Explore the eXtreme Scale-based caching service options in IBM PureApplication System

Cloud-related performance and scalability issues can be addressed with a caching service

Cheng-Chieh Cheng (cheng1@us.ibm.com)  
Software Services for WebSphere consultant  
IBM

Caching services are a popular solution to address performance and scalability issues for cloud enterprise applications. Explore three caching options available with the IBM® PureApplication® System cloud system: One built-in, one based on WebSphere® eXtreme Scale that uses a virtual system pattern on a cluster, and one based on eXtreme Scale that uses a VSP with a core OS image.

Caching services are a popular solution to address performance and scalability issues for cloud enterprise applications (like those apps that run on IBM® PureApplication® System, a cloud application platform for enterprise applications). PureApplication System has a built-in shared caching service for virtual deployments in the cloud to store, share, and access cached information among components. This caching service is based on code from WebSphere® eXtreme Scale — a fully elastic memory-based storage grid — and provides functionalities similar to the WebSphere DataPower® XC10 Appliance — a caching platform that supports data-oriented, distributed caching scenarios with little or no changes to existing applications and, which adds elastic caching functions that enable applications to scale more cost-effectively.

To take full advantage of WebSphere eXtreme Scale technology, you can create your own caching service that is based on eXtreme Scale:

- You can deploy the eXtreme Scale application into a WebSphere cluster that becomes an eXtreme Scale caching service.
- You can create an eXtreme Scale caching service by using a collection of virtual machines with eXtreme Scale installed and some scripts to configure the caching servers. (This combination is also known as a WebSphere eXtreme Scale virtual system pattern.)

In this article, I describe these caching options in the PureApplication System:
• Using the built-in shared caching service.
• Using the eXtreme Scale caching service that employs a WebSphere cluster virtual system pattern (VSP).
• Using the eXtreme Scale virtual system pattern with a core OS image.

Use the built-in shared caching service

The built-in caching service in PureApplication System is a shared service for virtual deployments in a PureApplication System cloud group to store, share, and access caching information among components. Each cloud group must have its own caching service.

The caching service automatically scales based on a policy that can be slightly adjusted by the user. The maximum number of instances is configurable so that the capacity of the caching service can be capped.

The caching service provides caching capabilities similar to the WebSphere DataPower XC10 Appliance. It supports three types of data grids:

• A simple data grid that can be used for anything.
• A WebSphere Application Server dynamic cache replication grid (to allow container-level caching of the output of servlets, JSPs, web services, and WebSphere Application Server commands into memory).
• A WebSphere Application Server HTTP session replication grid (to maintain user information and the state of the user session for the period of a user's interaction with the application).

The following procedure illustrates generally how to use the caching service for HTTP session management:

1. You deploy the caching Service into target cloud group.
2. You select the XC10 feature in the Deployment Manager part of WebSphere Application Server (WAS) cluster pattern.
3. You configure WAS servers or applications to enable session management by using XC10 (caching service) caching.

Now look at using the eXtreme Scale caching service that employs a WebSphere cluster virtual system pattern.

Use eXtreme Scale caching service and WebSphere cluster VSP

WebSphere eXtreme Scale supports running both catalog and container servers in a WebSphere Application Server Network Deployment (WAS-ND) environment when the eXtreme Scale server component is installed on WAS-ND and WebSphere profiles are augmented with eXtreme Scale. 

(Catalog servers are the JVMs that comprise the catalog service that maintains the healthy operation of grid servers and containers. Container servers host grid containers that hold the shards, which make up the data that is contained in the whole grid.) The WAS-ND virtual image in PureApplication System does not include the eXtreme Scale server component.
To support eXtreme Scale servers in a WAS-ND environment in PureApplication System, you must install the eXtreme Scale server component on the WAS cluster parts. Use a script package to install the eXtreme Scale server component that runs when the WAS cluster pattern is deployed.

Once the eXtreme Scale server component is available in WAS-ND, you can create eXtreme Scale caching service on the WebSphere cluster.

