Elastic Caching with IBM WebSphere eXtreme Scale

IBM Redbooks Solution Guide

IBM® WebSphere® eXtreme Scale provides an extensible framework to simplify the caching of data that is used by an application. It can be used to build a highly scalable, fault-tolerant data grid with virtually unlimited scaling capabilities beyond *terabyte* capacity. Because its capacity can be dynamically increased to an extreme size, it can be thought of as an *elastic cache*.

Figure 1 illustrates many business and application challenges that suggest an elastic caching solution. WebSphere eXtreme Scale enables infrastructure with the ability to deal with extreme levels of data processing and performance. When the data and resulting transactions experience incremental or exponential growth, the business performance does not suffer because the grid is easily extended by adding more capacity.

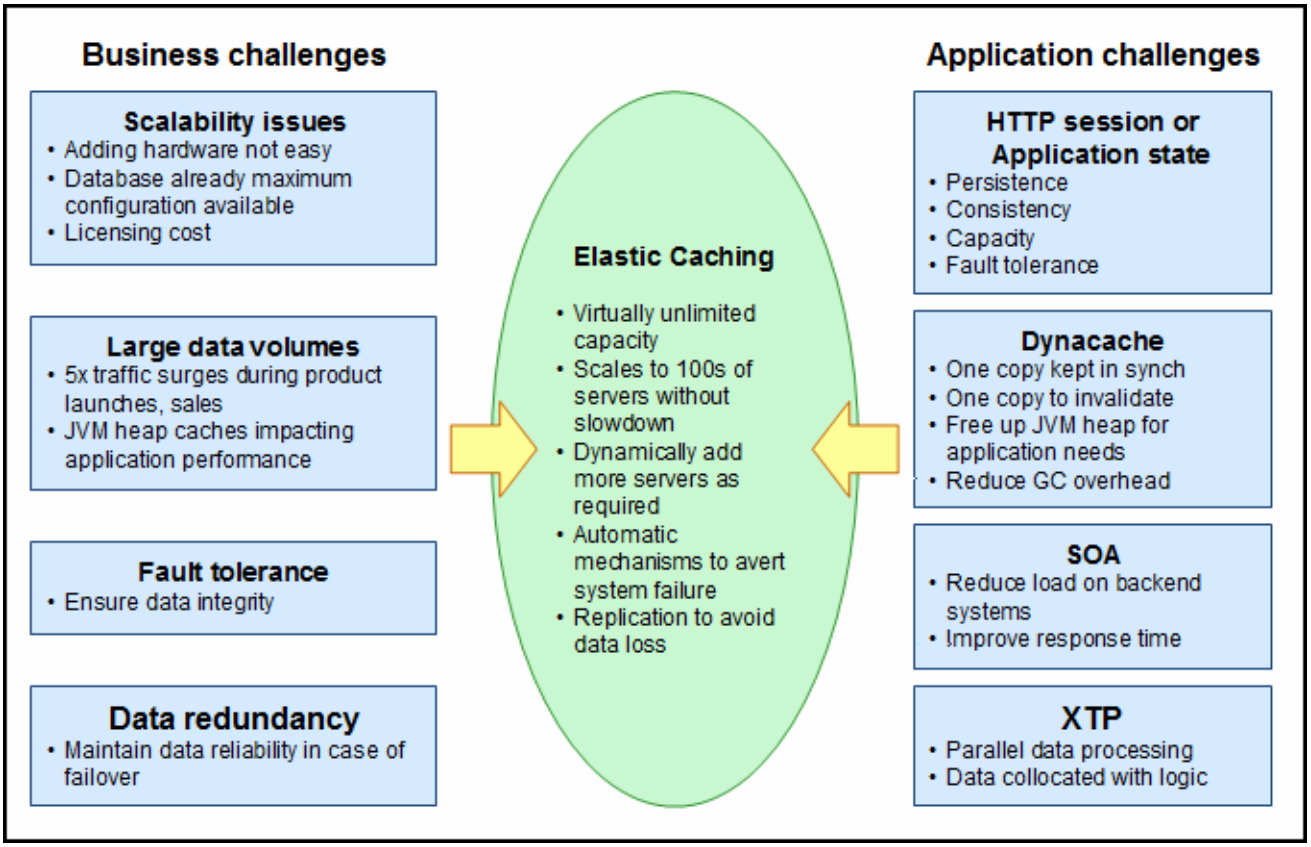

Figure 1. Challenges that lead to an elastic caching solution

Did you know?

