Dynamic Caching and Replication in WebSphere Application Server V8

Varun Tallapragada (varunt@us.ibm.com)
WebSphere Application Server L2 Support
15 December 2011

Agenda

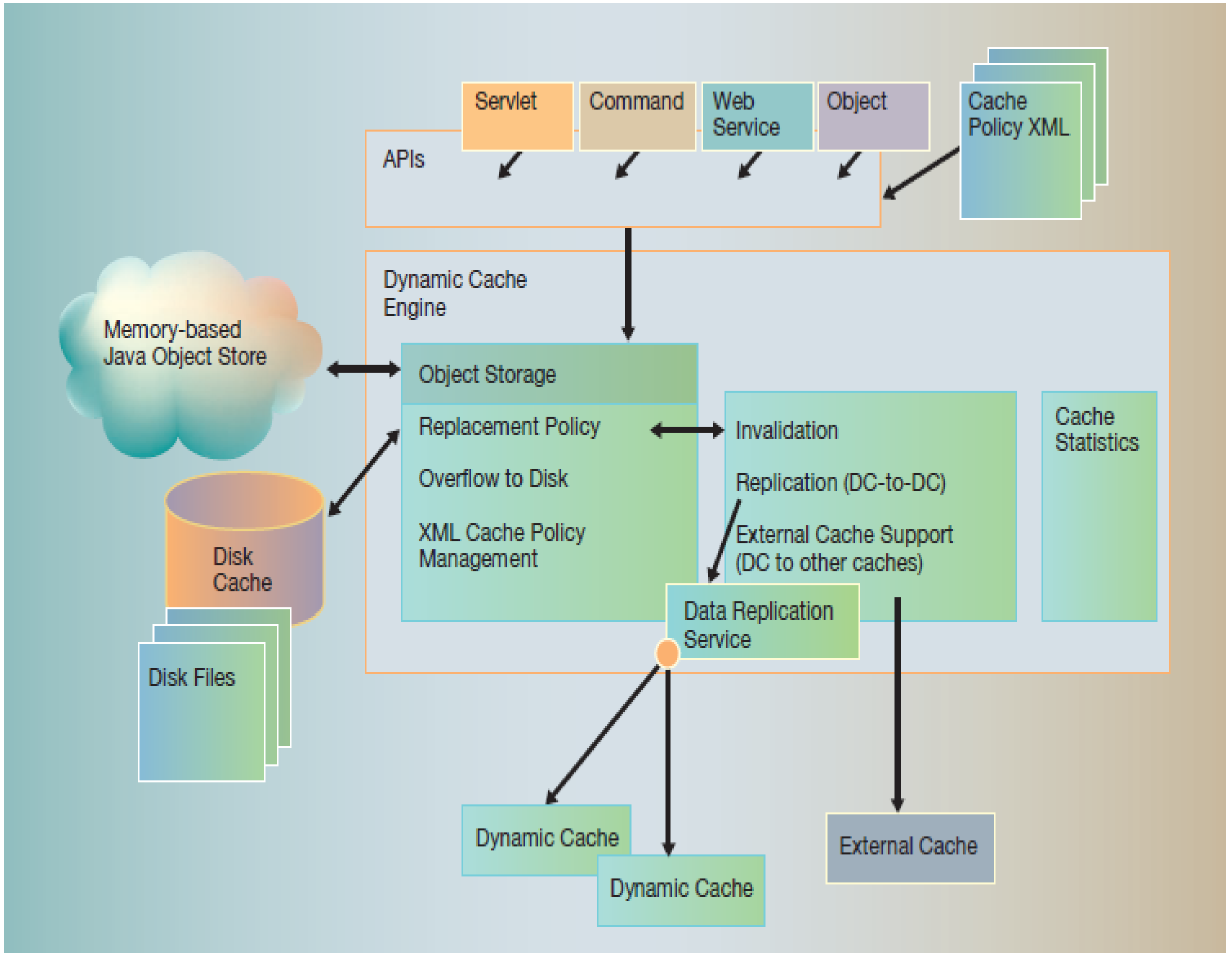
- Dynamic caching
- Policy based caching
- Object caching
- Cache replication
- Monitoring cache
- Dynamic caching tuning
- Problem determination
- Q&A

Dynamic caching

Dynamic caching

- Dynamic cache is part of the IBM® solution for improving performance of J2EE applications running within WebSphere Application Server.
- Dynamic cache improves the performance of J2EE applications by caching objects and results at every level.
- By default dynamic caching is enabled in WebSphere Application Server.
- Provides in memory cache and disk cache.
- Replication support to improve performance.
- There is no time-consuming installation and integration work needed to activate the dynamic cache service.

Dynamic Cache overview:



- Operates in JVM memory and/or disk.
- Distributed replication support for "expensive to recreate" entries/objects..
- Inter-operates with external caches.
- Access via Dynamic cache APIs or cachespec.xml.

Caching types:

Policy based (cachespec.xml)
- Servlet / JSP / Portlet caching.
- Struts and tiles response caching.
- Webservices caching.
- Webservices client(JAXRPC) caching

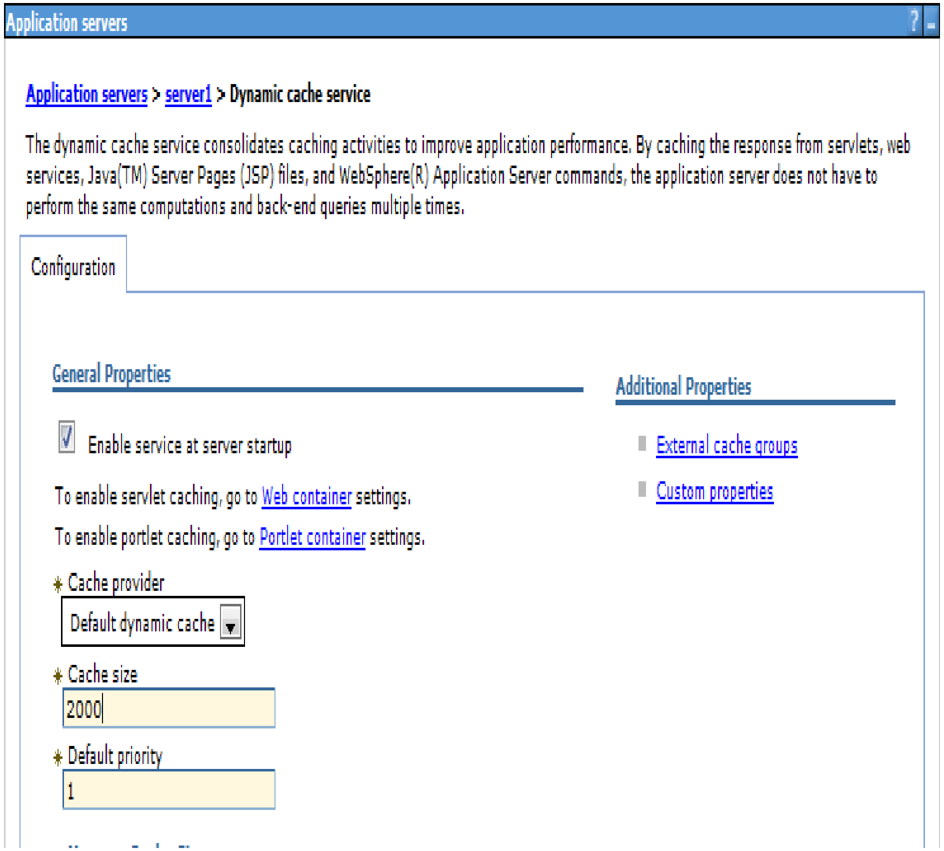
API based
- Distributed map
- Cacheable servlet
- Command cache

