Redpaper

Nidhi Nijhawan

Elastic Dynamic Caching with the IBM WebSphere DataPower XC10 Appliance

This IBM® Redpaper™ publication provides an overview of the IBM WebSphere® DataPower® XC10 Appliance. It describes the problem for which the appliance provides a solution and the architecture and the placement of the XC10 in typical topologies. This paper also provides a comparison of the XC10 and IBM WebSphere eXtreme Scale, which is a software-based offering from IBM for caching needs.

This chapter includes the following topics:

- Overview of the XC10
- Caching scenarios
- Comparing the XC10 with WebSphere eXtreme Scale
- IBM DataPower XC10 V2 additions

Overview of the XC10

Three-tier application topologies that consist of web server, application server, and database tiers are common today. As the load on an application grows, scaling to accommodate this increase is necessary. This scaling is traditionally performed at the web and application server tiers by increasing the number and size of the servers, with load balancing mechanisms to manage the traffic to the servers. The proliferation of servers at the database tier is more complex due to concerns for data integrity. Although you can scale the database server up through the use of additional hardware resources, eventually you will reach the physical and cost limitations and the database becomes a bottleneck for transaction processing.

The XC10 is a powerful distributed cache that accelerates application access to data and services. The XC10 provides a solution for addressing the scaling needs of applications and databases with elastic data grids that allow you to scale out transaction volumes quickly and easily, with minimally invasive changes to the application and architecture, as shown in Figure 1. This approach also drastically reduces reads and writes on the database, cutting back on time and resource-intensive calls that can create bottlenecks.

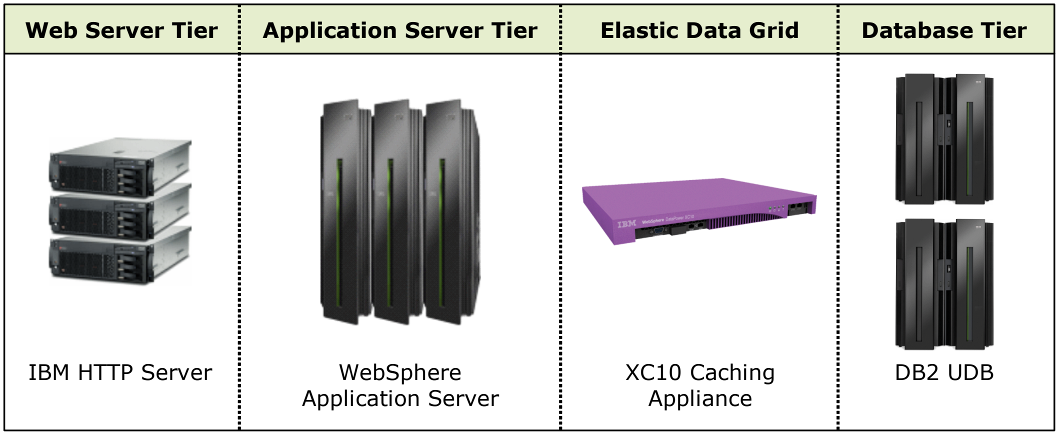

Figure 1 Positioning the XC10

A data grid stores data in a grid style, as illustrated in Figure 2 on page 3, using a large amount of loosely coupled cooperative caches to store data. The XC10 can hold one or more data grids. Each grid is associated with a specific application or set of applications.

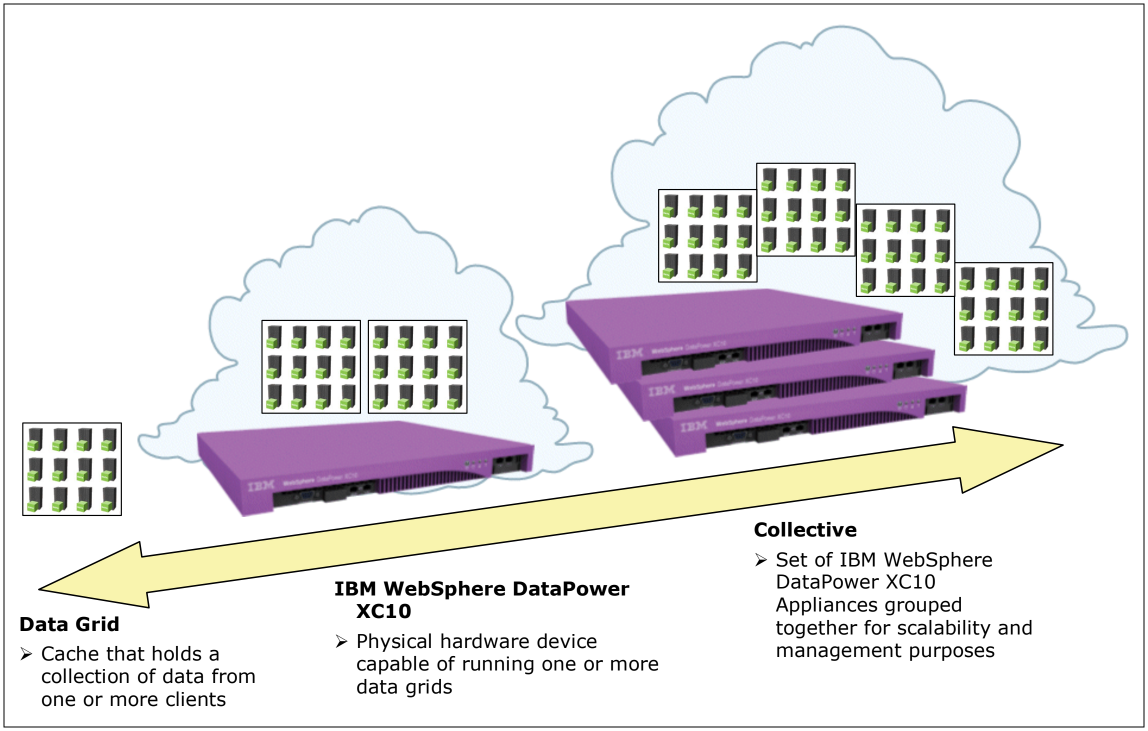

Figure 2 Data grids in the XC10

The XC10 (shown in Figure 3 on page 4) is designed for simplified deployment and hardened security at the caching tier of an enterprise application infrastructure. It contains 240 GB of storage for grid capacity. If this capacity is not enough, you can expand the grid by adding one or more appliances to the configuration, which creates a collective of appliances that host data. This capability of grouping appliances allows you to increase cache capacity and data throughput quickly and easily as your needs grow.

