

IBM WebSphere eXtreme Scale V8.6

Overview



© 2012 IBM Corporation

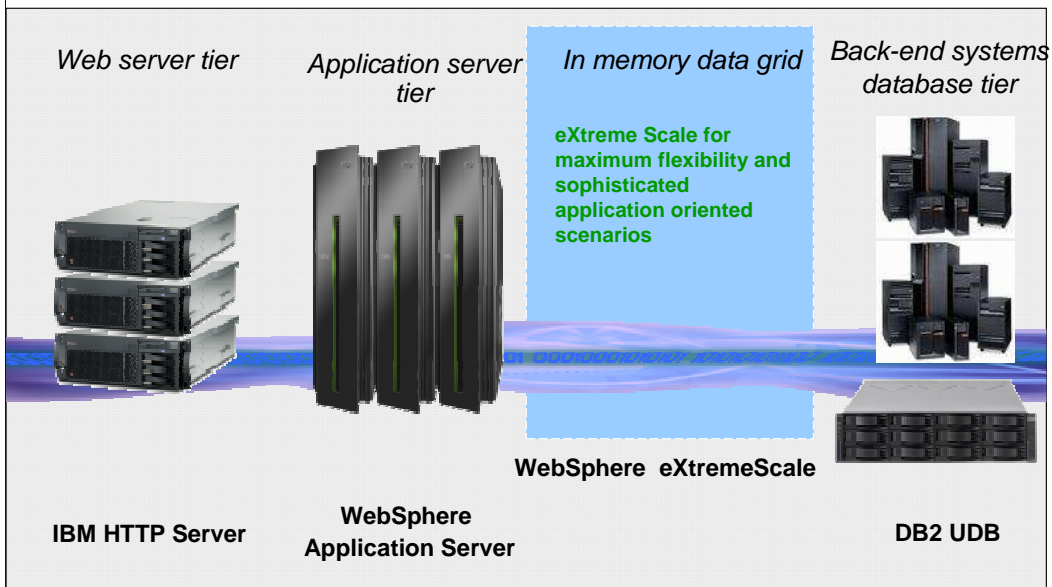
This presentation highlights WebSphere eXtreme scale version 8.6 features.

Table of contents

- Overview
 - Liberty support
 - Monitoring enhancements - Message center
 - Enterprise Data Grid and .Net
 - Near cache invalidation
 - Continuous query
- Summary

WebSphere eXtreme Scale version 8.6 continues to support the Liberty profile. There are also enhanced monitoring capabilities for better grid management and visualization with the message center. Java and .NET applications can now interact natively with the same data in the same data grid, leading the way toward a true enterprise-wide data grid. Near cache invalidation and continuous query are the other new features in version 8.6. Built in pub/sub capabilities enable WebSphere eXtreme Scale version 8.6 to update client “near caches” whenever data is updated, deleted, or invalidated on the server side. API enhancements enable continuous query of data that is inserted and updated in the grid.

Recap: Elastic caching solves the problem of transaction overload



Here is a quick recap on what eXtreme Scale is and where it sits in relation to WebSphere Application Server and databases.

WebSphere eXtreme Scale is an elastic, scalable, in-memory data grid. The data grid dynamically caches, partitions, replicates, and manages application data and business logic across multiple servers. WebSphere eXtreme Scale can be used in different ways. You can use the product as a very powerful cache, as an in-memory database processing space to manage application state, or to build Extreme Transaction Processing (XTP) applications. These XTP capabilities include an application infrastructure to support your most demanding business-critical applications. Typically, WebSphere eXtreme Scale sits between WebSphere application server tier and backend systems.

Liberty support

One of the features that was added in WebSphere eXtreme scale version 8.5, is support for Liberty. Specifically WebSphere eXtreme Scale first supported Liberty in version 8.5.0.1 (the first fixpack of version 8.5). WebSphere eXtreme Scale version 8.6 continues to support the Liberty profile.