Cache instance

- Storing, retrieving and sharing the objects in dynamic cache is done by the application with cache instances.
- Separate cache instance can be configured for each application with different options to provide application better flexibility and tuning of the cache.
- There is one default cache instance in each application server ,known as base cache

- Types of cache instances:
  - Default cache instance for each application server
    - Policy based   - Default is services/cache/basecache
      - Creted only when servlet caching is enabled.
    - Object caching   - Default is services/cache/distributedmap
      - Created during the server startup.
  - Custom cache
    - Servlet cahce instance (cachespec.xml)
    - Object cache instance (com.ibm.websphere.cache.DistributedMap)
- Ways to create custom cache instance.:
  - Admin console
  - Property file
  - web.xml file

Default cache:



Application servers

Application servers > server1 > Dynamic cache service

The dynamic cache service consolidates caching activities to improve application performance. By caching the response from servlets, web services, Java(TM) Server Pages (JSP) files, and WebSphere(R) Application Server commands, the application server does not have to perform the same computations and back-end queries multiple times.

Configuration

General Properties

☑ Enable service at server startup

To enable servlet caching, go to Web container settings.

To enable portlet caching, go to Portlet container settings.

* Cache provider
Default dynamic cache

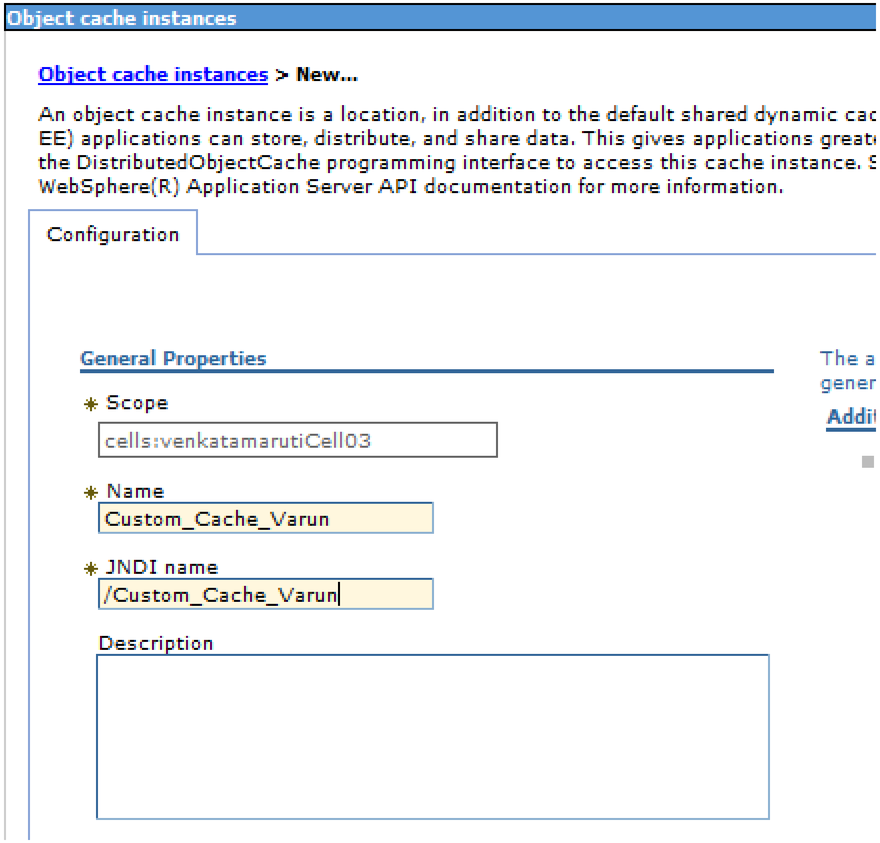
* Cache size
2000

* Default priority
1

Additional Properties

- External cache groups
- Custom properties

Custom cache:



Object cache instances

Object cache instances > New...

An object cache instance is a location, in addition to the default shared dynamic cac
EE) applications can store, distribute, and share data. This gives applications great
the DistributedObjectCache programming interface to access this cache instance. S
WebSphere(R) Application Server API documentation for more information.

Configuration

General Properties

* Scope
cells:venkatamarutiCell03

* Name
Custom_Cache_Varun

* JNDI name
/Custom_Cache_Varun

Description

The a
gener

Addi

Properties file:

cacheinstances.properties:

```
cache.instance.0=/services/cache/instance_one
cache.instance.0.cacheSize=1000
cache.instance.0.enableDiskOffload=true
cache.instance.0.flushToDiskOnStop=true
cache.instance.0.useListenerContext=true
cache.instance.0.enableCacheReplication=false
cache.instance.0.disableDependencyId=false
cache.instance.0.htodCleanupFrequency=60
cache.instance.1=/services/cache/instance_two
cache.instance.1.cacheSize=1500
cache.instance.1.enableDiskOffload=false
cache.instance.1.flushToDiskOnStop=false
cache.instance.1.useListenerContext=false
cache.instance.1.enableCacheReplication=true
cache.instance.1.replicationDomain=DynaCacheCluster
cache.instance.1.disableDependencyId=true
```