Figure 3 XC10 appliance

The XC10 allows business applications to process billions of transactions per day with efficiency and near-linear scalability. It allows you to take advantage of the value of existing infrastructure investments and to bring higher performance, fault tolerance, and scalability to common, distributed caching scenarios.

With the XC10, you can experience the following benefits:

- Rapid drop-in use without code changes

  The XC10 can be used in a broad range of Java and non-Java application environments where it can deliver a cost-effective, distributed caching solution in support of data-oriented distributed caching.

- Accelerated time to value

  Installation, setup, and configuration time is quick, with ready to use common data-oriented caching scenarios.

- Simplified management and administration

  The XC10 provides a simplified administrative and monitoring console to enable efficient configuration and management. Figure 4 on page 5 shows the simple web GUI of the XC10.

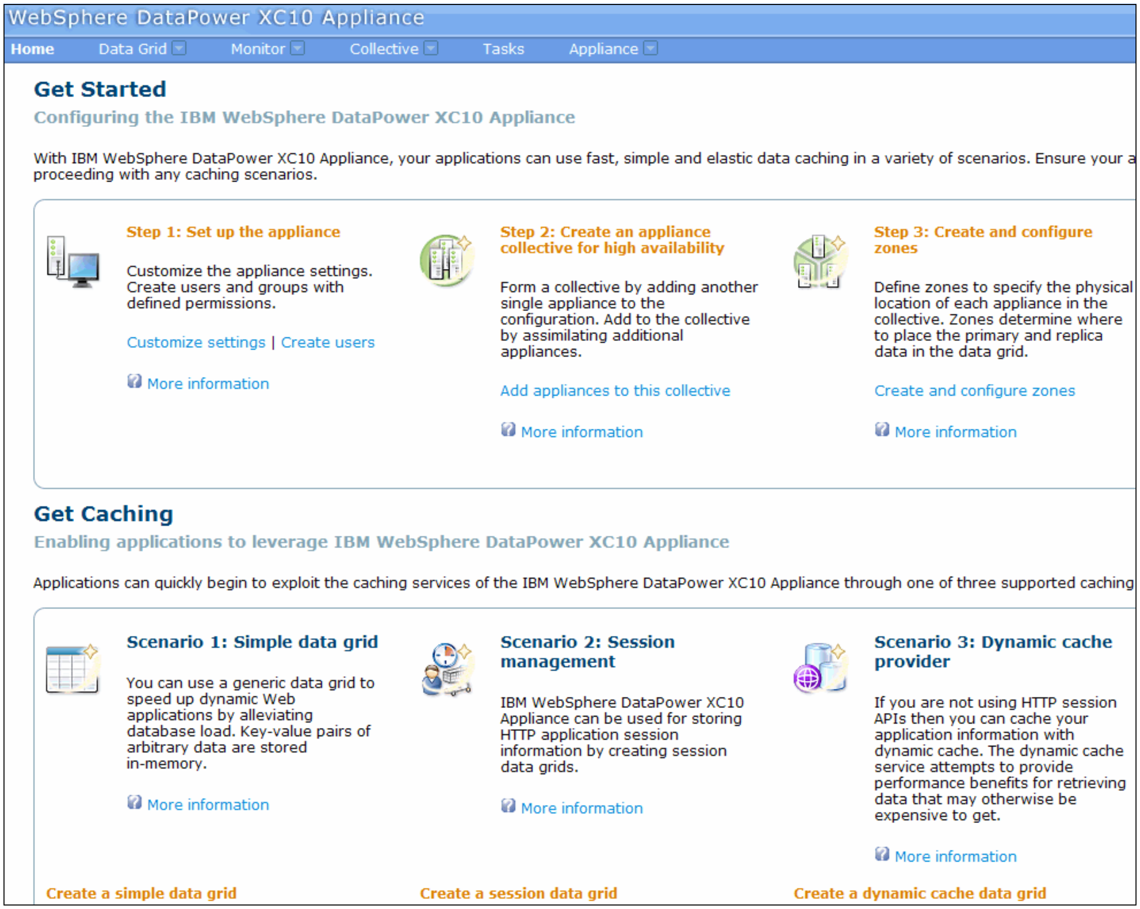

Figure 4 The XC10 web GUI

- Linear scalability

  Scale out without limitations. The XC10 contains a 240 GB elastic data grid that you can use to host data for business-critical applications. To add more memory to a data grid, you can add another appliance to your configuration, creating a collective of appliances that host data.

- Fault tolerance

  The data in data grids is replicated automatically, which lowers the risk of data loss.

Caching scenarios

The scenarios that are described in this section illustrate common cases for using the XC10. These scenarios show both how the scenario is implemented with DataPower caching technology and the reason that the XC10 is the preferred choice in each case.

This section examines the following use cases for using the XC10 as a caching solution:

- Using the XC10 as a side cache
- Using the XC10 for HTTP session management
- Using the XC10 for elastic dynamic caching

Using the XC10 as a side cache

It is common for applications to make redundant calls, doing something over and over again on expensive back-end systems, often to access data that does not change much (for example, user profiles). Applications that fit this description can benefit from using a simple data grid in the XC10 as a *side cache*. In this scenario, the application checks the cache every time that data is needed. If the value is not found (cache *miss*), the data is retrieved from the back-end database and inserted into the cache. The next time that the data is needed, it is retrieved from the cache.

