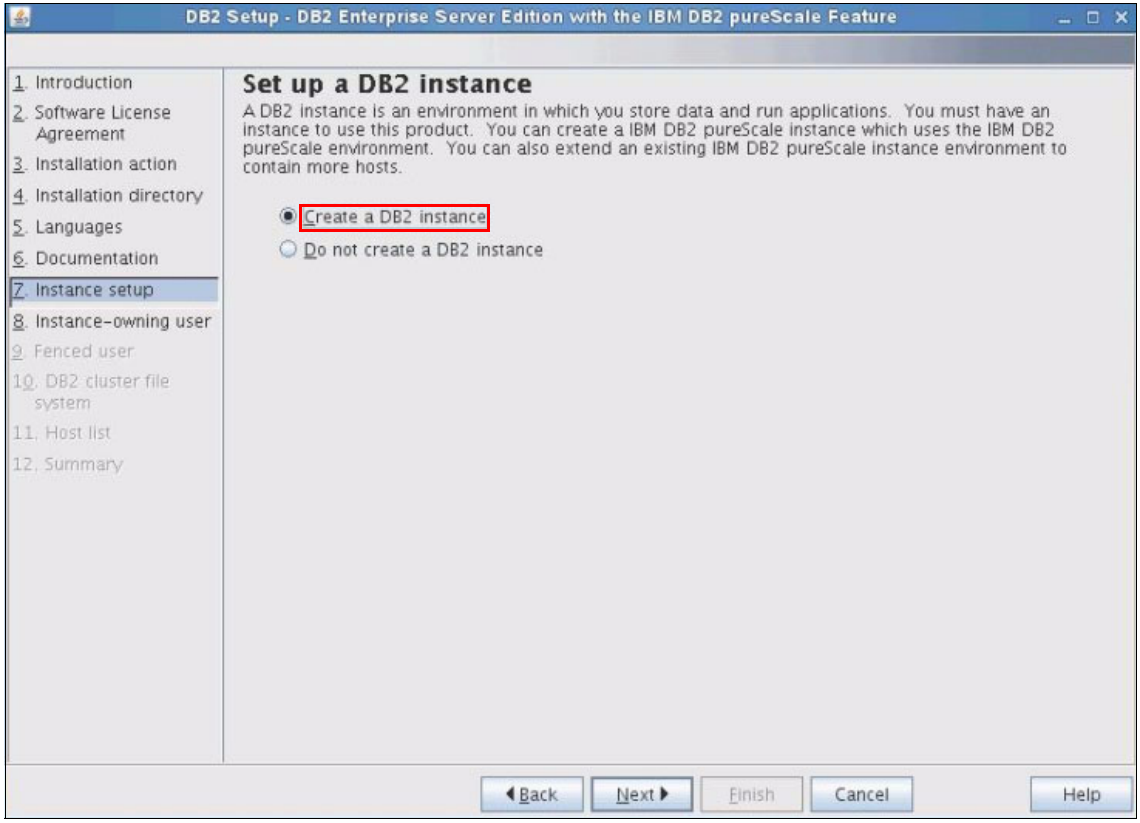

Figure 3-63 Set up a DB2 instance

9. Set the DB2 instance owner (Figure 3-64). Select **New user**. Then, enter db2sdin1 as the user name and db2iadm1 as the instance owner group. In a DB2 pureScale environment, this user and group are created on all hosts in the cluster.

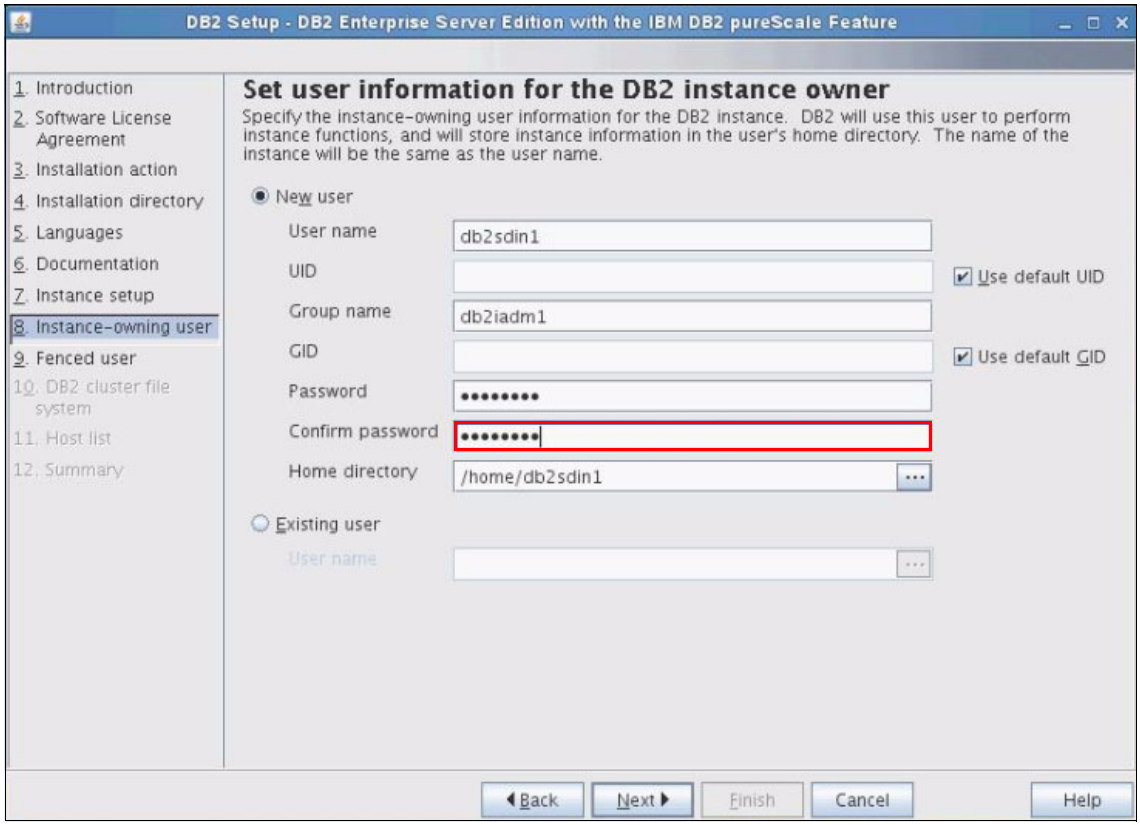

Figure 3-64 Set up the DB2 instance user

10. After setting the DB2 instance owner, set the information for the fenced user (Figure 3-65). Select **New user**. Then, enter db2sdfe1 as the DB2 fenced user name and db2fadm1 as the fenced user group. In a DB2 pureScale environment, this user and group are created on all hosts in the cluster.

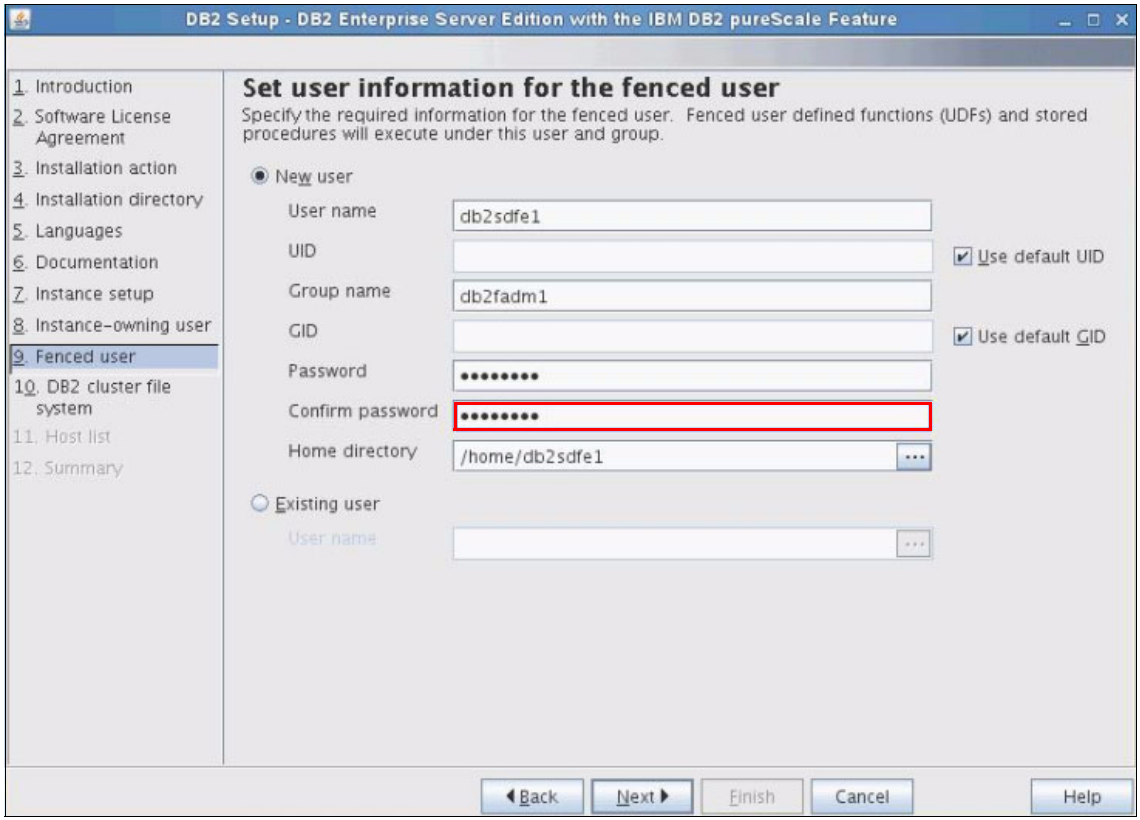

Figure 3-65 Set up a DB2 fenced user

11. Set up a DB2 Cluster File System (Figure 3-66). Enter the location of your shared disk partition device. The shared disk partition is where your DB2 files are (SQLLIB), and the disk path is a path such as `/dev/dm-1`. The cluster services tiebreaker is on a path such as `/dev/dm-0`.

> **Important:** Ensure that the device path matches the output of the **`multipath -l`** command, The minimum disk size for the tiebreaker is 25 MB.

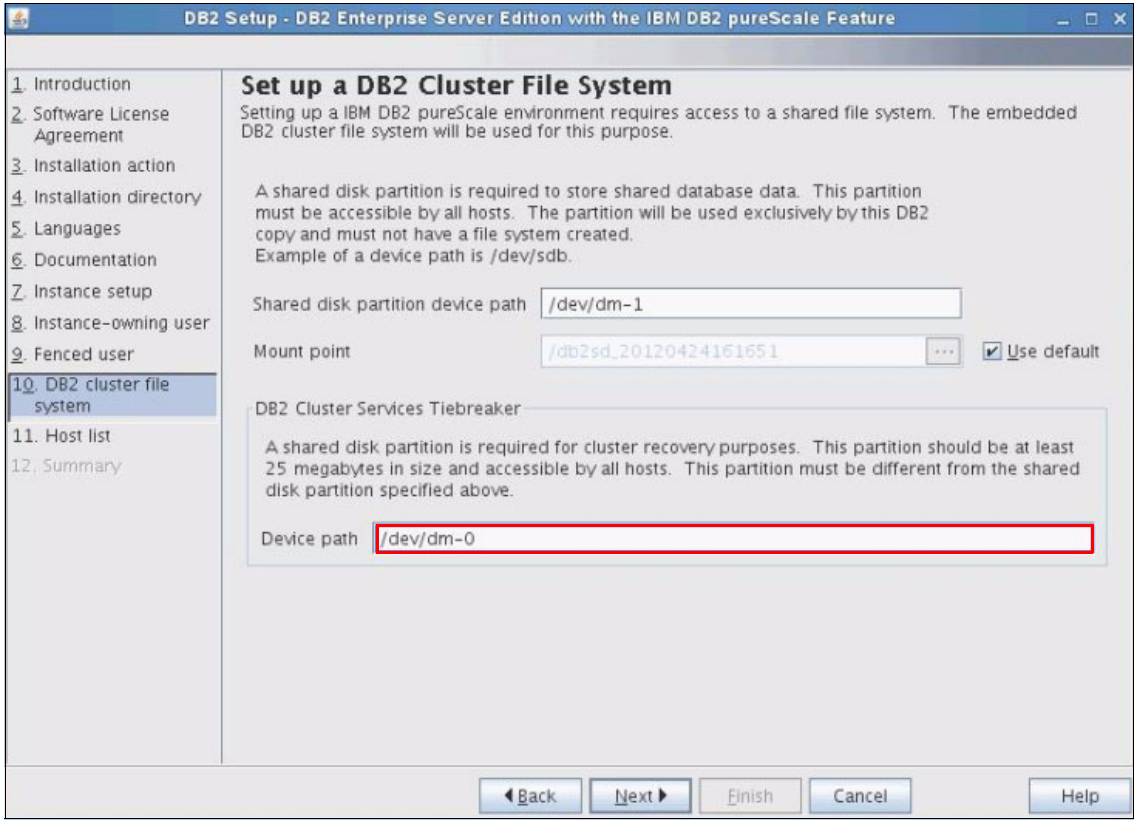

Figure 3-66 Set up the DB2 cluster file system

12. Add all the hosts that are part of the DB2 pureScale cluster. The installation-initiating host is already in the list. To add a host to the list, complete the following steps:

    a. Click **Add**. For example, you can add the node102 host in the cluster (Figure 3-67).

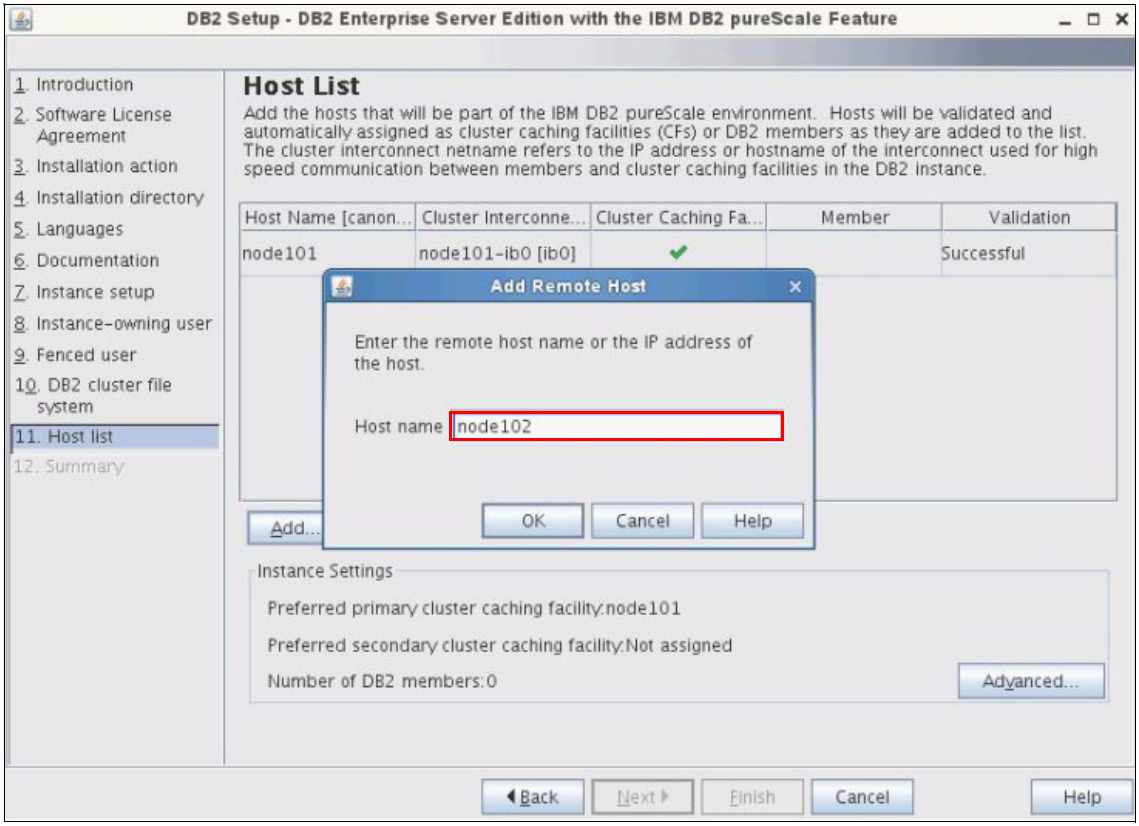

Figure 3-67 Add a second node

b. Select the cluster interconnect name upon validation of any host by the DB2 setup wizard (Figure 3-68).

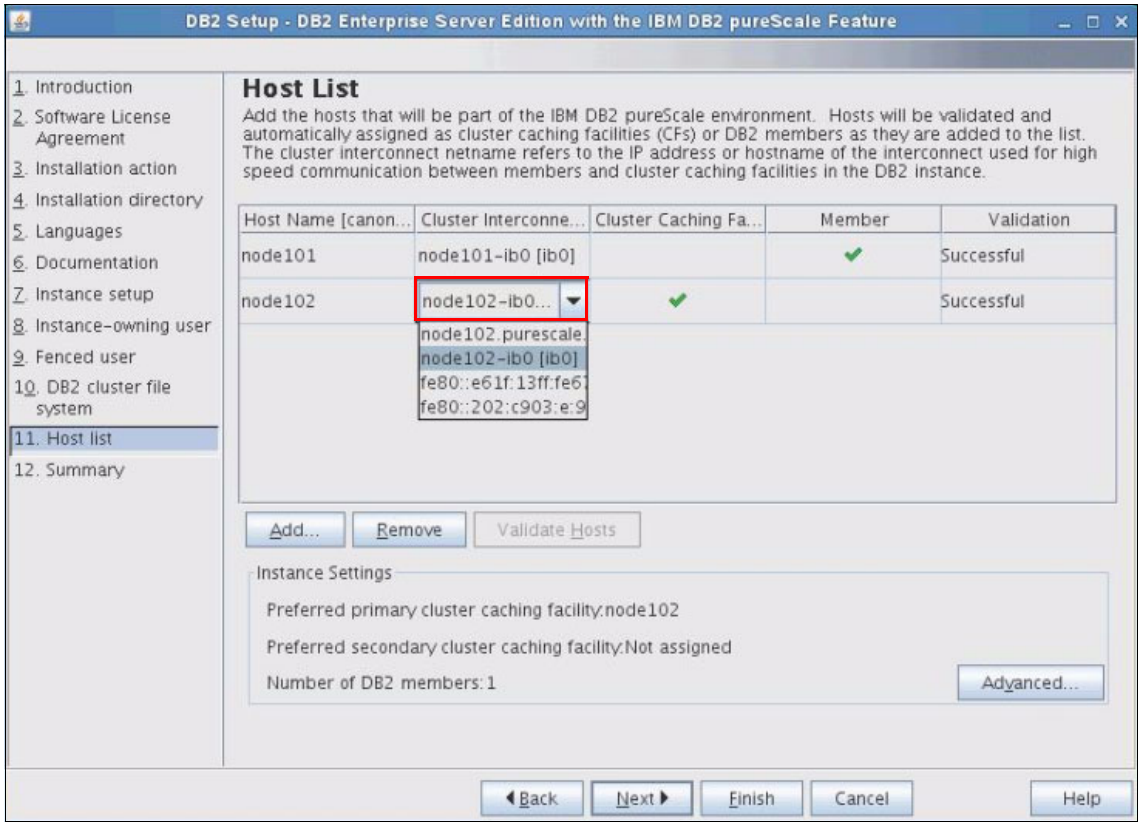

Figure 3-68 Select the cluster interconnect name for the second node

c. After adding all hosts in the cluster on the Host List page, modify the instance setting to assign CFs and DB2 members. On the Host List window of the wizard, click **Advanced** and then go to the **Cluster caching facility** tab (Figure 3-69). Complete the following information:

- Select **Manually assign cluster caching facility (CFs)**.
- From the **Preferred primary CF** list, select **node101**, and select **Configure the host as both a CF and a DB2 member**.
- From the **Preferred secondary CF** list, select **node102**, and select **Configure the host as both a CF and a DB2 member**.

Click **OK**.

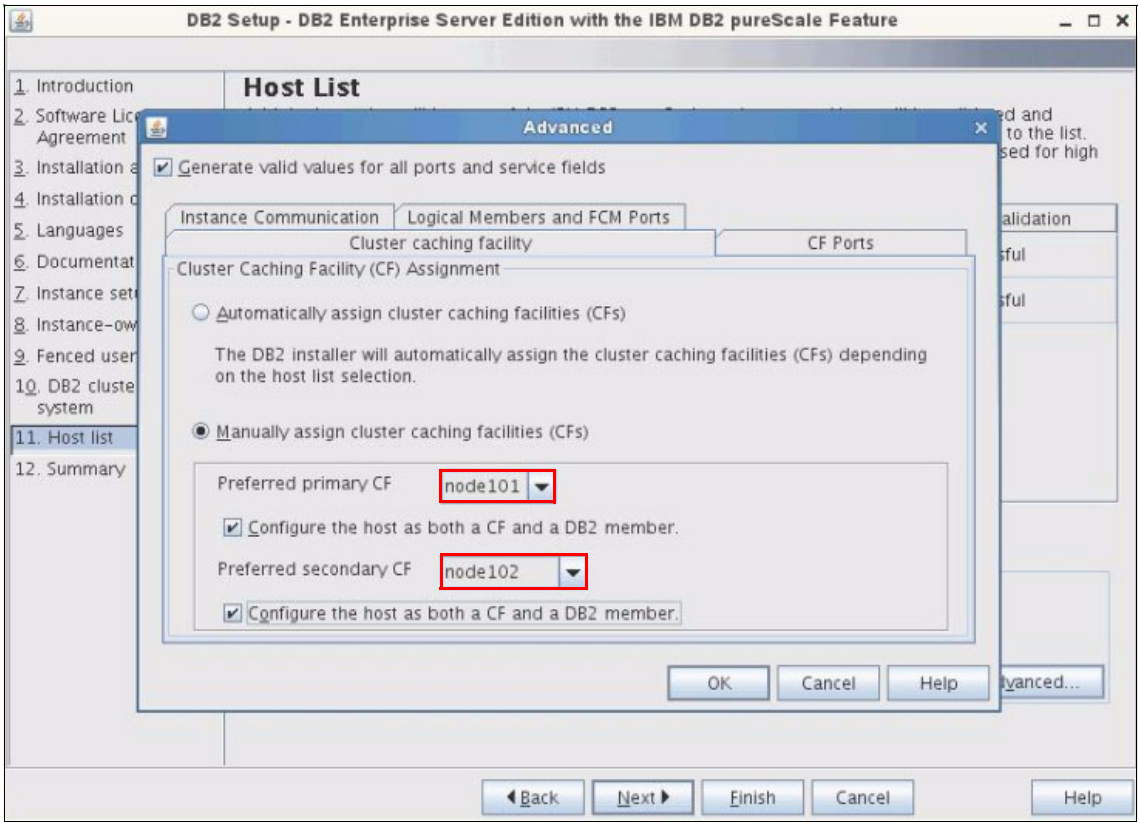

Figure 3-69 Advanced options

Figure 3-70 shows the second added host name in the host list.