Lookup:

```
InitialContext ic = new InitialContext();
 DistributedMap dm1 =

(DistributedMap)ic.lookup("services/cache/instance_one");
```

web.xml file:

web.xml and ibm-web-bnd.xmi:

```
<resource-ref id="ResourceRef_1">
   <res-ref-name>dmap/LayoutCache</res-ref-name>
   <res-type>com.ibm.websphere.cache.DistributedMap
   </res-type>
   <res-auth>Container</res-auth>
   <res-sharing-scope>Shareable</res-sharing-scope>
</resource-ref>
```

```
<webapp href="WEB-INF/web.xml#WebApp_ID"/>
<resRefBindings xmi:id="ResourceRefBinding_1"
jndiName="services/cache/instance_one">
<bindingResourceRef href="WEB- INF/web.xml
#ResourceRef_1"/>
</resRefBindings>
```

Lookup:

```
InitialContext ic = new InitialContext();
DistributedMap dma =(DistributedMap)ic.lookup
                           ("java:comp/env/dmap/LayoutCache");
```

Policy based caching

Policy file: cachespec.xml

- Policy based caching is done by the rules specified from cachespec.xml file.
- The rules defined in policy file will decide what should be cached , when to be cached along with various other settings.
- Cachespec.xml file is a deployable xml file and can be found in WEB-INF directory of a web module.
- Global cachespec.xml file can be found in the application server properties directory.
- Dynamic cache engine parses the cachespec.xml file when the server starts.

- Policy file will have the following information:
  - Specify the cache-instance name where the cache rules to be applied
  - What to be cached
  - How to identify an item in cache
  - Where to cache
  - When to invalidate
  - How to handle dependencies
  - How to share across servers in cluster

- **NOTE:** If there are two cachespec.xml files then both the cachespc.xml file entries are merged, If there are conflicting entries in the two files, the entries in the in the cachespec.xml file that is in the application override the entries in the global cachespec.xml file for that application

cachespec.xml :

```
<cache-entry>
      <class>servlet</class>
      <name>inventory</name>
      <cache-id>
        <component id=“pid" type="parameter">
           <required>true</required>
        </component>
        <component id=“action" type="parameter">
           <required>true</required>
           <value>getDetails</value>
      </cache-id>
     <dependency-id>update
        <component id="action" type="parameter" ignore-value="true">
            <required>true</required>
            <value>getDetails</value>
        </component>
        <component id="pid" type="parameter">
            <required>true</required>
        </component>
    </dependency-id>
    <invalidation>update
        <component id="action" type="parameter" ignore-
value="true">
            <required>true</required>
            <value>putDetails</value>
        </component>
        <component id="pid" type="parameter">
            <required>true</required>
        </component>
    </invalidation>
</cache-entry>
```

Cache-entry will have cacheId and cache data

Cache-entry will also have other information like dependency Id’s, invalidation rules, replication type etc.

Example:

```
<cache-entry>
     <class>servlet</class>
     <name>QuoteServlet</name>
     <cache-id>
       <component id=“locale"
                  type=“locale">
         <required>false</required>
       </component>
      </cache-id>
 </cache-entry>
```

URL:
http://www.hostname.com/**QuoteServlet**

Cache id =
/QuoteServlet:requestType=GET

Cache id =
/QuoteServlet:locale=en_US:requestType=GE
T

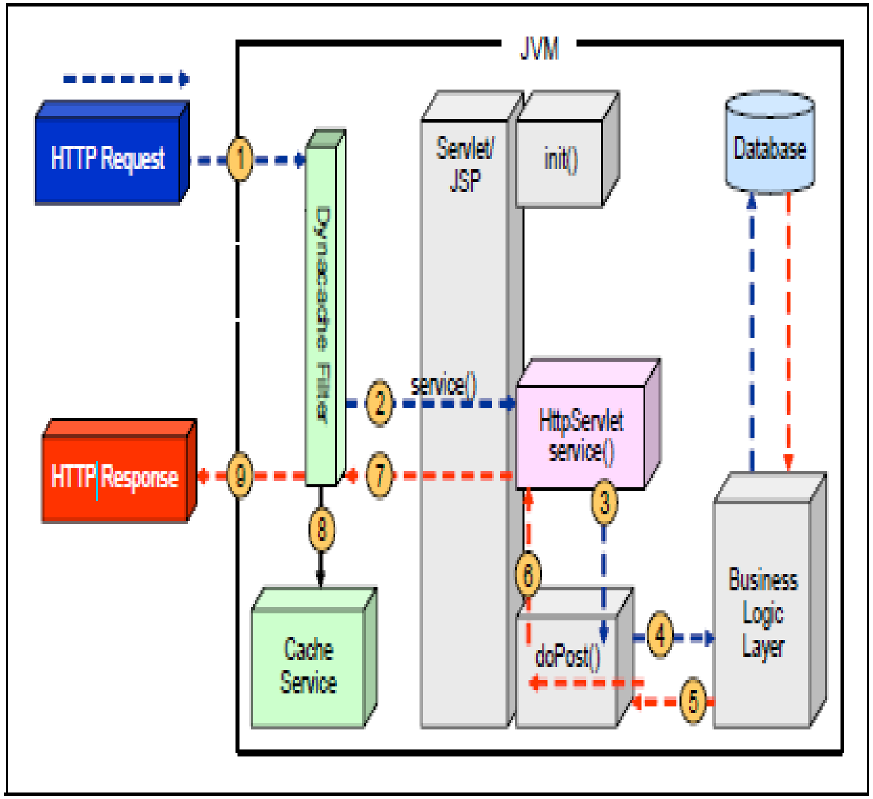
Cache id =
/QuoteServlet:locale=ja_JP:requestType=GET

Servlet/JSP caching:

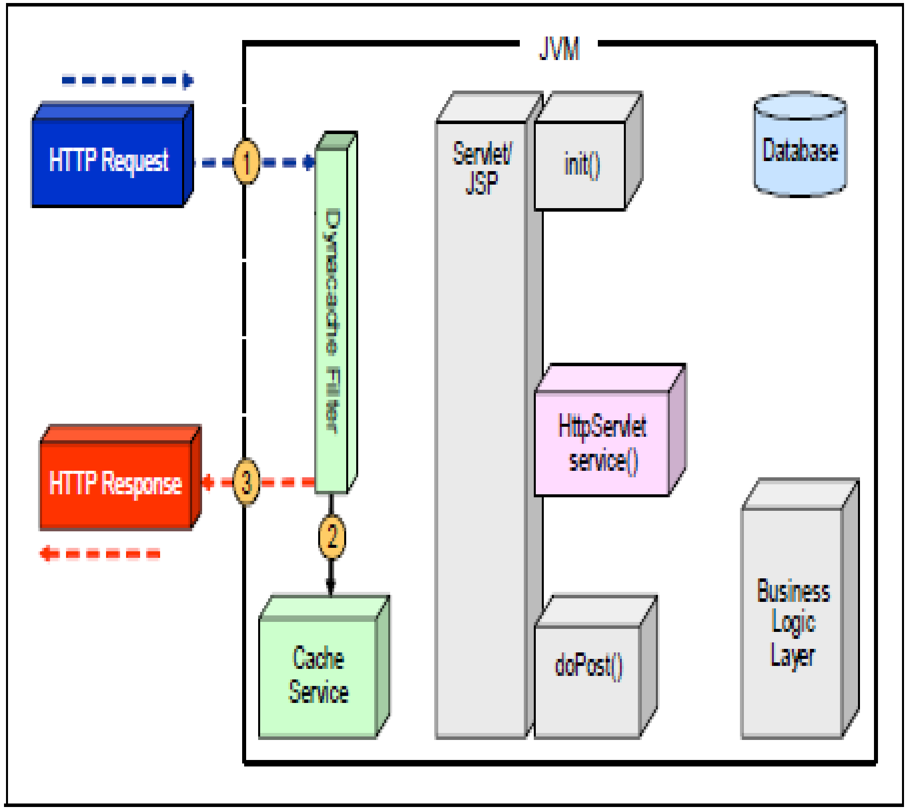
- Servlet/JSP caching needs to be explicitly enabled from admin console.
- Servlet/JSP response caching is based on the rules defined in the policy file – cachespec.xml.
- The default cache instance will be used if custom cache instance is NOT defined.
- Dynamic cache intercepts the service() method and decides, whether the invocation can be served from cache.
- The servlet is invoked to generate the output that will be cached. The resulting cache entry contains the output.
- Servlet/JSP result caching can be based on:
    - Request parameters and attributes
    - The URI used to invoke the servlet or JSP
    - Session information
    - Other options, including cookies

CACHE MISS & CACHE HIT

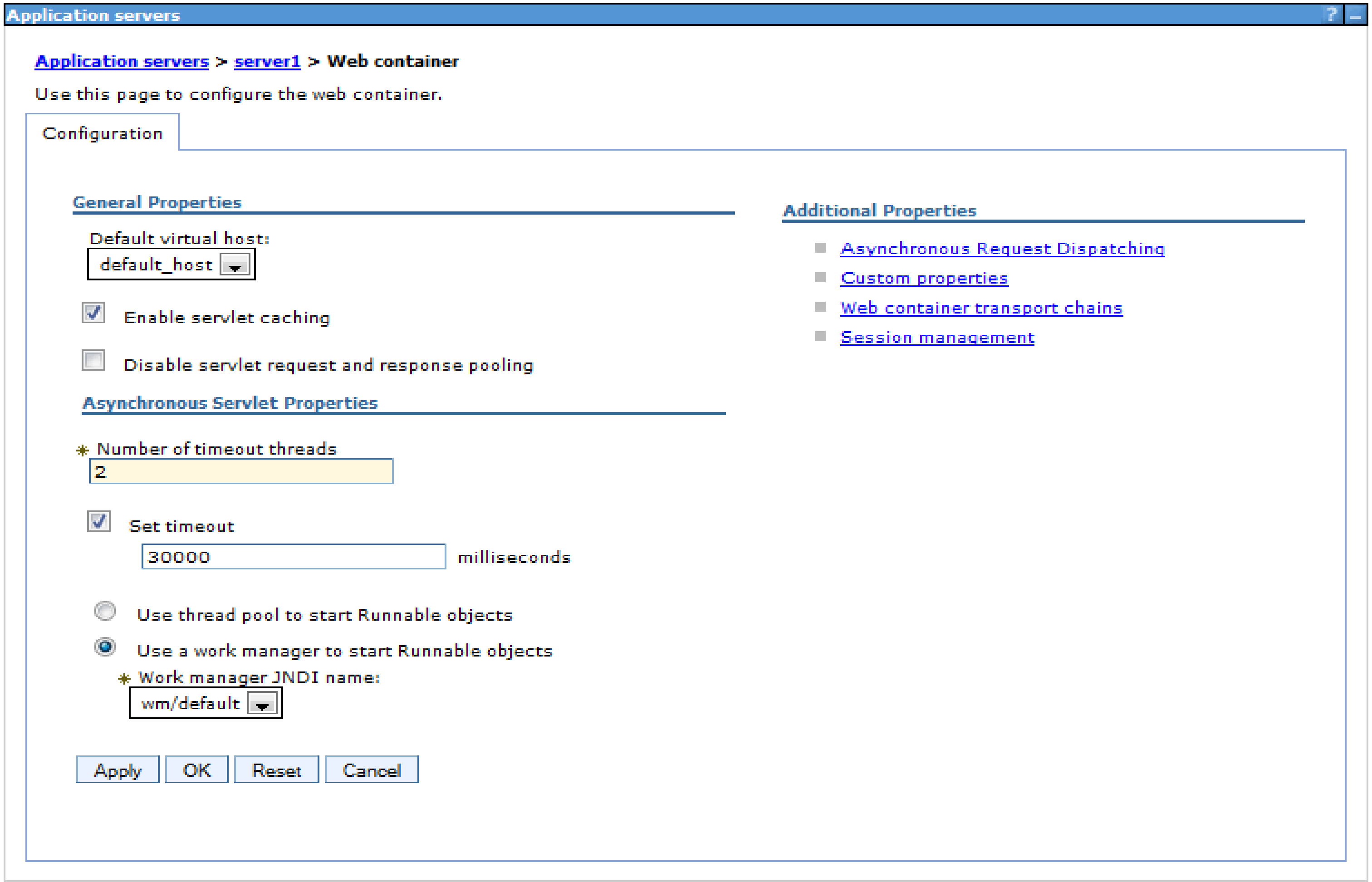
CACHE MISS (1,2,3,4,5,6,7,8,9)



CACHE HIT (1,2,3)



Servlet caching



Application servers > server1 > Web container

Use this page to configure the web container.