Simple data grids can be used with WebSphere Application Server or with a stand-alone Java application. In both cases, the WebSphere eXtreme Scale client is installed on the Java virtual machine (JVM) to enable access to the XC10.

The application uses the ObjectMap API to store key-value pairs of data in memory, reducing expensive database queries. The keys can be any existing Java type, such as `java.lang.String` or `Integer`. The values can be any serialized object type. Every time that data is needed, the simple data grid on the appliance is checked first. If the appliance does not have the data, the data is retrieved from the database and inserted into the simple data grid.

As an example of this type of scenario, consider a shopping website where potential customers can view information about furniture items that are available. The information about each furniture item is stored in a database and is updated on a regular basis with price changes, new pictures, and offers. These updates are infrequent, but users viewing the site need to know about the status of the furniture item availability.

Accessing the database with every request has proven to be expensive in terms of response time. When the volume of requests is high, the database becomes a bottleneck, reducing performance significantly. Use of the site and purchases dropped as these problems grew. The application was modified to use the XC10 as a simple cache. When a user requests information about a furniture item, the data is retrieved from the database and is cached for future retrievals. Caching the data significantly improved performance and provided the opportunity for future growth.

Figure 5 shows the web GUI that is used to configure a simple data grid in the XC10 appliance.

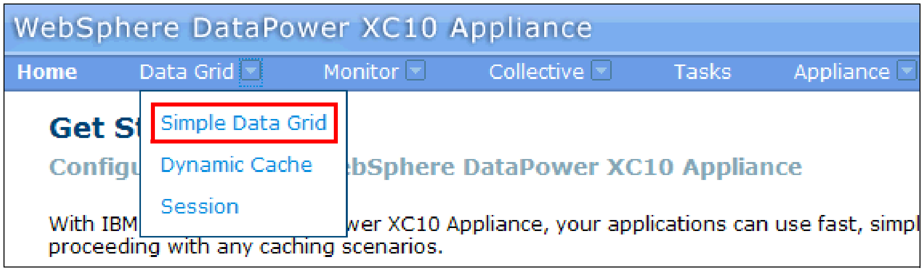

Figure 5 Configuring a simple data grid in the XC10

XC10 REST Gateway feature

The Representational State Transfer (REST) Gateway feature of XC10 enables non-Java clients to access simple data grids. It is supported on the XC10 firmware V1.0.0.4 or later. This feature allows integration with the following components:

- IBM WebSphere DataPower XI50 appliances
- Microsoft .NET applications
- Clients that cannot host the IBM Object Request Broker (ORB)

XC10 as a side cache for XI50

Using the XC10 REST Gateway feature, you can integrate the elastic caching tier with the IBM WebSphere DataPower Integration Appliance XI50. The DataPower XI50 is a hardware appliance that provides enterprise service bus (ESB) features. Traditionally, the elastic caching tier is inserted between the application server tier and the database tier. However, in a configuration that includes an XI50, the elastic caching tier acts as a side cache for the ESB.

A simple data grid that is hosted on the XC10 functions as an service-oriented architecture (SOA) results cache for the ESB. In an SOA, all application requests pass through the ESB before they are routed to the application. Therefore, if the result of an application request is retrieved from the elastic caching tier, the application processing and processing latency for that request are eliminated. The result is a significant decrease in the response time and a reduction in the application processing.

As incoming client application requests are received, the XML proxy configured on the XI50 inspects the URI, the XML body contents, or both to determine whether the request meets the criteria for being cached, based on the caching policy rules that are defined in the XI50. If the request is cacheable, the XML proxy performs a standard side-cache operation. Using the REST-based HTTP GET method, the XML proxy determines whether the request is cached in the simple data grid.

If the HTTP GET returns an `HTTP 404 NOT FOUND` error, which means a cache miss, the XML proxy allows the request to pass through the existing processing flow to the application that is hosted in the back-end systems. The XML proxy then caches the response as it flows back through the XML proxy to the client application. The XML proxy uses the REST-based HTTP POST method to insert the response into the side cache. If the incoming request is found in the cache, the result is retrieved from the cache, bypassing the back-end systems and removing the latency that was introduced by the application and data layers.

Figure 6 illustrates the use of the XC10 as a side cache for the XI50.

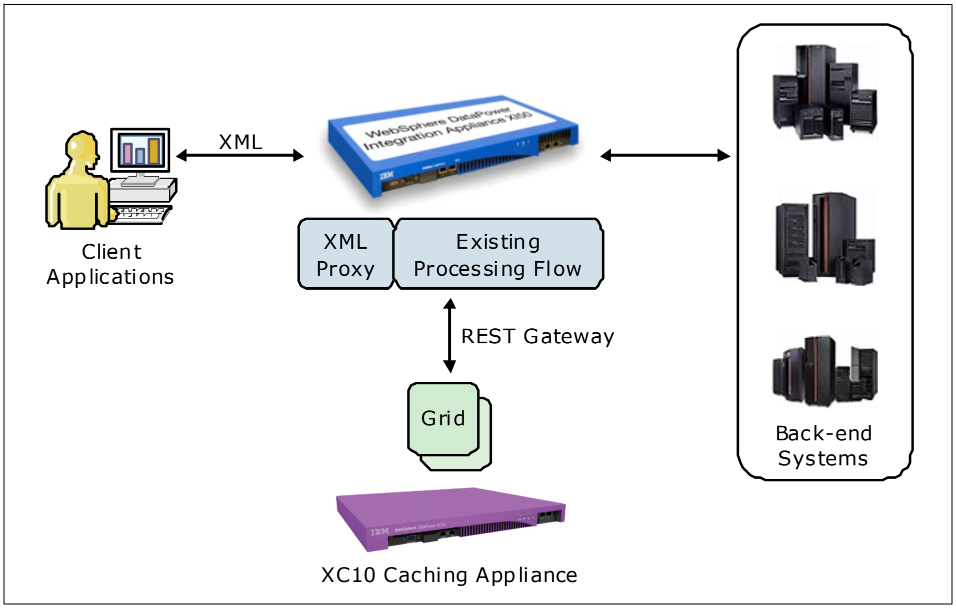

Figure 6 The XC10 as a side cache for the XI50

Using the XC10 for HTTP session management

The HTTP session state management features that are provided by Java Platform, Enterprise Edition (Java EE) application servers are used by many applications. *Session state management* allows the application to maintain user information and the state of the user session for the period of a user's interaction with the application. When an application serving environment supports high availability and failover, session state information must be available to all application servers. Typically, the application servers achieve high availability and failover of session data by storing the state information in a shared database or in the application server's memory.