Much of the functionality that is provided by the WebSphere eXtreme Scale software product is also available in an easy-to-install “appliance” called the IBM WebSphere DataPower® XC10 Appliance. The key point to consider about the WebSphere DataPower XC10 Appliance is that, as a pre-configured and secure ready-to-go “black box”, it does not allow you to install any software onto it. You can use its administration console to configure it, but you cannot install any of your own software on it. This situation means that you cannot use some of the more advanced features that you might want to use WebSphere eXtreme Scale for, such as in-line loaders, agents, or parallel grid processing. Each of these advanced features requires that your unique code is available on the grid side.

Business value

WebSphere eXtreme Scale is an essential tool for elastic scalability and provides the following valuable benefits:

- Processes massive volumes of transactions with extreme efficiency and linear scalability.
- Rapidly builds a seamless, flexible, and highly available elastic grid that scales out as applications scale, removing the performance limits of the database.
- Provides high availability and security with redundant copies of cache data, and authentication schemas that help ensure system security.
- Enables your existing back-end systems to support significantly more applications, reducing your total cost of ownership (TCO).

WebSphere eXtreme Scale provides an extensible framework to simplify the caching of data that is used by an application. It can be used to build a highly scalable, fault-tolerant data grid with nearly unlimited horizontal scaling capabilities. WebSphere eXtreme Scale creates infrastructure that can deal with extreme levels of data processing and performance. When the data and resulting transactions experience incremental or exponential growth, the business performance does not suffer because the grid is easily extended by adding additional capacity in the form of Java virtual machines and hardware.

The key features of WebSphere eXtreme Scale include the following ones:

- A highly available elastic and scalable grid
- Extreme transaction support
- Security
- Easy integration into existing solutions
- Monitoring
- Support for JSE, Java Platform, Enterprise Edition, ADO.NET data services, and REST capable client applications

Solution overview

Scalability is the ability of a system to handle an increasing load in a graceful manner. This ability implies that a system can be readily extended. A system has perfectly linear scaling capabilities if doubling its processor capacity also doubles its maximum throughput. In general, there are two ways an IT system can be scaled:

- Horizontally, by adding more hosts to a tier. This is called *scale out*.
- Vertically, by enlarging the capabilities of a single system, for example, adding processors. This is called *scale up*.

Consider a classical three-tier application, such as the one shown in Figure 2.