Liberty (WebSphere Liberty profile) support

- What is liberty?
 - Lightweight application server
 - Very fast startup time, simple administration, easy to use
- WebSphere eXtreme Scale integrates seamlessly with Liberty runtime
- Easy to develop WebSphere eXtreme scale applications using Liberty within Eclipse
 - start a WebSphere eXtreme scale grid
 - start a WebSphere eXtreme scale client
 - start the Liberty server

Liberty, also known as the WebSphere Application Server Liberty profile, is a lightweight application server intended to compete with products like Tomcat and Jboss. It ships as a separate profile type with WebSphere Application Server V8.5 and above. It has a very fast startup time and you can choose what features you want loaded. It also has simple administration, including drag-and-drop application management, simplified server configuration, and it is easy to use for application development. It does not come with all features that traditional WebSphere does. For example, Liberty lacks many Java EE components and WebSphere extensions such as: Enterprise Java Beans, Resource Adapters, BLAs, and the administrative console.

WebSphere eXtreme Scale integrates seamlessly with the Liberty runtime and feature model. WebSphere eXtreme Scale container servers can now run on Liberty profile. Stand-alone Liberty servers can maintain HTTP session failover and high availability by leveraging a WebSphere eXtreme Scale grid making it very simple to make use of WebSphere eXtreme Scale use cases. It is easy to develop WebSphere eXtreme Scale applications using Liberty within Eclipse. All you have to do is start up a WebSphere eXtreme Scale grid, start a WebSphere eXtreme Scale client, and start the Liberty server all within a single runtime environment.

Monitoring enhancements

This section highlights some monitoring enhancements done in WebSphere eXtreme Scale version 8.6.

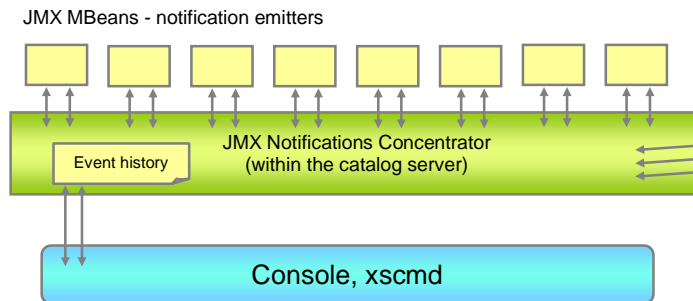
Monitoring

- Message center
 - messaging hub built into each catalog server for monitoring
- New xscmd command
 - With scripting support
- New MBean api
 - new MBean operations and notifications
 - History of notifications

From time to time, the health of a grid might be in jeopardy and the administrator needs real time notifications for monitoring the health of a grid. WebSphere eXtreme Scale Version 8.6 introduces monitoring enhancements to provide event notifications for logs and “first failure data capture” (FFDC) messages. These notifications are integrated through JMX. These event JMX notifications can be viewed with the new message center in the web console, the XSCMD utility, or programmatically through MBeans. The monitoring web console retrieves and displays events in a sortable and filterable list, showing priority, source, and NLS enabled messages for each event. It shows real time events by popping up all severe messages, but no more than one every 30 seconds. With the XSCMD utility, you can view event history. You can subscribe to real time events and pipe them out to standard output for scripted automatic response.

Monitoring enhancements

- The new JMX notifications built into the WebSphere eXtreme scale servers running in each JVM
- The notification hub built into every catalog server process
- Console **message center** built into existing eXtreme Scale and XC10 monitoring consoles
- JMX Mbean act as a concentrator (optionally running in each catalog server) for all JMX notifications



The new JMX notifications are built into the WebSphere eXtreme Scale servers running in each Java virtual machine (JVM). The notification hub, which maintains a queryable history of all recent notifications, is built into every catalog server process (although you can turn this off). The web console **message center** is built into the existing WebSphere eXtreme Scale and XC10 monitoring consoles, which means that there is nothing new to acquire, install, or configure. The new JMX Mbean subscribes to all log and FFCD notifications generated by all servers in the domain, in addition to existing and new JMX notifications. All notifications received are then forwarded to the subscribers, while maintaining a recent notification history that is also displayed in the monitoring console.