When session data is stored in a shared database, scalability is limited by the database server. The disk I/O operations are an expensive performance cost to the application. When the transaction rate exceeds the capacity of the database server, the database server must be scaled up.

When session data is kept in memory, the session data is replicated between application servers to keep the data in sync. In this case, the limits to scalability vary depending on the replication scheme. Commonly, a simple scheme is used in which each application server holds a copy of all user session data. In this scheme, the total amount of state information cannot exceed the available memory of any single application server. Memory-to-memory replication schemes often trade consistency for performance, which means that in cases of application server failure or user sessions being routed to multiple application servers, the user experience might be inconsistent and confusing.

Installing the WebSphere eXtreme Scale Client in a WebSphere Application Server configuration provides an extension beyond the database and in-memory HTTP session management caching mechanisms to support XC10 session management caching. With the XC10, session data is kept in the grid, providing high-speed access to the data from all application servers. If one server goes down, the session can be continued on another server with the session data kept intact.

The XC10 provides a quick and non-invasive option for handling HTTP session management by minimizing the need for costly application changes. The XC10 can provide a number of benefits, including higher qualities of service across scenarios that span application server cell boundaries and heterogeneous application server environments.

You can create the cache using one of the following methods:

- Create the cache on the appliance, and then point WebSphere Application Server to the cache.
- Create the cache on the appliance directly from the WebSphere Application Server administrative console.

As a practical example of this type of scenario, consider an insurance company that provides many types of insurance, including life, health, home, auto, and so on. It has an Internet site that allows customers to get a price quote on auto or home insurance. As potential customers use the site, they navigate through a series of pages, enter information about the assets that they want to insure, and make selections regarding coverage. This information is collected and stored by the web application as session data to be used to create the price quote. At the end of the process, the user reviews the information and submits it for processing.

In this scenario, user performance is critical, because users tend to become frustrated with slow movement between pages. Although the default in-memory session management that was used by the application serving environment worked, the replication that was required to keep the data synchronized for availability purposes put a load on the application servers and caused performance delays for the user. In addition, the accumulation of session data due to the length of time that it took users to complete the application became a burden on the memory of the application servers.

By moving this session data to a collective of XC10 appliances, several critical improvements occurred. The response times to the user became consistent and faster. Memory is freed for executing the application rather than storing data. Synchronization of data between the application servers is no longer necessary, which reduced the load on the application servers. Reliability was also improved. The collective contained replicas of the data so that if an application server went down, the data was not lost with it. The remaining servers continued the session without losing the data.

Figure 7 show the use of an XC10 collective for HTTP session management.

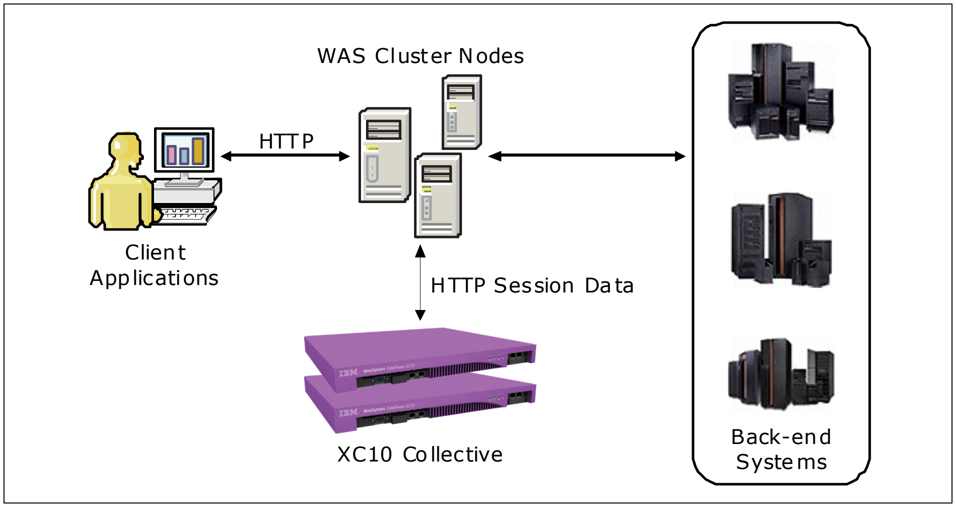

Figure 7 HTTP session management using an XC10 collective

Using the XC10 for elastic dynamic caching

WebSphere Application Server provides a dynamic cache service that allows the container-level caching of the output of servlets, JavaServer Pages (JSP), web services, and WebSphere Application Server commands into memory. Java EE applications that are deployed in WebSphere Application Server also have access to the dynamic cache through the Dynamic Cache API, which is informally known as *DynaCache*. The dynamic cache service uses the default dynamic cache provider of WebSphere Application Server. The default provider stores the cached data in the JVM memory with the option of offloading cached data to disk in an overflow situation. The cached data is replicated among the other JVMs in the application server cluster.

As an alternative, you can configure the XC10 as the dynamic cache provider for WebSphere Application Server (Figure 8 on page 12). By setting up this capability, you can enable applications that are written with the Dynamic Cache API or applications using container-level caching to use the features and performance capabilities of the appliance, including replication over the network, high availability, scalability, and cache partitioning. The memory that was being used for caching in application servers can now be used for other purposes. The data is cached and managed in the grid, rather than having the multiple redundant synchronized copies stored by the default dynamic cache provider.