The eXtreme Scale caching service has two parts: catalog servers and container servers. Dedicate one WAS cluster for the eXtreme Scale catalog servers and another cluster for the container servers. For catalog clusters, each cluster member should run on a dedicated virtual machine (VM). For container clusters, cluster members can reside on the same VM. However, each VM should have identical computing resource and host the same number of cluster members to achieve best load balancing.

**Catalog server cluster**

The cluster that is dedicated to the eXtreme Scale catalog servers does not need to install anything. However, you must create a eXtreme Scale catalog service domain, by using a script package or manual method after deployment using the WebSphere Integrated Solutions Console. All of the cluster members of the catalog server cluster need to be added to the catalog service domain.

**Container server cluster**

You must install the eXtreme Scale application on the cluster that is dedicated to container servers. The eXtreme Scale application is a Java™ Platform, Enterprise Edition (JEE) application that has only one JEE module. It includes two designated eXtreme Scale configuration XML files, objectGrid.xml and objectGridDeployment.xml, in the META-INF folder of the JEE module in the eXtreme Scale application.

After the eXtreme Scale application is installed on the container server cluster and started, cluster members will host eXtreme Scale container servers that are based on the definitions in the included eXtreme Scale configuration XML files.

**Enabling eXtreme Scale caching service in WAS cluster pattern**

A WAS cluster pattern contains both Deployment Manager and Custom Node parts. Parts are components that are configured on a virtual machine:

- A deployment manager's configuration profile provides a base for an environment with multiple VMs. A database part is not required for configuring a deployment manager.
- Custom nodes provide unconfigured nodes to use with a basic function control node, a full function control node, or a deployment manager. A deployment manager part must already be deployed on another VM before you can configure a Custom Node part.

Enabling eXtreme Scale caching service can be achieved through running script packages to encapsulate the following configuration tasks:
• Install eXtreme Scale server component
• Augment profiles with eXtreme Scale
• Create eXtreme Scale catalog server cluster
• Create eXtreme Scale container server cluster
• Create eXtreme Scale catalog service domain and add all catalog server cluster members
• Install eXtreme Scale application on eXtreme Scale container server cluster
• Start eXtreme Scale catalog server cluster
• Start eXtreme Scale container server cluster

The first two tasks — install and augment profiles — must run on both Deployment Manager and Custom Node parts. The rest of the tasks only run on Deployment Manager part.

Extract these configuration tasks at creation time of pattern deployment. After the WAS cluster pattern is deployed, the eXtreme Scale caching service is ready for business. The provisioned eXtreme Scale caching service is referred as eXtreme Scale system in WAS.

Use WebSphere eXtreme Scale VSP with a core OS image

WebSphere eXtreme Scale does not require WebSphere Application Server. It can be installed and run from a Java Standard Edition JVM.

An eXtreme Scale virtual system pattern can be created by using a core OS image as its starting point. After the WebSphere eXtreme Scale VSP is deployed, you can use it like a regular WebSphere eXtreme Scale system that serves as an elastic caching service in the PureApplication System.

The WebSphere eXtreme Scale VSP can be built based on core OS image with script packages used to install eXtreme Scale software and set up eXtreme Scale system configuration (that has scripts for controlling the eXtreme Scale system lifecycle).

The primary configuration tasks for building an eXtreme Scale caching service with a core OS image are:

- Installing eXtreme Scale software. (An alternative is to use the extend-and-capture approach to build a custom core OS image with eXtreme Scale installed. Extending a virtual image is the process of starting with a copy of a virtual image, modifying it. Capturing it consists of storing the modified image as a new virtual image in the catalog.)
- Setting up an eXtreme Scale server configuration. Two eXtreme Scale configuration XML files, objectGrid.xml and objectGridDeployment.xml, are required for starting container servers.
- Starting cataloger servers.
- Starting container servers.

Execute these configuration tasks during the creation time of pattern deployment. After the WebSphere eXtreme Scale virtual system pattern is deployed, the eXtreme Scale caching service is ready for business. The provisioned eXtreme Scale caching service is referred as a stand-alone eXtreme Scale system.
Built-in shared caching service versus custom eXtreme Scale caching service

Both built-in and custom caching service provides the same quality of runtime system. From an application perspective, the difference between these two is the supported functions:

- A custom eXtreme Scale caching service has full eXtreme Scale functions
- The built-in caching service supports only a subset of eXtreme Scale functions

A custom eXtreme Scale caching service can support any scenario the built-in caching service can support. The limitation of built-in caching service is because it does not allow users to modify the configuration and Java class path.