Improved logging and tracing (HPEL/ISA)

- Built-in High Performance Event Logging (**HPEL**) offers improved performance for trace and logging
- Same facility that WebSphere Application Server provides
- Data stored in binary format
- The log Viewer command-line utility allows querying and filtering the log files
- Adds many more options for tracing and logging
- Improved log gathering using IBM Support Assist Data collector (**ISA DC**)

High Performance Extensible Logging (HPEL) is a log and trace facility available in both stand-alone and WebSphere Application server-managed WebSphere eXtreme scale. It provides a convenient mechanism for storing and accessing log, trace, System.err, and System.out information produced by the application server or applications. It is an alternative to the basic log and trace facility, which provides the JVM logs, diagnostic trace, and service log files commonly named SystemOut.log, SystemErr.log, trace.log, and activity.log. HPEL provides a log data repository, a trace data repository, and a text log file.

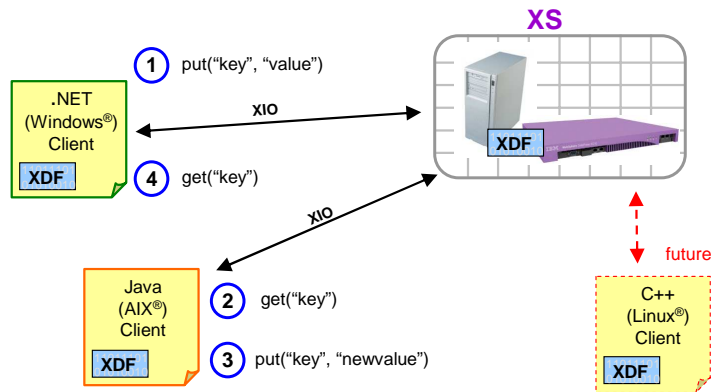
Rather than replacing the existing logging facility, HPEL is being provided as an alternative that can be enabled using the server properties file for container and catalog servers. HPEL is not enabled by default. Once enabled, HPEL will handle all WebSphere eXtreme scale logging and the resulting log files are placed in the specified HPEL repository location. This location is also specified in the server properties files for container and catalog servers.

ISA DC stands for IBM Support Assist Data Collector. Log gathering for problem determination is a function that has been in eXtreme Scale for multiple releases in the form of ISALite, which is shipped with the product and can be found in the dir_wxs_home/isalite_wxs folder. ISA has recently been updated to be more interactive and has also changed the name of the feature to ISA DC. ISA DC installs with both WebSphere eXtreme scale stand-alone and WebSphere Application server-managed eXtreme scale. One of the key features of ISA DC is that it improves ease of use by remembering various inputs. ISA DC provides a feature that reduces the amount of repetitive typing during console input.

Enterprise Data Grid and .Net

Enterprise Data Grid and .Net support in WebSphere eXtreme Scale are features added in version 8.6 of WebSphere eXtreme Scale.

Enterprise Data Grid



- 1 The .NET client saves data in its format to XS
- 2 XS stores the data in a universal format, so when the Java client requests this data, it gets converted to Java format
- 3 The Java client can then update the data and resave it
- 4 The .NET client can once again gain access to the data, and it gets converted to .NET format

Enterprise Data Grid provides grid and cache access to multiple clients. For example, a Java client can integrate with the grid and a .Net client can integrate with the same grid.

Here are some terminologies to keep in mind: In the example shown, XDF is the new data format. This is a cross-platform serialization format. XIO is new TCP/IP-based transport and it removes the need for the Object Request Broker. .Net is a new client in the native eXtreme Scale Object Grid API support.