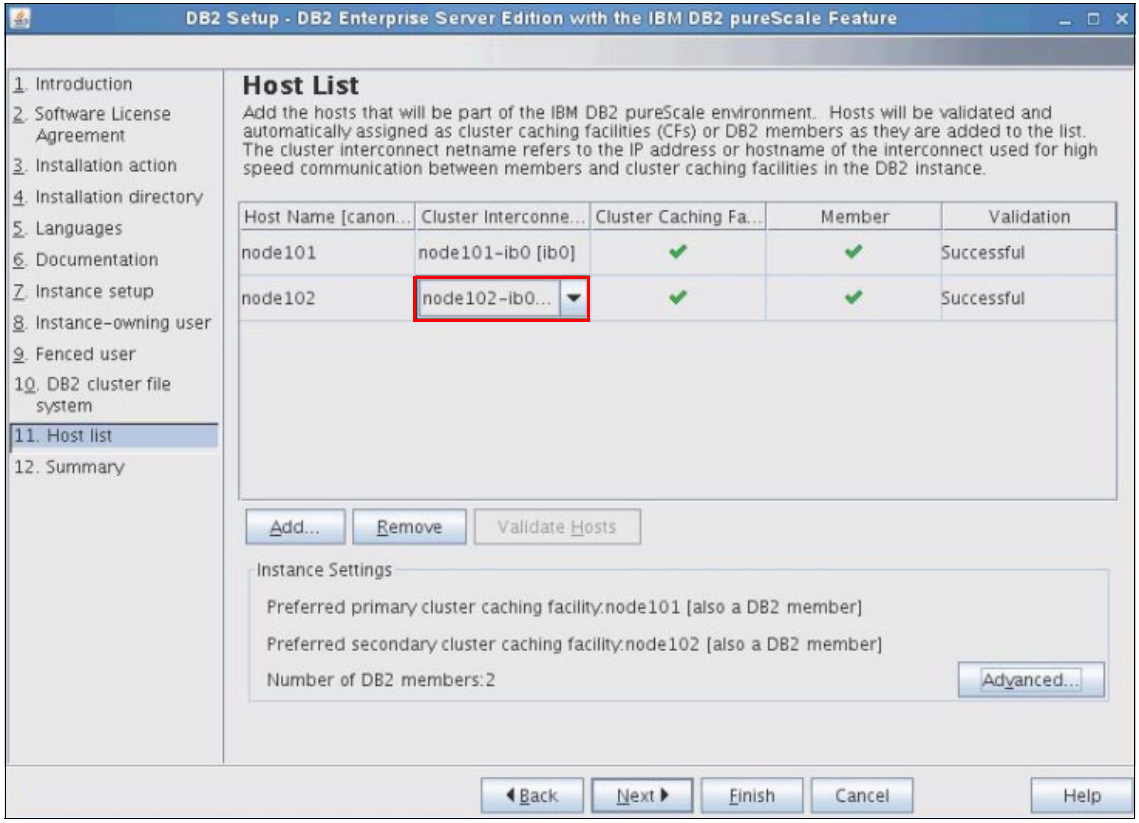

Figure 3-70 Added host name in the host list

Ignore warnings: If you receive warnings about the CF and members being on the same host, you can ignore them. Having a CF and a member on the same host decreases redundancy and increases the number of components that can fail at the same time. Having the CF and the member on separate hosts decreases the number of components that are affected when a hardware or power failure occurs.

13. Review the summary, and then click **Finish** to begin the installation.

14. After the installation completes, review the content on the Post-install steps and the Log file tabs (Figure 3-71).

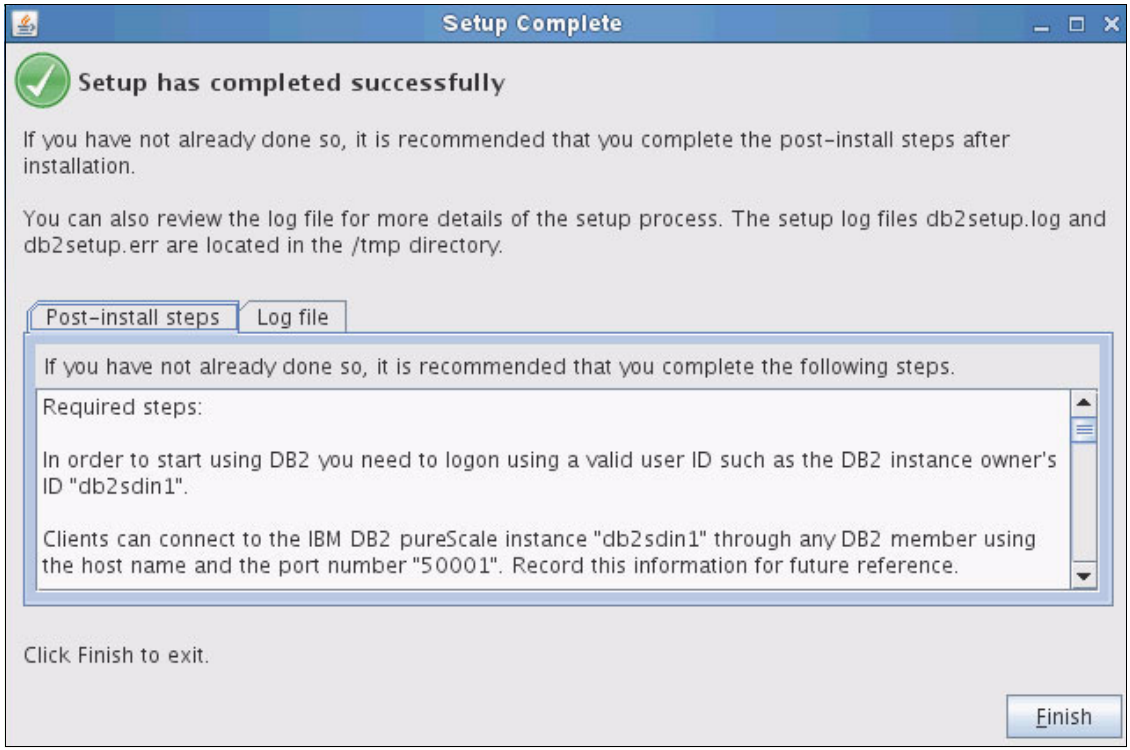

Figure 3-71 Setup completed successfully

> **Important:** You have finished the DB2 pureScale Feature installation on a Linux system and can continue to 3.5, “Post-installation tasks” on page 187.

3.4.4 Using the command line installation in an AIX environment

Installing the DB2 pureScale Feature on an AIX environment is similar to installing it on a Linux system. This section describes how to install the DB2 pureScale Feature using the command line. The process includes:

- Running **`db2_install`** to install the DB2 pureScale Feature on the installation-initiating host.
- On the installation-initiating host, running **`db2icrt`** to create a member and cluster caching facility. The **`db2icrt`** command copies the DB2 product to the second host and creates the member and cluster caching facility.

- Running **db2iupdt** to add extra members and cluster caching facilities to the cluster. The **db2iupdt** command copies the DB2 product to the new host and creates a member and a cluster caching facility.

Complete the following steps:

1. Log in as root on the first node and extract the DB2 pureScale `tar.gz` file.
2. Ensure that the required DB2 instance/fenced user and administrator group are created on all hosts with the same user/group ID (Example 3-39).

Example 3-39 Check the required users on all hosts

```
# ssh demom0 id db2sdin1
uid=1004(db2sdin1) gid=999(db2iadm1)
# ssh democf1 id db2sdin1
uid=1004(db2sdin1) gid=999(db2iadm1)
# ssh demom0 id db2sdfe1
uid=1003(db2sdfe1) gid=998(db2fadm1)
# ssh democf1 id db2sdfe1
uid=1003(db2sdfe1) gid=998(db2fadm1)
......
```

3. Navigate to the directory that was extracted, and run the following command as root to install the DB2 pureScale Feature on the installation-initiating host (IIH):

```
./db2_install –t /tmp/db2_install.trc –l /tmp/db2_install.log
```

Example 3-40 shows the manual installation panel.

Example 3-40 Manual installation prompt

```
DBI1324W  Support of the db2_install command is deprecated. For more
information, see the DB2 Information Center.
Default directory for installation of products - /opt/IBM/db2/V10.1
***********************************************************
Install into default directory (/opt/IBM/db2/V10.1) ? [yes/no] yes
Specify one of the following keywords to install DB2 products.
ESE
CLIENT
RTCL
Enter "help" to redisplay product names.
Enter "quit" to exit.
***********************************************************
ESE
***********************************************************
Do you want to install the DB2 pureCluster Feature? [yes/no] yes
DB2 installation is being initialized.
```

4. After the installation, you can run the following command to verify the installation:

   ```
   /usr/local/bin/db2ls
   ```

   Example 3-41 shows a sample output of this command.

Example 3-41 Verify the DB2 installation

```
/usr/local/bin/db2ls
Install Path Level  Fix Pack  Special Install Number Install Date Installer        UID
-----------------------------------------------------------------------------------------
/opt/IBM/db2/V10.1 10.1.0.0  0                      Tue Mar 20 11:37:15 2012 EDT   0
```

5. Use the following syntax to create the DB2 instance on the DB2 pureScale Feature product. A member and a CF are required when you create the instance.

   db2icrt –d –m <*member_hostname*> -mnet <*member_netname*> –cf <*cf_hostname*> -cfnet <*cf_netname*> –instance_shared_dev <*disk name*> -tbdev <*tb disk name*> -u <*DB2 fence user name*> <*DB2 instance user name*>

   For example, if you want to create a member on the demom0 host and a CF on the democf1 host, run the command shown in Example 3-42.

   Example 3-42 Create the DB2 instance

   ```
   db2icrt –d –m demom0 -mnet demom0-ib0 –cf democf1 -cfnet democf1-ib0
   –instance_shared_dev /dev/hdisk4 -tbdev /dev/hdisk6 -u db2sdin1
   db2sdfe1
   ```

6. To add the demom1 host as the second DB2 member to the DB2 pureScale cluster environment, run the following command:

   ```
   db2iupdt –d –add –m demom1 -mnet demom1-ib0 db2sdin1
   ```

7. To add the democf2 host of the standby cluster caching facility to the DB2 pureScale cluster environment, run the following command:

   ```
   db2iupdt –d –add –cf democf2 -cfnet democf2-ib0 db2sdin1
   ```

8. After you create the DB2 instance, verify its creation by running **db2start** (Example 3-43).

Example 3-43 Start the DB2 manager

```
$ db2start
03/21/2012 11:21:01     1   0   SQL1063N  DB2START processing was
successful.
03/21/2012 11:21:02     0   0   SQL1063N  DB2START processing was
successful.
SQL1063N  DB2START processing was successful.
```

Important: You have finished the DB2 pureScale installation on an AIX system. Proceed to the next section to continue.

3.5 Post-installation tasks

This section describes the following necessary and optional post-installation tasks that can help improve performance:

- Validate the DB2 pureScale Feature installation by creating the SAMPLE database and run some queries against this database.
- Create shared file systems to store the directories for the data and log files. This configuration can separate I/O workload and improve your database performance.
- Change the default database path to be on a shared file system.
- Change the default log file path to be on a shared file system.
- Specify the number of processor cores to a cluster caching facility or a DB2 member.
- Enable SCSI-3 PR for the DB2 pureScale Feature to provide faster failover support.

3.5.1 Validating the installation by creating the SAMPLE database

You can validate the DB2 pureScale Feature installation by creating the SAMPLE database and running queries.

Attention: Complete this task from one host in your DB2 pureScale cluster.

To validate the installation, complete the following steps:

1. Log in as a DB2 instance user, for example, db2sdin1.
2. Start DB2 by running **db2start** (Example 3-44).

Example 3-44 Start the DB2 manager

```
db2sdin1@node101:~> db2start
03/11/2012 17:21:16     0   0   SQL1063N  DB2START processing was
successful.
03/11/2012 17:21:16     1   0   SQL1063N  DB2START processing was
successful.
SQL1063N  DB2START processing was successful.
```

3. Run **db2sampl** to create the SAMPLE database (Example 3-45).

Example 3-45 Create the SAMPLE database

```
db2sdin1@node101:/root> db2sampl
Creating database "SAMPLE"...
Connecting to database "SAMPLE"...
Creating tables and data in schema "DB2SDIN1"...
Creating tables with XML columns and XML data in schema
"DB2SDIN1"...
'db2sampl' processing complete.
```

4. Connect to the SAMPLE database by running **db2 connect** (Example 3-46).

Example 3-46 Connect to the SAMPLE database

```
db2sdin1@node101:/root> db2 connect to SAMPLE
   Database Connection Information
Database server        = DB2/LINUXX8664 10.1.0
SQL authorization ID   = DB2SDIN1
Local database alias   = SAMPLE
```

5. Run the queries shown in Example 3-47 to verify that the database is created successfully.

Example 3-47 Fetch data from the SAMPLE database

```
db2sdin1@node101:/root> db2 list tables
Table/View                      Schema          Type  Creation time
------------------------------- --------------- ----- --------------------------
ACT                             DB2SDIN1        T     2012-03-11-17.24.08.259797
ADEFUSR                         DB2SDIN1        S     2012-03-11-17.24.09.627426
CATALOG                         DB2SDIN1        T     2012-03-11-17.24.10.653997
CL_SCHED                        DB2SDIN1        T     2012-03-11-17.24.07.240606
```

```
CUSTOMER                        DB2SDIN1        T     2012-03-11-17.24.10.409045
DEPARTMENT                      DB2SDIN1        T     2012-03-11-17.24.07.339562
DEPT                            DB2SDIN1        A     2012-03-11-17.24.07.490265
EMP                             DB2SDIN1        A     2012-03-11-17.24.07.631344
EMPACT                          DB2SDIN1        A     2012-03-11-17.24.08.255705
EMPLOYEE                        DB2SDIN1        T     2012-03-11-17.24.07.492346
EMPPROJACT                      DB2SDIN1        T     2012-03-11-17.24.08.199933
....

46 record(s) selected.
db2sdin1@node101:/root> db2 "select count(*) from employee"
1
-----------
42
1 record(s) selected.
```

3.5.2 Creating shared file systems

During the DB2 pureScale Feature installation, only one shared file system (db2fs1) was created. This file system is used to store all the required DB2 files. This section describes how to create two additional shared file systems to store data and logs for the database. For better database performance, you can have separate I/O for data and log.

Note: Complete this task from one host in your DB2 pureScale cluster.

To create shared file systems, complete the following steps:

1. To determine the correct device path on your system for your DB2 data and log, run the following command:

   ```
   multipath -l
   ```

   Example 3-48 shows the output for this command.

   Example 3-48 List multipath devices

   ```
   node101:~ # multipath -l
   360080e50001b757e000093b74f3a3e15 dm-0 IBM,1746      FAStT
   size=100M features='3 queue_if_no_path pg_init_retries 50'
   hwhandler='1 rdac' wp=rw
   |-+- policy='round-robin 0' prio=-1 status=active
   | `- 4:0:0:0 sdf 8:80  active undef running
   `-+- policy='round-robin 0' prio=-1 status=enabled
   `- 3:0:0:0 sdb 8:16  active undef running
   ```

```
360080e50001b70e800001b0b4f3a3e7d dm-1 IBM,1746      FAStT
size=10G features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 rdac' wp=rw
|-+- policy='round-robin 0' prio=-1 status=active
| `- 3:0:0:1 sdc 8:32  active undef running
`-+- policy='round-robin 0' prio=-1 status=enabled
`- 4:0:0:1 sdg 8:96  active undef running
360080e50001b70e800001b634f5f123c dm-3 IBM,1746      FAStT
size=250G features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 rdac' wp=rw
|-+- policy='round-robin 0' prio=-1 status=active
| `- 3:0:0:3 sde 8:64  active undef running
`-+- policy='round-robin 0' prio=-1 status=enabled
`- 4:0:0:3 sdi 8:128 active undef running
360080e50001b757e000093bb4f3a3e7c dm-2 IBM,1746      FAStT
size=500G features='3 queue_if_no_path pg_init_retries 50'
hwhandler='1 rdac' wp=rw
|-+- policy='round-robin 0' prio=-1 status=active
| `- 4:0:0:2 sdh 8:112 active undef running
`-+- policy='round-robin 0' prio=-1 status=enabled
`- 3:0:0:2 sdd 8:48  active undef running
```

2. Create two shared file systems for the DB2 data and log:
   - The shared file system for the DB2 data uses the 500 GB disk and is mounted at `/db2fs/db2data`.
   - The shared file system for the DB2 log uses the 250 GB disk and is mounted at `/db2fs/db2log/`.

   Run the following command as root to create a file system:

   ```
   /opt/ibm/db2/V10.1/bin/db2cluster -cfs -create -filesystem <file
   system name> -disk <storage path> -mount <mount point>
   ```

   The commands and output looks similar to Example 3-49.

Example 3-49 Create shared file systems

```
node101:~ # /opt/ibm/db2/V10.1/bin/db2cluster -cfs -create
-filesystem db2data -disk /dev/dm-2 -mount /db2fs/db2data

File system 'db2data' has been successfully created.

node101:~ # /opt/ibm/db2/V10.1/bin/db2cluster -cfs -create
-filesystem db2log -disk /dev/dm-3 -mount /db2fs/db2log

File system 'db2log' has been successfully created.
```

3. List all the file systems on your server by running the following command:

   ```
   db2cluster -cfs -list -filesystem
   ```

   Example 3-50 shows that there are three shared file systems:

   - db2fs1
   - db2data
   - db2log

   Example 3-50 List shared file systems

   ```
   node101:~ # /opt/ibm/db2/V10.1/bin/db2cluster -cfs -list -filesystem
   FILE SYSTEM NAME                      MOUNT_POINT
   --------------------------------      -------------------------
   db2data                               /db2fs/db2data
   db2fs1                               /db2sd_20120311151645
   db2log                                /db2fs/db2log
   ```

3.5.3 Changing the default database path

After you create the shared file system, you can change the default database location to use the new shared file system.

Attention: Complete this task from one host in your DB2 pureScale cluster.

To change the default database path, complete the following steps:

1. Log in as a DB2 instance user, for example, db2sdin1.
2. Verify the current database path by running the following command:

   ```
   db2 get dbm cfg | grep DFTDBPATH
   ```

   The output resembles the output shown in Example 3-51.

   Example 3-51 Check the database path

   ```
   db2sdin1@node101:~> db2 get dbm cfg | grep DFTDBPATH
   Default database path (DFTDBPATH) = /db2sd_20120311151645/db2sdin1
   ```

3. Change the `/db2sd_20120311151645/db2sdin1` path to `/db2fs/db2data`, which is the file system you created for the DB2 data. Use the following syntax to update the DFTDBPATH parameter.

   ```
   db2 update dbm cfg using DFTDBPATH <New database path>
   ```

To change the default database path, run the command shown in Example 3-52.

Example 3-52 Change the database path

```
db2sdin1@node101:~> db2 update dbm cfg using DFTDBPATH
/db2fs/db2data
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command
completed successfully.
```

4. DB2 picks up the new parameter dynamically and any new databases are created on the new database path by default. Execute the command shown in Example 3-53. The default database path is now `/db2fs/db2data`. Now, all databases created are placed under this path.

Example 3-53 Check the database path

```
db2sdin1@node101:~> db2 get dbm cfg | grep DFTDBPATH
Default database path (DFTDBPATH) = /db2fs/db2data
```

3.5.4 Changing the default log path

You also need to change the default log path to the log file system that you created earlier to store DB2 logs on the shared storage.

Attention: Complete these tasks from one host in the DB2 pureScale cluster.

To change the log file path, create the database first, and then change the `NEWLOGPATH` parameter to tell DB2 to use the new path for the log files. Use the SAMPLE database as the example in this section.

To change the default log path, complete the following steps:

1. Log in as a DB2 instance user and start DB2 by running **`db2start`**.

2. Connect to the SAMPLE database by running the command shown in Example 3-54.

Example 3-54 Connect to the SAMPLE database

```
db2sdin1@node101:~> db2 connect to SAMPLE

   Database Connection Information

Database server        = DB2/LINUXX8664 10.1.0
SQL authorization ID   = DB2SDIN1
Local database alias   = SAMPLE
```

3. Run the command in Example 3-55 to update the `NEWLOGPATH` parameter.

Example 3-55 Update the NEWLOGPATH parameter

```
db2sdin1@node101:~> db2 update db cfg using NEWLOGPATH /db2fs/db2log
DB20000I  The UPDATE DATABASE CONFIGURATION command completed
successfully.
SQL1363W  Database must be deactivated and reactivated before the
changes to one or more of the configuration parameters will be
effective.
```

4. Disconnect from the database by running a **`db2 terminate`**.
5. Deactivate and reactivate the SAMPLE database so that DB2 can pick up the new parameter (Example 3-56).

Example 3-56 Reactivate the SAMPLE database

```
db2sdin1@node101:~> db2 deactivate database SAMPLE
DB20000I  The DEACTIVATE DATABASE command completed successfully.
db2sdin1@node101:~> db2 activate database SAMPLE
DB20000I  The ACTIVATE DATABASE command completed successfully.
```

6. Connect to the SAMPLE database again, and run the command to get the log path information. The path is set to the shared `db2log` file system (Example 3-57).

Example 3-57 Retrieve the log file path information

```
db2sdin1@node101:~> db2 connect to SAMPLE
   Database Connection Information
Database server        = DB2/LINUXX8664 10.1.0
SQL authorization ID   = DB2SDIN1
Local database alias   = SAMPLE
db2sdin1@node101:~> db2 get db cfg | grep 'Path to log files'
Path to log files = /db2fs/db2log/NODE0000/LOGSTREAM0000/
```

3.5.5 Processor binding

When a cluster caching facility and a DB2 member coexist on a single host, the CF processes can be assigned to some processor cores to improve performance.

Attention: Complete this task from one host in the DB2 pureScale cluster.

To assign CF processor cores, complete the following steps:

1. Log in as an instance user, for example, db2sdin1.
2. As shown in Example 3-58, run **db2stop** to stop the database manager.

Example 3-58 Stop the DB2 manager

```
db2sdin1@node101:~> db2stop
03/13/2012 15:35:49     0   0   SQL1064N  DB2STOP processing was
successful.
03/13/2012 15:35:55     1   0   SQL1064N  DB2STOP processing was
successful.
SQL1064N  DB2STOP processing was successful.
```

3. Determine the primary CF ID in your cluster by running **db2instance -list** (Example 3-59, “Output of the db2instance -list command” on page 194). It is usually 128.

Example 3-59 Output of the db2instance -list command

```
db2sdin1@node101:~> db2instance -list
ID   TYPE           STATEHOME_HOSTCURRENT_HOSTALERTPARTITION_NUMBERLOGICAL_PORT
NETNAME
```

```
--   ----           ------------------------------------------------------------
-------
0  MEMBER         STARTED  node101     node101   NO               0           0
node101-10gige
1  MEMBER         STARTED  node102     node102   NO               0           0
node102-10gige
128CF          PRIMARY  node101     node101   NO               -           0
node101-10gige
129CF          CATCHUP  node102     node102   NO               -           0
node102-10gige
HOSTNAME   STATEINSTANCE_STOPPEDALERT
--------   --------------------------
 node102  ACTIVE              NO   NO
 node101  ACTIVE              NO   NO
```

4. Use the following syntax to assign some processor cores to the CF or member:

   ```
   db2set -i <DB2 Instance user> <CF ID> DB2_CPU_BINDING=<Value>
   ```

   For example, if you want to assign four processor cores to CF 128, run the command shown in Example 3-60.

   Example 3-60 Assign four processor cores to CF 128

   ```
   db2sdin1@node101:~> db2set -i db2sdin1 128
   DB2_CPU_BINDING="NUM_CORES=4"
   ```

   You can also specify processor cores to a CF or member (Example 3-61).

   Example 3-61 Specify the processor cores to CF 128

   ```
   db2sdin1@node101:~> db2set -i db2sdin1 128
   DB2_CPU_BINDING="PROCESSOR_LIST=2,10,6,14"
   ```

   Important: Repeat this step for other CFs as well, for example, CF 129.

3.5.6 Enabling SCSI-3 PR on a Linux environment

Enabling SCSI-3 PR for DB2 pureScale provides faster failover support.

To enable SCSI-3 PR, complete the following steps:

1. Log in as an instance user, for example, db2sdin1.

2. Run **db2stop** to stop the database manager (Example 3-62).

Example 3-62 Stop the DB2 manager

```
db2sdin1@node101:~> db2stop
03/13/2012 15:55:31 0   0   SQL1064N  DB2STOP processing was
successful.
03/13/2012 15:55:36 1   0   SQL1064N  DB2STOP processing was
successful.
SQL1064N  DB2STOP processing was successful.
```

3. List all the file systems by running **db2cluster -cfs -list -filesystem** (Example 3-63).

Example 3-63 List the file systems

```
node101:~ # /opt/ibm/db2/V10.1/bin/db2cluster -cfs -list -filesystem
FILE SYSTEM NAME                      MOUNT_POINT
---------------------------------     -------------------------
db2data                               /db2fs/db2data
db2fs1                                 /db2sd_20120311151645
db2log                                 /db2fs/db2log
```

4. Discover information about all the disks in the file system, and note the path names for the corresponding file systems being used in DB2 (Example 3-64).

Example 3-64 List the shared file systems

```
db2sdin1@node101:~> db2cluster -cfs -list -filesystem db2fs1
PATH ON LOCAL HOST                     OTHER KNOWN PATHS
---------------------------------     -------------------------
(*) /dev/dm-1
db2sdin1@node101:~> db2cluster -cfs -list -filesystem db2data
PATH ON LOCAL HOST                     OTHER KNOWN PATHS
---------------------------------     -------------------------
/dev/dm-2
db2sdin1@node101:~> db2cluster -cfs -list -filesystem db2log
PATH ON LOCAL HOST                     OTHER KNOWN PATHS
---------------------------------     -------------------------
/dev/dm-3
```

5. Run **tsprinquiry** as root to discover information about the disk (Example 3-65). The devices should correspond to the output given in step 4 on page 196. Save the output to the `/var/mmfs/etc/prcapdevices` file. Ensure that you save the output on all hosts in your cluster.