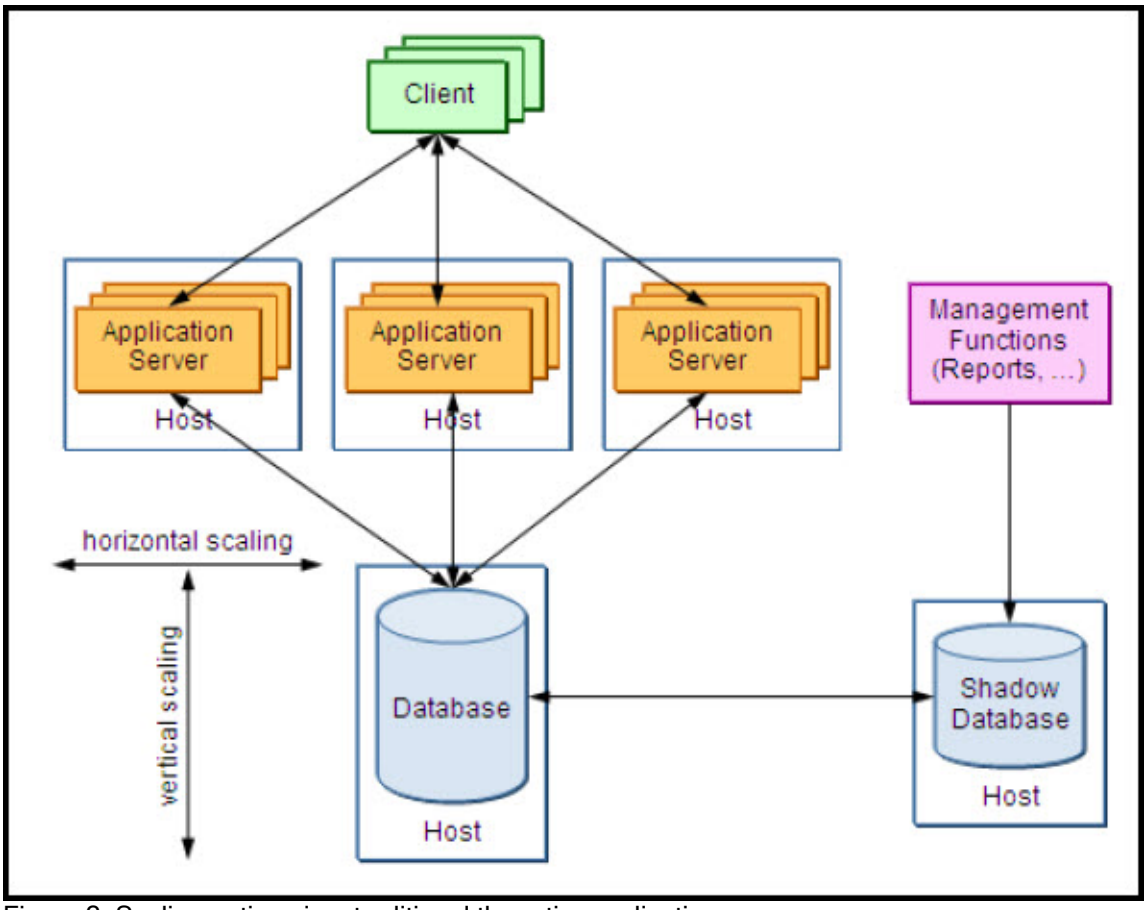

Figure 2. Scaling options in a traditional three-tier application

The application server tier is both scaled out by having three hosts and scaled up by having three application servers on each host. The database tier is scaled up by using a single powerful system with many processors. The database tier is scaled out by having a shadow database that uses log shipping capability to support reports, analysis, and so on.

Scaling is successful if all involved resources can cope with the increased load indefinitely. But at some point a resource reaches its maximum throughput, limiting the overall throughput of the system. This point is called the *saturation point,* and the limiting resource is called a *bottleneck resource*.

The obvious approach to solve the scalability challenge is to reduce the number of requests that are made to the bottlenecked resource or resources. Often, the most practical way to accomplish this task is by introducing a *cache*. A cache can be defined as a copy of frequently accessed data that is held in a relatively nearby location, such as within process memory. The intent of any caching mechanism is to reduce response time by reducing access time to data, and to increase scalability by reducing the number of requests to a constrained resource.

Virtually every major business application incorporates some caching techniques. A frequent example is a database cache, as illustrated in Figure 3. Here, the cache sits between the application and the database to reduce the load on the database.