With the example shown, in step one the .NET client saves data in its format to eXtreme Scale. In step two, eXtreme scale stores the data in a universal format so that when the Java client requests this data, it gets converted to Java format. In step three, the Java client can then update the data and save it. Finally, in step four, the .Net client can access the data that gets converted to .NET format.

eXtreme I/O (XIO) - Overview

- What is it?
 - A new Transport protocol deprecating the Object Request Broker (ORB)
- Why introduce a new protocol?
 - The ORB binds XS to Java only native clients
 - The ORB is a general purpose transport, you can do better with a custom transport targeted for data caching
- XIO Benefits
 - Enables new language integration: for example, .NET
 - Performance and reliability enhancements

In the previous slide, XIO, or eXtreme I/O was mentioned. This is the new transport protocol. Object Request Broker protocol has worked for many years; however, to use other clients beyond Java, the new eXtreme IO transport has been introduced. Object Request Broker is still supported but may eventually be phased out in favor of eXtreme IO. This new eXtreme IO transport works for all clients - Java and non-Java – has better performance, and is more reliable.

eXtreme data format (XDF) - Overview

- What is it?
 - New cross platform serialization format
- Why introduce a new serialization format?
 - The current default is Java serialization which is relatively slow and binds XS to java
- XDF benefits
 - Faster performance
 - More compact representation
 - Enables cross language supports

Similarly, XDF, which stands for eXtreme data format, is a new data format that is serializable across platforms. This new data format allows you to communicate well with different clients. In the past, creating custom serializers was supported for eXtreme Scale. Since the current default Java serialization is slow and binds eXtreme scale to Java, the use of new data formats like XDF is a better option. XDF provides faster performance, more compact representation, and cross-language support.

.NET client – What is it?

- A new self contained 32-bit Windows installer
- A set of .NET assemblies that provide
 - A general WebSphere eXtreme Scale data access programming interface
 - Create, retrieve, update and delete operations plus transactional support
 - One or more plug-in implementations
 - User ID and password credential generator
- Developer assistance
 - Visual studio integration
 - API online documentation
 - Samples
- Updated IBM information center publications

The .net client is a new, self-contained, 32-bit Windows Java-based installer. It is a WebSphere eXtreme Scale data access programming interface with standard create, remove, update, and delete operations and transactional support, contains one or more plug-in implementations with user ID and password credentials similar to Java. It intergrates well with the visual studio. The WebSphere eXtreme Scale .NET client provides a Visual Studio IntelliSense extension. Some tutorials and samples are provided in the information center.

.NET Client API - overview

- Java Client API differences
 - Connection is still TCP/IP based, but all client/server traffic flows over eXtreme IO (XIO)
 - EDG Java client is capable of supporting XIO and non-XIO traffic
 - .NET Client provides generics based APIs.
 - Developer specifies data type of grid keys and values
 - More “batch” APIs (that is... GetAll(), RemoveAll(), and so on.)
 - Exception and non-exception throwing APIs. For example
 - void Remove(key)
 - bool TryRemove (key)
- .NET client limitations
 - COPY_TO_BYTES* copy mode only
 - No NearCache
 - No EntityManager

The .NET client API is similar to the Java API but initially limited features are available. As an example, connection is still TCP/IP based, but all client/server traffic flows over XIO. The Enterprise Data Grid Java client is capable of supporting XIO and non-XIO traffic. Also note that the .NET client provides full support for generic APIs and more of the batch APIs like GetAll and DeleteAll.

The .Net client has some limitations including no availability of near cache or EntityManager. Also even though WebSphere eXtreme Scale supports various "copy" modes in this release, the .Net client is limited to COPY_TO_BYTES.