Example 3-65 Run the tsprinquiry command

```
node101:~ # /usr/lpp/mmfs/bin/tsprinquiry dm-1 >>
/var/mmfs/etc/prcapdevices
node101:~ # cat /var/mmfs/etc/prcapdevices
IBM     :1746      FAStT :1070
```

6. Run the following command to stop the resource pool domain:

```
/opt/ibm/db2/V10.1/bin/db2cluster –cm –stop –all -force
```

7. Stop IBM GPFS on all hosts by running the following command:

```
/opt/ibm/db2/V10.1/bin/db2cluster -cfs -stop -all
```

8. Enable SCSI-3 PR on all nodes by running **/usr/lpp/mmfs/bin/mmchconfig usePersistentReserve=yes** (Example 3-66).

Example 3-66 Enable SCSI3-PR

```
node101:~ # /usr/lpp/mmfs/bin/mmchconfig usePersistentReserve=yes
Verifying GPFS is stopped on all nodes ...
mmchconfig: Processing disk gpfs3nsd
mmchconfig: Processing disk gpfs1nsd
mmchconfig: Processing disk gpfs4nsd
mmchconfig: Command successfully completed
mmchconfig: Propagating the cluster configuration data to
allaffected nodes.  This is an asynchronous process.
```

9. Start the resource pool domain by running the following command:

```
/opt/ibm/db2/bin/db2cluster –cm –start –all
```

10. Start GPFS on all hosts by running the following command:

```
/opt/ibm/db2/V10.1/bin/db2cluster -cfs -start -all
```

11. Run **/usr/lpp/mmfs/bin/mmlsnsd -X** to check that SCSI3-PR is enabled on all file systems. The Remarks column shows `pr=yes` (Example 3-67).

Example 3-67 The output of the mmlsnsd command

```
node101:~ # /usr/lpp/mmfs/bin/mmlsnsd -X
Disk name    NSD volume ID      Device         Devtype  Node name            Remarks
---------------------------------------------------------------------------------------
gpfs1nsd     C0A801654F5CFA00   /dev/dm-1      dmm      node101.purescale.demo pr=yes
gpfs3nsd     C0A801654F5E29F1   /dev/dm-2      dmm      node101.purescale.demo pr=yes
gpfs4nsd     C0A801654F5E2A31   /dev/dm-3      dmm      node101.purescale.demo pr=yes
```

12. Start DB2 by running **db2start** (Example 3-68).

Example 3-68 Start the DB2 manager

```
db2sdin1@node101:~> db2start
03/13/2012 16:18:17 1   0   SQL1063N  DB2START processing was
successful.
03/13/2012 16:18:17 0   0   SQL1063N  DB2START processing was
successful.
SQL1063N  DB2START processing was successful.
```

4

Working with a DB2 pureScale cluster

This chapter provides information about how to work effectively with a DB2 pureScale cluster. It demonstrates how to perform basic operations with individual cluster components and describes the maintenance tasks that can be performed on the cluster. Finally, it shows how to monitor a cluster for performance and health purposes.

This chapter includes the following topics:

- Operations
- Maintenance
- Monitoring

4.1 Operations

This section provides information about how to perform basic operations against DB2 pureScale cluster components. This information can be used when performing maintenance tasks.

> **Tip:** Although the subsequent sections show how to start individual components, running a global **db2start** command starts all the DB2 pureScale cluster components simultaneously.

4.1.1 Querying cluster components

In this section, you learn how you can use the monitoring function to obtain the overall state of the health of a cluster. This section also highlights important areas when assessing the output of a monitoring command.

To query the overall state of a cluster, run **db2instance -list**, which returns an output similar to that shown in Example 4-1.

Example 4-1 Sample output of db2instance -list showing the overall cluster state

```
ID   TYPE     STATE    HOME_HOST   CURRENT_HOST   ALERT   PARTITION_NUMBER  LOGICAL_PORT  NETNAME
--   ------   -------  ---------   ------------   ----    ----------------  ------------  -------
0    MEMBER   STARTED  node102     node102        NO      0                 0             node102
1    MEMBER   STARTED  node103     node103        NO      0                 0             node103
128  CF       PRIMARY  node102     node102        NO      -                 0             node102
129  CF       PEER     node103     node103        NO      -                 0             node103

HOSTNAME                  STATE                INSTANCE_STOPPED        ALERT
--------                  -----                ----------------        -----
node103                   ACTIVE               NO                      NO
node102                   ACTIVE               NO                      NO
```

The following important pieces of information from the output are worth mentioning:

ID Refers to the identifier of a component. It is used and referenced when starting and stopping a component.

STATE Represents the current state of the component and can be helpful when troubleshooting issues. For more information about what states can occur, see “Interpretation of DB2 status information” on page 236.

ALERT	Indicates whether an alert is raised against the component. For more information about why an alert might be raised for a component, see “Interpretation of DB2 status information” on page 236.
CURRENT_HOST	Represents the host that a component is currently on. It is important to understand that when a member is performing crash recovery, its current host might differ from its home host. As such, this information can be a vital piece of information when assessing issues or problems with a cluster.
INSTANCE_STOPPED	Indicates whether the instance on the host is stopped.

4.1.2 Working with a cluster caching facility

This section describes how to start, stop, and query the status of a cluster caching facility (CF).

Starting a caching facility

The first cluster CF that starts successfully assumes the role of the *primary* CF, which manages, coordinates, and distributes pages and locks to the appropriate members, thus playing a vital role in the DB2 pureScale cluster. If there is already an existing CF that assumed the role of primary CF, the subsequent CF start assumes the role as secondary CF upon successfully starting.

Although the primary CF takes responsibility to interact with the members' requests and updates, a copy of that information is also sent to the secondary CF to ensure that the secondary is synchronized with the primary CF. When the secondary CF is synchronized with the primary one, it has the `PEER` state. For more information about the `PEER` state, see “Interpretation of DB2 status information” on page 236.

To start a CF, run the following command:

```
db2start cf [CF-identifier]
```

For example, if you want to start a cluster CF with ID 129, run the following command:

```
db2start cf 129
```

Stopping a caching facility

You might need to stop a cluster CF. Although this task can be done rather easily, there must be at least a single cluster CF started and running. Otherwise, the cluster cannot function.

You cannot stop the primary cluster CF if you do not have a secondary cluster caching facility in the `PEER` state. A **`FORCE`** option stops the primary cluster CF if you do not have a secondary cluster caching facility in the `PEER` state; however, this option triggers a Group Restart of the database instance.

You can stop the secondary cluster CF by using the **`FORCE`** option, even if any of the following situations are true:

- There are active members in the instance.
- The primary cluster caching facility contains dirty pages.
- The primary cluster caching facility contains locks.

To stop a CF, run the following command:

```
db2stop cf [CF-identifier]
```

For example, if you want to stop the secondary CF in a cluster with ID 129 and ensure that the CF is in the `PEER` state, run the following command:

```
db2stop cf 129
```

Querying the caching facility for its status

You can monitor the functionality of elements to obtain information that pertains to the CF. To view the status of all the cluster caching facilities in a DB2 pureScale cluster, run the following command:

```
SELECT * FROM SYSIBMADM.DB2_CF
```

Example 4-2 shows the output of this command.

Example 4-2 Sample output of the DB2_CF administrative view

```
ID  CURRENT_HOST STATE                          ALERT
--- -----------  ------------------------------ ---------
128 node102      PRIMARY                        NO
129 node103      PEER                           NO

2 record(s) selected.
```

4.1.3 Working with a member

This section explains how to start, stop, and query the status of a member.

Starting a member

When starting a member, the database manager starts the member on the host and on any idle processes that are required for crash recovery. After the member is started successfully, it is introduced to the cluster and is available to accept client connections.

> **Important:** The databases within the quiesced instance are activated to do authorization checking for all connection attempts to the database. This authorization is necessary to determine if the connecting user ID has DBADM authority. This authority is stored in the database catalog, and the database must be activated to determine if the user ID has the authority. To prevent this authorization checking from happening, specify the restricted access option.

To start a member, run the following command:

```
db2start member [member-identifer]
```

For example, if you want to start member with ID 2, run the following command:

```
db2start member 2
```

Stopping a member

Stopping a member stops the member but keeps other instance processes on the host up and running, implying that any other member or cluster CFs on the same host are not affected. However, when you perform maintenance, stopping a member is not the only prerequisite for performing maintenance on a host, because the host can remain a viable failover target for other members that must perform crash recovery. For more information about what processes you need to stop for host maintenance, see 4.3.1, “Performance monitoring” on page 219.

Similar to stopping a cluster CF, there are important conditions to consider when stopping a member. You cannot stop a member if there are any active database connections on the member. You also cannot stop a member if the member is in a restart light mode and has not yet completed member crash recovery. These conditions prevent the use of the **db2stop** command. You can stop the member by using the **FORCE** option, but use *extreme caution* when using this option. Do not stop a member that is undergoing crash recovery.

To stop a member, run the following command:

```
db2stop member [member-identifier]
```

For example, if you want to stop a member with ID 2, run the following command:

```
db2stop member 2
```

Querying for member status

This section explains how to view specific details that pertain to all of the members in a DB2 pureScale cluster. To view the status of the members in a DB2 pureScale environment, run the following command:

```
SELECT * FROM SYSIBMADM.DB2_MEMBER
```

Example 4-3 shows the output of this command.

Example 4-3 Sample output of the DB2_MEMBER administrative view

```
ID     HOME_HOST    CURRENT_HOST    STATE          ALERT
---    ----------   -------------   -----------    -------------
0      node102      node102         STARTED        NO
1      node103      node103         STARTED        NO

2 record(s) selected.
```

4.1.4 Activating and deactivating DB2 pureScale databases

In a DB2 pureScale environment, a database can be activated explicitly or implicitly when the first client connects. Furthermore, a database that was explicitly activated (by running the **activate database** command) or implicitly activated (by a user connection) can be deactivated only by running a **deactivate database** command.

Activating a database

To explicitly activate a database on all active members in a DB2 pureScale instance, run the following command:

```
db2 activate database [database-name]
```

For example, if you want to explicitly start a database called *test*, run the following command:

```
db2 activate database test
```

Deactivating a database

Even if you have no users connected to the database or if the last user disconnects, the database remains activated until an explicit **deactivate database** command is run.

To explicitly deactivate a database, run the following command:

```
db2 deactivate database [database-name]
```

For example, if you want to deactivate a database named *test*, run the following command:

```
db2 deactivate database test
```

In addition, if there is a particular need to deactivate the database from a specific member, run the following command:

```
db2 deactivate database [database-name] member [member-id]
```

For example, if you wanted to deactivate member 2 from the same database, run the following command:

```
db2 deactivate database test member 2
```

4.1.5 Quiescing an operation

When you quiesce an operation, you can perform maintenance by draining all activity that is directed against a member, a database, or an instance without needing to disrupt it with a forced stop action. This section shows how to perform a quiesce operation for a member, a database, and an instance.

Quiescing a member

You might need to perform maintenance on a member in the DB2 pureScale instance. For example, operating system updates or hardware updates might be required over time. To ensure the least amount of impact to running transactions, DB2 offers an optional **quiesce** parameter that drains current activity that is running against a specific member.

While the member is being drained, new transactions or connections are redirected through automatic client rerouting to the least loaded member. Thus, after the member's existing connections or transactions are completed, you can perform maintenance tasks against this member. This method is an effective way to ensure that all currently running transactions on the affected member finish and that you have as much time as needed to work on the member. The quiesce stay in effect until you start the member by running **db2start**.

The **quiesce** parameter also has an optional **timeout** value, specified in minutes, that can be used to ensure that the current activity is finished in a timely manner. The **timeout** parameter includes the following values:

- If you specify a **timeout** value when you are quiescing a member, applications have up until that specified **timeout** value to conclude their units of work. After the timeout is reached, the remaining connections are forced off.

- If no **timeout** value or a value of -1 is specified, the server waits indefinitely until all active transactions or associated connections on the member end.
- If you specify a value of 0 for the **timeout** value, connections are forced off immediately.

Thus, depending on the urgency to remove the member from the instance, you can elect to select the criteria that meets your wanted expectations.

To quiesce a member, run the following command:

```
db2stop member [member-id] quiesce
```

For example, if you want to perform some minor maintenance to the hardware on a member with ID 1 but you want to give all current connections or transactions 30 minutes to conclude before they are forced off, run the following command:

```
db2stop member 1 quiesce 30
```

To unquiesce a member, you need to run **db2start**. For example, if the maintenance for a host with ID 1 concludes and you want to bring it back up, run the following command:

```
db2start member 1
```

Quiescing a database

If you need to remove all users from a specified database to perform maintenance tasks, you can place your database in quiesce mode. While quiesced, only users that have the appropriate permission can access the database and its objects. Any other users that attempt to access the database are prompted with a message that states that the database is quiesced.

You can run **quiesce db** on any member. The database manager ensures that the database is quiesced based on the user's specifications. For example, if a database administrator wants to place a database in quiesce mode immediately, the DBA uses the **immediate** option. The database manager then forces all current transactions to terminate immediately, rolling back any uncommitted transactions and ensuring that no new requests to process transactions or connections are accepted. Thus, just like quiescing a member, you can elect when the database is quiesced and how to handle all current transactions.

After the database is in a quiesced state, the database remains in this state for all members until you run an explicit **unquiesce db** command, ensuring that the database remains in a quiesced state until it is explicitly unquiesced by you.

To place a database in a quiesced state, run the following command:

```
db2 quiesce db
```

Important: A connection must be made to the database before a quiesce operation can be run. The default behavior when a **quiesce** command is run is to force off all connections. If this is not the behavior that you want, specify another option that meets your criteria. For example, you can specify a timeout period to allow all current connections or transactions to potentially finish their unit of work before being forced off.

To unquiesce a database, run the following command:

```
db2 unquiesce db
```

After the database is successfully unquiesced, the database is accessible for all members and attempts to connect to the database by clients can proceed.

Quiescing an instance

An instance in DB2 can also be placed into quiesced mode. When the instance is placed in to this state, any databases that are in the instance are also placed into the quiesce state. Furthermore, if an instance is in a quiesced state, any database that is within the instance cannot be explicitly quiesced. You must first unquiesce the instance before interacting with any dependent database.

After an instance is quiesced, user access to the instance is prevented, unless the user is granted the authority to access the quiesced instance (for example, if the user has system administrator authority).

Important: The databases within the quiesced instance are activated to do authorization checking for all attempts to connect to the database. This authorization is necessary to determine if the connecting user ID has DBADM authority. This authority is stored in the database catalog, and the database must be activated to determine if the user ID has the authority. To prevent this authorization checking, specify the restricted access option.

To place an instance in a quiesced state, run the following command:

```
db2 quiesce instance [instance-name]
```

For example, Paul, the system administrator wants to quiesce the instance named db2inst1, while allowing a user named Doug to access the database. Furthermore, Paul does not want to wait for all transactions to commit and wants to force all connections immediately. To accomplish these tasks, Paul runs the following command:

```
db2 quiesce instance db2inst1 user doug immediate force connections
```

After you complete the tasks that must be done against the instance, reintegrate the instance to restore user access and databases in the instance. To perform this unquiesce operation, you must have SYSADM or SYSCTRL authority. Without this level of authority this operation cannot be performed against DB2.

To unquiesce the instance, run the following command:

```
db2 unquiesce instance [instance-name]
```

For example, when Paul, the system administrator, completes all the relevant tasks against the instance and wants to bring the instance named db2inst1 back online. Paul runs the following command:

```
db2 unquiesce instance db2inst1
```

4.2 Maintenance

Regardless of the type of environment that you have deployed, some type of maintenance is required, whether it is adding more space to the file system, adding or removing a member or CF, applying security patches to the operating system, and so on. This section provides an understanding of the various maintenance tasks that can be performed against a DB2 pureScale cluster.

4.2.1 Adding or removing components

This section explains how to add or remove a member or cluster CF to or from a cluster. The focus of subsequent sections illustrates a command-line Interface (CLI) approach by running the **db2iupdt** command with an existing instance. For more information about how to install components by running the **db2setup** command, see Chapter 1, “An overview of the DB2 pureScale Feature” on page 1.

Important: Any time you modify the topology of a DB2 pureScale environment by adding or removing a member, you must take an offline backup before you can access the database. Any attempt to access the database before an offline backup results in a warning. The warning notifies the user that the database is in a backup pending state. You can, however, add or remove multiple members without having to take a backup after each successive change to the topology of a cluster.

Adding or removing a member

One maintenance task that you might encounter is the necessity to add or remove a member from your instance. When you add a member to your DB2 pureScale cluster, only a few simple pieces of information are required. The host name and cluster interconnect name must be identified for the additional member. The DB2 pureScale Feature **instance update** command simplifies the addition by installing the required binary files and applies any necessary licenses on the new host.

In addition to the binary files installation, the command verifies that the prerequisites are met on the new host before it begins this process to ensure a high degree of success. This approach is a simplified approach that reduces the number of manual tasks the DBA would otherwise perform on every host.

To add a member to a DB2 pureScale Feature instance, run the following command:

```
db2iupdt -add -m [host1] -mnet [netname] [instance-name]
```

For example, if you have a host named purescale-mem1 with a cluster interconnect netname member1-ib0, and an instance with the name db2inst1, run the following command:

```
db2iupdt -add -m purescale-mem1 -mnet member1-ib0 db2inst1
```

Removing a member from the instance can be done by running a similar command. DB2 handles the removal of the member from the DB2 pureScale Feature instance, the removal of the host from the Reliable Scalable Cluster Technology (RSCT) peer domain, and the removal of the host from the GPFS cluster. You must ensure that all of the DB2 processes are stopped by running **db2stop** before trying to remove the member. After the member is removed, remove the binary files by running **db2_deinstall**.

To remove a member, run the following command:

```
db2iupdt -drop -m [hostname] [instance_name]
```

For example, if you have a host name named purescale-mem1 with instance named db2inst1, run the following command to remove the member from the instance:

```
db2iupdt -drop -m purescale-mem1 db2inst1
```

Adding or removing a cluster caching facility

A DB2 pureScale cluster must have at least one active cluster CF for the cluster to remain operational. You can elect to have a second cluster CF to duplex global buffer pool pages and lock requests and internal metadata used by the recovery process, should anything happen to the primary CF. A second CF might improve the high availability characteristics of a DB2 pureScale instance; however, a minimum of one CF is required. You can have a maximum of two CFs.

A cluster CF can be added to an instance by running the following command:

```
db2iupdt -add -cf [host1] -cfnet [netnames] [instance-name]
```

For example, if you want to add another cluster CF with the host name purescale-cf2 and cluster interconnect net names cf2-ib0 and cf2-ib1 to an existing instance named db2inst1, run the following command:

```
db2iupdt -add -cf purescale-cf2 -cfnet cf2-ib0, cf-ib1 db2inst1
```

You can remove a cluster CF from the instance by running a similar command and specifying the **drop** option instead of the **add** option. Before you run the command, you must ensure that this cluster CF is not the last cluster CF in the cluster and that you stopped all DB2 processes on the host by running a **db2stop** command. DB2 prevents you from dropping the last CF from the DB2 pureScale instance. After the CF is dropped, you can then remove the binary files, if this was the last component on the host.

To remove a cluster CF, run the following command:

```
db2iupdt -drop -cf [host-name] [instance-name]
```

For example, if you want to drop the secondary cluster CF and ensured that all the processes are stopped (by running a **db2stop** command) on the purescale-cf2 host name on the db2inst1 instance, run the following command:

```
db2iupdt -drop -cf purescale-cf2 db2inst1
```

4.2.2 Putting components in to maintenance mode

When you perform updates to DB2 cluster services, you must put the target host into maintenance mode. You can also put a host into maintenance mode if you want to ensure that the specified member or cluster CF is not restarted on the host if you are applying updates to the operating system or hardware. Maintenance mode is an explicit set of commands intended to prevent the accidental restarting of DB2 by either the DB2 cluster services or DBAs.

Putting a host in to maintenance mode

In this chapter, we explained how to stop and quiesce a member from an instance to perform maintenance tasks. Putting a member into maintenance mode is slightly different. This method ensures that the member or CF (or both) are not restarted on the host on which you are performing the maintenance and that the host is not a viable host on which to perform crash recovery. Maintenance mode also stops the cluster manager and the cluster file system.

When putting a host into maintenance mode, consider the following items:

- Do not place the hosts of both of the cluster CFs into maintenance mode at the same time. At least one active cluster CF for the DB2 pureScale instance must remain operational.
- Do not place all hosts with members into maintenance mode at the same time. At least one active member for the DB2 pureScale instance must remain operational.
- Do not put more than 50% of the hosts in the DB2 pureScale instance in to maintenance mode at the same time. Operational quorum rules require that at least 50% of hosts in the cluster are active.

Furthermore, a host cannot enter maintenance mode if any DB2 processes are still active on the host.

To put a host into maintenance mode, complete the following steps:

1. Identify the host on which you are performing maintenance. If this host is one with DB2 members, you can start the process to drain all the current activities off the affected members to ensure that all active work is processed. If this host is a cluster CF host, you can omit this step.

   To start the drain process for a member, run the following command:

   ```
   db2stop member [member-id] quiesce
   ```

2. Remove the host from active operations in the DB2 pureScale cluster. This command takes the host out of the failover targets that are usable by the DB2 instance.

   To prevent the selected host from becoming a recovery target for the instance, run the following command:

   ```
   db2stop instance on [host-name]
   ```

3. Stop the cluster manager on the selected host. The cluster manager is responsible for managing the health of the components that are on the host. As such, stopping the cluster manager on the host ensures that the components that are on it are not restarted and makes the host a non-viable option to perform crash recovery.

To put the cluster manager into maintenance mode, run the following command:

```
db2cluster -cm -enter -maintenance
```

4. Stop the cluster file system service, which is the last step before the host itself can be deemed in maintenance mode. The purpose of stopping this service is to ensure that the host can unload the file system kernel module if the module needs to be updated, for example, for DB2 Cluster Services updates or if operating system patches are being applied.

   To stop the cluster file system service, run the following command:

   ```
   db2cluster -cfs -enter -maintenance
   ```

The host is now in maintenance mode and any operation to update the operating system or hardware can be done. Upon conclusion of any update, you must reintegrate the host and any relevant DB2 pureScale component that is on the host back into the cluster. Use the following process (which is *not* the opposite of the steps used to put the host *into* maintenance mode):

1. Restart the cluster manager service so that the host can reintegrate with the DB2 cluster. To start the cluster manager service, you must exit maintenance mode by running the following command:

   ```
   db2cluster -cm -exit -maintenance
   ```

   After you start the cluster manager service, the cluster file system restarts on its own. If, for some reason, the cluster file system does not restart, run the following command to make it restart:

   ```
   db2cluster -cfs -exit -maintenance
   ```

2. With all important background processes started and the host itself now out of maintenance mode, integrate the DB2 pureScale components back into the instance. Run the following command to notify DB2 that the host can be reintegrated into the cluster:

   db2start instance on [*host-name*]

3. Finally, start any relevant DB2 pureScale components by running the following command. In this case, the member or CF needs to be resumed.

   db2start member [*member-id*]

Putting a cluster in to maintenance mode

There might be cases where putting the entire DB2 pureScale cluster into maintenance mode is more efficient than rolling the maintenance through the cluster. For example, for a standby or development cluster that does not need to be constantly available, you can place the entire cluster into maintenance mode using fewer steps.

To place your entire cluster into maintenance mode, complete the following steps:

1. As the instance user, stop the database manager on all hosts by running the appropriate **db2stop** command. You can either stop the instance gracefully after disconnecting all applications or you can use a **quiesce** option to drain connections as necessary.
2. Stop the instance on all the hosts by running the following command once for each host:

   ```
   db2stop instance on [host-name]
   ```

   This command can be run by the DB2 pureScale instance user on any of the hosts. In other words, you do not need to be logged in to the host that will be stopped for this command to work. Do not forget to do this step for every host that belongs to the cluster.
3. Place both the cluster manager and cluster file system services in maintenance mode by running the following commands:

   – **db2cluster -cm -enter -maintenance -all**
   – **db2cluster -cfs -enter -maintenance -all**

   Placing the cluster manager into maintenance mode stops any potential automation of DB2 components, and prevents the services from restarting accidentally. Placing the cluster file system service in maintenance mode ensures that the host cannot interact with the shared storage and allows the kernel module used by the shared storage to be unloaded if necessary.
4. After you complete your updates, reintegrate all components by following the same logic that was applied to individual hosts. Start by exiting maintenance on the cluster manager, followed by the cluster file system (if needed), and then run **db2start instance on *[hostname]*** for all hosts before running **db2start** to restart the DB2 pureScale instance.

4.2.3 Managing disk space on a shared storage file system

Managing your file system is necessary. As data continues to grow exponentially, you must manage capacity. This section shows how you can add and remove a disk or disks and how to rebalance your file systems.