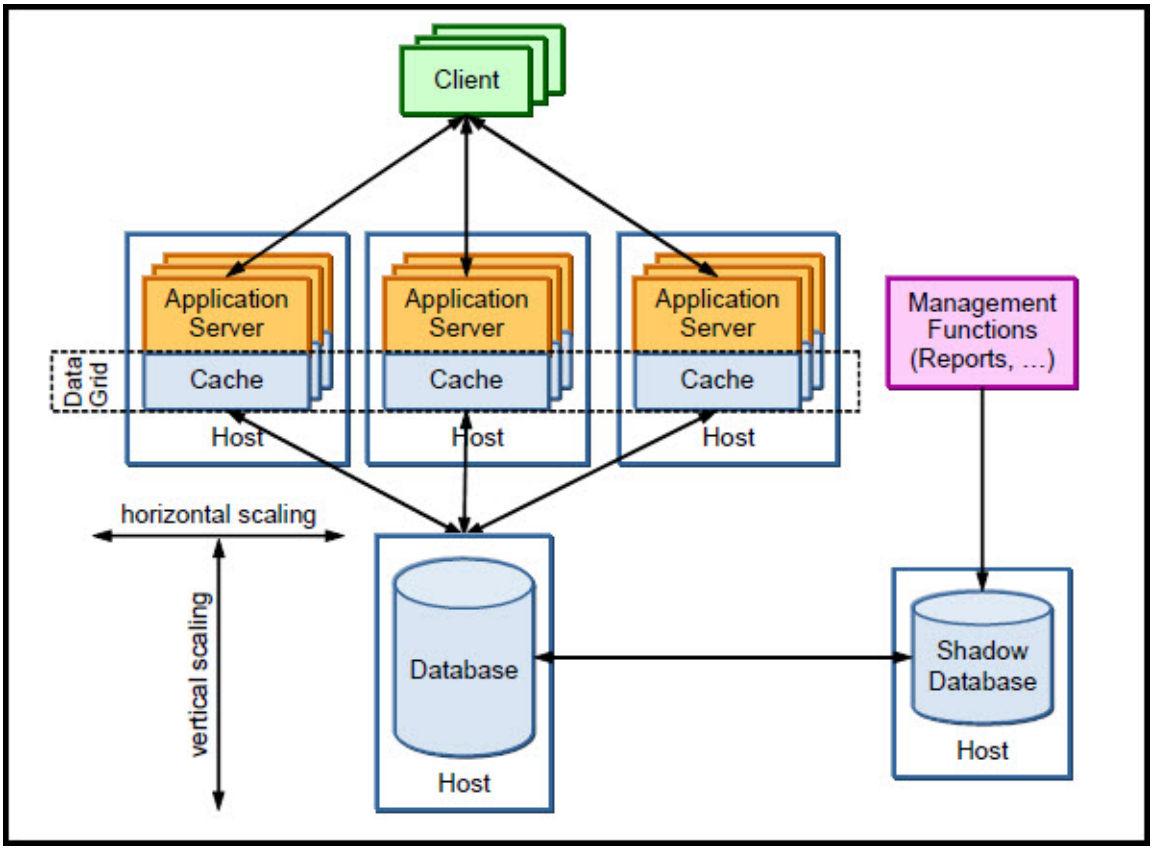

Figure 3. Introducing caching as a response to the scalability challenge

A cache is used to hold the syntax of prior requests to the resource and its response. This situation allows subsequent requests of the same syntax to be quickly served up from the cache rather than making another request to the resource. Caches can be introduced at any tier, and in multiple places within each tier. In an end-to-end request from a browser to an e-commerce system, for example, there might be significant caching that is used:

- Within the browser, for images and CSS files.
- In a caching proxy within the user's company intranet, for identical image requests made by other users in that company.
- In an edge-of-network service as provided by, for example, Akamai.
- At the web server tier, for static content.
- In the application server tier, for caching various oft-used items, and copies of browser pages or page fragments already rendered for a request.
- In the database, caching past read and write data to avoid having to read them from disk.
- At other back-end systems that are used to complete the request. For example, you can use caches in a server that is used to handle a web service call to return appropriate municipal and state taxes for a locale.

Caching is a universally accepted and commonly implemented practice to increase performance while reducing system requirements. The appropriate uses of caching are ever expanding.

After you determine that caching is valuable at a particular tier, the next decision is where to hold the cached data. Usually, the easiest and fastest place is within the local addressable memory of the component that is using the cache. However, being easiest does not make it correct for every circumstance.

At the application server tier in particular, the amount of cached data can be substantial, in the hundreds of megabytes. It often includes caching of HTTP session data, servlet/JSP page fragments, commands, inventory, prices, and many other application objects, all working to reduce back-end system loads while improving responsiveness. If the number or sizes of these caches can be increased, the cache hit ratio can be further improved, saving even more calls to constrained back-end resources.

The amount of cached data at the application tier can lead to several issues:

- How to ensure all Java virtual machines (JVMs) within the application server tier have a consistent copy of each cached object.
- How to propagate a cache invalidation from one JVM to the other JVMs so that none of them use a stale cache entry.
- Performing both of the above tasks in a timely manner so that caches are synchronized as changes are made.
- How to hold all of the cached data within the limited application server JVM heap size.
- Excessive garbage collection (GC) pause times because of increased application server JVM heap size.

In a modern WebSphere Application Server Network Deployment environment, built-in Data Replication Service (DRS) features are available to address the first three bullets. But this service comes at extra cost in terms of processor and network bandwidth usage. As the number of application server JVMs in the cluster is increased, the DRS “network chatter” to keep all of the caches synchronized can become a significant portion of the overall work that is done by each JVM. A point of negative return can be reached, where adding another application server JVM to the cluster decreases the responsiveness of the overall application.

Furthermore, if the cached data is held within the application server’s JVM heap, it can become constrained in the total size of its caches. In 32-bit mode, there is an addressability limit of around 2 GB as a usable heap size. Switching to 64-bit mode is not a cure either, as there can be excessive GC impact as heap sizes grow over 4 GB or more. As the caches grow bigger, GC activity and pause times can rise to 10% or more of the total processor usage. Users can also experience excessive response time delays when lengthy full-GC operations freeze the entire JVM, sometimes for several seconds at a time.

WebSphere eXtreme Scale provides an extensible framework to simplify the caching of data that is used by an application. It can be used to build a highly scalable, fault-tolerant data grid with virtually unlimited scaling capabilities beyond *terabyte* capacity. Because its capacity can be dynamically increased to an extreme size, it can be thought of as an *elastic cache*. WebSphere eXtreme Scale enables infrastructure with the ability to deal with extreme levels of data processing and performance. When the data and resulting transactions experience incremental or exponential growth, the business performance does not suffer because the grid is easily extended by adding more capacity (JVMs and hardware).