.NET client API - Configuration

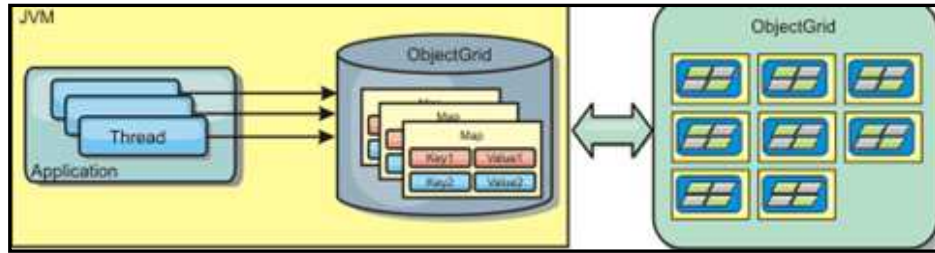
- Client side properties file
 - Connection characteristics
 - Security parameters
 - Plugin assembly specification
- Server side grid configuration
 - Default configuration values for
 - Transaction timeout
 - Transaction isolation level
 - Grid item time to live
- Programmatic API override
- Override priority
 - Server side value (if applicable)
 - Client side properties file
 - Client side programmatic API

Just like the Java client, there are some properties you can configure on the .Net client. Like Java, there are three tiers for configuration: client side properties, server side configuration, and programmatic API. Note that, just like the Java client, the .Net client uses the same override priority.

Near cache invalidation

Near cache invalidation is another new feature in eXremeScale version 8.6.

Near cache



- local copy of the stored data in the eXtreme Scale Grid
- Independent ObjectGrid on each client serving as a cache for the remote, server-side cache
- Very fast – provides in-memory access to a subset of the entire cached data set that is stored remotely
- Not partitioned and contains data from any of the remote eXtreme Scale partitions
- Fast response time because all access to the data is local
- Disadvantage – stale data
 - Use when response time is important and stale data can be tolerated

18

Overview

© 2012 IBM Corporation

Near cache is essentially a local copy of the stored data in the eXtreme Scale grid. It is an independent ObjectGrid on each client, serving as a cache for the remote server-side cache.

Near cache is very fast because it provides in-memory access to a subset of the entire cached data set that is stored remotely in the eXtreme Scale servers. It is not partitioned and contains data from any of the remote eXtreme Scale partitions.

The advantage of near cache is that it gives fast response time because all access to the data is local. Looking for the data in the near cache first saves a trip to the grid of servers, thus making even the remote data locally accessible. The disadvantage of the near cache is that it increases the chance of stale data because the near cache at each tier can be out of synchronization with the current data in the data grid. It also relies on an evictor to invalidate data to avoid running out of memory. Near cache is recommended for use when response time is important and stale data can be tolerated.