Adding a disk to a shared storage file system

After creating your shared storage file system for the DB2 pureScale cluster, you might need to add space to accommodate a growing database. Adding more space to a file system and making it available to be used by the cluster is easy.

After the disk is available to the host, you can run the **db2cluster** command to add the disk to the file system:

```
db2cluster -add -filesystem [filesystem-name] -disk [disk-name]
```

The disk is added to the shared storage file system and is available to be used by the DB2 pureScale cluster. When a disk is added to the file system, it is used immediately. As a result, consider rebalancing the file system at a time when it is more convenient, because rebalancing is an I/O-intensive operation. For more information, see “Rebalancing a file system” on page 215.

Removing a disk from the shared file system

There might be cases where, in the initial setup of a shared storage file system, you allocate more disk space than is needed. For this reason, you might want to remove a disk from the file system to be used elsewhere. Similar to adding a disk to the shared storage file system, you can remove a disk by running the **db2cluster** command. However, be aware of the following considerations before you run this command:

- The instance owner can perform this task only if the instance owner created the file system; otherwise, only the DB2 cluster services administrator (or root) can remove a disk from a file system.
- If there is only a single disk in the file system, you cannot remove it by running the **db2cluster** command. If you do not intend to delete the file system, you must *add* a disk to the file system before you can remove the current one.
- You cannot remove the tiebreaker disk. If you want to remove a disk that is also the tiebreaker disk for the shared file system, you must change the tiebreaker disk to another disk before it can be removed.
- Only one disk at a time can be removed from a file system.
- There must be enough space left over on the remaining disks to hold the data that was contained by the disk that is being removed.

To remove a specific disk from the shared storage, run the following command:

```
db2cluster -cfs -remove -filesystem [filesystem-name] -disk [disk-name]
```

Upon running the command, the disk is successfully removed from the shared storage file system if there are no errors.

Important: Disk removal is an I/O-intensive operation. Removal of a disk requires that any data held on that disk be moved to the remaining disks in the file system. If there is not enough space, the disk is not removed. After you remove the disk from the file system, you can remove it immediately from the operating system or reuse it for other purposes.

The removal of a disk from the shared storage file system requires a rebalancing operation to ensure that the file system is balanced over all the remaining disks in the file system.

Rebalancing a file system

Perform a rebalance operation when you add space or remove one or more disks from an existing file system. This operation ensures that the file system data is balanced evenly across all the disks in the file system by restripping the data for all disks in the file system.

Important: A balance operation is I/O-intensive. Data is moved from one or more disks to one or more other disks. For this reason, perform this operation only when there is reduced system activity on the cluster. Furthermore, only the instance owner can rebalance the file system if it was created by the instance owner; otherwise, only the DB2 cluster services administrator can rebalance the file system.

To perform a rebalance operation on your file system, run the following command:

```
db2cluster -cfs -rebalance -filesystem [filesystem-name]
```

4.2.4 Advanced operations

This section describes how to manage processors and memory allocations to address performance issues or to reallocate resources. Each section is divided into two areas (for the AIX and Linux operating systems) to provide detailed information when addressing the situation for an operating system.

Managing processors for members

You might need to add or remove processors for a member when trying to address performance or reallocate resources. This section highlights how to manage processor capacity when dealing with AIX or Linux.

AIX

If you have extra processor capacity that is not being used or want to reassess your allocation, this task can be accomplished fairly easy on AIX. Because a member is on an LPAR on AIX, you can dynamically reassess processor capacity for a member without having to reboot the server, resulting in no downtime.

AIX has *physical processors* that are assigned to an LPAR and *virtual processor* that the operating system uses. The physical processors that are assigned can never be more than the number of virtual processors defined for the operating system. If you need to use more physical processors than are assigned, you need to have enough virtual processors assigned to the operating system. If you do not, you must reboot the system to activate any extra processors. This task can be accomplished in a rolling fashion through the DB2 pureScale cluster so that you do not need to shut down DB2 completely to accomplish this task.

This process of adjusting processor capacity is performed by using the Hardware Management Console (HMC). See the AIX documentation to learn how this task can be done.

Linux

You can add extra capacity for a member that is on a Linux operating system, but the process is not as dynamic. Linux does not use the concept of virtual processors, so if you want to add additional processors to the server, you must shut down the host to add physical processors to the host, and then activate the extra capacity as needed. If the extra processors are already on the system, it might be a boot option that you need to adjust, after which you can then roll through the members in the DB2 pureScale instance to activate these extra processors.

Managing memory for members

You might need to add or remove memory for a member when you are trying to address performance or reallocate resources. This section shows how you can manage memory capacity when dealing with AIX or Linux.

AIX

When allocating additional memory to be used by a member on AIX, the increase of capacity is dynamic. Thus, the server does not need to be rebooted, and the member can slowly start using the additional memory. However, if you are removing memory from a member, this process requires that you first stop the member by running the **`db2stop member`** command.

After you stop the member stopped, you can change the LPAR memory allocation accordingly and then restart the member on the LPAR. Memory de-allocation requires that the memory to be de-allocated is not in use by the LPAR. Thus, you need to shut down the member from the LPAR from which the memory will be removed.

Linux

In most instances, adding additional memory in Linux requires physically installing more memory on the host that can be used by DB2. This situation means that the server must be shut down to perform this task. When removing memory from a member, you can adjust the memory allocations by using the **`instance_memory`** configuration parameter, but this process is a slow, progressive decrease coordinated by the Self-tuning Memory Manager (STMM) to ensure stability. This situation also assumes that the removal is not a physical removal of memory from the host. If you plan to remove physical memory from a Linux host, you must shut down the associated DB2 member before you can remove the physical memory.

4.2.5 Backup and restore

This section describes important regular maintenance operations that you must perform to ensure that a database can be restored when an unrecoverable failure of a cluster occurs.

Backup

Backing up your database should be part of basic, regular maintenance on any DB2 instance, regardless of the DB2 pureScale Feature. This maintenance ensures that you can restore your database if there is catastrophic failure.

In the DB2 pureScale cluster environment, when a **`backup database`** command is run on any member, it initiates a backup operation on behalf of all members. Because the DB2 pureScale cluster consists of only a single partition, a backup operation has only one set of data to process, and as such, produces a single backup image for the entire cluster. The database metadata and transaction logs of other members are automatically processed and included in the single backup image, without any extra work by you. A backup image for a DB2 pureScale database includes data from all table spaces, any required metadata, configuration information for all currently defined members, and all transactions logs for the defined members.

You can use the following types of backup operations against a DB2 pureScale database:

- Offline backup

  All members must be in a consistent state before you attempt an offline backup operation. Only a single offline backup can occur at a time, as the backup utility acquires super-exclusive access to the database across all members.

- Online backup

  All members are not required to be online when an online backup is occurring. In other words, members that are stopped or that failed for any reason do not impede the progress of a database backup. However, the backup waits for a short duration while member crash recovery is completed on an offline and inconsistent member before proceeding to ensure that the backup image can be used for recovery later. All logs (including logs from stopped members) are included in the database backup to ensure a consistent image. Furthermore, concurrent online backups can occur; however, backup operations are not allowed to copy the same table spaces simultaneously, and each backup process must wait its turn.

Important: Backup and restore operations are not supported between an environment where the DB2 pureScale Feature is installed and an environment where the DB2 pureScale Feature is not installed. Furthermore, any direct changes to the topology of your DB2 pureScale cluster, such as removing or adding members, places the database in a backup pending state. In this case, a full offline backup is required to remove the pending backup and to allow users to connect to the database.

To back up a database, run the **backup** command. For example, if you want to perform a backup of a database called *sample*, run the following command:

```
backup db sample
```

Restore

If a catastrophic failure cannot be resolved, you must restore your database back to a healthy state. Similar to the backup process, a single **restore db** command can restore the database and member-specific metadata for all members. There is no need perform additional restore operations on any other member to restore the cluster.

For example, if you wanted to restore a database named *test,* run the following command from any of the hosts:

```
restore db test
```

4.3 Monitoring

The DB2 pureScale Feature offers a flexible, highly available database environment. With this type of environment comes the necessity to monitor its functionality and performance.

You can use functional monitoring to become aware of possible issues and areas of concern. Although in most cases issues in a DB2 pureScale cluster can be contained automatically by DB2 with no manual intervention required, there are situations where this recovery automation can potentially mask issues that are occurring, thus leading to unwanted performance.

For example, one problem that can impact your cluster is a hardware-related issue with one of your hosts. If you are aware of the issue, rectifying the problem is easily manageable, and in most cases you can quickly bring your host back to a healthy state. However, if you are unaware of the underlying issue and it goes unrecognized, DB2 recovery automation handles this problem by failing over to another host. This example takes into account only a single failure, so imagine a scenario where the troubled host repeatedly fails. DB2 then expends many cycles to provide recovery automation. Therefore, actively monitoring your environment can aid in discovering problems that are occurring so that you can address them before they become an issue.

You can use performance monitoring to assess the performance and capacity of your cluster. You cannot fine-tune performance only for the present, but also accurately assess capacity and requirements for the future.

4.3.1 Performance monitoring

This section highlights monitoring elements that can help you gain a basic understanding of performance monitoring and then elaborates on possible preferred practices for performance monitoring.

Interfaces and tools

In addition to viewing the overall status and state of the components in the DB2 pureScale cluster, you can examine specific aspects of the operation of the cluster caching facilities and members by using the DB2 monitoring capabilities.

As the database administrator, you can use table functions and administrative views to get information that you need. Within DB2 itself, there are defined sets of table functions that use in-memory monitoring elements to provide detailed information about any specific point in time. The DB2 pureScale Feature extends the existing DB2 monitoring capabilities by letting you view specific aspects of the cluster caching facilities and members in a DB2 pureScale instance.

The subsequent sections demonstrate some of the important table functions and administrative views that you can use when monitoring for performance. For more information, see the DB2 Information Center at the following website:

http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/index.jsp

You also can use tools such as IBM Data Studio or Optim™ Performance Manager to assist in daily tasks and monitor your DB2 pureScale instance.

IBM Data Studio provides an integrated development environment for database administration for any DB2 environment. Database administrators can perform tasks against a DB2 pureScale environment, which include quiescing a member, placing a host into maintenance mode, and so on. For more information, see the IBM Data Studio Information Center at the following website:

http://publib.boulder.ibm.com/infocenter/dstudio/v1r1m0/index.jsp

Optim Performance Manager provides database administrators with a tool to monitor and tune DB2 environments. It also provides you with the ability to view the overall health of databases and drill down for more information. For more information, see the Integrated Data Management Information Center at the following website:

http://pic.dhe.ibm.com/infocenter/idm/v2r2/index.jsp

Buffer pool monitoring

A fundamental concept of assessing performance in any DB2 environment is calculating the extent to which required pages are found in memory as opposed to on disk. This situation is called the *buffer pool hit ratio*. Although this type of assessment is an important factor when assessing performance, you must investigate other factors as well. In general, a low buffer pool hit ratio might be an indicator that an environment is under performing. Adjusting the memory allocations might improve performance.

However, in a DB2 pureScale cluster, knowing the buffer pool hit ratio does not provide enough information to accurately assess a cluster's performance. Two additional hit ratios provide more information when dealing exclusively with a DB2 pureScale environment:

- The *local buffer pool (LBP) hit ratio* is a representation of requested pages that are found in the memory of the member as opposed to having to request the page from the group buffer pool or retrieve it from disk.

  In a DB2 pureScale cluster, if a page cannot be found in the LBP, the member requests the page from the *group buffer pool (GBP)*. A low LBP hit ratio can indicate performance issues. Allocating additional memory to the local buffer pool might help, because more pages can be cached locally, thus improving the LBP hit ratio.

- The *group buffer pool (GBP) hit ratio* is a representation of pages that are requested by members to the CF that are found in memory.

  If the page does not exist in the GBP, then there is no choice but to retrieve the page directly from disk. A low GBP hit ratio can indicate that performance can be improved by increasing the GBP memory allocation (if the workload has a high write component). A low GBP hit ratio in a highly read-only workload is not unusual. More pages can be cached locally on the GBP as opposed to having to make I/O calls to the disk, which is the slowest operation that can be done.

Buffer pool monitoring elements

The following buffer pool monitoring elements are available:

POOL_DATA_LBP_PAGES_FOUND	Number of times a data page was resolved in the local buffer pool.
POOL_DATA_GBP_L_READS	Number of times a GBP-dependent data page was attempted to be read from the group buffer pool because the page was either invalid or not present in the local buffer pool.
POOL_DATA_GBP_P_READS	Number of times a group buffer pool-dependent data page was read in to the local buffer pool from disk because it was not found in the GBP.
POOL_DATA_GBP_INVALID_PAGES	Number of times a data page was invalid in the local buffer pool and was read from the group buffer pool instead.
POOL_ASYNC_DATA_GBP_L_READS	Number of times a group buffer pool-dependent data page was attempted to be read from the group buffer pool by a prefetcher because the page was either invalid or not present in the local buffer pool.

Buffer pool monitoring examples

As the GBP plays a vital role in the coordination of pages among members, general monitoring of buffer pool hit rates and ratios is important. Here are a few examples that you can use to monitor your cluster:

- The following query returns the sum of all data pages that were requested from the buffer pool and the sum of all data pages that were read in from disk:

  ```
  SELECT SUM(POOL_DATA_L_READS) AS POOL_DATA_L_READS,
  SUM(POOL_DATA_P_READS) AS POOL_DATA_P_READS FROM
  TABLE(MON_GET_BUFFERPOOL('', -2))
  ```

  Example 4-4 shows the output of this query.

  Example 4-4 Sample output of overall cluster logical and physical reads

  ```
  POOL_DATA_L_READS    POOL_DATA_P_READS
  -------------------- --------------------
                 469325                19762

    1 record(s) selected.
  ```

- The following query returns the sum of all data pages that were resolved in the local buffer pool and the sum of all data pages that were present in the local buffer pool when a prefetcher attempted to access it for each member in the DB2 pureScale instance:

  ```
  SELECT MEMBER, SUM(POOL_DATA_LBP_PAGES_FOUND) AS
  POOL_DATA_LBP_PAGES_FOUND, SUM(POOL_ASYNC_DATA_LBP_PAGES_FOUND) AS
  POOL_ASYNC_DATA_LBP_PAGES_FOUND FROM TABLE(MON_GET_BUFFERPOOL('',
  -2)) GROUP BY MEMBER
  ```

  Example 4-5 shows the output of this query.

  Example 4-5 Sample output of sum of data pages found within the LBP

  ```
  MEMBER POOL_DATA_LBP_PAGES_FOUND POOL_ASYNC_DATA_LBP_PAGES_FOUND
  ------ ------------------------- -------------------------------
       0                    127427                              21
       1                    130315                              15

    2 record(s) selected.
  ```

- The following query returns the sum of group buffer pool-dependent data pages that were attempted to be read from the group buffer pool because the pages were either invalid or not present in the local buffer pool. The query also returns the sum of group buffer pool-dependent data pages read into the local buffer pool from disk were not found in the group buffer pool.

  ```
  SELECT SUM(POOL_DATA_GBP_L_READS) AS POOL_DATA_GBP_L_READS,
  SUM(POOL_DATA_GBP_P_READS) AS POOL_DATA_GBP_P_READS FROM
  TABLE(MON_GET_BUFFERPOOL('', -2))
  ```

  Example 4-6 shows the output of this query.

  Example 4-6 Sample output of GBP logical and physical reads

  ```
  POOL_DATA_GBP_L_READS POOL_DATA_GBP_P_READS
  --------------------- ---------------------
                 213326                 19765

    1 record(s) selected.
  ```

Leading practice

When you are assessing buffer pool performance, the first place to start is to obtain the overall buffer pool hit ratio. This hit ratio is a great for tuning because it is based on the entire cluster and not individual components. If the ratio does not provide what you need, then you can start here to refine your search to locate and resolve the problem.

To get a basic understanding of the overall cluster hit ratio, you must know the logical reads and physical reads. When you have these two values, use the following formula to determine the overall buffer pool hit ratio:

((Logical Reads - Physical Reads) / Logical Reads) * 100

The second place to assess is the local buffer pool hit ratios. As each member in the DB2 pureScale cluster has its own local buffer pool, a low LBP hit ratio in general indicates that the member must defer to the group buffer pool to get a page more often than it must get the page in the LBP. In general, LBP ratios higher than 85% for data and 90% for indexes reflect great LBP performance. LBP hit ratios of 70 - 80% for indexes and 65 - 80% for data reflect good performance. If the ratios are below these values, it might indicate that the memory allocations for the LBP need to be increased. However, merely adding more memory for the LBP without adjusting the GBP can hinder GBP performance.

To determine the LBP hit ratio, you can use the following formula:

((LBP Pages Found - Async LBP Pages Found) / Logical Reads) * 100

The last step is to inspect the group buffer pool hit ratio. As the GBP plays a vital role in the coordination and distribution of pages, the current allocation of memory might not be sufficient, thus potentially affecting the cluster's performance. In general, GBP hit ratios of 90% for indexes and 80% for data reflect great GBP performance. GBP hit ratios of 65 - 80% for indexes and 60 - 75% for data reflect good GBP performance. However, because the GBP caches only modified or new pages, there are cases with high read activity in which the GBP does not have much to do.

To determine the GBP hit ratio, use the following formula:

((GBP Logical Reads - GBP Physical Reads) / GBP Logical Reads) * 100

The following extra checks can help show whether the GBP is under performing and whether adding additional memory to the GBP might help:

- Determine if there is a low GBP dependency, in which case tuning the GBP size might be less valuable than intended, if the following situation is true:

  pool_data_l_reads > 10 x pool_data_gbp_l_reads

- Determine if more than 25% of GBP reads are because of invalidated LBP pages. If so, adding more memory to the GBP can potentially help with performance. Use the following formula:

  pool_data_gbp_invalid_pages > 25% of pool_data_gbp_l_reads

Locking

In any DB2 environment, efficient lock management plays a critical role in both maintaining data integrity and high levels of concurrency. In a DB2 pureScale cluster, lock control across members is managed by the *global lock manager* (GLM), a component of the cluster CF. When a member requires a lock for an object, the *local lock manager* (LLM) within the member works with the GLM to attain the appropriate lock. Thus, the global lock manager mediates the distribution of locks and is aware of the locks that are currently being used within the cluster.

Therefore, in addition to examining locking for individual members, you need to view cross-member lock information to get an overall sense of how your DB2 pureScale cluster is operating.

Because lock management can affect performance, there is a wealth of lock monitoring elements that exist in DB2. But there are also some special monitoring elements to help with DB2 pureScale cluster-specific lock management.

The following query illustrates the monitoring elements that were introduced into the DB2 pureScale cluster and shows how you can effectively monitor your cluster against lock requests between members:

```
SELECT MEMBER,SUM(LOCK_ESCALS_GLOBAL) AS LOCK_ESCALS,
SUM(LOCK_TIMEOUTS_GLOBAL) AS LOCK_TIMEOUTS, SUM(LOCK_WAIT_TIME_GLOBAL)
AS LOCK_WAIT_TIME, SUM(LOCK_WAITS) AS LOCK_WAITS FROM TABLE
(MON_GET_CONNECTION('',-2)) GROUP BY MEMBER
```

Example 4-7 shows the output of this query.

Example 4-7 Sample output of lock monitoring elements

```
MEMBER LOCK_ESCALS  LOCK_TIMEOUTS  LOCK_WAIT_TIME  LOCK_WAITS
------ -----------  -------------  --------------  ----------
0      0            0              557             23
1      0            0              673             20

2 record(s) selected.
```

In addition to the previous example, there is a monitor report for lock waits called `monreport.lockwait` that can be used to assess each lock wait that is currently in progress. You can determine who the lock holder is, who the lock requestor is, and characteristics of the lock itself. To access this report, run the following command:

```
CALL monreport.lockwait()
```

Example 4-8 shows the output of this command:

Example 4-8 Sample output when calling monreport.lockwait()

```
------------------------------------------------------------------
  Monitoring report - current lock waits
------------------------------------------------------------------
  Database:               DTW
  Generated:              2012-04-24-23.55.03

  ==================================================================
  Part 1 - Summary of current lock waits

  ------------------------------------------------------------------

   REQ_APPLICATION  LOCK_MODE  HLD_APPLICATION  LOCK_  LOCK_OBJECT_TYPE
#  HANDLE           REQUESTED  _HANDLE          MODE
-- ---------------  ---------  ---------------- -----  ----------------
1  65651            X          -                -      ROW
```

```
2  65661            X          -                -      ROW
3  65674            X          -                -      ROW
4  65641            X          -                -      ROW
5  65660            X          -                -      ROW


==========================================================================
  Part 2 - Details for each current lock wait

  lock wait #:1

--------------------------------------------------------------------------
  -- Lock details --

  LOCK_NAME            = 040004000900B60C0000000052
  LOCK_WAIT_START_TIME = 2012-04-24-23.54.48.306784
  LOCK_OBJECT_TYPE     = ROW
  TABSCHEMA            = DB2LCO
  TABNAME              = STOCK
  ROWID                = 9
  LOCK_STATUS          = W
  LOCK_ATTRIBUTES      = 0000000000480000
  ESCALATION           = N

  -- Requestor and holder application details --

  Attributes           Requestor                      Holder
  ----------           ----------------------------   ------
  APPLICATION_HANDLE   65651                          -
  APPLICATION_ID       172.16.42.101.47944.120425035  -
  APPLICATION_NAME     WD                             -
  SESSION_AUTHID       DB2LCO                         -
  MEMBER               1                              0
  LOCK_MODE            -                              -
  LOCK_MODE_REQUESTED  X                              -

  -- Lock holder current agents --

  -- Lock holder current activities --

  -- Lock requestor waiting agent and activity --

  AGENT_TID            = 118
  REQUEST_TYPE         =
  ACTIVITY_ID          =
```

```
  UOW_ID               =
  LOCAL_START_TIME     =
  ACTIVITY_TYPE        =
  ACTIVITY_STATE       =

  STMT_TEXT            =
  update stock  set s_order_cnt=s_order_cnt+2 WHERE s_i_id = ? AND
s_w_id = ?
```

Page reclaiming

Page reclaiming is an important concept to understand when you are working with the DB2 pureScale Feature. It refers to the process where one member requests and is granted a page of data that is being used by another member before that member is finished with the page. When different members require access to the same page, the cluster CF manages who gets access to the page and when.

Important: This example assumes that two members intend to update different rows on the same page. Updating the same row of data in the same table is prevented, because only one DB2 agent can update any one row at a time.

To demonstrate how page reclaiming works, here is an example of the general process when a member wants a page that is being used by another member:

1. Member 1 is updating row R1 within a page of data. It has exclusive access to the page that contains the row of data it wants to update.
2. Member 2 requires exclusive access to the same page to update row R2. So it then makes the request to the CF and waits while the request is processed.
3. The CF knows that Member 1 already has exclusive access to the page and so issues a request to reclaim the page from Member 1. Meanwhile, Member 2 is still waiting.
4. Member 1 processes the reclaim request by writing the page back to the GBP and then releasing the page. Member 1 still retains any row or table locks it might acquire, and it finishes its update to the row that it intended to update.
5. The CF grants Member 2 access to the page. Member 2 reads the page from the GBP and then performs the updates to row R2.

It is important to understand that any lock that Member 1 receives is retained until the unit of work is completed, even if another member reclaims the page and begins to use it before Member 1 finishes its unit of work. As such, different members can work with the same page without compromising data integrity. Any attempts by different members to update the same row still serializes on the row lock for that row.

Page reclaiming monitoring

Although the concept of page reclaiming helps eliminate lock waits, an excessive amount of page reclaims can have a negative impact. Having too many page reclaims creates much page contention, resulting in low processor usage and the need to expend extra cycles to deal with these occurrences.

For this reason, a DBA can use many different monitoring elements to help gain optimal performance. You can run the following monitoring queries examples against a DB2 pureScale instance to gauge whether there are performance concerns:

- The following query captures the total page reclaims and total wait time across all members:

  ```
  SELECT
  SUM(PAGE_RECLAIMS_X+PAGE_RECLAIMS_S+SPACEMAPPAGE_PAGE_RECLAIMS_X+SPA
  CEMAPPAGE_PAGE_RECLAIMS_S)AS PAGE_RECLAIMS, SUM(RECLAIM_WAIT_TIME)
  AS RECLAIM_WAIT_TIME FROM TABLE(MON_GET_PAGE_ACCESS_INFO('','', -2))
  ```

  Example 4-9 shows the output of this query.