What major functions does the eXtreme Scale caching service support that the built-in caching service doesn’t?

- **In-line cache topology.** During the eXtreme Scale application design phase, one of the important decisions is to choose between in-line and side-cache topology. With in-line cache topology, the application needs to use the eXtreme Scale Loader plug-in to integrate with the back-end data store and use eXtreme Scale as the primary means for interacting with the data.
- **DataGrid API.** The DataGrid API provides a simple programming interface to run business logic over all or a subset of the data grid in parallel with where the data is located. It supports two common grid programming patterns: parallel map and parallel reduce.
- **Query API.** eXtreme Scale Query API provides a powerful mechanism to perform SELECT type queries over an entity or object-based schema by using the eXtreme Scale query language. The application needs this critical feature for searching data by multiple criteria.
- **EntityManager API.** The EntityManager API simplifies the interaction with the data grid by providing an easy way to declare and interact with a complex graph of related objects. eXtreme Scale EntityManager API is similar to JPA EntityManager API, but there are differences.
- **JPA Loaders.** eXtreme Scale has a built-in Java Persistence API (JPA) Loader plug-in implementation to interact with any database supported by your chosen loader. When the built-in JPA loaders are used, the system becomes an in-line cache topology.
- **JPA L2 Cache.** eXtreme Scale has built-in OpenJPA and Hibernate L2 cache implementation that can be used to improve performance of OpenJPA and Hibernate application. Using eXtreme Scale L2 cache implementation for OpenJPA and Hibernate does not require any application code change. This approach is the simplest to improve performance.
- **Data Grid customization.** eXtreme Scale allows applications to customize the data grid through defined plug-ins. A plug-in is a component that provides a function to the pluggable components, which include ObjectGrid and BackingMap in the eXtreme Scale system. Applications can maximize performance, change runtime behavior, and add desired functions with available plug-ins.
- **Multiple data center topologies.** eXtreme Scale has the multi-master asynchronous replication mechanism to build multiple data center topologies. Using multi-master asynchronous replication mechanism, two or more data grids can become exact copies of...
each other. Each data grid is hosted in an independent catalog service domain, with its own catalog service, container servers, and a unique name.

If your application use cases require any of these functions, employ the custom caching service.

**Conclusion**

Explore setting up a caching service as a tool to enhance application performance and scalability in your cloud environment. Caching services can reduce much of the overhead demand in access bandwidth for the most-commonly-requested data resources on your system.

I detailed how an enterprise cloud system (PureApplication System) supports caching management with built-in tools. I also showed how you can create and integrate even more powerful caching management by building a custom system that engages WebSphere eXtreme Scale's caching functions.
Resources

Learn

- These technology products converge in this article:
  - IBM PureApplication System / the IBM PureSystems family of expert integrated systems
  - IBM WebSphere eXtreme Scale
  - IBM WebSphere DataPower XC10 Appliance
- In the developerWorks cloud developer resources, discover and share knowledge and experience of application and services developers by building their projects for cloud deployment.
- These IBM technology topics converge with cloud to make this article possible:
  - WebSphere products
  - Big data and analytics knowledge
  - Java and programming language knowledge
- The author is a part of the IBM Software Services for WebSphere expert team; watch them in action.

Get products and technologies

- SoftLayer trial: Try a free cloud server for a month.

Discuss

- Get involved in the developerWorks community. Connect with other developerWorks users while you explore the developer-driven blogs, forums, groups, and wikis.
About the author

Cheng-Chieh Cheng

Cheng-Chieh Cheng has an MS in Computer Science and an MBA and much experience as a developer for the WebSphere Application Server and WebSphere eXtreme Scale. He is a published author of six technical articles and holder of four patents. Currently, Cheng is a member of the expert team of consultants in the IBM Software Services for WebSphere group.

© Copyright IBM Corporation 2014
Trademark
(www.ibm.com/developerworks/ibm/trademarks/)