Configuration

General Properties

Default virtual host:
default_host

☑ Enable servlet caching

□ Disable servlet request and response pooling

Asynchronous Servlet Properties

* Number of timeout threads
2

☑ Set timeout
30000 milliseconds

○ Use thread pool to start Runnable objects

● Use a work manager to start Runnable objects

* Work manager JNDI name:
wm/default

Additional Properties

- Asynchronous Request Dispatching
- Custom properties
- Web container transport chains
- Session management

Object caching

Dynamic caching: Object caching

- WebSphere dynamic caching provides simple API (DistributedMap and DistributedObjectCache) to cache objects.
- Using these interfaces, J2EE applications can inspect and manipulate the cache.
- Dynamic cache API provides many methods to manipulate the cache. Please refer information center for the Dynamic cache API.
- If the custom cache instance is **NOT** created, default cache is used.

  JNDI name - “services/cache/distributedmap”
- JNDI lookup for custom cache instance “instance_one”

  ```
  InitialContext ic = new InitialContext();
  DistributedMap dm = (DistributedMap)ic.lookup("services/cache/instance_one");
  ```

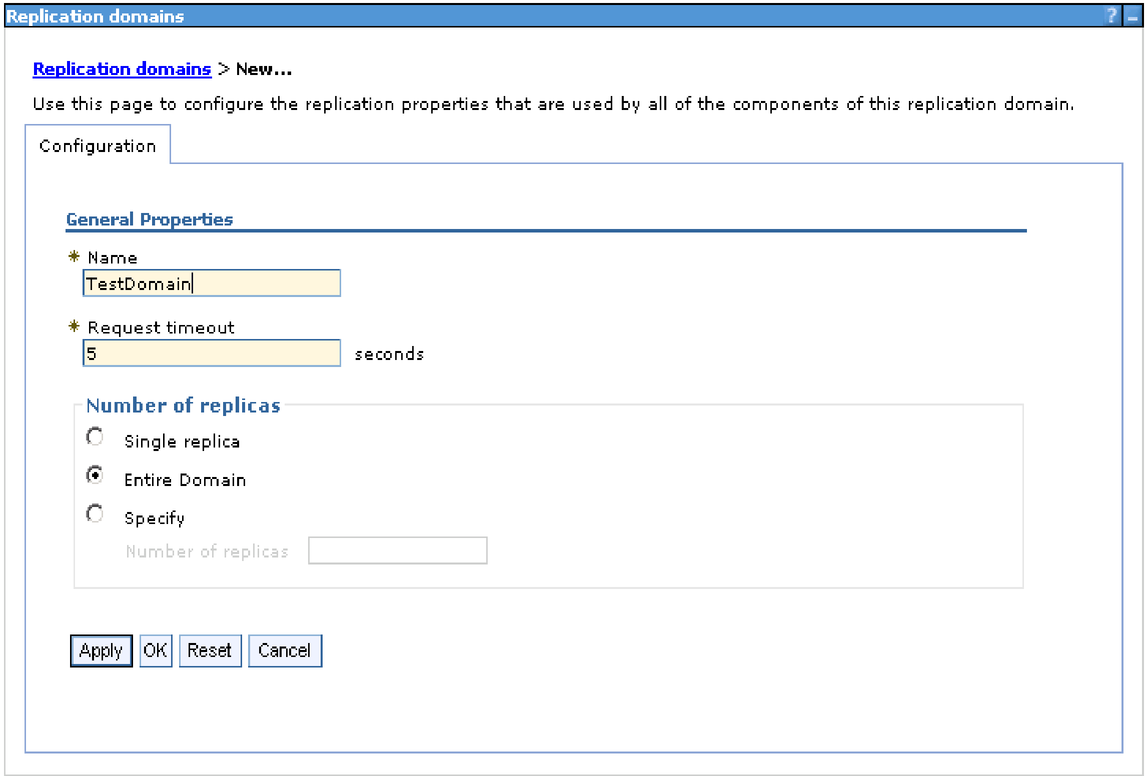
DRS and Cache replication

Data replication service (DRS)

- The data replication service (DRS) is an internal WebSphere Application Server Component.
- Replication is a service that transfers data, objects, or events among application servers.
- Dynamic cache replication uses the DRS to replicate data across many application Servers in that domain.
- Responsible for serialization and de-serialization of messages.
- Http Session,SIP and Stateful session beans are other consumers of DRS.
- Replication domain should be created (Either manually or while creating the cluster) before configuring the DRS.
- Replication domain is required to configured with “Entire Domain” as number of replicas.
- DRS uses HAManager internally to replicate the data across servers in that domain.

NOTE: Do not use same replication domain for Dynamic caching and sessions/Stateful.

Replication domain



Replication domains

Replication domains > New...

Use this page to configure the replication properties that are used by all of the components of this replication domain.