  Example 4-9 Sample output of total page reclaims and total wait time in the cluster

  ```
  PAGE_RECLAIMS        RECLAIM_WAIT_TIME
  -------------------- --------------------
                    52                  934

    1 record(s) selected.
  ```

- The following query captures the overall shared and exclusive page reclaim requests:

  ```
  SELECT SUBSTR(TABNAME,1,8) AS NAME, SUBSTR(OBJTYPE,1,5) AS TYPE,
  PAGE_RECLAIMS_X AS PGRCX, PAGE_RECLAIMS_S AS PGRCS,
  SPACEMAPPAGE_PAGE_RECLAIMS_X AS SMPPGRCX,
  SPACEMAPPAGE_PAGE_RECLAIMS_S AS SMPPGRCS FROM TABLE(
  MON_GET_PAGE_ACCESS_INFO(NULL, NULL, NULL) ) AS WAITMETRICS ORDER BY
  NAME
  ```

Example 4-10 shows the output of this query.

Example 4-10 Sample output of overall shared and exclusive page reclaims

```
NAME     TYPE     PGRCX    PGRCS    SMPPGRCX   SMPPGRCS
-------- -------  -------- -------- --------- -----------
STOCK    TABLE    39       0        0         0

  1 record(s) selected.
```

The columns in the sample output are defined as follows:

PGRCX Represents the number of times a page related to an object was reclaimed by another member that required exclusive access to the page

PGRCS Represents the number of times a page related to an object was reclaimed by another member that required shared access to the page

SMPPGRCX Represents the number of times a space map page was reclaimed by another member that required exclusive access to the page

SMPPGRCS Represents the number of times a space map page was reclaimed by another member that required shared access to the page

Leading practice

An excessive amount of page reclaims can cause much contention and directly impact the performance of cluster. Thus, when it comes to assessing and evaluating the amount of page reclaims, the rule of thumb is to see if you have more than one reclaim per 10 transactions. This guideline might or might not apply to your situation, because every environment has different factors.

If you determine that there are too many page reclaims and want to reduce them, consider the following suggestions:

- Smaller page sizes might eliminate the number of false sharing conflicts and reduce reclaims on tables and indexes.
- Small tables with frequent updates might benefit from increased PCTFREE. This setup distributes rows over more pages, but also increases overall space consumption.
- As reclaims can affect metadata pages, increased extent size can reduce the number of space map page reclaims.

CF monitoring

Actively monitoring your CF can help you recognize current performance-related issues and also gauge capacity needs for the future. In this section, we initially describe the important memory allocations that you need to understand to be able to fine-tune your CF in accordance with your cluster demands. Then we illustrate some important hardware-related monitoring elements or items that are critical for the overall performance of your cluster.

Viewing configured GBP memory allocations

As the group buffer pool (GBP) plays a vital role in the coordination of lock and page distribution, it is always a preferred practice to periodically view the memory allocations for your GBP and make adjustments. To use a specific table function to view the allocation of memory for the GBP, run the following query:

```
SELECT SUBSTR(HOST_NAME,1,8) AS HOST,SUBSTR(DB_NAME, 1,8) AS DBNAME,
CURRENT_CF_GBP_SIZE,CONFIGURED_CF_GBP_SIZE, TARGET_CF_GBP_SIZE FROM
TABLE( MON_GET_CF( cast(NULL as integer) ) ) AS CAMETRICS ORDER BY HOST
```

Example 4-11 shows the output of this query.

Example 4-11 Sample output of GBP memory allocations

```
HOST     DBNAME   CURRENT_CF_GBP_SIZE CONFIGURED_CF_GBP_SIZE TARGET_CF_GBP_SIZE
-------- -------- ------------------- --------------------- -----------------
node102  DTW      20642               142080                142080
node103  DTW      15026               142080                142080

  2 record(s) selected.
```

The output provides three important memory size allocations:

CURRENT_CF_GBP_SIZE The group buffer pool memory currently in use

CONFIGURED_CF_GBP_SIZE The group buffer pool memory currently allocated and reserved

TARGET_CF_GBP_SIZE The dynamic resize target

Viewing configured lock size memory allocations

When you notice performance problems related to a locking issue, adjusting the lock size memory allocation directly within the CF can sometimes rectify things. To use a table function to view specific information regarding the lock size memory allocation, run the following query:

```
SELECT SUBSTR(HOST_NAME,1,8) AS HOST,SUBSTR(DB_NAME, 1,8) AS DBNAME,
CURRENT_CF_LOCK_SIZE,CONFIGURED_CF_LOCK_SIZE, TARGET_CF_LOCK_SIZE FROM
TABLE( MON_GET_CF( cast(NULL as integer) ) ) AS CAMETRICS ORDER BY HOST
```

Example 4-12 shows the output of this query.

Example 4-12 Sample output of lock size memory allocations

```
HOST    DBNAME CURRENT_CF_LOCK_SIZE CONFIGURED_CF_LOCK_SIZE TARGET_CF_LOCK_SIZE
----    ------ -------------------- ----------------------- -------------------
node102 DTW    328                  29696                   29696
node103 DTW    24                   29696                   29696

  2 record(s) selected.
```

The output provides the following important memory size allocations:

CURRENT_CF_LOCK_SIZE The lock size that is currently in use

CONFIGURED_CF_LOCK_SIZE The currently allocated and reserved memory allocation for the lock size on the CFs

TARGET_CF_LOCK_SIZE The dynamic resize target for the lock size

Viewing the configured SCA memory allocations

To view the memory allocations for the share communications area (SCA) on a CF, run the following query:

```
SELECT SUBSTR(HOST_NAME,1,8) AS HOST,SUBSTR(DB_NAME, 1,8) AS DBNAME,
CURRENT_CF_SCA_SIZE,CONFIGURED_CF_SCA_SIZE, TARGET_CF_SCA_SIZE FROM
TABLE( MON_GET_CF( cast(NULL as integer) ) ) AS CAMETRICS ORDER BY HOST
```

Example 4-13 shows the output of this query.

Example 4-13 Sample output of SCA memory allocations

```
HOST     DBNAME   CURRENT_CF_SCA_SIZE CONFIGURED_CF_SCA_SIZE TARGET_CF_SCA_SIZE
-------- -------- ------------------- ---------------------- ------------------
node102  DTW      30                  23296                  23296
node103  DTW      30                  23296                  23296

  2 record(s) selected.
```

The output provides the following import memory size allocations:

CURRENT_CF_SCA_SIZE The total memory in use for the shared communications area (SCA)

CONFIGURED_CF_SCA_SIZE The currently allocated and reserved memory allocation for the SCA on the CFs

TARGET_CF_SCA_SIZE The dynamic resize target for the shared communications area

CF performance monitoring

The cluster CF plays a vital role in a DB2 pureScale cluster, so sometimes you want to see CF server information for assessing possible hardware and memory improvements.

You can use the following administrative view to see the server memory and the processor load, which can be a vital piece of information. If you notice that your CF is constantly running at the maximum processor level, your cluster might benefit by moving to a more powerful server. However, do not confuse the processor used by the CF and the processor that you see being used in a tool, such as **top** or **topas**. The OS resource-monitoring tool cannot determine how much of the CF polling is actual work versus how much of it is polling for work. You must use the DB2 method to gauge the actual amount of work the CF is performing.

To see the CF server information, run the following SQL statement:

```
SELECT SUBSTR(NAME,1,20) AS NAME, SUBSTR(VALUE,1,10) AS VALUE,
SUBSTR(UNIT,1,20) AS UNIT, ID FROM SYSIBMADM.ENV_CF_SYS_RESOURCES
```

Example 4-14 shows the output of this query.

Example 4-14 Sample output of the ENV_CF_SYS_RESOURCES administrative view

```
NAME                 VALUE        UNIT      ID
-----------------    ----------   --------  ------
HOST_NAME            node102.pu   -         128
MEMORY_TOTAL         3943         MB        128
MEMORY_FREE          136          MB        128
MEMORY_SWAP_TOTAL    8189         MB        128
MEMORY_SWAP_FREE     8135         MB        128
VIRTUAL_MEM_TOTAL    12133        MB        128
VIRTUAL_MEM_FREE     8272         MB        128
CPU_USAGE_TOTAL      98           PERCENT   128
HOST_NAME            node103.pu   -         129
MEMORY_TOTAL         3943         MB        129
MEMORY_FREE          198          MB        129
MEMORY_SWAP_TOTAL    8189         MB        129

MEMORY_SWAP_FREE     8189         MB        129
VIRTUAL_MEM_TOTAL    12133        MB        129
VIRTUAL_MEM_FREE     8387         MB        129
```

```
CPU_USAGE_TOTAL       99            PERCENT   129

  16 record(s) selected
```

Detecting interconnect bottlenecks

Although InfiniBand provides excellent throughput, which provides more desirable response times and performance, there are cases where a potential interconnect bottleneck can occur. When these situations occur, you have limited cluster throughput with ample available processors on members and CFs, higher CF response times, and an increase in member processor wait time. These symptoms are not noticeable unless you are investigating them directly. However, you indirectly notice an unusual increase in response times when interacting with the cluster.

Fortunately, you can use DB2 to monitor CF response times using metrics that were introduced specifically for the DB2 pureScale Feature. The two metrics that are important to note are *CF_WAITS*, which represents the approximate number of CF calls, and *CF_WAIT_TIME*, which represents the time accumulated when communicating with the CF. To obtain an estimate of CF wait time per CF call, do the following simple calculation:

CF_WAIT_TIME / CF_WAITS

Unfortunately, there is no set number that can determine if your cluster CF wait time requires attention, as many different factors can be involved. However, one good way to judge whether the number is acceptable is to gather a history to assess if there has been a drastic change in your system's normal operation. A rough guideline for good response time for a busy system is less than 200 ms. If your response times are higher than the 200 ms, adding another CF HCA can potentially help lower the amount of CF wait time within the cluster.

4.3.2 Functional monitoring

Functional monitoring can be thought of as troubleshooting your DB2 pureScale instance, whether the problem is a current one or something that occurred in the past. When you notice an issue with your cluster, it is best to take an approach that is well-defined to isolate the problem. This action can save you a considerable amount of time and assist you in bringing your cluster back to a healthy state as soon as possible.

There are two categories that a problem can fall under. Knowing these categories can help you cut down troubleshooting time.

Current problems

When problems are currently affecting your cluster, users often complain about performance slowdowns or sporadic errors. These cases tend to be issues with a member or cluster CF, so you can investigate them directly. However, if your entire cluster is unavailable, then there are different avenues that you must investigate while keeping the topology of your DB2 pureScale cluster in mind. The topology of the DB2 pureScale cluster can affect its reliability and availability.

For example, if you have only a single cluster CF, its failure results in errors to the application or the application can stall while waiting for DB2 to recover the entire instance. A loss of all cluster CFs results in all members being restarted in a procedure called a *group restart*, which is necessary to resynchronize the information, both at the members and the CF, to a known state.

If you have more than one cluster CF but only one member, the loss of the member also results in application errors or stalls while waiting for DB2 to restart the member.

These examples produce information that can be found in the `DIAGPATH` directory and indicate that the cluster CF restarted (in the first case) or that the member restarted (in the second case).

The `DIAGPATH`, or the `sqllib/db2dump` path as it is commonly called, is where diagnostic data for DB2 is placed when something goes wrong. The DB2 pureScale Feature does not change this location from previous versions of DB2. However, the DB2 pureScale Feature added an `ALT_DIAGPATH` path as a secondary path to write diagnostic data should the initial `DIAGPATH` be unavailable.

In addition to the `DIAGPATH`, and as with previous versions of DB2, the operating system logs can also hold information about the events that surround failures in DB2. However, with the DB2 pureScale Feature, there are some new directories that should be considered as well.

As described in Chapter 1, “An overview of the DB2 pureScale Feature” on page 1, the DB2 pureScale Feature has multiple components, and each component has its own diagnostic logs that can be created. The cluster CF shares diagnostic directories with DB2 itself, but this configuration can be changed with the **`CF_DIAGPATH`** parameter in the database manager configuration. Other components, such as IBM Tivoli System Automation and Reliable Scalable Cluster Technology (RSCT), have their logs in the `/var/ct` directory, while GPFS puts its diagnostic tests in a directory called `/var/adm/ras`. Depending on the problem, you might need to review all of these diagnostic locations.

The response is dependent on the problem itself. Many times, to get the correct diagnostic tests, you must understand and define the problem itself. Understanding and defining the problem might require talking with users to understand what they see, looking at the DB2 pureScale Feature diagnostic tools, or monitoring the output to see if the problem description matches the end data.

Past problems

If you notice in the log files that a problem occurred in the past, it can still be worthwhile to investigate it, so the same issue can be avoided in the future. Finding answers to problems that occurred in the past is important in a high-availability, fault-resistant environment because it can sometimes unmask underlying issues.

If a problem occurs that can be handled by DB2, DB2 rectifies the issue. Whether you lost a member or primary CF, DB2 restarts the member or cluster CF as necessary. Even in cases where your components are repeatedly failing, DB2 tries to handle the issue without your intervention. You might not even know that the problem occurred until you look at the logs, or someone mentions the problem to you. This situation is why taking an active approach to monitoring your cluster can be so beneficial.

Whether the problems are current or in the past, the diagnostic directories and DB2 pureScale diagnostic tooling and monitoring output helps you determine the causes and solutions.

Problem determination

Problem determination is not so much a science as it is an art. The method you take to determine and resolve a problem depends on the issues encountered and the amount and quality of the input provided by the user who is experiencing the problem.

In a clustered environment, some of the rules of problem diagnosis are different. You are no longer looking at a single machine that could be the culprit, but rather a cluster of machines with multiple methods and paths of interaction between them.

The following sections provide general suggestions for where to start looking for problems that are reported by users. In all cases, common sense should prevail and the logic you use should match the situation encountered.

System down scenarios

System down scenarios are harmful for business and can be difficult to diagnose. The problem could be in the database system, or it could be in some other component above the database layer.

Consider connecting to the database from another client to see if it is available; if it is, the problem is likely not with the database itself. If you cannot connect through DB2, consider pinging the hosts in the DB2 pureScale cluster. Check to see if there is a network outage or power outage that affected the entire DB2 pureScale cluster. The possibilities are numerous and beyond the scope of this book to cover completely, but if you can follow some common-sense strategies for ruling out issues, you can find the issue.

Performance problems

Performance problems can be classified into multiple scenarios. Some performance problems are because of a lack of resources, while other problems are because of issues in the DB2 pureScale cluster.

You need to discover if the performance problem is constant, intermittent, or sporadic. If it is sporadic, it is more likely, though not certain, that there is a problem in the DB2 pureScale cluster. Intermittent or constant problems point usually to resource issues, and some of the monitoring steps mentioned previously might be helpful.

If you think about how the DB2 pureScale Feature works, there are issues that can lead to performance problems, such as members that are not failing back to their home host. The next few sections provide information about how to identify DB2 issues in a DB2 pureScale instance. Then we look at some other operating system areas that could be useful in problem diagnosis.

Interpretation of DB2 status information

When you query hosts, members, or cluster caching facilities for their status, you are presented with state and alert information to assist you in determining if there is an issue. You want to examine all of this information to understand what is happening in the system.

Host statuses

Table 4-1 lists situations that you might discover when examining a host.

Table 4-1 Host state and status interpretation

State	Instance stopped	Alert	Description
ACTIVE	NO	NO	The host is active and operating normally.
		YES	The host is active (it responds to system commands), but there might be a problem that prevents it from participating in the DB2 pureScale instance. For example, there might be a file system problem or a network communication issue, or the idle processes that the DB2 pureScale Feature requires for performing failovers might not be running.
	YES	NO	The host is active. The instance stopped explicitly on this host by the administrator running the **`db2stop instance on hostname`** command
		YES	The host is active, but an alert exists for the host that has not been cleared. The administrator explicitly stopped the instance.
INACTIVE	NO	NO	Not applicable. A host cannot be `INACTIVE` when both `INSTANCE_STOPPED` and `ALERT` are set to `NO`.
		YES	The host is not responding to system commands. The instance was not stopped explicitly by the administrator, but there is an alert. This combination of status information indicates the abnormal shutdown of a host. Such a shutdown might arise, for example, from a power failure on a host.
	YES	NO	This state is the normal state when the instance is stopped by the administrator. Such a combination of status information might arise when the host is being taken offline for the installation of software updates.
		YES	The host is not responding to system commands. An alert exists for the host that has not been cleared, but the instance was stopped explicitly by the administrator (the system did not shut down abnormally).

Member statuses

Table 4-2 lists situations that you might discover when examining a member.

Table 4-2 Member state and status interpretation

State	Alert	Description
STARTED	NO	The member is started in the instance and is operating normally.
	YES	The member is started in the instance. However, at some point, there was an unsuccessful attempt to fail over to another host. Because of that unsuccessful attempt to fail over, the member was able to fail over successfully to another host or it failed back to its home host. If the member is running on its home host, it is running normally; if it is running on a guest host, it is running in light mode. Investigate the alert to determine what happened.
STOPPED	NO	The member was stopped by the administrator running the **`db2stop`** command.
	YES	The member was stopped by the administrator running the **`db2stop`** command; however, the alert field has not yet been cleared.
RESTARTING	NO	The member is starting.
	YES	The member is starting. However, at some point, there was an unsuccessful attempt to start the member on the home host or to fail over to another host. The alert field has not yet been cleared.
WAITING_FOR_FAILBACK	NO	The member is running in light mode on a guest host, and is waiting to fail back to the home host. You might want to examine the status of the home host to see if anything is preventing the member from failing back to the home host (for example, a failed network adapter).
	YES	An attempt to restart the member on the home host might have failed, automatic failback is disabled, or crash recovery might have failed. You need to resolve the problem and clear the alert manually before the member can automatically fail back to its home host. If automatic failback is disabled, you can manually clear the alert and enable automatic failback by running the **`db2cluster`** command.
ERROR	YES	The DB2 cluster services were not able to start the member on any host. You need to resolve the problem and clear the alert manually before attempting to restart the instance.

Cluster caching facility statuses

Table 4-3 lists situations that you might discover when examining a cluster CF.

Table 4-3 CF state and status interpretation

State	Alert	Description
STOPPED	NO	The cluster CF was manually stopped by the administrator running the **db2stop** command.
	YES	There was an unsuccessful attempt by the CF to become the primary CF. The cluster CF was manually stopped in the instance by the administrator running the **db2stop** command.
RESTARTING	NO	The CF is restarting, either as a result of running the **db2start** command or after a primary CF failure.
	YES	The CF is restarting, but there is a pending alert from a previous failed attempt by the CF to take on the primary role. The alert must be cleared manually.
BECOMING_PRIMARY	NO	The CF takes on the role of primary CF if there is no other primary CF already running in the instance.
	YES	Not applicable. The CF cannot attempt to take on the primary role with an alert condition set.
PRIMARY	NO	The CF took on the role of primary CF and is operating normally.
	YES	Not applicable. The CF cannot be acting as the primary CF with an alert condition set.
CATCHUP(n%)	NO	This non-primary CF is in the process of copying information from the primary CF that it needs to operate in PEER mode. When you view the status of the non-primary CF by running the **db2instance -list** command, that non-primary CF is in the CATCHUP state until a connection is made to the database. After the first connection is made, the process of copying data from the primary CF begins.
	YES	This non-primary CF is in the process of copying information from the primary CF that it needs to operate in PEER mode. There is a pending alert from a previous failed attempt by this CF to take on the primary role that must be cleared manually.

State	Alert	Description
PEER	NO	This non-primary CF is ready to assume the role of primary CF if the current primary CF fails.
	YES	This non-primary CF is ready to assume the role of primary CF if the current primary CF fails. There is a pending alert from a previous failed attempt by this CF to take on the primary role that must be cleared manually.
ERROR	YES	The CF could not be started on any host in the instance. You need to resolve the problem and clear the alert manually before attempting to restart the instance.

Viewing and clearing alerts

When alerts are raised against any of your cluster components, it indicates that there is an issue that might need further investigation. Alerts can be about minor or major issues. Thus, investigate each alert that is raised.

To view the component status, run the **db2instance -list** command, as shown here, to see an overview of your cluster, paying special attention to component status and state information:

```
db2cluster -cm -list -alert
```

Example 4-15 shows the output from this command.

Example 4-15 Sample output of the db2cluster -cm -list -alert command

```
Alert: Host 'node103' is INACTIVE. Ensure the host is powered on and
connected to the network.

Action: This alert will clear itself when the host is ACTIVE.
Impact: While the host is INACTIVE, the DB2 members on this host will
be in restart light mode on other hosts and will be in the
WAITING_FOR_FAILBACK state. Any CF defined on the host will not be able
to start, and the host will not be available as a target for restart
light.
```

Each alert is presented in three parts:

- The first part describes the alert.
- The second part describes the action required to clear the alert.
- The third part describes the impact of the alert if it not resolved.

This information provides the user with a good understanding of the problem and the consequences of not resolving it.

Important: When you request alert information, the most important and severe alerts are presented at the top. As you move through the list, the severity of the alerts decreases.

In most cases when an alert is raised, DB2 clears the alert after it is resolved. But there are some situations when an alert must be cleared manually to ensure that the administrator knows that the problem needs intervention. The alert remains until the problem is resolved by an administrator and the alert is cleared manually.

To clear an alert manually, run the following command:

```
db2cluster -cm -clear -alert [-member member-id | cf cf-id | -host
host-name]
```

For example, after you resolve an issue with a member in the DB2 pureScale instance with the ID `1`, clear the alert by running the following command:

```
db2cluster -cm -clear -alert -member 1
```

Examining the log files

Like any DB2 environment, diagnostic log files are created to assist database administrators in problem determination. Depending on the type of error that is encountered, additional files are created with additional technical details about the issue. This section helps you understand these log files and provide examples of using log files in certain scenarios to more easily spot DB2 pureScale Feature-specific problems.

The db2diag.log and notify log files

The DB2 diagnostic log files, commonly referred to as the `db2diag` and `notify` log files, are the primary locations where problem information is captured and stored. The `notify` log file should be the first log file that you inspect when confronting an issue; `db2diag.log` is the second.

The `notify` log file contains notification messages that pertain to the DB2 instance intended for the database administrator. In this file, you can find the start and stop times of the db2 members, any restart light messages, and database recovery messages.

Both log files allow for several levels of logging functionality. The more diagnostic descriptions required (for example, log level 4), the more entries that are written (potentially slowing down the system). The default is level 3, which shows errors and severe messages but leaves out the informational message.

Each entry in `db2diag.log` or the `notify` log file has a header that provides information about the process ID, the member, and the component that wrote the message. This information can help you determine which part of DB2 the error or warning occurred in.

First occurrence data capture

First occurrence data capture (FODC) collects information about an instance, host, or member when specific problems occur. It can be started manually when you observe a problem or automatically whenever a specific scenario or symptom is detected. The information that is collected can assist a database administrator in determining potential causes of the problem. Included in this directory are things such as stack traces and event log files. Sometimes operating system logs are copied here, or traces that are relevant to the problem are initiated and deposited here.

Using logs for problem determination

The next sections describe events that can occur in the DB2 pureScale cluster and shows how to diagnose the events. These scenarios can help familiarize you with the logs, but they are not intended to be the only diagnostic tools that you use.

Log scenario 1: Local restart of member

A local restart of a member occurs when there is a software failure that can be contained and handled automatically by DB2. DB2 restarts the member on its home host. After the restart, DB2 reactivates the database for connections.

For this scenario, a software failure is replicated by stopping the appropriate db2sysc process on a host that has a member. After you stop the process, DB2 tries to contain the failure automatically and restarts the member on its home host. Run the **`db2pd -edus`** command to attain the appropriate db2sysc process.

Example 4-16 shows the output of the **`db2pd -edus`** command. The output presents the relevant DB2 processes on the given host. To replicate the software failure, stop the 15529 process by running a **`kill -9 15529`** command.

Example 4-16 Sample output of the db2pd -edus command

```
Database Member 1 -- Active -- Up 0 days 00:02:47 -- Date 04/26/2012 23:51:01

List of all EDUs for database member 1

db2sysc PID: 15529
db2wdog PID: 15765
db2acd  PID: 15899
```

After you stop the ID 15529 process, the DB2 member restart is initiated. To observe this process, run the **db2instance -list** command (Example 4-17).

Example 4-17 Sample output of the db2instance -list command when a member is stopped

```
ID   TYPE     STATE      HOME_HOST   CURRENT_HOST   ALERT   PARTITION_NUMBER  LOGICAL_PORT  NETNAME
--   ------   -------    ---------   ------------   ----    ----------------  ------------  -------
0    MEMBER   STARTED    node102     node102        NO      0                 0             node102
1    MEMBER   RESTARTING node103     node103        NO      0                 0             node103
128  CF       PRIMARY    node102     node102        NO      -                 0             node102
129  CF       PEER       node103     node103        NO      -                 0             node103

HOSTNAME                  STATE                INSTANCE_STOPPED        ALERT
--------                  -----                ----------------        -----
node103                   ACTIVE               NO                      NO
node102                   ACTIVE               NO                      NO
```

You can obtain more information by viewing the resource state associated with the failed member by running the **lssam** command. As you can see in Example 4-18, the failed member is in the `Pending Online` state, which means that DB2 is in the process of restarting the member.