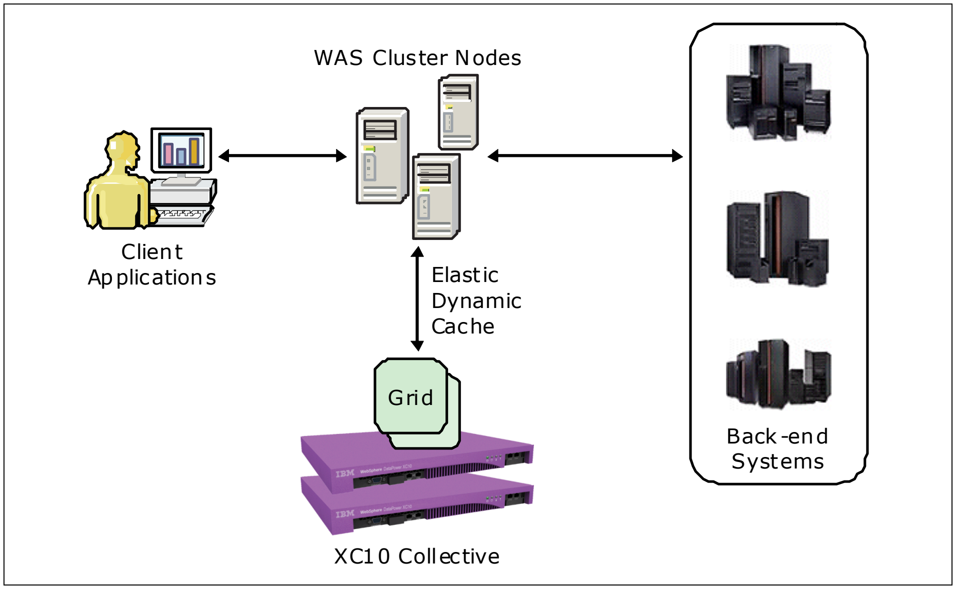

Figure 8 Elastic Dynamic Cache in XC10

As a practical example of this type of scenario, consider a large airline that posts information about current and future flights on its website. The site is used by customers who are looking for available flight information, by passengers checking to see whether flights are delayed or canceled, and by others checking for flight arrival information. Users can view a list of flights by date, select a specific flight for more information, or select multiple flights for comparison.

The flight information is stored in a database and is accessed by the web application that is deployed on WebSphere Application Server. Although the information about flights for the current day tends to change frequently, the data for future flights is fairly static. The high volume of requests, however, puts a significant load on the application servers, increasing response times and making the performance erratic.

Presently, dynamic caching is used to store the data that is accessed frequently but that is changed infrequently, thus reducing the time and resources that are required to build the web pages. However, the data is stored in the JVM memory, reducing the amount of memory resources available to the application.

By introducing a collective of XC10 appliances, the company can offload the memory and computing resources that are required to maintain the cache from the application servers, increasing both throughput and performance. The XC10 provides a consistent, distributed cache for the airline enterprise application that is running on WebSphere Application Server.

Figure 9 shows that each record is stored once in the grid and shared by all clients.

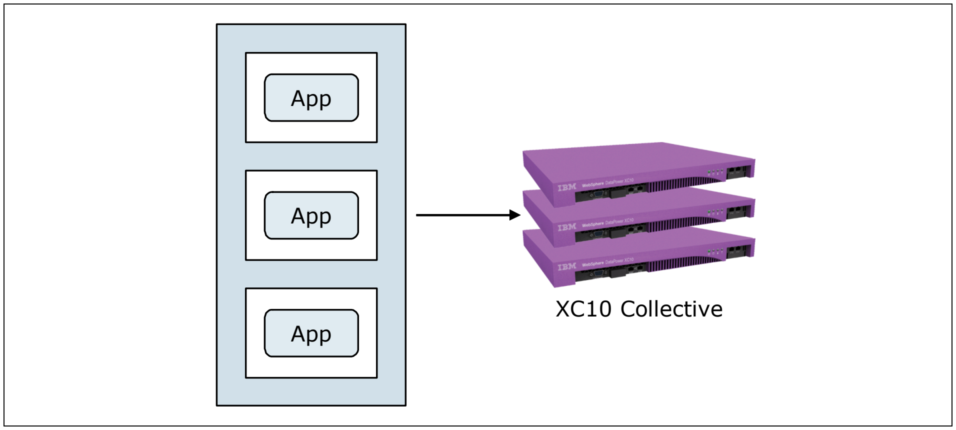

Figure 9 Using an XC10 collective for dynamic caching

Comparing the XC10 with WebSphere eXtreme Scale

The IBM WebSphere caching family includes the following products:

- WebSphere eXtreme Scale (software)
- The XC10 appliance (hardware)

Both products allow business applications to process billions of transactions per day with extreme efficiency and near-linear scalability. They are designed to work in heterogeneous environments across all leading application server platforms and virtualization environments.

WebSphere eXtreme Scale operates as an in-memory grid that dynamically processes, partitions, replicates, and manages application data and business logic across hundreds of servers. It provides transactional integrity and transparent failover to ensure high availability, high reliability, and consistent response times. WebSphere eXtreme Scale provides the technology to enhance business applications by extending the data-caching concept with advanced features.

The XC10 appliance is designed for simplified deployment and hardened security at the caching tier of the enterprise application infrastructure, adding elastic caching functions that enable your business-critical applications to scale cost effectively with consistent performance. IBM WebSphere DataPower XC10 is designed for rapid, “drop-in” use in conjunction with WebSphere Application Server and other WebSphere family products where it can deliver a cost-effective, distributed caching solution in support of data-oriented, distributed caching scenarios.