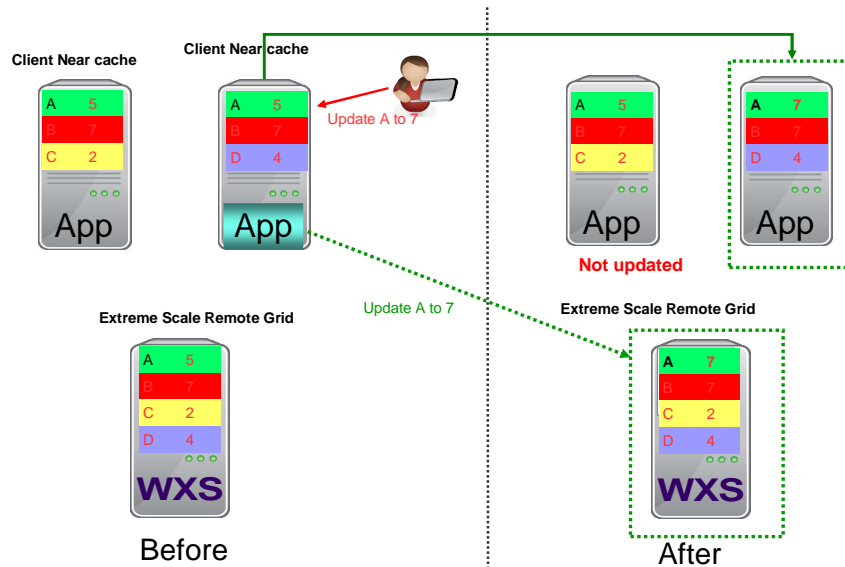
Near cache invalidation key concepts

- Near cache – A local copy of the data stored in an eXtreme Scale grid
 - This copy can be out of date with respect to the data in the grid
- Near cache invalidation – Removing stale data from the near cache as quickly as possible instead of using time to live eviction
 - As a WebSphere eXtreme Scale developer you want your client's Java virtual machine to have a relatively current near cache. This is to avoid remote calls to the grid while limiting exposure to stale data

In a distributed WebSphere eXtreme Scale environment, the client side has a near cache by default when using the optimistic locking strategy or when locking is disabled. This data can become stale from time to time - thus the need for near cache invalidation. Near cache invalidation removes stale data from the near cache as quickly as possible instead of using TimeToLive eviction.

As a developer, you want your client JVM to have a current near cache to avoid remote calls to the grid while limiting exposure to stale data. You can configure near cache invalidation to remove stale data from the near cache as quickly as possible. When an update, deletion, or invalidation operation is run against the remote data grid, an asynchronous invalidation gets triggered in the near cache. This mechanism works more quickly than the other option of using time-to-live (TTL) eviction in the near cache.

Example: No near cache invalidation feature



20

IBM Confidential

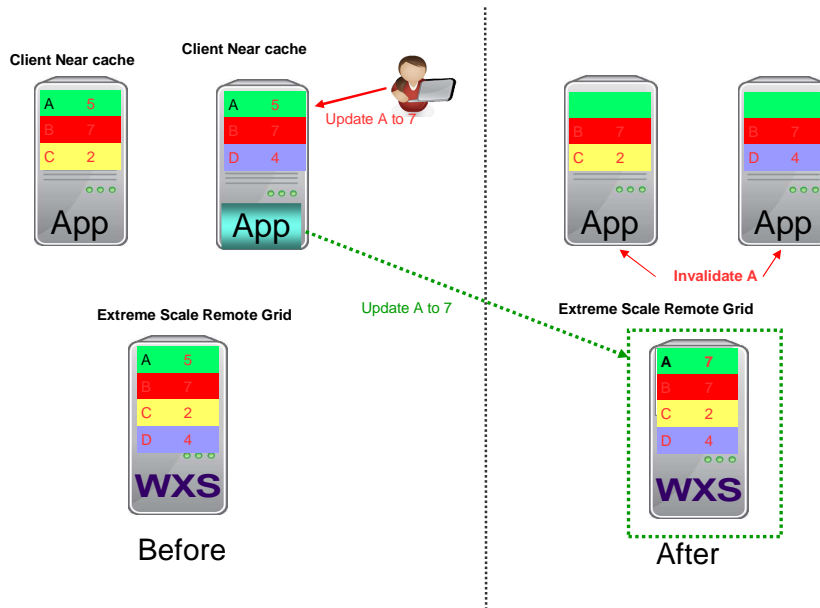
© 2012 IBM Corporation

Here is an example of how data can become stale.

Starting from the left – There is a grid and some clients. These clients have near caches, and so have cached a few values locally.

As an example, suppose someone updates key A from five to seven on one of the clients. The remote grid gets updated to seven and the client that did the update now has a value of seven as shown on the right. However for the other client that had cached the key A value the cache still has a value of five and is not aware of the update. This is the problem that near cache invalidation solves.

Example: With near cache invalidation feature



With Near cache invalidation, if someone updates the value of key A from five to seven on one of the clients, the remote grid is updated with a value of seven. The remote grid will issue an invalidation message to all the clients to invalidate their key A. This means that now these clients will need to go to the remote Grid to access the key A since it is no longer available in the near cache. The result is that stale data is removed from the near cache as quickly as possible.