Example 4-18 Sample output of the lssam command to see the resource state of a failed member

```
Pending online IBM.ResourceGroup:db2_db2lco_1-rg Nominal=Online
     '- Pending online IBM.Application:db2_db2lco_1-rs
          |- Offline IBM.Application:db2_db2lco_1-rs:node102
          '- Pending online IBM.Application:db2_db2lco_1-rs:node103
```

DB2 restarts of this nature can be quick, and you might not notice the changes in the state if you are not quick enough to capture them. You know that the member has restarted when you recheck the PID of the process and it is not the same as it was at the beginning of this exercise.

Example 4-19 shows an additional short snippet from the `db2diag` logs when the member fails and then restarts back on its home host. DB2 recognizes the failure and starts the cleanup process of the failed member.

Example 4-19 Snippet from db2diag showing events corresponding to the failed member

```
2012-03-29-02.51.07.605563-240  I26443E510   LEVEL: Event
PID: 22258 TID: 47128015876064  PROC: db2rocm 1 [db2lco]
INSTANCE: db2lco                NODE: 001
HOSTNAME: node103
FUNCTION: DB2 UDB, high avail services, db2rocm_main, probe:1266
DATA #1: SQLHA Event Recorder header data (struct sqlhaErPdInfo),
PD_TYPE_SQLHA_ER_PDINFO, 80 bytes
```

```
Original timestamp: 2012-03-29-02.51.06.71889000
DATA #2: String, 30 bytes
db2rocm 1 DB2 db2lco 1 CLEANUP
DATA #3: String, 5 bytes
BEGIN

2012-03-29-02.51.07.607349-240   I30428E513   LEVEL: Event
PID: 22258 TID: 47128015876064   PROC: db2rocm 1 [db2lco]
INSTANCE: db2lco                 NODE: 001
HOSTNAME: node103
FUNCTION: DB2 UDB, high avail services, db2rocm_main, probe:2178
DATA #1: SQLHA Event Recorder header data (struct sqlhaErPdInfo),
PD_TYPE_SQLHA_ER_PDINFO, 80 bytes
Original timestamp: 2012-03-29-02.51.07.605434000
DATA #2: String, 30 bytes
db2rocm 1 DB2 db2lco 1 CLEANUP
DATA #3: String, 7 bytes
SUCCESS
```

After the cleanup process is complete, the member is restarted, the database is reactivated, and it is starts accepting connections from clients after it is fully recovered. Example 4-20 illustrates the return to a healthy state when a **db2instance -list** command is run, which shows that the member is started.

Example 4-20 Sample output of the db2instance -list command showing member integration

```
ID   TYPE     STATE    HOME_HOST   CURRENT_HOST   ALERT   PARTITION_NUMBER  LOGICAL_PORT  NETNAME
--   ------   -------  ---------   ------------   ----    ----------------  ------------  -------
0    MEMBER   STARTED  node102     node102        NO      0                 0             node102
1    MEMBER   STARTED  node103     node103        NO      0                 0             node103
128  CF       PRIMARY  node102     node102        NO      -                 0             node102
129  CF       PEER     node103     node103        NO      -                 0             node103

HOSTNAME                  STATE                INSTANCE_STOPPED        ALERT
--------                  -----                ----------------        -----
node103                   ACTIVE               NO                      NO
node102                   ACTIVE               NO                      NO
```

Also, you notice that the resource state that is associated with the member within the resource group returns to a healthy online state when a **lssam** command is run (Example 4-21).

Example 4-21 Sample output of the lssam command showing the healthy resource state

```
Online IBM.ResourceGroup:db2_db2lco_1-rg Nominal=Online
        '- Online IBM.Application:db2_db2lco_1-rs
```

```
                    |- Offline IBM.Application:db2_db2lco_1-rs:node102
                    '- Online IBM.Application:db2_db2lco_1-rs:node103
```

Log scenario 2: CF takeover

This scenario demonstrates a situation where the host that has the primary CF fails momentarily but comes back online. You notice that the DB2 pureScale Feature has mechanisms to recover from this failure by failing over the primary CF responsibilities to the second CF.

Example 4-22 shows that after a **`db2instance -list`** command is run, the secondary CF's state changes from `PEER` to `BECOMING_PRIMARY`, indicating that it is taking the appropriate actions to assume the role as the primary CF in the cluster. Furthermore, as this takeover process is being conducted, the failed primary CF is also reintegrating itself back into the cluster and assumed the role of `RESTARTING`.

Example 4-22 Sample output of the db2instance -list command

```
ID   TYPE    STATE           HOME_HOST CURRENT_HOST  ALERT PARTITION_NUMBER LOGICAL_PORT  NETNAME
--   ------  -------         --------- ------------  ----  ---------------- ------------  -------
0    MEMBER  STARTED          node102  node102       NO    0                0             node102
1    MEMBER  STARTED          node103  node103       NO    0                0             node103
128  CF      RESTARTING       node102  node102       NO    -                0             node102
129  CF      BECOMING_PRIMARY node103  node103       NO    -                0             node103

HOSTNAME                  STATE                INSTANCE_STOPPED        ALERT
--------                  -----                ----------------        -----
node103                   ACTIVE               NO                      NO
node102                   ACTIVE               NO                      NO
```

If you explore the snippet from the `db2diag` logs shown in Example 4-23, you might gain a better understanding of the cleanup process by the failed CF and the takeover process by the secondary CF. You can see the failed CF is in the process of restarting, but is currently going through the clean-up process.

Example 4-23 Snippet from db2diag showing the cleanup process for the failed CF

```
2012-03-30-04.52.40.509678-240 I290053E540           LEVEL: Event
PID     : 7620                 TID : 47240765626688  PROC : db2rocme
128 [db2lco]
INSTANCE: db2lco               NODE : 128
HOSTNAME: node102
FUNCTION: DB2 UDB, high avail services, db2rocm_main, probe:1266
DATA #1 : SQLHA Event Recorder header data (struct sqlhaErPdInfo),
PD_TYPE_SQLHA_ER_PDINFO, 80 bytes
Original timestamp: 2012-03-30-04.52.31.130952000
DATA #2 : String, 56 bytes
```

```
/home/db2lco/sqllib/adm/db2rocme 1 CF db2lco 128 CLEANUP
DATA #3 : String, 5 bytes
BEGIN

2012-03-30-04.52.42.831262-240 I291238E542          LEVEL: Event
PID     : 7620                 TID : 47240765626688  PROC : db2rocme
128 [db2lco]
INSTANCE: db2lco               NODE : 128
HOSTNAME: node102
FUNCTION: DB2 UDB, high avail services, db2rocm_main, probe:2178
DATA #1 : SQLHA Event Recorder header data (struct sqlhaErPdInfo),
PD_TYPE_SQLHA_ER_PDINFO, 80 bytes
Original timestamp: 2012-03-30-04.52.40.509553000
DATA #2 : String, 56 bytes
/home/db2lco/sqllib/adm/db2rocme 1 CF db2lco 128 CLEANUP
DATA #3 : String, 7 bytes
SUCCESS
```

Example 4-24 shows that the primary role is initiated on the secondary CF.

Example 4-24 Snippet from db2diag showing the primary role initiated on the secondary CF

```
2012-03-30-04.52.34.372318-240 I275608E392          LEVEL: Event
PID     : 5645                 TID : 47182079875392  PROC : db2rocme
900 [db2lco]
INSTANCE: db2lco               NODE : 900
HOSTNAME: node103
FUNCTION: DB2 UDB, high avail services, db2rocm_main, probe:1266
DATA #1 : String, 59 bytes
/home/db2lco/sqllib/adm/db2rocme 1 PRIMARY db2lco 900 START
DATA #2 : String, 5 bytes
BEGIN

2012-03-30-04.52.34.458831-240 E278405E384          LEVEL: Event
PID     : 5645                 TID : 47182079875392  PROC : db2rocme
900 [db2lco]
INSTANCE: db2lco               NODE : 900
HOSTNAME: node103
FUNCTION: DB2 UDB, high avail services, rocmPrimaryCAStart, probe:911
MESSAGE : ADM7531I  A CF changed state. CF: "129". New state:
          "BECOMING_PRIMARY".
CHANGE  : CF

2012-03-30-04.52.42.614704-240 I291781E394          LEVEL: Event
```

```
PID     : 5645                  TID : 47182079875392  PROC : db2rocme
900 [db2lco]
INSTANCE: db2lco                NODE : 900
HOSTNAME: node103
FUNCTION: DB2 UDB, high avail services, db2rocm_main, probe:2178
DATA #1 : String, 59 bytes
/home/db2lco/sqllib/adm/db2rocme 1 PRIMARY db2lco 900 START
DATA #2 : String, 7 bytes
SUCCESS
```

Finally, you can observe that, as shown in Example 4-25, the failed CF restarts and assumes the role of secondary CF in the cluster.

Example 4-25 Snippet from dbdiag showing the failed CF reintegration process

```
2012-03-30-04.52.43.356364-240 I292176E387           LEVEL: Event
PID     : 8020                  TID : 47284219590976  PROC : db2rocme
128 [db2lco]
INSTANCE: db2lco                NODE : 128
HOSTNAME: node102
FUNCTION: DB2 UDB, high avail services, db2rocm_main, probe:1266
DATA #1 : String, 54 bytes
/home/db2lco/sqllib/adm/db2rocme 1 CF db2lco 128 START
DATA #2 : String, 5 bytes
BEGIN

2012-03-30-04.52.45.455218-240 E292564E361           LEVEL: Event
PID     : 8020                  TID : 47284219590976  PROC : db2rocme
128 [db2lco]
INSTANCE: db2lco                NODE : 128
HOSTNAME: node102
FUNCTION: DB2 UDB, high avail services, db2rocm_main, probe:911
MESSAGE : ADM7531I  A CF changed state. CF: "128". New state:
"RESTARTING".
CHANGE  : CF

2012-03-30-04.52.51.712937-240 E307143E348           LEVEL: Event
PID     : 8020                  TID : 47284219590976  PROC : db2rocme
128 [db2lco]
INSTANCE: db2lco                NODE : 128
HOSTNAME: node102
FUNCTION: DB2 UDB, high avail services, db2rocm_main, probe:911
MESSAGE : ADM7533I  Cluster caching facility (CF) has started.
START   : CF
```

```
2012-03-30-04.52.51.740243-240 I307492E389          LEVEL: Event
PID     : 8020                 TID : 47284219590976  PROC : db2rocme
128 [db2lco]
INSTANCE: db2lco               NODE : 128
HOSTNAME: node102
FUNCTION: DB2 UDB, high avail services, db2rocm_main, probe:2178
DATA #1 : String, 54 bytes
/home/db2lco/sqllib/adm/db2rocme 1 CF db2lco 128 START
DATA #2 : String, 7 bytes
SUCCESS
```

The db2pd command

You can use the **db2pd** command to troubleshoot DB2, whether you are using the DB2 pureScale Feature. This tool can provide detailed information for assessing the DB2 environment, both at the instance or database level. The following sections describe important **db2pd** command parameters that can be used in a DB2 pureScale environment.

stack

Use the **stack** parameter when you assess the state of the DB2 environment. The stack trace files generated by this parameter can assist in troubleshooting DB2. Stack files provide information about the functions that were being run by each thread or process at the time the **db2pd** command was run. In most cases, you must contact IBM Support to interpret and understand the files that are generated.

cfinfo

Use the **cfinfo** parameter when diagnosing issues with a CF. Example 4-26 shows a sample of the output for this parameter.

Example 4-26 Sample output of the db2pd command with the cfinfo parameter

```
CF Server (128) Information
    Hostname                                = node102.purescale.demo
    Management Port                         = 56000
    Role                                    = Primary
    CF Host Information
       Virtual Memory Used                  = 3771924480
       Virtual Memory Available             = 8857186304
       Real Memory Used                     = 3771924480
       Real Memory Available                = 270000128
       Swap File Used                       = 0
       Swap File Available                  = 8587186176
       CPU Usage                            = 0
```

```
    Total Space (4KB)                       = 192256
    Free Space (4KB)                        = 256
    Frame Size (4KB)                        = 256
    Configured Size (4KB)                   = 0

CF Server (129) Information
    Hostname                                = node103.purescale.demo
    Management Port                         = 56000
    Role                                    = Secondary
    CF Host Information
       Virtual Memory Used                  = 3681800192
       Virtual Memory Available             = 8947310592
       Real Memory Used                     = 3681800192
       Real Memory Available                = 360124416
       Swap File Used                       = 0
       Swap File Available                  = 8587186176
       CPU Usage                            = 0
    Total Space (4KB)                       = 192256
    Free Space (4KB)                        = 256
    Frame Size (4KB)                        = 256
    Configured Size (4KB)                   = 0
```

cfpool

The **cfpool** parameter gives database administrators the ability to monitor the CF connection pool entry on the current member and its status, including whether it is being used. It also can provide information about the number of connections and HCA port mapping, which can help you validate that load balancing between HCA ports is behaving as intended.

serverlist

The **serverlist** parameter can provide information about the available members in the cluster and the relative priority of each available member. The higher the priority number, the more likely connections and transactions for workload balancing are sent to that host. Example 4-27 shows sample output from using this parameter.

Example 4-27 Sample output of the db2pd command with the serverlist parameter

```
Server List:
   Time:          Fri Mar 30 04:48:18
   Database Name: DTW
   Count:         2

   Hostname                   Non-SSL Port   SSL Port   Priority
   node102.purescale.demo     50001          0          81
```

```
   node103.purescale.demo    50001          0          100
```

totalmem

The **totalmem** parameter can provide the total amount of memory allocated on a DB2 host. It also can provide memory information for the member itself, and the amount of reserved restart light memory that is pre-allocated on the host. Example 4-28 shows a sample output from using this parameter.

Example 4-28 Sample output of the db2pd command with the totalmem parameter

```
Total Memory Statistics in KB

                     Controller   Memory     Current  HWM       Cached
                     Automatic    Limit      Usage   Usage      Memory
==================== =========== ==========  ======= =======    =======
Member 0               Yes       1744848     874880  875136     278912
Restart Light Memory   Yes       235924      296448  296448     27648

Total current usage:  1171328
Total cached memory:  306560
```

Operating system logs

Operating system logs can potentially provide valuable information when you are trying to determine and solve a problem. Operating system logs are used primarily in the initial stages of identifying a problem because, in most cases, the information that is captured might not be relevant to the problem encountered within DB2. However, if you can correlate a system-captured problem with a DB2 error, this information can help identify the problem and save you a considerable amount of time.

For example, imagine that you notice, when running a **db2instance -list** command, that one of your members is currently on another host in a `waiting_for_failback` state. You go to the failed member, look at the operating system logs, and notice that there was a message that concerns a network failure. Based on this information, you can narrow down the issue to explicitly check for hardware issues.

Thus, although the relevance of the system logs might not be useful when addressing issues with DB2, they are always worth considering within your initial assessment as a way to eliminate hardware and operating system problems.

On AIX, you can look through the `errpt` logs, and the `syslog` if it is configured. On Linux, the equivalent files would be in the `/var/log` directory, and are presented as the messages file.

5

Upgrading to DB2 10.1

This chapter describes the process of upgrading to DB2 10.1. It explains the planning that is required to perform an upgrade, provides support information to consider, and describes how to upgrade related DB2 components. It presents the upgrade path from DB2 9.8 to DB2 10.1 and describes how to add the DB2 pureScale Feature to your instance.

This chapter includes the following topics:

- Benefits of upgrading
- Supported upgrade paths
- Planning for an upgrade
- Performing the upgrade to DB2 10.1
- Upgrading from DB2 9.5 or 9.7 to DB2 10.1
- Enabling the DB2 pureScale Feature
- Upgrading a DB2 9.8 server to DB2 10.1
- Post-upgrade steps

5.1 Benefits of upgrading

Whether you upgrade from an older version of DB2 or switch from a different database solution, DB2 10.1 offers the following benefits:

- Lower database costs: DB2 offers full control over processors, and fewer processors are required, meaning that there is less hardware and software to license and maintain.
- Lower storage costs: The compression features in DB2 10.1 help reduce the cost of ownership because there is less storage capacity to purchase and maintain. Some customers report compression rates of up to 83%, which means that they require much less hardware to store their data.
- Easier administration: Using the various tools and available automatic administration features of DB2 10.1, such as memory management and other tasks, can increase efficiency among database administrators.
- Improved performance: DB2 10.1 offers performance improvements over previous editions, including reductions in query execution time.
- Productivity improvements: Functions are available for pureXML, data warehousing, and traditional OLTP applications.
- Improved security: DB2 10.1 offers simplified security management with database roles, label-based access control (LBAC), and row and column access control (RCAC). The security and audit features of DB2 ensure that you can meet current needs and evolve to meet future needs.
- Business resiliency innovations: Database on-demand capabilities allow information design to be changed dynamically, often without database outages.
- High availability: DB2 10.1 includes improved failover and recovery, with more functions for schema evolution, logging enhancements, and online utilities. It also offers automated object recovery management and enhancements to automated maintenance configuration.

Upgrading your current architecture to DB2 10.1 requires an understanding of upgrade concepts, upgrade restrictions, and upgrade recommendations, all of which are explained in this chapter.

5.2 Supported upgrade paths

The most important part of upgrading to DB2 10.1 is ensuring that the upgrade is planned and supported. An unsupported upgrade can result in errors that cost time and potentially force you to restructure the upgrade or even start over. The following list of supported upgrades can help ensure that a plan is compliant with the basic requirements for upgrading to DB2 10.1:

- Upgrading directly to DB2 10.1 is supported from DB2 9.5, DB2 9.7, and DB2 9.8 (even if there are multiple copies of DB2 on the server). Upgrading from earlier releases requires you to first upgrade to DB2 9.5.
- Upgrading to a DB2 10.1 non-root installation is supported from similar non-root installations of DB2 9.5 and DB2 9.7. Upgrading from non-root DB2 9.5 or DB2 9.7 installations to a DB2 10.1 root installation is not supported.
- Upgrading from a partitioned database environment with multiple database partitions is supported.
- Restoring full database offline backups from a version lower than DB2 10.1 copies is supported. Rolling forward logs from the previous version is not supported. For a full list of backup and restore operations, review the **RESTORE DATABASE** command.

In addition, to ensure a successful upgrade, pay attention to the following special conditions:

- Ensure that the user that is upgrading the instance has the level of operating system authorization that is required to perform the upgrade.
- Ensure that all instances are stopped and inactive before starting an upgrade. You might have to run **db2stop** to stop an instance.
- If an instance is already at DB2 10.1 or later, run **db2iupdt** to update to different Fix Pack levels or copies of DB2 10.1.
- If you are trying to migrate from DB2 10.1 back to DB2 9.7 or DB2 9.5, you can reverse the DB2 server upgrade. However, you must meet the prerequisites and follow the steps documented in the *Reversing DB2 server upgrade* section of the DB2 10.1 Information Center at the following website:

  http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/index.jsp

- The instance that you are upgrading to DB2 10.1 must be supported. Table 5-1 describes the upgrade support for each type of instance, sorted by DB2 database product.

This list of supported upgrades defines the fundamental rules that must be followed to achieve successful results.

Table 5-1 Upgrade paths by instance type

Instance type	Node types	Upgrade supported
Client (default type for DB2 clients)	Client	Upgrade to a client, a stand-alone, a Workgroup Server Edition, or an Enterprise Server Edition instance is supported.
Stand-alone	Database server with local clients	Upgrade to a stand-alone, a Workgroup Server Edition, or an Enterprise Server Edition instance is supported. Upgrade to a client instance is unsupported.
WSE (default type for DB2 Workgroup Server Edition)	Database server with local and remote clients	Upgrade to a Workgroup Server Edition or an Enterprise Server Edition instance is supported. Upgrade to a stand-alone instance creates a stand-alone instance (Linux and UNIX only). Upgrade to a client instance is unsupported.
ESE (default type for DB2 Enterprise Server Edition)	Partitioned database server with local and remote clients or Enterprise Server Edition with local and remote client	Upgrade to an Enterprise Server Edition instance is supported. Upgrade to a stand-alone or a Workgroup Server Edition instance from single database partition environments is supported and creates a stand-alone or Workgroup Server Edition instance (Linux and UNIX only). Upgrade to a client instance is unsupported.

If you are adding the DB2 pureScale Feature, you first must upgrade your instance to Enterprise Server Edition. If you have a partitioned database server, you must remove the partitions, except for the last one, before adding the DB2 pureScale Feature. Support for adding the DB2 pureScale Feature is available in DB2 9.8; however, this chapter assumes that you upgraded to DB2 10.1 before adding the DB2 pureScale Feature to your instance.

5.3 Planning for an upgrade

Creating a plan for an upgrade is one of the most effective ways to ensure that all components are considered and accounted for, including DB2 servers, DB2 clients, tools, and database applications, which can include scripts, routines, and so on. To ensure success and minimize costs, devise a strategy that follows leading practices and that complies with any important restrictions.

Before you begin: Back up your databases and configuration settings before starting the upgrade of your DB2 servers.

When planning for an upgrade, complete the following steps:

1. Review the new, deprecated, and discontinued functionality for DB2 10.1 or for any releases between the release you are upgrading from and DB2 10.1. You can find this information at the following website:

   http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/index.jsp?topic=%2Fcom.ibm.db2.luw.wn.doc%2Fdoc%2Fc0052036.html

   Tip: Remove deprecated functions early on and move to new functions for your database products, applications, and routines to ensure that functionality is enhanced to the latest releases and to help improve overall performance.

2. After determining the feature changes you will be making, take time to plan the logistics for modifying database applications and routines so that they run successfully in DB2 10.1. Cross-reference all instances where deprecated or unsupported routines are called and think about how they will be modified.
3. Set up a DB2 10.1 test server and create test databases against which to test the database applications and routines.
4. If you plan to enable the DB2 pureScale Feature, confirm that the database meets the requirements for the DB2 pureScale Feature.

5. After testing your application and ensuring that it works with the DB2 10.1 test server, test the upgrade of your DB2 servers with an upgraded DB2 client. This test can help you resolve any issues before putting your test environment into production.
6. Upgrade only one of the clients in the production or test environment to ensure that it is running as expected. Using only one client allows you to identify issues quickly.
7. After you resolve any issues and ensure that all tests run successfully on all of the components independently, you can turn the test environment into a production environment or upgrade the production environment to the tested level. If the plan is to upgrade the production environment, ensure that you tested all upgrade steps in a separate environment so that you are prepared for any unexpected issues.

As with any plan, separate the work into detailed tasks or subprocesses. Consider the following subprocesses:

- Upgrade prerequisites: Tasks to complete to ensure that the environment supports the upgrade
- Pre-upgrade tasks: A list of any changes that must be made to allow the upgrade to occur
- Upgrade tasks: A step-by-step plan for the upgrade
- Post-upgrade tasks: Any other tasks that need to be performed after the upgrade is complete

See the DB2 10.1 Information Center for additional resources, updated information, instructional materials, white papers, and web casts about upgrading to DB2 10.1:

http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/index.jsp

Also, consider having a contingency plan in case things go wrong. Contingency plans are for worst-case scenarios that you hope to never encounter, but you need to plan for them just in case, especially when upgrading a production system.

5.4 Performing the upgrade to DB2 10.1

An upgrade plan must consider all components that are involved in the upgrade, including the DB2 servers, DB2 clients, tools, and database applications. This section reviews each of these components as they relate to an upgrade to DB2 10.1.

5.4.1 Upgrading DB2 servers

To upgrade DB2 servers to DB2 10.1 on Linux and UNIX operating systems, you must install a new DB2 10.1 copy. When it is installed, you need to manually upgrade any existing instances and databases to the new copy.

Regardless of the instance bit size and whether you perform the upgrade from DB2 9.5, DB2 9.7, or DB2 9.8 to DB2 10.1, you need to determine the tasks that apply best to your particular environment. A successful upgrade requires that you consider the base components that are related to a successful installation of DB2 10.1. To achieve a successful upgrade, ensure that the following factors are in place before you begin:

- You have root access to your machines.
- You meet the hardware and software installation requirements for DB2 10.1 described in 2.2, “Hardware requirements and considerations” on page 26 and 2.3, “Software requirements and considerations” on page 37.
- You reviewed the disk space requirements described in 2.2.2, “Hardware considerations” on page 31.
- You performed any of the required pre-upgrade tasks described in 3.2, “Configuring a Linux system to install the DB2 pureScale Feature” on page 80 and 3.3, “Configuring an AIX system to install the DB2 pureScale Feature” on page 150.