Both WebSphere eXtreme Scale and the XC10 appliance provide distributed object caching. However, the set of features that they offer differs. The XC10 appliance stores cached data in the appliance, and WebSphere eXtreme Scale builds grids using JVMs. In addition, whereas the XC10 appliance is designed for the three data-oriented caching solutions (dynamic cache, session management, and simple side cache), WebSphere eXtreme Scale supports these scenarios and offers additional flexibility for application-oriented caching and how the cache is loaded, as illustrated in Figure 10.

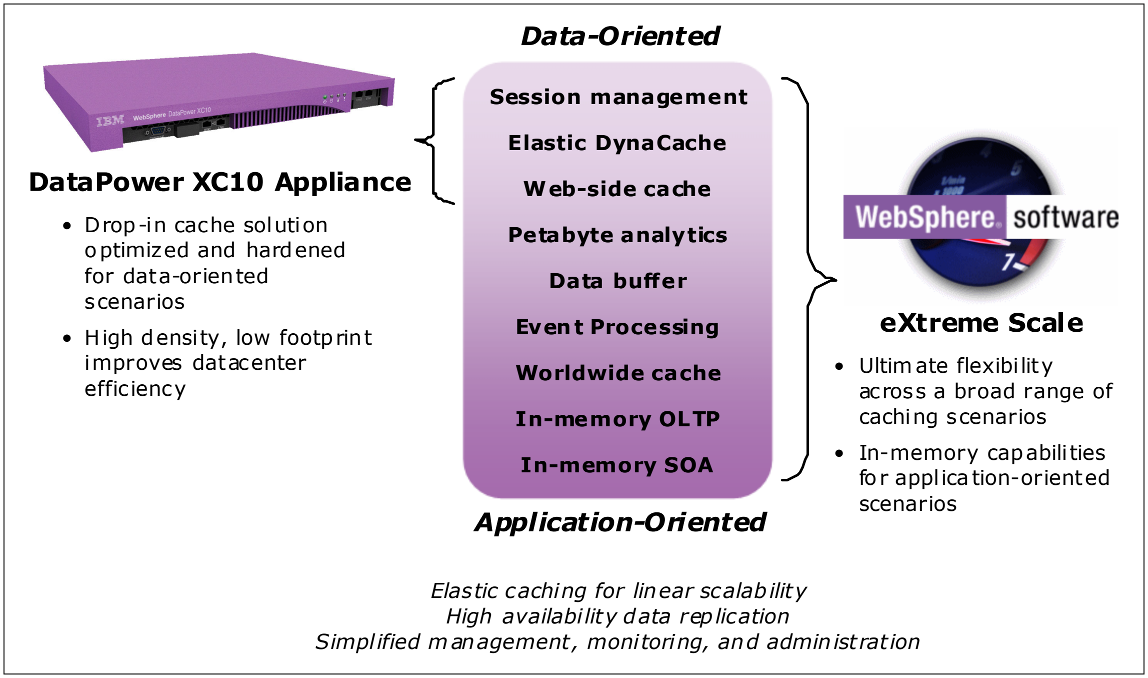

Figure 10 The XC10 compared to WebSphere eXtreme Scale

The following list compares the features of the XC10 and WebSphere eXtreme Scale:

- Reporting and monitoring

  WebSphere eXtreme Scale includes implementations of metric access adapters to improve integration with IBM Tivoli® Monitoring or Hyperic HQ, enabling comprehensive insight into the operational behavior of business solutions.

  The XC10 includes a built-in console for ease of management and metric tracking.

- Simplified monitoring of the run time and health of the appliance (the XC10 appliance only)

  The XC10 includes status widgets to report key metrics pertaining to your transaction load and memory. Two examples of the reported metrics are memory/disk usage and average response time.

- Real-time data and event mining (WebSphere eXtreme Scale only)

  When working with real-time data flows, the first challenge is filtering and organizing the data so that the applications can use it. A partitioned WebSphere eXtreme Scale configuration can subscribe to events and apply them to partitioned data, thus supporting near-linear scalability and deterministic latency for these applications.

- Map/reduce support (WebSphere eXtreme Scale only)

  WebSphere eXtreme Scale clients can invoke agents that run against massive amounts of data on multiple nodes in parallel. Clients can then aggregate and further process the results that are stored in the grid by the nodes. WebSphere eXtreme Scale caches can potentially span thousands of JVMs and support extremely large data sets.

  In addition, WebSphere eXtreme Scale includes efficient new algorithms to allow in-memory caches to grow elastically as the number of available JVMs or physical machines changes. Traditional distributed cache products use Map APIs as their primary programming model. WebSphere eXtreme Scale offers Map APIs and also allows graphs of objects to be easily pushed to the cache.

  WebSphere eXtreme Scale allows simple Java objects to be annotated and uses a simpler API to transparently allow these graphs to be fetched from the grid and to push any changes made by the application back to the grid. This design significantly simplifies programming compared with the older JCache or map-based APIs, improving the productivity of the application developers by allowing them to focus on the core business logic.

- Client/grid with near cache (WebSphere eXtreme Scale only)

  Having the data in memory is a proven step to higher performance. When dealing with large volumes of data, applications can perform even better. A JVM can have a local WebSphere eXtreme Scale grid, which sits in front of a remote grid serving as a “near cache” for a subset of the data, allowing a client to use a large remote cache to offload back-end processing or to accelerate access to cached results.

  The near cache is in the same JVM as the application and provides local, in-process access to data. It contains a subset of all the data in the grid and is checked first when a record is requested. If the record is not in the near cache, the record is retrieved from the grid and put into the near cache. The response time is reduced the next time that the same record is accessed. The faster response times for the records that you access often lead to faster response time for the user. The near cache is also updated when data writes to the grid. Applications can use the distributed locking services that are provided by the remote grid to coordinate access to shared data across clients.

- Accelerated Time to Value (the XC10 appliance only)

  The XC10 reduces the time that is necessary for installation, setup, and configuration, with “drop-in” use for the HTTP session replication and WebSphere Application Server dynamic cache service.