Near cache invalidation

New!

- Triggered by server side update, remove, invalidate, or clear
- Affected keys are sent to interested clients
 - Affected keys are removed from client's near cache
- No changes to client code required
 - Configured through objectgrid.xml
- Does not completely prevent stale data in near cache
 - Invalidation notice is asynchronous
- Must use optimistic or no locking
 - Near caches are not available with pessimistic locking
- Near cache evictor still recommended
 - Ensure near cache does not grow unbounded
 - For example - Size based least recently used eviction policy

When an update occurs, affected keys are sent to the interested clients and the keys are removed from the clients near cache. This helps lower the occurrence of stale data in the near cache. It does not require any changes to the client code; instead it is configured through the objectgrid.xml. Note that near cache invalidation does not completely prevent stale data in near cache because it might take some time before the invalidation notice is received by clients since the invalidation notice is asynchronous. Also note that near caches are not available with pessimistic locking. They only use optimistic or no locking mechanism.

You should also use the near cache evictor to prevent the near cache from growing indefinitely. Use eviction policies like the leastRecentlyUsed eviction policy to contain the size of the near cache.

Near cache invalidation options

- Time to live evictor – Remove cache entry after X minutes
 - Based on create time, last access time, or last update time
- JMSObjectGridEventListener
 - Existing eXtreme Scale plug-in
 - Synchronizes client caches
 - Sends all updates, not just invalidation
 - Requires external JMS bus configuration and deployment
- Roll your own
 - Trigger near cache invalidations based on some event (like DB update)

Several near cache invalidation options are available. You can configure the Time-to-live evictor. You remove cache entries after a certain period of time based on when the cache was created, or when last accessed or updated. The JMSObjectGridEventListener is a plug-in to eXtreme scale that synchronizes client caches and sends all updates and invalidations. The downside of this plug-in is that it is network intensive and requires an external JMS bus configuration and deployment. The third option is to create your own near invalidations triggered by some event.

Continuous query

Continuous query is yet another new feature in eXtreme Scale version 8.6.

Continuous query key concepts

- Continuous query – proactively notifying clients of new cache entries that match a user defined filter chain
 - *As a WebSphere eXtreme scale developer, you want the ability to find a set of continuously updated keys that match a set of criteria. This criteria is one that you have defined without having to make an explicit remote call to the grid . This is so that you can continuously process this set to provide a service to the user*
- Filter – component that can pass or not pass a cache entry based on an attribute of that cache entry
 - Example: Customer.age > 17
- Filter chain – combination of filters to allow for more sophisticated filter operations

Continuous query is the ability to proactively notify clients of new cache entries that match a user defined filter chain. This is done by running a query in a polling mode.

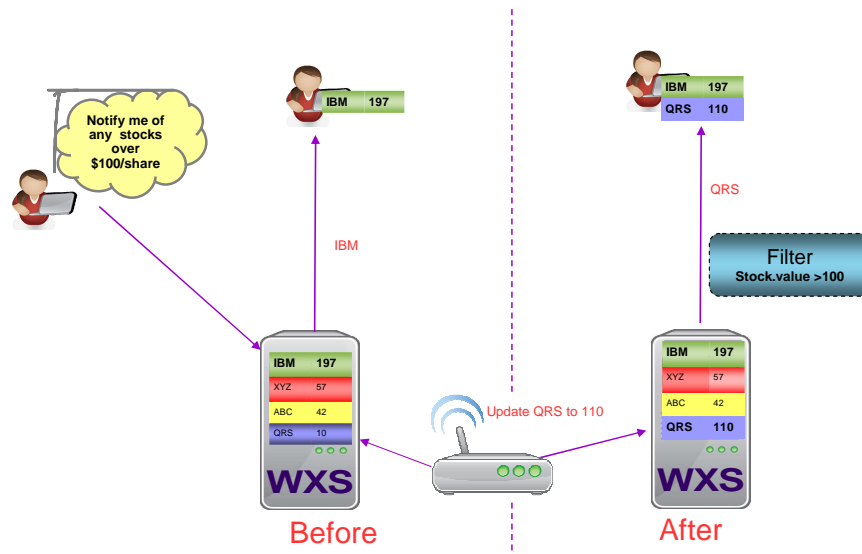
A filter is different for cache purposes in that you are looking at one cache entry and saying does this one cache entry have a set of attributes? A combination of filters can yield more sophisticated results.