Integrating WebSphere eXtreme Scale with other middleware

Business applications are not the only way to take advantage of the features of WebSphere eXtreme Scale. Several IBM products can also take advantage of these performance and scalability enhancements:

- IBM WebSphere Commerce
- IBM WebSphere Portal Server
- IBM WebSphere Application Server
- IBM WebSphere Business Events
- IBM Rational® Jazz™

Usage scenarios

WebSphere eXtreme Scale addresses two major concerns for scalability: cost savings and adaptation to growing business needs.

Cost savings

Integration of applications and application serving products with WebSphere eXtreme Scale can help you scale your solution with minimal cost. Specifically, major cost savings can be realized in session offloading and dynamic cache offloading.

- Session offloading

  Session offloading enhances efficiency in application performance and minimizes the need for additional JVM capacity to hold the session data. A specific example of this can be seen in the integration of WebSphere eXtreme Scale and WebSphere Portal Server. Offloading sessions from session heavy customer portlets can reduce the need for storage in the JVMs, thus increasing the efficiency of each JVM and reducing the need for additional Portal JVMs (and additional Portal licenses).

- Dynamic cache offloading

  The dynamic cache service in WebSphere Application Server is used extensively by WebSphere Commerce Server for caching pages, catalogs, and other application-critical objects. Offloading the dynamic cache from the JVM memory into eXtreme Scale grids reduces the load on the Commerce JVMs, reducing the need for additional JVMs and additional Commerce server licenses.

Adapting to growing business needs

Adapting your application infrastructure according to the future needs of your business is critical and requires precise engineering. Designing your scalability solution to be dynamic enough to keep pace with your growing business requirements is important for keeping your business competitive. WebSphere eXtreme Scale provides you with the capability to do this task.

Two major areas where eXtreme Scale can provide benefits for meeting business growth requirements are in extreme transaction processing and complex event processing.

- Extreme transaction processing

  Scalability that is combined with performance is something every business wants. Conducting transactions using WebSphere eXtreme Scale caching is much faster than transactions that span multiple back-end systems. Although positioning WebSphere eXtreme Scale for extreme transaction processing requires changes to your existing architecture, the cost of this change is minimal compared to the benefits that can be realized over the long run.

- Complex event processing

  WebSphere eXtreme Scale can be positioned as the first layer in event processing architectures to filter out events that are not business relevant. WebSphere eXtreme Scale also provide event enrichment by efficiently caching event-related data. Using WebSphere eXtreme Scale increases the scalability of the event processing platform by leveraging a routing optimization technique. The integration of WebSphere eXtreme Scale and WebSphere Business Events is a good example of this use.

Ordering information

You can order WebSphere eXtreme Scale, by accessing the WebSphere eXtreme Scale product page and clicking **Ready to Buy** at the following website:

http://www.ibm.com/software/webservers/appserv/extremescale

Related information

For more information, see the following documents:

- *WebSphere eXtreme Scale V8.6: Key Concepts and Usage Scenarios*, SG24-7683
  http://www.redbooks.ibm.com/abstracts/sg247683.html
- *Scalable, Integrated Solutions for Elastic Caching Using IBM WebSphere eXtreme Scale*, SG24-7926
  http://www.redbooks.ibm.com/abstracts/sg247926.html
- *WebSphere eXtreme Scale Best Practices for Operation and Management*, SG24-7964
  http://www.redbooks.ibm.com/abstracts/sg247964.html
- *Enterprise Caching Solutions using IBM WebSphere DataPower SOA Appliances and IBM WebSphere eXtreme Scale*, SG24-8043
  http://www.redbooks.ibm.com/abstracts/sg248043.html
- WebSphere eXtreme Scale product page
  http://www.ibm.com/software/webservers/appserv/extremescale
- IBM Offering Information page (announcement letters and sales manuals):
  http://www.ibm.com/common/ssi/index.wss?request_locale=en

  On this page, enter WebSphere eXtreme Scale, select the information type, and then click **Search**. On the next page, narrow your search results by geography and language.

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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

This document was created or updated on December 19, 2013.

Send us your comments in one of the following ways:

- Use the online **Contact us** review form found at:
  ibm.com/redbooks
- Send your comments in an e-mail 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.

This document is available online at http://www.ibm.com/redbooks/abstracts/tips1059.html .

Trademarks