- Simplified management and administration (the XC10 appliance only)

  The XC10 offers a built-in, simplified administrative and monitoring console to enable the efficient setup, configuration, and management of the appliance and transaction load within your data center.

- Write-through caching (WebSphere eXtreme Scale only)

  Write-through caching immediately propagates all changes from the in-memory cache to the back-end database as part of the transaction. This method results in longer response times but guarantees all changes are persisted to the back-end database. This approach also provides synchronization between the cache and the back-end system. This caching is valuable in situations where the data must be hardened to the back-end data store for the transactions to be considered complete.

- Write-behind caching (WebSphere eXtreme Scale only)

  Write-behind caching batches data updates, sending them to the back-end data store at a configured interval. This caching improves transaction response times, because they no longer need to deal with the back-end data store in a synchronous fashion. This approach reduces the load on the data store and shields the application from back-end outages. The grid holds the updates in memory in a fault-tolerant manner until the back end comes back online.

- Java Persistence API (JPA) loaders (WebSphere eXtreme Scale only)

  Loaders are needed to read and write data from the data store when using the WebSphere caching family product as an in-memory cache. Starting with WebSphere eXtreme Scale V6.1.0.3, there are two built-in loaders that interact with JPA providers to map relational data to the WebSphere eXtreme Scale maps: the JPLoader and JPAEntityLoader. The JPALoader is used for caches that store Plain Old Java Objects (POJOs), and the JPAEntityLoader is used for caches that store WebSphere eXtreme Scale entities. These loaders reduce the burden on the application programmer.

- Multimaster replication (WebSphere eXtreme Scale V7.1 only)

  This feature allows you to host a grid in multiple locations that connect through user-defined links. Each grid is fully independent and runs its own catalog service. The locations need to have the same grid that is defined with the same number of partitions and map/template definitions.

  You can create a link between two locations and, from that point forward, WebSphere eXtreme Scale attempts to make both locations identical. If you create a link between a location with data and another empty location, WebSphere eXtreme Scale copies the data from the non-empty location to the empty location automatically. These replication links are bidirectional. Changes made on either side of the link are propagated to the other side. In addition to the simple topology of two locations, more complex topologies can be constructed.

IBM DataPower XC10 V2 additions

The following features are available in the IBM DataPower XC10 V2:

- REST Gateway
- Simple Network Management Protocol (SNMP) monitoring support
- Integration with other products:
  - WebSphere Commerce
  - WebSphere Portal
  - WebSphere Enterprise Service Bus
  - XI50 (XC10 appliance as a side cache for XI50)

REST Gateway

The 1.0.0.4. release of the WebSphere DataPower XC10 firmware introduces a REST Gateway that provides non-Java-based clients access to simple data grids using a set of HTTP-based operations. This new feature expands the range of clients capable of utilizing the XC10 appliance for elastic caching to any client with HTTP capabilities, including PHP and .NET clients. Using the REST Gateway feature, the XC10 can be used as an SOA results side cache for the WebSphere DataPower Integration Appliance XI50. Using the XC10's simple data grid as a side cache for an XI50 can significantly reduce the load on the back-end systems by eliminating redundant requests to the back-end systems, improve the response time to the clients, and increase the total system throughput.

SNMP monitoring

Simple Network Management Protocol is commonly known as SNMP. SNMP is a User Datagram Protocol (UDP)-based network protocol that is commonly used to communicate with hardware devices on a computer network. SNMP provides a mechanism for monitoring hardware devices, and altering their configurations by requesting information from a service, which is called an agent, that is running on the hardware and sending the agent requests to alter the hardware's configuration. Hardware devices that are commonly monitored and managed using SNMP include computer hosts, routers, switches, IP telephones, and network printers. Using an SNMP client to communicate with the hardware's SNMP agent, information about the current state of the hardware can be determined. Based on this information, requests can be sent to the device using SNMP to alter its configuration.

WebSphere DataPower XC10 V2.0 has a configurable SNMP agent. The SNMP agent supports SNMPv1 and SNMPv2c specifications. System administrators can configure a network management system to monitor WebSphere DataPower XC10 using Management Information Base (MIB) files that are downloaded from the appliance web console.

WebSphere DataPower XC10's SNMP agent runs on the appliance as a daemon. The port number is configurable, and system administrators that configure a network management system to monitor WebSphere DataPower XC10 can access the MIB files from the appliance console.

DataPower XC10 as a side cache for XI50

Support for DataPower XC10 as a side cache for XI50 was added in V2.