Restrictions

Before starting the upgrade, it is important to ensure that the supported configurations mentioned in 5.2, “Supported upgrade paths” on page 253 are met. In addition to these requirements, keep in mind that on Linux and UNIX operating systems, your existing 32-bit or 64-bit instances are required to upgrade automatically to DB2 10.1 64-bit instances. The operating system and DB2 10.1 database product that you installed together determine the instance bit size.

General procedure

To upgrade a DB2 server to DB2 10.1, complete the following steps:

1. Log on to the DB2 server as root.
2. Ensure that a default gateway exists on the Ethernet network that is used by the DB2 pureScale Feature. If the gateway does not exist, the binary installation of DB2 10.1 fails. You can check for a gateway by running **`netstat -r`**.

Example 5-1 shows the output of this command.

Example 5-1 Running the netstat command

```
node101:/opt/ibm/db2/V10.1 # netstat -r
Kernel IP routing table
Destination   Gateway     Genmask      Flags  MSS Window irtt Iface
link-local    *           255.255.0.0  U      0   0         0 eth0
172.16.0.0    *           255.255.0.0  U      0   0         0 eth0
loopback      *           255.0.0.0    U      0   0         0 lo
default       172.16.0.1  0.0.0.0      UG     0   0         0 eth0
```

3. Run **db2setup**.
4. If you are not planning to use the DB2 pureScale Feature, select Install New under the DB2 Enterprise Server Edition Version 10.1 option.

   If you plan to use the DB2 pureScale Feature, select Install New under the DB2 Enterprise Server Edition Version 10.1 with the IBM DB2 pureScale Feature option.

   Ensure that you select all the add-on products that were installed in the previous DB2 copy.

   > **Note:** You cannot use the “Working with Existing” option if you have the DB2 pureScale Feature installed already. If so, see 5.6, “Enabling the DB2 pureScale Feature” on page 270.

5. Go through the installation panes. Read and accept the license. You are then prompted for the directory in which you want to install DB2 10.1. You might need to run **db2rmln** to remove the links added in to the `/usr/lib` directory by DB2 in previous installations before continuing.
6. When presented with the option of creating a DB2 instance, select Do not create a DB2 instance. When presented with the Host List pane, do not add any hosts. You add the hosts that are used for the DB2 pureScale instance in a later step.

With these steps completed, you can upgrade the previous DB2 (9.5, 9.7, or 9.8) instances from the same installation path that you indicated during the DB2 10.1 installation. You can also upgrade your database administration server if you want to keep the existing DAS configuration and use new functionality available in DB2 10.1.

This general procedure covers the process of installing DB2 10.1 binary files on the host that is upgraded to DB2 10.1. If you do not want to update clients at this time, see 5.4.3, “Upgrading a DB2 instance” on page 261 for details about upgrading the DB2 instance to Version 10.1. A more detailed explanation of upgrading from DB2 9.5 and 9.7 to DB2 10.1 is presented in 5.5, “Upgrading from DB2 9.5 or 9.7 to DB2 10.1” on page 267, along with an explanation of upgrading from DB2 9.8 when the instance already has the DB2 pureScale Feature.

5.4.2 Upgrading DB2 clients

When upgrading to DB2 10.1, upgrade the clients and test the application using the new clients against the DB2 10.1 servers. This upgrade involves installing a DB2 10.1 client copy and then upgrading the client instance. After the upgrade, it is important to ensure that the client instance keeps the previous information about your client configuration, your cataloged nodes, and your cataloged databases.

The current level of the client that you installed determines the way to upgrade to DB2 10.1. You can directly upgrade to DB2 10.1 clients from DB2 9.5, 9.7, or 9.8. If you have DB2 9.1 or earlier clients, you must upgrade to a DB2 9.5 client first.

To ensure that your upgrade goes successfully, confirm that the following conditions are met:

- You have root user authority.
- You have SYSADM, SYSCTRL, or SYSMAINT authority and root access to run **`db2iupgrade`** and the **`db2icrt`**.
- You meet the installation requirements for DB2 database products. Some operating systems require a 64-bit kernel.
- You reviewed the supported connectivity between clients and DB2 database servers.

Important: You do not need to update all your DB2 clients at the same time. You can still run older clients with DB2 10.1; however, your client cannot access the new functionality until it is upgraded appropriately.

Restrictions

As with upgrading DB2 servers, keep in mind the following criteria when upgrading to a DB2 10.1 client:

- Upgrades are possible only from a DB2 9.5 or DB2 9.7 Data Server Client to a DB2 10.1 Data Server Client.
- Upgrades are possible only from a DB2 9.5 or DB2 9.7 Data Server Runtime Client to a DB2 10.1 Data Server Runtime Client.
- On Linux and UNIX operating systems (except for Linux on x64), existing 32-bit or 64-bit client instances are upgraded to DB2 10.1 64-bit client instances. The bit size of the client instance is determined by the operating system where you install the DB2 10.1 client.
- On Linux and UNIX operating systems, you must not set up the instance environment for the root user. Running **db2iupgrade** or **db2icrt** when you set up the instance environment is not supported.

Complying with these criteria can help you achieve a successful upgrade to DB2 10.1 clients.

General procedure

The general procedure to upgrade a DB2 client to Version 10.1 is outlined here. An upgrade is possible either by installing and creating a DB2 10.1 client instance or by upgrading an existing client instance. Creating a DB2 10.1 client instance is useful when you want to keep multiple client copies running on the same machine.

These steps provide the general guidelines for upgrading the DB2 client by upgrading an existing client instance:

1. Log on to the system with root user authority.
2. Install the appropriate DB2 10.1 client as a new copy by running db2setup, and then select Install New on the Install a Product pane. If upgrading from a DB2 9.5 or DB2 9.7 Data Server Client, install a new DB2 10.1 Data Server Client. If upgrading from a previous Data Server Runtime client, install a new DB2 10.1 Data Server Runtime client copy.
3. Upgrade your existing client instances by running **db2iupgrade**:

   ```
   $DB2DIR/instance/db2iupgrade <InstName>
   ```

   Where:

   – *DB2DIR* refers to the location that is specified during the DB2 10.1 client installation.

   – *InstName* refers to the login name of the client instance owner.

The default installation path for UNIX is the `/opt/IBM/db2/V10` directory and for Linux it is the `/opt/ibm/db2/V10.1` directory.

A second method is to create a DB2 client instance. Run the following command:

```
$DB2DIR/instance/db2icrt -s client <InstName>
```

In this case, *DB2DIR* is set to the location that you specified during the DB2 10.1 client installation and *InstName* is the login name of the instance owner.

After upgrading the DB2 client to Version 10.1, remember to import the same client connectivity environment you had previously, including the database manager configuration parameter and DB2 profile registry settings. Run **db2cfimp** with the configuration profile that you backed up as one of the pre-upgrade tasks. After performing this command, compare the before and after values to ensure that they are compatible with your application.

5.4.3 Upgrading a DB2 instance

Upgrading a DB2 instance depends on various **db2iupgrade** parameters. DB2 makes the upgrade process easier by automatically performing additional upgrades when the database directory is accessed for the first time, either implicitly through the **LIST DATABASE DIRECTORY** command or explicitly through the **UPGRADE DATABASE** command. After this upgrade occurs, DB2 automatically modifies the following components to ensure that the operation succeeds:

- Database configuration file: This file is created for each database and contains values for the following types of configuration parameters that affect the usage of the database:
  - Parameters that are specified or used when creating the database, such as database code page, collating sequence, and DB2 database release level
  - Parameters that indicate the current state of the database, such as backup pending flag, database consistency flag, and rollforward pending flag
  - Parameters that define the amount of system resources that the operation of the database can use, such as buffer pool size, database logging, and sort memory size
- Log file header: This header contains information that details the log record and transaction information of the log record writer.
- Table root page for all tables: This directory contains a list of all the tables for your particular database.

- Index root page for all tables: This directory contains a list of the indexes for all the tables that are updated.
- Catalog tables: Each database includes a set of system catalog tables that describe the logical and physical structure of the data.
- Buffer pool files: These files specify which buffer pools are used and which ones are started or stopped.
- History file: This file provides the history of items that were run against the database.

Restrictions

Before starting an upgrade, ensure that it fits within the supported configurations described in 5.2, “Supported upgrade paths” on page 253.

General procedure

When you run the **db2iupgrade** command, the following actions occur:

1. The **db2ckupgrade** command is called. If it finds any condition that prevents the upgrade from succeeding, the command fails and returns the `DBI1205E` error code.
2. The installation path to the `db2iupgrade` binary is used to determine the level of code to which the instance is upgraded.
3. The instance profile registry variables are upgraded. Global profile registry variables that are created by the user are not upgraded.
4. The database manager configuration is updated, including updates to configuration parameters such as the `JDK_PATH`.
5. If the audit facility is enabled, the `db2audit.cfg` audit configuration file is updated.
6. The SSL parameter value in the `SSLconfig.ini` configuration file (if it exists) is used to set the new database manager configuration parameters, and the instance profile registry setting `DB2COMM=SSL` is updated.

When these actions are completed, you can upgrade the database

Note that you have not enabled the DB2 pureScale Feature yet. This step is performed after the database is upgraded to V10.1.

5.4.4 Upgrading the database

After you upgrade the instances to DB2 10.1, upgrade each database under each instance. Before starting this process, ensure that each of these factors is in place:

- You have SYSADM authority.
- All of the local databases that you want to upgrade are cataloged.
- You installed DB2 10.1 and upgraded the instance to DB2 10.1.

One of the most helpful tools to make upgrades easier is the **db2ckupgrade** command. It verifies that your database is ready for upgrade by cross-referencing a list of conditions.

> **Tip:** Run **db2ckupgrade** with the **-l** parameter to write warning messages for certain conditions to a log file.

Here is a breakdown of the possible **db2ckupgrade** parameter values:

```
>>-db2ckupgrade--+-database-+-- -l--filename--+--------+-------->
                 '- -e------'                 '- -not1-'
>--+-------------------------------+-------------------------><
   '- -u--userid--+---------------+-'
                  '- -p--password-'
```

The values are defined as follows:

- The **database** tag specifies the database on which to check the upgrade. Using the **-e** option specifies that all local catalog databases must be scanned.
- The **-l filename** option specifies the log file name to be used.
- With **-not1**, you can disable the type-1 indexes, and the **-p** and **-u** options are used to specify the password and user name for the specific database to which you are connecting.

The **db2ckupgrade** command verifies that each of the following conditions for a successful upgrade is true:

- A cataloged database exists.
- A database is not in an inconsistent state.
- A database is not in a backup pending state.
- A database is not in a restore pending state.
- A database is not in a rollforward pending state.
- Tables are not in a load pending state.

- Tables are not in a redistribute pending state.
- For DB2 9.8 or later, table space container paths are not nested mount points. For DB2 9.8 Fix Pack 3 or later, the I/O write operations for the database are not suspended or being suspended.
- There are no materialized query tables (MQTs) that depend on system views.
- Table spaces are in a normal state.
- No database contains user-defined types (UDTs) with the names ARRAY, BINARY, CURSOR, DECFLOAT, ROW, VARBINARY, or XML.
- No database contains the built-in DATALINK data type.
- No database has a schema with the name SYSPUBLIC.
- No database has orphan rows in system catalog tables (which would cause a database upgrade to fail).
- Any database enabled as a high availability disaster recovery (HADR) primary database allows successful connections.
- No HADR database role is standby.
- If SYSCATSPACE is a DMS table space and `AUTORESIZE` is not enabled, SYSCATSPACE has at least 50% free pages of total pages.
- No database is enabled for XML Extender.

Understanding the checks that are performed and ensuring that your database is compliant with them makes the process of upgrading much simpler.

Restrictions

Because they are directly related, the same restrictions that apply to the DB2 servers apply for a database upgrade. For universal conditions that specify the types of upgrades that can be performed, see 5.2, “Supported upgrade paths” on page 253.

General procedure

Before starting the procedure to upgrade a database to DB2 10.1, remove any `db2diag.log` files and any existing dump files, trap files, and alert log files from the directory indicated by the **`diagpath`** parameter. Removing these files can help you more easily diagnose and isolate issues that might occur after the upgrade process begins. After you remove these files, complete the following steps:

1. Log on to the DB2 server as the instance owner or a user with SYSADM authority.
2. Recatalog the database by running **`CATALOG DATABASE`**:

```
db2 CATALOG DB database_name as database_alias
```

3. Run **LIST DATABASE DIRECTORY** to ensure that the database is in the list of all cataloged databases in the current instance.
4. Initiate the upgrade by running **UPGRADE DATABASE**:

   ```
   db2 UPGRADE DATABASE database-alias USER username USING password
   ```

In this example, `database-alias` refers to the name or the alias of the database you want to upgrade and the user name and password needed to authenticate a user with SYSADM authority. In addition to this command, you might want to consider adding the **REBIND ALL** parameter, which specifies that a rebind of all packages is performed during the upgrade.

It is possible that the database upgrade fails (return an SQL1704N error). To determine the cause of failure, see Table 5-2, which lists some of the common errors and what they mean.

Table 5-2 Upgrade failure errors and possible causes

Upgrade failure	Cause / Explanation
Returns SQL1704N Database upgrade failed. Reason code 3.	One of the most common causes of upgrade failure is that the log file space is not large enough. After increasing the log file size, run the **UPGRADE DATABASE** command again. After it completes, reset the value of the **logfilsiz**, **logprimary**, and **logsecond** database configuration parameters.
Returns SQL1243W warning message.	You need to drop or rename the `SYSTOOLS.DB2LOOK_INFO` table. Otherwise, the `ALTER TABLE` and `COPY SCHEMA` statements fail to run. To rename it, run this command: `db2 RENAME SYSTOOLS.DB2LOOK_INFO TO new-table-name`
Returns SQL1499W warning message followed by DM7535W warning message.	The **UPGRADE DATABASE** command failed to refresh the table space attributes in the catalog table. However, the database still upgrades successfully.
Returns SQL1499W warning message followed by ADM4003E warning.	The **UPGRADE DATABASE** command failed to upgrade the DB2 Text Search catalogs or indexes because of an error in a stored procedure.
Returns SQL1499W warning message followed by ADM7534W warning.	The **UPGRADE DATABASE** command failed to refresh the table space attributes in the catalog table. However, the database still upgrades successfully.
Returns SQL1499W warning message followed by ADM4102W warning.	This warning is returned when you qualify or delimit with quotation marks the identifiers called NULL in your SQL statements to avoid conflict with the NULL keyword.

Upgrade failure	Cause / Explanation
Returns SQL1499W warning message and writes the ADM4106W warning message.	This warning requires you to drop all references to the XML Extender user-defined data types and drop all XML Extender database objects under the DB2XML schema. The XML Extender was discontinued starting with DB2 9.7.
Returns SQL1499W warning message followed by the ADM4105W warning message.	This warning requires you to create new WebSphere MQ functions for the XML data type by running **enable_MQFunctions** with the **-xml** parameter.
Returns the SQL1499W warning message followed by the ADM9516W warning message.	This warning requires you to verify that the **indexrec** configuration parameter is set to RESTART and to run **RESTART DATABASE** to rebuild indexes marked as invalid during the database upgrade. If you do not do this action, the index rebuild starts upon your first access to the table and you might experience an unexpected degradation in response time.
Returns SQL0473N error message.	This warning requires you to reverse the database migration and re-create all user-defined data types that use a system built-in data type name with a different name that is not restricted. When finished, verify that your databases are ready for upgrade.
Returns ADM4003E error message.	Requires an upgrade the DB2 Text Search catalog and indexes manually. For details, see SYSTS_UPGRADE_CATALOG and SYSTS_UPGRADE_INDEX.

Remember: If you use identifiers called NULL in an SQL statement for column names, routine parameter names, or variable names, and if you do not fully qualify or delimit these identifiers with quotation marks, the identifier name might resolve to the **NULL** keyword instead. This resolution results in a change in behavior from previous releases.

5.5 Upgrading from DB2 9.5 or 9.7 to DB2 10.1

As part of the upgrading a DB2 database server to DB2 10.1, you must also upgrade your instances and databases.

To successfully upgrade from DB2 9.5 or 9.7, ensure that the following factors are in place before you begin:

- You have root user authority on the Linux and UNIX operating systems.
- You installed any DB2 database add-on products that were installed in the DB2 copy from which you are upgrading.
- The following recommended conditions are met before the **db2iupgrade** command is issued:
  - On Linux and UNIX operating systems, there is 5 GB of free space in the `/tmp` directory (where the instance upgrade trace file is written).
  - If the upgrade is part of a process to enable the DB2 pureScale Feature, there is a configured default gateway on all hosts that is part of the DB2 pureScale cluster.
  - You gathered pre-upgrade diagnostic information to help diagnose any problems that might occur after the upgrade. For details, see the *Gathering pre-upgrade diagnostic information* section of the DB2 Information Center at the following website:

    http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/index.jsp

Important: This section does not explain the process of enabling the DB2 pureScale Feature. To enable it, upgrade instances and databases first, and then complete the steps in 5.6, “Enabling the DB2 pureScale Feature” on page 270. Upgrade DB2 instances to Version 10.1 before the enabling the DB2 pureScale Feature.

5.5.1 Restrictions

Use the steps provided here to upgrade *only* from DB2 9.5 or DB2 9.7. As always, the standard upgrade restrictions apply, as described in 5.2, “Supported upgrade paths” on page 253. On Linux and UNIX operating systems, do *not* set up the instance environment for the root user.

You can use the **-not1** special case parameter with the **db2ckupgrade** command for DB2 9.5 databases. If this parameter is omitted, the **db2ckupgrade** command calls the **db2IdentifyType1** command to identify type-1 indexes and to generate a script to convert type-1 indexes to type-2 indexes for a specific database.

5.5.2 General procedure

To upgrade existing DB2 9.5 or DB2 9.7 instances to DB2 10.1, run **db2iupgrade** and complete the following steps:

1. Determine the node type, as shown here, to learn whether you can upgrade existing instances to a DB2 10.1 copy that you installed:

   ```
   db2 GET DBM CFG | grep 'Node type'
   Node type = Partitioned database server with local and remote
   clients.
   ```

   Review Table 5-1 on page 254 to determine the instance type and whether an instance upgrade is supported. In this example, the node type is called *Partitioned database server with local and remote clients*, which correlates to the ESE instance type.

   As shown in Table 5-1 on page 254, on Linux and UNIX operating systems, you can upgrade to a DB2 10.1 copy of DB2 Workgroup Server Edition, but your instance is re-created with type WSE by using default configuration values. If you cannot upgrade an instance to any DB2 10.1 copy that you installed, install a copy of the DB2 10.1 database product that supports an upgrade of your instance type before you proceed with the next step.

2. Disconnect all users, stop the application processes, and stop your existing DB2 instances by running the following commands:

   – **db2 force applications all**

   – **db2stop**

3. Log on to the DB2 database server with root user authority (on Linux and UNIX operating systems) or Local Administrator authority (on Windows operating systems).

4. Upgrade your existing instances by running **db2iupgrade** from the target DB2 10.1 copy location. You must run **db2iupgrade** only on the host where the DB2 instance exists (or the host that contains a local copy of the `sqllib` directory for partitioned databases).

   Run the following command:

   ```
   $DB2DIR/instance/db2iupgrade [ -u fencedID ] InstName
   ```

   Where:

   – $*DB2DIR* refers to the location that is specified during the DB2 10.1 installation.

   – *fencedID* refers to the user name under which the fenced user-defined functions (UDFs) and stored procedures are run.

   – *InstName* refers to the login name of the instance owner.

> **Important:** If you did not install all DB2 database add-on products that were installed in the DB2 copy from which you are upgrading, the instance upgrade fails and returns a warning message. If you plan to install these products later or if you no longer need the functionality provided by these products, use the **-F** parameter to upgrade the instance.

The **db2iupgrade** command calls the **db2ckupgrade** command with the **-not1** parameter to verify that the local databases are ready for upgrade. The `update.log` file is specified as the log file for the **db2ckupgrade** command, and the default log file created for the **db2iupgrade** command is `/tmp/db2ckupgrade.log.processID`. On Linux and UNIX operating systems, the log file is created in the instance home directory. The **db2iupgrade** command does not run if the **db2ckupgrade** command reports errors. Check the log file if you encounter any errors.

5. Log on to the DB2 server as a user with sufficient authority to start the instance, and then run **db2start** to start the instance.
6. After upgrading the existing DB2 9.5 or DB2 9.7 copy, the database log directories automatically change.

   Review the `db2diag.log` file, which contains entries that detail the new log directories. If a user-defined log directory, such as `/usr/logpath`, is used, the location of the log files is `/usr/logpath/NODE0000/LOGSTREAM0000` after the upgrade, and the old log directory contains only renamed log files. If the default database directory is being used, for example, `/home/db2user/db2inst/NODE0000/SQL00001/SQLOGDIR`, the location of the log files is `/home/db2user/db2inst/NODE0000/SQL00001/LOGSTREAM0000` after the upgrade, and the old log directory contains only renamed log files.

After you complete these steps, run a quick check to verify that the instance is now running on DB2 10.1 by running **db2level**, which should return a set of tokens that include a string such as `DB2 10.1.`*`X.X`* (where *X* is a number).

5.6 Enabling the DB2 pureScale Feature

At this stage, you should have a single node DB2 10.1 instance. If you have a partitioned database and want to change it to a DB2 pureScale instance, you must remove all of the data partitions from all of the databases until you have only one data partition left. The instructions to accomplish this task are beyond the scope of this book, but you can find information about how to remove data partitions from a partitioned database in the DB2 Information Center at the following website:

http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/index.jsp

The next steps for adding the DB2 pureScale Feature to your instance depend on whether the database meets the requirements for the DB2 pureScale Feature. If there is no database created, you can continue to 5.7, “Upgrading a DB2 9.8 server to DB2 10.1” on page 276.

5.6.1 Restrictions

A DB2 pureScale Feature-enabled database must use Automatic Storage table spaces, and both the data and the logs must be on a General Parallel File System (GPFS).

5.6.2 General procedure

This procedure includes the following tasks:

- Enabling the DB2 pureScale Feature in DB2 10.1 instances without databases
- Enabling the DB2 pureScale Feature in DB2 10.1 instances with databases

If you have a database in the DB2 10.1 instance but you have not placed it into a GPFS file system yet, see 5.6.3, “Resolving possible IBM DB2 pureScale Feature database enablement issues” on page 273. Then continue with “Enabling the DB2 pureScale Feature in DB2 10.1 instances with databases” on page 271.

Enabling the DB2 pureScale Feature in DB2 10.1 instances without databases

To enable the DB2 pureScale Feature when there are no databases cataloged in the instance, complete the following steps:

1. Run **db2iupdt** from the DB2 10.1 installation directory:

   ```
   $DB2DIR/instance db2iupdt -cf host2 -cfnet host2-ib0 -m host1 -mnet
   host1-ib0 -instance_shared_dev /dev/hdisk1 -tbdev /dev/hdisk2 -u
   fencid db2sdin1
   ```

   Where:

   – $*DB2DIR* is the location of the DB2 10.1 binary files.

   – *host2* and *host2-ib0* are the host name and InfiniBand host name for the CF host.

   – *host1* and *host1-ib0* are the host name and InfiniBand host name for the member

   – */dev/hdisk1* is the disk used for the `sqllib_shared` directory.

   – */dev/hdisk2* is used for the DB2 cluster services tiebreaker disk.

   – *fencid* is the name of the fenced user.

   – *db2sdin1* is the instance name.

   This command creates the Tivoli System Automation cluster, the GPFS cluster, and the first shared file system. It also sets up the first member and the first cluster caching facility.

2. Add another member by running the following command:

   ```
   $DB2DIR/instance/db2iupdt -add -m host3 -mnet host3-ib0 db2sdin1
   ```

3. Add a second cluster caching facility by running the following command:

   ```
   $DB2DIR/instance/db2iupdt -add -cf host4 -mnet host4-ib0 db2sdin1
   ```

Enabling the DB2 pureScale Feature in DB2 10.1 instances with databases

To enable the DB2 pureScale Feature when there are databases cataloged, complete the following steps:

1. Run **db2checkSD** to identify any issues that might prevent access to the database after the DB2 pureScale Feature is enabled. If you encounter any errors or warnings from this command, correct those issues before continuing.

To run **db2checkSD**, use the following syntax for every database that is cataloged:

```
db2checkSD <database> -l <logfileName> -u <user> -p <password>
```

This command creates a log file in the current directory with a list of warnings and errors that it encounters. Solve these problems before continuing with this procedure. For more information, see 5.6.3, “Resolving possible IBM DB2 pureScale Feature database enablement issues” on page 273.