Continuous query

- User defines Filter chain which passes the keys/values they care about
 - Users can implement their own filters or use the provided basic filters
 - EQ, NEQ, GT, GTE, NotNull, IsNull, LT, LTE, RegEx, NotRegEx, And, Not, Or
- New Filter registration adds filter to transaction listener/publisher
 - Each log entry in a transaction is processed by each filter, publish matches
 - Insert
 - Update – did not match, matches now
 - Update – did match, does not match now
 - Update – did match, still matches, but value changed - optional
 - Delete/Clear/Invalidate
 - Keys, and optionally Values, published to subscribed clients
- Client maintains map datastructure of matching entries
 - Client code can access the map of matches directly - optional
 - Map supports notification listener registration

Continuous query works by defining filters. You define filter chains that can either pass keys or values. Initially, the query will start once on all containers and then will notify those containers that are interested. When you register a new filter, it creates a listener on the server side. Each log entry in a transaction is processed by each filter. When the value changes beyond the filter value, what is sent to the client is the keys that no longer match, NOT the values. On the client side, the client maintains keys that match the filter.

Continuous query notification – stock value example



27

IBM Confidential

© 2012 IBM Corporation

This is an example of a continuous query in a simple scenario.

The Grid contains stock symbols as keys, and stock prices as values. A user has come along and wants to be notified of any stock prices of over 100 dollars a share. They are going to submit this as a continuous query and initially will get a result of IBM stock of over 100 dollars a share. As the value of the other stocks change, the grid is updated. Suppose QRS goes from 10 to 110 - the grid is updated and the continuous query filter will detect that and send notification of the second stock that matches the filter condition.

Section

Summary

This section contains a summary.

Summary

- Support for the new IBM WebSphere Application Server **V8.5 Liberty profile**, proves a smaller installation footprint
- Addition of the **message center** to monitor grid health in real time
- **Enterprise data grid** for support of the .Net clients
- **Near cache invalidation** feature for removing stale data from the near cache as quickly as possible
- **Continuous query** for proactively notifying clients of new cache entries that match a user defined filter chain

WebSphere eXtreme Scale V8.6 continues to support liberty profile initially introduced in eXtreme Scale version 8.5. Monitoring enhancements include the new **message center** to monitor grid health in real time. **Enterprise Data Grid** supports the .Net clients using eXtreme Data format (XDF) and eXtreme IO (XIO) transport. **Near cache invalidation** removes stale data from the near cache as quickly as possible. **Continuous query** proactively notifies clients of new cache entries that match a user defined filter chain.

Java and .NET applications can now interact natively with the same data in the same data grid, leading the way toward a true enterprise-wide data grid. WebSphere eXtreme scale 8.6 delivers a faster, more compact serialization format called eXtreme Data Format (XDF), which is neutral to programming languages. A new transport mechanism, eXtreme IO (XIO), removes the dependency on the IBM ORB, enabling easier integration with existing environments. Built-in pub/sub capabilities enable WebSphere eXtreme Scale version 8.6 to update client “near caches” whenever data is updated, deleted, or invalidated on the server side. API enhancements enable continuous query of data that is inserted and updated in the grid. There are also enhanced monitoring capabilities for better grid management and visualization.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_XS86_Overview.ppt

This module is also available in PDF format at: [../XS86_Overview.pdf](#)

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, AIX, DB2, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. Windows, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

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

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2012. All rights reserved.