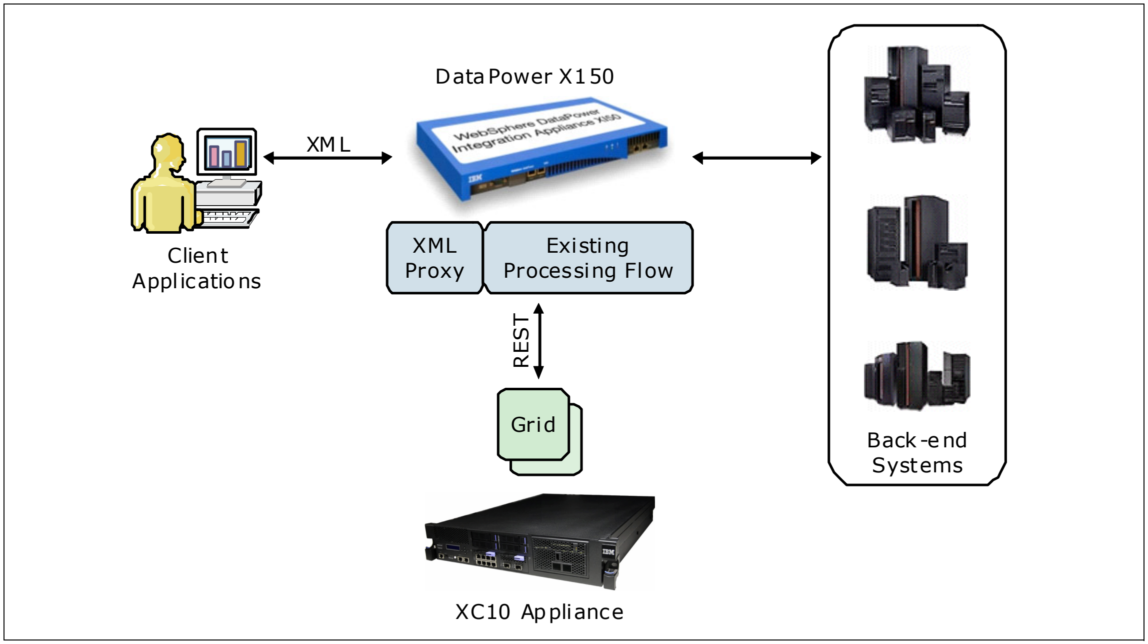

Figure 11   XC10 as a side cache to XI50

Figure 11 shows the high-level design of the XI50 using the REST APIs to use XC10 simple data grids as an SOA results side cache. As incoming client application requests are received, the XML proxy inspects the URL and the XML body contents to determine if the request meets the criteria for being cached, based on the caching policy rules. If the request is cacheable, the XML proxy will perform a standard side cache operation. Using the REST-based HTTP GET method, the XML proxy looks to see if the request is cached in the simple data grid. If the HTTP GET returns an `HTTP 404 NOT FOUND` error, signifying a cache miss, the XML proxy allows the request to pass through to the existing processing flow to the application hosted in the back-end systems. The XML proxy uses the REST-based HTTP POST method to insert the request into the request cache. The XML proxy also caches the result to the result cache as it flows back through the XML proxy to the client application. If the incoming request was found in the request cache, the result is retrieved from the result cache, bypassing the back-end systems, thus removing the latency introduced by the application and data layers.

WebSphere Commerce integration with DataPower XC10

WebSphere Commerce can use the IBM WebSphere DataPower XC10 Appliance to store data from WebSphere Application Server dynamic cache instead of caching the data in local memory. This setup allows for lower local memory requirements, enabling the WebSphere Commerce JVMs to run more efficiently. Because all data is cached in the collective, individual WebSphere Commerce JVMs are less likely to have “stale” data, and they do not have to call back-end systems that generate the data as often, therefore reducing the load. This setup also allows for the high availability of cache data and improved performance.

WebSphere Portal integration with DataPower XC10

The integration of WebSphere Portal with DataPower XC10 provides multiple benefits to portal customers. DataPower XC10 appliances can be used to offload HTTP sessions to the grid, which otherwise are stored and persisted either in memory or in a shared database. The integration of WebSphere Portal with DataPower XC10 is simple and helps reduce response times, thus increasing the throughput of the application. Replication across the data grid provides high availability, and moving the session replication off of the database reduces the load on the database. If session data is currently being held in the JVM memory, introducing the DataPower XC10 frees up the JVM heap for other application data.

WebSphere Enterprise Service Bus integration with the XC10

WebSphere Business Process Management and Connectivity products integrate with back-end systems, such as IBM CICS®, web services, databases, or JMS topics and queues. You can add the WebSphere DataPower XC10 to the configuration to cache the output of these back-end systems, increasing the overall performance of your configuration.You can integrate WebSphere DataPower into your configuration without changing the business process by using the mediation flows that are provided by WebSphere Enterprise Service Bus.

The IBM WebSphere DataPower XC10 Appliance is an easy-to-use caching appliance. It provides simplified deployment at the caching tier of the enterprise application infrastructure. DataPower XC10 client code is provided. It easily integrates non-intrusively into existing foundational ESB applications, whether running on a WebSphere Application Server or running stand-alone applications.

For more information

For more information, consult the following resources:

- Integrating WebSphere Portal with DataPower XC10

  http://www.redbooks.ibm.com/abstracts/redp4822.html?Open

- Integrating WebSphere Commerce with IBM WebSphere DataPower XC10

  http://www.redbooks.ibm.com/abstracts/redp4823.html?Open

- Innovations within reach: Enhance your ESB with the REST Gateway feature in the WebSphere DataPower XC10 Appliance

  http://www.ibm.com/developerworks/websphere/techjournal/1102_inreach/1102_inreach.html?ca=drs-

- Using the WebSphere DataPower XC10 Appliance as a side cache for the WebSphere DataPower XI50 Integration Appliance

  https://www.ibm.com/developerworks/wikis/display/extremescale/Using+the+WebSphere+DataPower+XC10+Appliance+as+a+side+cache+for+the+WebSphere+DataPower+XI50+Integration+Appliance

The author

This paper was produced by a specialist working at the International Technical Support Organization, Raleigh Center.

Nidhi Nijhawan is an IBM Certified IT Specialist working with the WebSphere Worldwide Business Partner Team as a technical sales specialist. She has eight years of experience in IT. She has worked with IBM for the last five years in the role of an EAI Consultant handling DataPower SOA Appliances, WebSphere Transformation Extender, and Business Process Management (BPM). She is a member of TEC India (Academy Of Technology Affiliates) and actively contributes to IBM developerWorks® and other forums.

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

Margaret Ticknor, Carla Sadtler, Tamikia Barrow, Debbie Willmschen, Linda Robinson
International Technical Support Organization, Raleigh Center

Chao M Beck
IBM US

Cynthia Glass
IBM US

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

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 IBM 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

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 illustrates 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. 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 IBM's application programming interfaces.

This document REDP-4851-00 was created or updated on March 14, 2012.

Send us your comments 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:
  redbook@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 U.S.A.

Redpaper™

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

CICS®	IBM®	Redbooks (logo) ®
DataPower®	Redbooks®	Tivoli®
developerWorks®	Redpaper™	WebSphere®

The following terms are trademarks of other companies:

Microsoft, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other company, product, or service names may be trademarks or service marks of others.