2. Run **db2iupdt** to upgrade the instance to the DB2 pureScale Feature:

```
$DB2DIR/instance db2iupdt -cf host2 -cfnet host2-ib0 -m host1 -mnet
host1-ib0 -instance_shared_dir /dev/db2fs1 -tbdev /dev/hdisk2 -u
fencid db2sdin1
```

   Where:

   – $*DB2DIR* is the location of the DB2 10.1 binary files.
   – *host2* and *host2-ib0* are the host name and InfiniBand host name for the CF host.
   – *host1* and *host1-ib0* are the host name and InfiniBand host name for the member.
   – */dev/db2fs1* is the file system used for the `sqllib_shared` directory.
   – */dev/hdisk2* is used for the DB2 cluster services tiebreaker disk.
   – *fencid* is the name of the fenced user.
   – *db2sdin1* is the instance name.

   The shared file system used in this command is the one created by the **db2cluster_prepare** command described in “Creating a GPFS cluster and GPFS file systems” on page 273. The command creates the Tivoli System Automation cluster and sets up the first member and the first cluster caching facility. The GPFS cluster and file system were prepared before you got to this section.

3. Add another member by running the following command:

```
$DB2DIR/instance/db2iupdt -add -m host3 -mnet host3-ib0 db2sdin1
```

4. Add a second cluster caching facility by running the following command:

```
$DB2DIR/instance/db2iupdt -add -cf host4 -mnet host4-ib0 db2sdin1
```

5.6.3 Resolving possible IBM DB2 pureScale Feature database enablement issues

Databases must have logs and data on GPFS file systems and use automatic storage table spaces before the DB2 instance can be enabled for the DB2 pureScale Feature.

Creating a GPFS cluster and GPFS file systems

You can create a GPFS and GPFS cluster using DB2 tools. These tools are intended to reduce the time it takes to set up a working DB2 pureScale instance when you have an existing database that must be upgraded.

Before you begin, make sure that these components are in place:

- Install DB2 Enterprise Server Edition Version 10.1 with the DB2 pureScale Feature.
- Identify the disks that are available on shared storage that are used for the `sqllib_shared` directory and for the database and logs.

  These disks should be accessible by the other hosts that are added to the DB2 pureScale cluster. The disk used for the **`db2cluster_prepare`** command must be at least 10 GB. Prepare a larger disk because it also (by default) holds the diagnostic data for the DB2 pureScale instance (also known as the `db2dump` directory). The sizes of disks used for data and logs vary according to the requirements of your database or databases.

With the necessary components in place, complete these steps to create a GPFS file system and GPFS cluster:

1. Log in as root to the DB2 server host where the current DB2 10.1 database is.
2. Run the following command:

   ```
   $DB2DIR/instance/db2cluster_prepare -instance_shared_dev /dev/hdisk1
   -l /tmp/db2cluster_prepare.log
   ```

   In this command, $*DB2DIR* is the DB2 10.1 installation directory. The command creates a GPFS cluster on this host using `/dev/hdisk1` as the first disk in the first file system created on GPFS. This file system is used later for the `sqllib_shared` directory when you enable the DB2 pureScale Feature. You can use the log file that is created to diagnose errors that occur during the creation of the GPFS cluster or file system.
3. Query the GPFS cluster for the file system name (and where it is mounted) by running the following command:

   ```
   $DB2DIR/bin/db2cluster -cfs -list -filesystem
   ```

4. Create a directory for the instance that you are using. Then, change the ownership of that directory to the DB2 10.1 instance owner and change the group to the primary group for the DB2 10.1 instance. Set the permissions on this directory to read and run for user, group, and others, but give write permission only to the user and group. In other words, the UNIX permissions should be 775.
5. Create additional file systems for the data and log directories, as described in 3.5.2, “Creating shared file systems” on page 189. Then continue with the appropriate procedures as described in the sections that follow.

Database does not have logs on GPFS

When the database does not have logs on the GPFS, you can move the logs to the GPFS. This step does result in downtime for the database, but after the logs are moved, you can enable the DB2 pureScale Feature.

To move the logs, a GPFS cluster and an available file system must exist with enough space to hold the database logs. In addition, the DB2 pureScale Feature does not allow for circular logging, which can prevent recovery from occurring properly. Use a file system that is created on disks with a high throughput capability, and do not share this file system with any other usage. Dedicate the file system to just the logs for the DB2 database.

To move logs from their current location to GPFS, complete the following steps:

1. Log on to the DB2 server as the instance owner with DBADM authority.
2. Run the following command to change the location of the log files:

   ```
   db2 update db cfg for <databaseName> using LOGPATH <newLogPath>
   ```

3. Disconnect all users from the database, and then deactivate the database by running the following command:

   ```
   db2 deactivate database <databaseName>
   ```

4. Reactivate the database to activate the new log path by running the following command. This step can take time to complete, depending on the number of log files that must be created.

   ```
   db2 activate database <databaseName>
   ```

Database does not have data on GPFS

In addition to the requirement that the database data be on GPFS, the database table spaces must use automatic storage. Databases cannot have system-managed table spaces or database-managed table spaces.

If your database is already using automatic storage, the following options are available to move to GPFS:

- Automatic storage databases
- Databases not using automatic storage table spaces

Automatic storage databases

The first option is to back up the current database and perform a redirected restore of the database directly onto GPFS. This method is the simplest method of getting data on to a GPFS; however, this method requires extra space to hold the database backup and can take a long time to back up and restore a large database.

Backing up the database is described in 4.2.5, “Backup and restore” on page 217. The backup process is the same for both DB2 pureScale Feature and non-feature environments.

A redirected restore requires that you have access to your backup image, and that you have the appropriate authorizations to run the command. To start the redirected restore, run the following command:

```
db2 restore db SAMPLE on /db2data/db2sdin1/ DBPATH on $HOME NEWLOGPATH
/db2logs/db2sdin1/logs
```

In this example, the database `SAMPLE` is being restored so that the data is placed in the GPFS file system mounted at `/db2data` in the `/db2data/db2sdin1` directory. The `DBPATH` is in the default home directory and the logs are placed on another GPFS file system mounted on `/db2logs` in the `/db2logs/db2sdin1/logs` directory.

Other options include the following ones:

- Export and import (or load) the data into a new database that uses GPFS file systems. This action effectively rebuilds the database from scratch.
- Add GPFS file systems to the database automatic storage paths and remove the non GPFS paths. Details of this process are available in the DB2 Information Center at the following website:

  http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/index.jsp

Databases not using automatic storage table spaces

In cases where the database is using system-managed table spaces or has system-managed catalog table spaces, the data must be exported and then loaded (or imported) in to the new database. Unfortunately, these types of table spaces do not have a direct conversion method to automatic storage. But there are tools such as IBM Data Studio that can help you move data from one DB2 database to another.

5.7 Upgrading a DB2 9.8 server to DB2 10.1

To achieve a successful upgrade from DB2 9.8, ensure that the following factors are in place before you begin:

- You have root user authority on the Linux and UNIX operating systems.
- You installed any DB2 database add-on products that were installed in the DB2 copy from which you are upgrading.
- The database is backed up before starting the upgrade.
- The following conditions are met in preparation for running **db2iupgrade**:
  - On Linux and AIX, there is 5 GB of free space in the `/tmp` directory, where the instance upgrade trace file is written.
  - The DB2 installation binary files on the hosts are *not* on the GPFS file system. The GPFS file system is unmounted as part of the software update and is unavailable before you start the actual installation of DB2 10.1.
  - Pre-upgrade diagnostic information is gathered to help diagnose any problems that might occur after the upgrade. For details, see the *Gathering pre-upgrade diagnostic information* section of the DB2 Information Center at the following website:

    http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/index.jsp
- The following steps are complete before installing the DB2 10.1 binary files:
  - On Linux and AIX, a default gateway is configured on all hosts that are part of the DB2 pureScale cluster.
  - On AIX, if you are using InfiniBand, the upgraded uDAPL packages are installed and the `/etc/dat.conf` file is configured appropriately.

On Linux and UNIX operating systems, you must manually upgrade DB2 9.8 instances.

5.7.1 Restrictions

The procedure described here is used only to upgrade from DB2 9.8. As always, the standard upgrade restrictions described in 5.2, “Supported upgrade paths” on page 253 apply.

On Linux and UNIX operating systems, do *not* set up the instance environment for the root user.

5.7.2 General procedure

To upgrade existing DB2 9.8 instances to DB2 10.1, copy the installation binary files on to all the hosts and perform a check on any existing database. Then, complete the following steps:

1. Log on to any member host of the DB2 server as the root user and copy the installation binary files to a non GPFS file system. You need to complete this step only on one host that is part of the DB2 pureScale Feature instance. You might need to extract the installation binary files so that you can access the needed files.
2. Log on to the DB2 pureScale Feature instance as the instance owner.
3. Run **db2stop** to stop the database manager. If there are any connections to the database, run **db2 force applications all** to disconnect all users. Then run **db2stop** after all users are disconnected from all databases.
4. From the current host, run **db2stop instance on <*host name*>** for each host. This command prevents the accidental start of the member or cluster caching facility processes in the instance. For example, if you have hosts A, B, C, and D in the DB2 pureScale instance, run this command four times, once for each host.
5. Log on to the DB2 server as root.
6. Put the cluster management software (Tivoli System Automation) into maintenance mode by running the following command on one of the hosts to put all of the hosts into maintenance:

   ```
   db2cluster -cm -enter -maintenance -all
   ```

   Run this command from the existing DB2 9.8 installation directory. It stops the peer domain services on all hosts and prevents the peer domain from restarting during system maintenance.
7. Run the following command to put the cluster file system in to maintenance mode on all hosts:

   ```
   db2cluster -cfs -enter -maintenance -all
   ```

   Run this command from the DB2 9.8 installation directory. It stops all hosts from accessing the cluster GPFS during system maintenance.

8. On each host, install DB2 10.1 by running **db2setup**. If you are using the graphical interface, make sure that you click **Install New** and place the installation in the same new path on all the hosts.

   As part of the installation process, the cluster management software and the cluster file system software are upgraded from the versions that were included in DB2 9.8. When a warning for the cluster manager upgrade displays, click **Continue**. When prompted to set up a DB2 instance, select the **Do not create a DB2 instance** option. On the Host List pane, add the host names of all the hosts that belong to the current DB2 pureScale feature instance.

   Watch for the Add Remote Host window, which might open behind the Host List window. You do not need to change the Cluster Caching Facility or Member options in the Host List window. These options are not used because you do not create an instance with this step.

9. After you complete the binary installation, exit cluster manager maintenance mode by running the following command:

   ```
   db2cluster -cm -exit -maintenance
   ```

   You can run this command, and all commands in the steps that follow, on any host from the new DB2 10.1 installation path.

10. Exit cluster file system maintenance mode by running the following command:

    ```
    db2cluster -cfs -exit -maintenance -all
    ```

11. Run the following command on any host to commit the GPFS cluster changes:

    ```
    db2cluster -cfs -commit
    ```

12. Log on to the DB2 server as the instance owner.

13. Restart the instance on each host by running the following command:

    ```
    db2start instance on <hostname>
    ```

    In this command, *<hostname>* specifies the host on which you want to start the instance. This command does not actually start DB2, but prepares the host for use by DB2. Run this command on any host, and it applies to every host in the DB2 instance. As with the similar command to stop the instance, you can run this command on all hosts.

14. Verify that the databases meet the list of conditions for a successful upgrade by running **db2ckupgrade** from the DB2 10.1 installation directory. The following example shows how to run this command on the `SAMPLE` database:

    ```
    db2ckupgrade SAMPLE -l db2ckupgrade.log -u adminuser -p password
    ```

The `db2ckupgrade.log` file is created in the current directory and includes details about errors and warnings. Review the errors and warnings and take any necessary corrective action. Note that each time you reissue this command, all previous log files are overwritten. You can rename the log files to avoid losing the error details.

15. Stop the DB2 instance, which you started to run the previous command, and then log on to the DB2 server host as the root user.
16. Now, upgrade the existing DB2 9.8 instances by running **db2iupgrade** from the target DB2 10.1 copy location. Run this command from the DB2 10.1 installation path on all hosts. The best approach is to run this command on all of the members first and then run the command on all of the CFs. Use the following command syntax:

    ```
    $DB2DIR/instance/db2iupgrade [ -u fencedID ] InstName
    ```

    Where:

    – $*DB2DIR* refers to the location specified during the DB2 10.1 installation.

    – *fencedID* refers to the user name under which the fenced user-defined functions (UDFs) and stored procedures run.

    – *InstName* refers to the login name of the instance owner.

    Important: If you did not install all DB2 database add-on products that were installed in the DB2 copy from which you are upgrading, the instance upgrade fails and returns a warning message. If you plan to install these products later or if you no longer need the functionality provided by these products, use the **-F** parameter to upgrade the instance.

17. Check to verify that the peer domain was migrated from the previous version to the newly installed version by running **lsrpdomain**. Notice that the online domain shows `No` in the MixedVersions column. If the column shows `Yes`, you need to complete this migration step by running the following command:

    ```
    runact -c IBM.PeerDomain CompleteMigration Options=0
    ```

    If this command runs successfully, the MixedVersions column changes to `No`.
18. Log in to the DB2 database server and start the instance as a user with sufficient authority.
19. Restart the instance on each host by running the following command:

    ```
    db2start instance on <hostname>
    ```

    Then run **db2start**.

20. If you find inconsistencies between the cluster manager resource model and the `db2nodes.cfg` file, repair the cluster manager resources by running the following command:

    ```
    db2cluster -cm -repair -resources
    ```

    When completed, you can verify that your instances are running on DB2 10.1 by running **`db2level`** to ensure that the output string contains a string, such as `DB2 10.1.X.X`, where *X* is a number.

21. Upgrade the database by running the following command:

    ```
    db2 upgrade database <dbname>
    ```

    For a step-by-step description of this process or for additional details about error messages that might arise, see 5.4.4, “Upgrading the database” on page 263. You can now connect to the database by running **`db2 connect to <dbname>`**.

5.8 Post-upgrade steps

This section covers several post-upgrade tasks that you can use to ensure that the DB2 servers, DB2 instances, and DB2 clients are operating on DB2 10.1 and perform as expected.

5.8.1 After upgrading the DB2 servers

After verifying that you are running with DB2 10.1, complete the following tasks on the DB2 servers:

- Verify that the `diaglevel` setting that is defined in the database manager configuration is back to the original value to eliminate any additional diagnosis logging and to allow the system to run faster.
- If you want to use adaptive compression, enable it now. Unless you do so, existing tables that had row compression enabled from a pre DB2 10.1 database now have classic row compression enabled.
- Adjust the log space size for the `logfilsiz`, `logprimary`, and `logsecond` database configuration parameters to their pre-upgrade values. Ensure that the amount of log space that you allocate is adequate for your DB2 server.
- Ensure that existing libraries for your external routines remain at the original location from before the upgrade. If necessary, restore these libraries from the backup that you performed.
- Start your database and all necessary database services.

- If necessary, set the media attributes using the `ALTER STOGROUP` statement. Automatic storage table spaces inherit media attribute values, including the overhead, device read rate, and data tag attributes, from the default storage group. After upgrading to DB2 10.1, the existing table spaces retain their settings and the overhead and device read rate attributes for the storage group are set to undefined.
- Manage any changes in DB2 server behavior caused by the new registry variables and configuration parameters (and their associated default values) that are introduced in DB2 10.1. Each of these settings can affect how the DB2 server operates after the upgrade.
- If automatic collection of statistics failed on certain system catalog tables during the database upgrade, update the statistics on those system catalog tables.
- Rebind packages in the upgraded databases if you did not use the **REBINDALL** option on the **UPGRADE DATABASE** command. Then rebind the packages in the upgraded database to validate the packages and to use the updated statistics or new index information.
- Refresh data in existing materialized query tables (MQTs) by using the `REFRESH TABLE` statement. MQTs on unicode databases using language aware collation, where the MQT definition involves a `LIKE` predicate or substring function involved in a basic predicate, must be refreshed.
- Migrate DB2 explain tables to retain any explain table information that you previously gathered.
- If there were tables with XML columns created in a DB2 9.5 release, convert the XML storage object to the DB2 10.1 format by re-creating the tables so that they have access to functions, such as compression on XML data and collection of statistics, to estimate the inline length for XML columns.
- If you obtained customized code page conversion tables from the DB2 support service, copy all of the files for those tables from the `DB2OLD/conv` directory to the `DB2DIR/conv` directory (where `DB2OLD` is the location of the DB2 9.5 or DB2 9.7 copy and `DB2DIR` is the location of the DB2 10.1 copy). You do not need to copy standard code page conversion tables.
- Use the new **EVMON_UPGRADE_TABLES** procedure to upgrade existing target tables for event monitors that write to tables and to unformatted event (UE) tables.
- Verify that the DB2 server upgrade was successful and test applications and tools.
- Back up databases.

- If you have recoverable databases, the **UPGRADE DATABASE** command renamed all of the log files in the active log path to use the .MIG extension. After verifying that the database upgrade is successful and backing up your databases, you can delete the `S*.MIG` files in the active log path.
- If you have not already done so, migrate SQL Replication to support the new LSN formats.
- If you upgraded a DB2 server that is running high availability disaster recovery (HADR) replication, initialize HADR replication during the upgrade to DB2 10.1. In an HADR replication environment, a database role is changed from primary to standby, and upgrading standby databases is not supported because such databases are in a rollforward pending state.

When your DB2 server performance is stable, update the statistics for your upgraded databases to take advantage of optimizer improvements and collect statistics for new functionality. During a database upgrade to DB2 10.1, the statistics collected from existing database tables retain their values. Statistics for new characteristics on tables and indexes have a value of -1 to indicate that there is no information gathered. However, you need only these statistics if you are using new functionality.

After updating the statistics for your upgraded databases, run the **REORGCHK** command to determine if index or table reorganization is necessary. Table and index reorganization can help improve performance.

You can resume all of your maintenance activities, such as backing up databases and updating statistics. You can also remove any DB2 9.5, DB2 9.7, or DB2 9.8 copies that you no longer need.

5.8.2 After upgrading the DB2 clients

After upgrading clients, complete these post-upgrade tasks to ensure that the clients work as expected:

- Manage any changes in DB2 server behavior by modifying settings as required. Some new configuration parameters might impact the behavior of an application.
- Connect to the cataloged DB2 databases to verify that the upgrade is successful.

5.8.3 After upgrading a database

After you complete a database upgrade as described in 5.4.4, “Upgrading the database” on page 263, compare the database configuration settings with the settings from before the upgrade. Verify the following settings:

- Database configuration parameters
- Table spaces information
- Packages information for applications only (there is no need to check this information for system-generated packages, which can change after an upgrade)

After you verify these settings, verify that your database upgrade is successful by connecting to it and issuing a small query (Example 5-2).

Example 5-2 Simple query to verify a successful database upgrade

```
db2 connect to sample
Database Connection Information
        Database server        = DB2/AIX64 10.1.0
        SQL authorization ID   = TESTDB2
        Local database alias   = SAMPLE
db2 "select * from syscat.dbauth"
```

Alternatively, if you have sample files installed, run the **testdata.db2** script (Example 5-3).

Example 5-3 Running the testdata.db2 script

```
cd <samplefile-dir-clp>
db2 connect to sample
db2 -tvf testdata.db2
```

In this example, `samplefile-dir-clp` is at $*DB2DIR*`/samples/clp` on Linux and UNIX operating systems, $*DB2DIR* represents the location specified during the DB2 10.1 installation, and `sample` is the database name.

Abbreviations and acronyms

10 GbE	Gigabit Ethernet
ACR	automatic client reroute
ACS	Advanced Copy Services
CDC	IBM InfoSphere Change Data Capture
CF	caching facility
CLP	command line processor
CPU	central processing unit
DB	database
DBA	database administrator
DBM	database manager
DR	disaster recovery
ESE	Enterprise Server Edition
ETL	Extract, Transform, and Load
FC	Fibre Channel
FDOC	first occurrence data capture
FP	Fix Pack
GA	generally available
GB	gigabyte
GBP	group buffer pool
GLM	global lock manager
GPFS	General Parallel File System
HADR	high availability and disaster recovery
HCA	Host Channel Adapter
HMC	Hardware Management Console
IOCP	I/O completion ports
ISCSI	Internet SCSI
ISV	independent software vendor
LBAC	Label-based access control
LBP	local buffer pool
LDAP	Lightweight Directory Access Protocol
LLM	local lock manager
LPAR	logical partition
LUN	logical unit number
MDC	multidimensional clustering
MPIO	multipath I/O
MQ	Messaging Queue
MQT	materialized query table
NTP	Network Time Protocol
ODBC	Open Database Connectivity
OFED	OpenFabrics Enterprise Distribution
OS	operating system
PCI	Peripheral Controller Interface
PVID	physical volume identifier
RAID	redundant array of independent disks
RCAC	row and column access control
RCST	IBM Reliable Scalable Cluster Technology
RDMA	Remote Direct Memory Access
RoCE	RDMA over Converged Ethernet
RPO	recovery point objective
RTO	recovery time objective
SAN	storage area network
SAS	Serial Attached SCSI
SCA	shared communications area
SCSI3-PR	Small Computer System Interface 3 - Persistent Reservation

SDDPCM	Subsystem Device Driver Path Control Module
SOA	service-oriented architecture
SP	service pack
SQL	Structured Query Language
SSH	Secure Shell
STMM	Self-tuning Memory Manager
TCP/IP	Transmission Control Protocol / Internet Protocol
TL	Technology Level
uDAPL	User Direct Access Programming Library
UDF	user-defined function
UDT	user-defined type
WLB	workload balancing
WLM	Workload Management
WSE	Workgroup Server Edition
XML	Extensible Markup Language

Related publications

The publications listed in this section are considered suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks publications

The following IBM Redbooks publications provide additional information about the topic in this document. Some publications referenced in this list might be available in softcopy only.

- *IBM BladeCenter HS22 Technical Introduction*, REDP-4538
- *IBM System x3650 M3*, TIPS0805
- *IBM System x3850 X5*, TIPS0817
- *IBM System x3690 X5*, TIPS0818

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- Configuring geographically dispersed DB2 pureScale clusters:

  http://www.ibm.com/developerworks/data/library/long/dm-1104purescalegdpc

- DB2 10 for Linux, UNIX, and Windows Information Center:

  http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/index.jsp

- IBM AIX:

  http://www.ibm.com/systems/power/software/aix

- *IBM BladeCenter HS22 Installation and User's Guide*:

  http://ibm.com/support/entry/portal/docdisplay?lndocid=MIGR-5079689

- IBM Power Systems:
  http://www.ibm.com/systems/power/
- *IBM System x3650 M3 Installation and User's Guide*:
  http://ibm.com/support/entry/portal/docdisplay?lndocid=MIGR-5083595
- *IBM System x3850 X5 Installation and User's Guide*:
  http://ibm.com/support/entry/portal/docdisplay?lndocid=MIGR-5085479
- *IBM System x3690 X5 Installation and User's Guide*:
  http://ibm.com/support/entry/portal/docdisplay?lndocid=MIGR-5085206
- IBM System x Support Portal:
  http://ibm.com/support/entry/portal/
- Red Hat Linux:
  https://www.redhat.com/wapps/store/allProducts.html
- SUSE Linux Enterprise Server:
  http://www.suse.com/products/server/

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

Delivering Continuity and Extreme Capacity with the IBM DB2 pureScale Feature

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

Delivering Continuity and Extreme Capacity with the IBM DB2 pureScale Feature

Describes how to use the IBM DB2 pureScale Feature

Includes information about tuning the DB2 pureScale environment

Provides detailed installation and setup instructions

The IBM DB2 pureScale Feature offers clustering technology that helps deliver high availability and exceptional scalability transparent to applications. The DB2 pureScale Feature helps you to meet your business needs around availability and scalability, and is also easy to configure and administer.

This IBM Redbooks publication addresses the DB2 pureScale Feature that is available in IBM DB2 10.1 for Linux, UNIX, and Windows operating systems. It can help you build skills and deploy the DB2 pureScale Feature. This book bundles all the information necessary for a in-depth analysis into the functions of the DB2 pureScale Feature, including the actual hardware requirements. It includes validated step-by-step hardware and software installation instructions. In addition, this book provides detailed examples about how to work effectively with a DB2 pureScale cluster and how to plan and run an upgrade for all DB2 related components to DB2 10.1.

This book is intended for database administrators (DBAs) who use IBM DB2 10.1 for Linux, UNIX, and Windows operating systems who want to explore and get started with the DB2 pureScale Feature.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-8018-00 ISBN 0738437174