Configuration

General Properties

* Name

TestDomain

* Request timeout

5 seconds

Number of replicas

○ Single replica

● Entire Domain

○ Specify

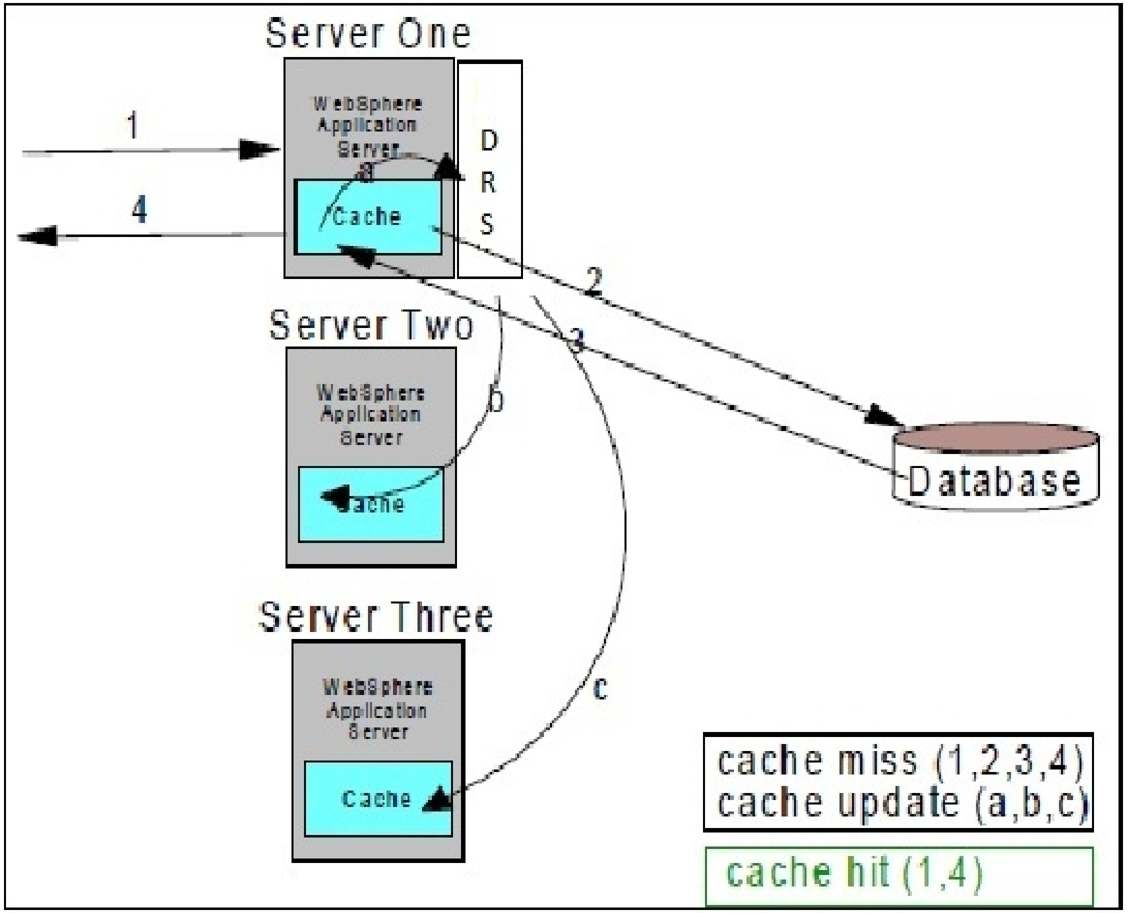
Number of replicas

Navigate to: Environment > Replication domains > New

Cache replication

- What happens, if cache replication is not configured:
  - Data need to be generated on each server.
  - Inconsistency of data across servers.
- With replication, data is generated one time and copied or replicated to other servers in the cluster, saving time and resources.
- Replication support further extends the value of caching in an e-business application.
- Cache replication maintains cache consistency across the domain.
- The replication can be turned on from container services of dynamic cache panel.
- Ways to configure Replication:
  - Admin console
  - Cachespec.xml (For Policy based)
  - Property file.
  - With API calls

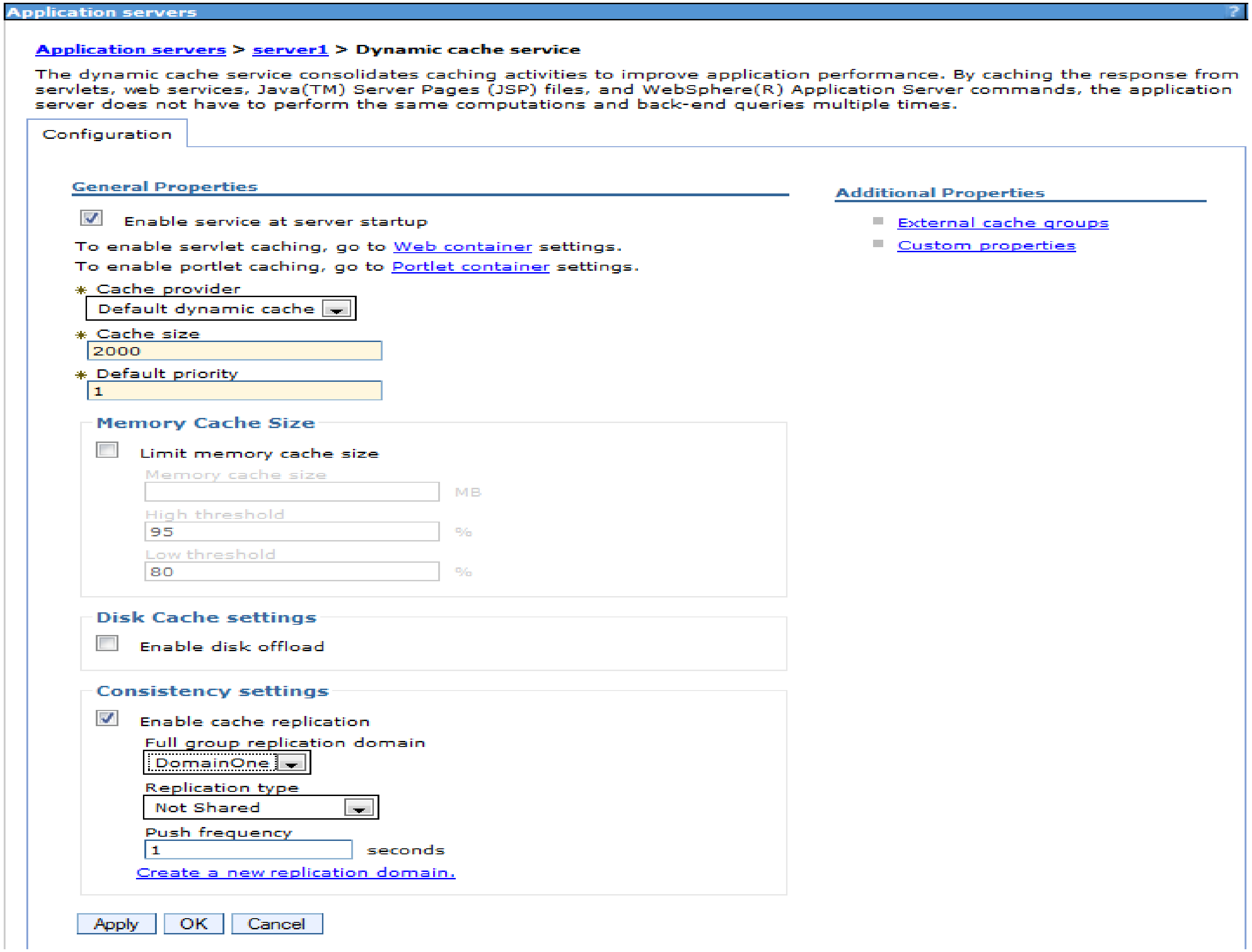
Cache replication



When a fresh request arrives, The request hits any of the three application servers. The requested data will be fetched from database and cached at server if the data is not found in the cache (1,2,3,4).

Dynamic cache detects the updates and replicates (with DRS) the updates to the other servers in that domain (a,b,c) based on the replication type.

Dynamic cache configuration with default cache



Application servers

Application servers > server1 > Dynamic cache service

The dynamic cache service consolidates caching activities to improve application performance. By caching the response from servlets, web services, Java(TM) Server Pages (JSP) files, and WebSphere(R) Application Server commands, the application server does not have to perform the same computations and back-end queries multiple times.

Configuration

General Properties

☑ Enable service at server startup

To enable servlet caching, go to Web container settings.

To enable portlet caching, go to Portlet container settings.

* Cache provider: Default dynamic cache

* Cache size: 2000

* Default priority: 1

Memory Cache Size

□ Limit memory cache size

Memory cache size: MB

High threshold: 95 %

Low threshold: 80 %

Disk Cache settings

□ Enable disk offload

Consistency settings

☑ Enable cache replication

Full group replication domain: DomainOne

Replication type: Not Shared

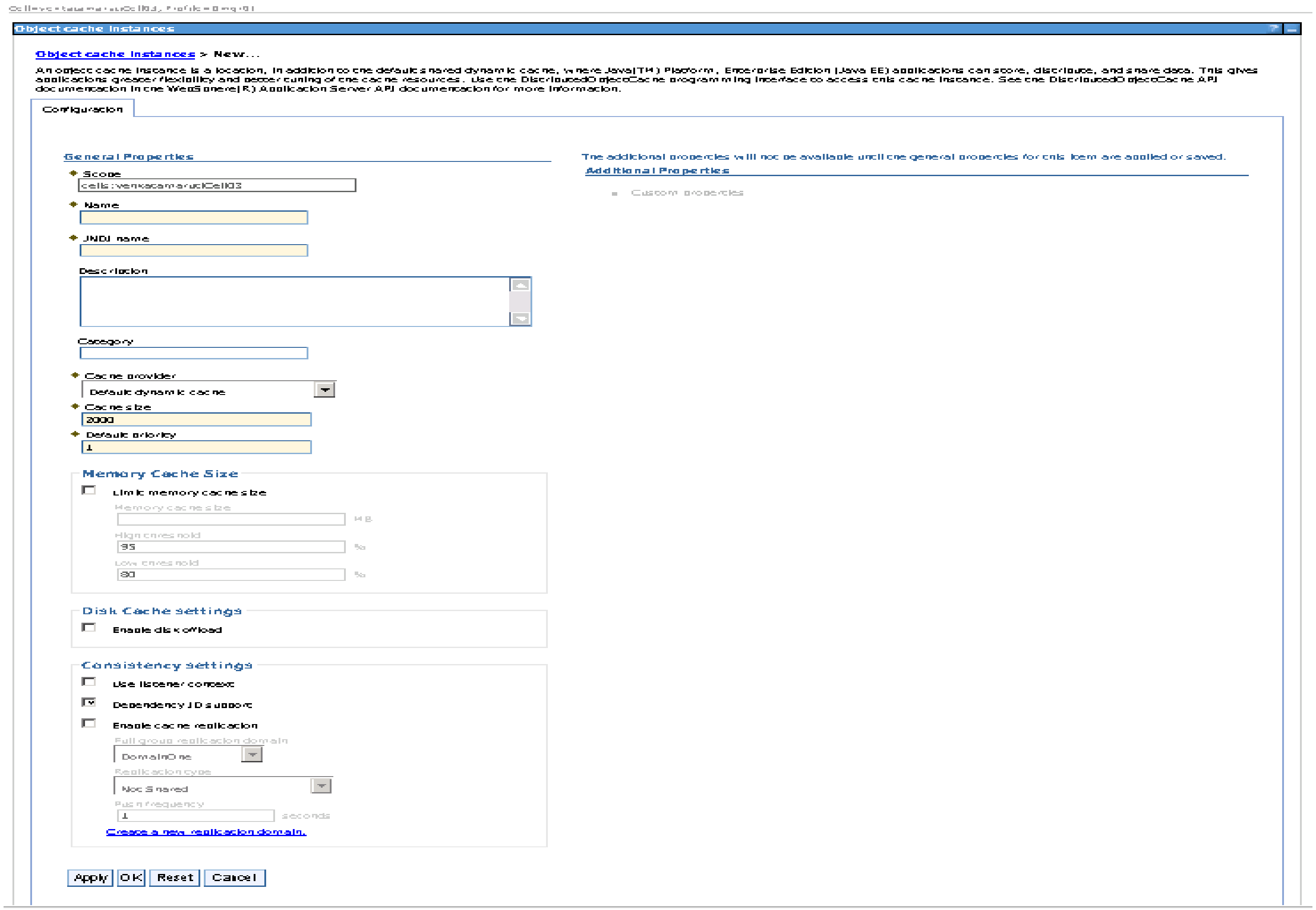
Push frequency: 1 seconds

Create a new replication domain.

Additional Properties

- External cache groups
- Custom properties

Dynamic cache configuration with custom cache



Cell=venkatamaruCell03, Profile=Dmgr01

Object cache instances

Object cache instances > New...

An object cache instance is a location, in addition to the default shared dynamic cache, where Java(TM) Platform, Enterprise Edition (Java EE) applications can store, distribute, and share data. This gives applications greater flexibility and better tuning of the cache resources. Use the DistributedObjectCache programming interface to access this cache instance. See the DistributedObjectCache API documentation in the WebSphere(R) Application Server API documentation for more information.

Configuration

General Properties

* Scope: cells:venkatamaruCell03

* Name:

* JNDI name:

Description:

Category:

* Cache provider: Default dynamic cache

* Cache size: 2000

* Default priority: 1

Memory Cache Size

☐ Limit memory cache size

Memory cache size: MB

High threshold: 95 %

Low threshold: 80 %

Disk Cache settings

☐ Enable disk offload

Consistency settings

☐ Use listener context

☑ Dependency ID support

☐ Enable cache replication

Full group replication domain: DomainOne

Replication type: Not Shared

Push frequency: 1 seconds

Create a new replication domain.

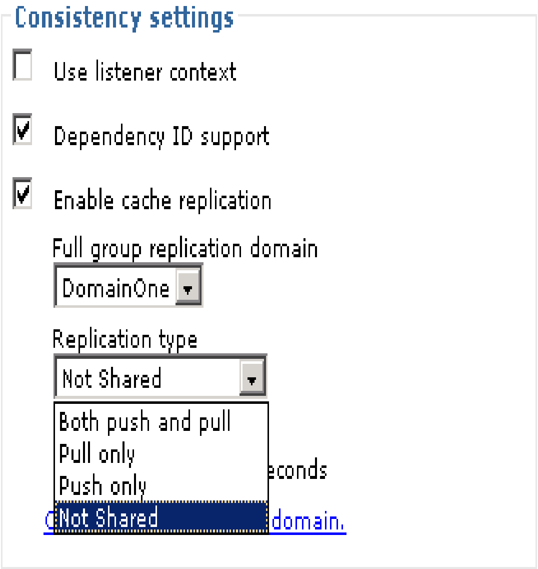
The additional properties will not be available until the general properties for this item are applied or saved.

Additional Properties

- Custom properties

Dynacache Configuration: Replication types

- Both push and pull
  - sends only cache ids for both new and updated content to all other servers in the replication domain.
  - Invalidation events are propagated
- Push only
  - sends cache id and content for both new and updated cache content to all other servers in the replication domain.
  - Invalidation events are propagated.
- Not shared
  - Data will not get replicate to other servers in the domain.
  - Explicit invalidation events are propagated to the other servers in the domain.
  - When enabling replication, the default value is Not Shared



Consistency settings

□ Use listener context

☑ Dependency ID support

☑ Enable cache replication

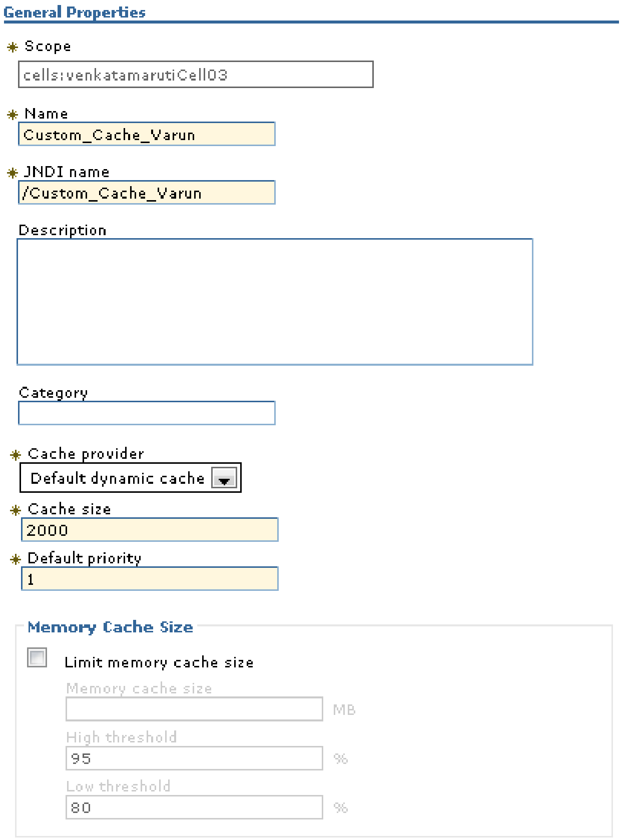
Full group replication domain: DomainOne

Replication type: Not Shared

Dynacache configuration: Properties

- Scope
- Name
- JNDI name
- Cache size
- Default priority

- Limit cache size
  - Memory cache size
  - High threshold (95%)
  - Low threshold (80%)



General Properties

* Scope

cells:venkatamarutiCell03

* Name

Custom_Cache_Varun

* JNDI name

/Custom_Cache_Varun

Description

Category

* Cache provider

Default dynamic cache

* Cache size

2000

* Default priority

1

Memory Cache Size

□ Limit memory cache size

Memory cache size MB

High threshold

95 %

Low threshold

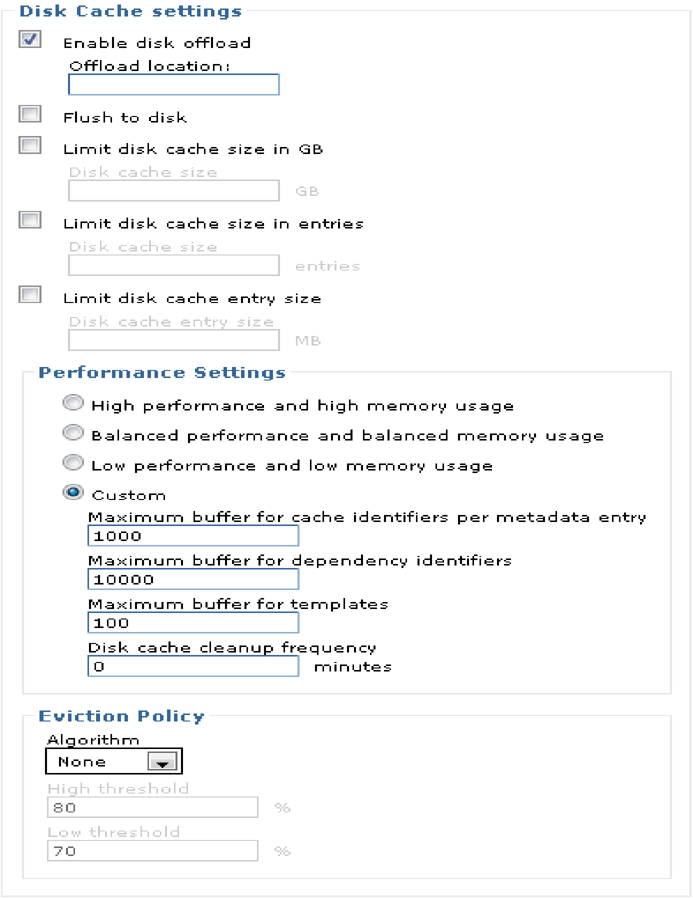
80 %

Dynacache configuration: Disk offload

- Offload location
- Flush to disk
- Limit disk cache size

- Performance settings
  - High performance and high memory usage
  - Balanced performance and balanced memory usage
  - Low performance and low memory usage
- Custom

- Eviction policy
  - Algorithm
    - None
    - Random
    - Size
  - High threshold
  - Low threshold



Disk Cache settings

☑ Enable disk offload

Offload location:

☐ Flush to disk

☐ Limit disk cache size in GB

Disk cache size GB

☐ Limit disk cache size in entries

Disk cache size entries

☐ Limit disk cache entry size

Disk cache entry size MB

Performance Settings

○ High performance and high memory usage

○ Balanced performance and balanced memory usage

○ Low performance and low memory usage

● Custom

Maximum buffer for cache identifiers per metadata entry: 1000

Maximum buffer for dependency identifiers: 10000

Maximum buffer for templates: 100

Disk cache cleanup frequency: 0 minutes

Eviction Policy

Algorithm: None

High threshold: 80 %

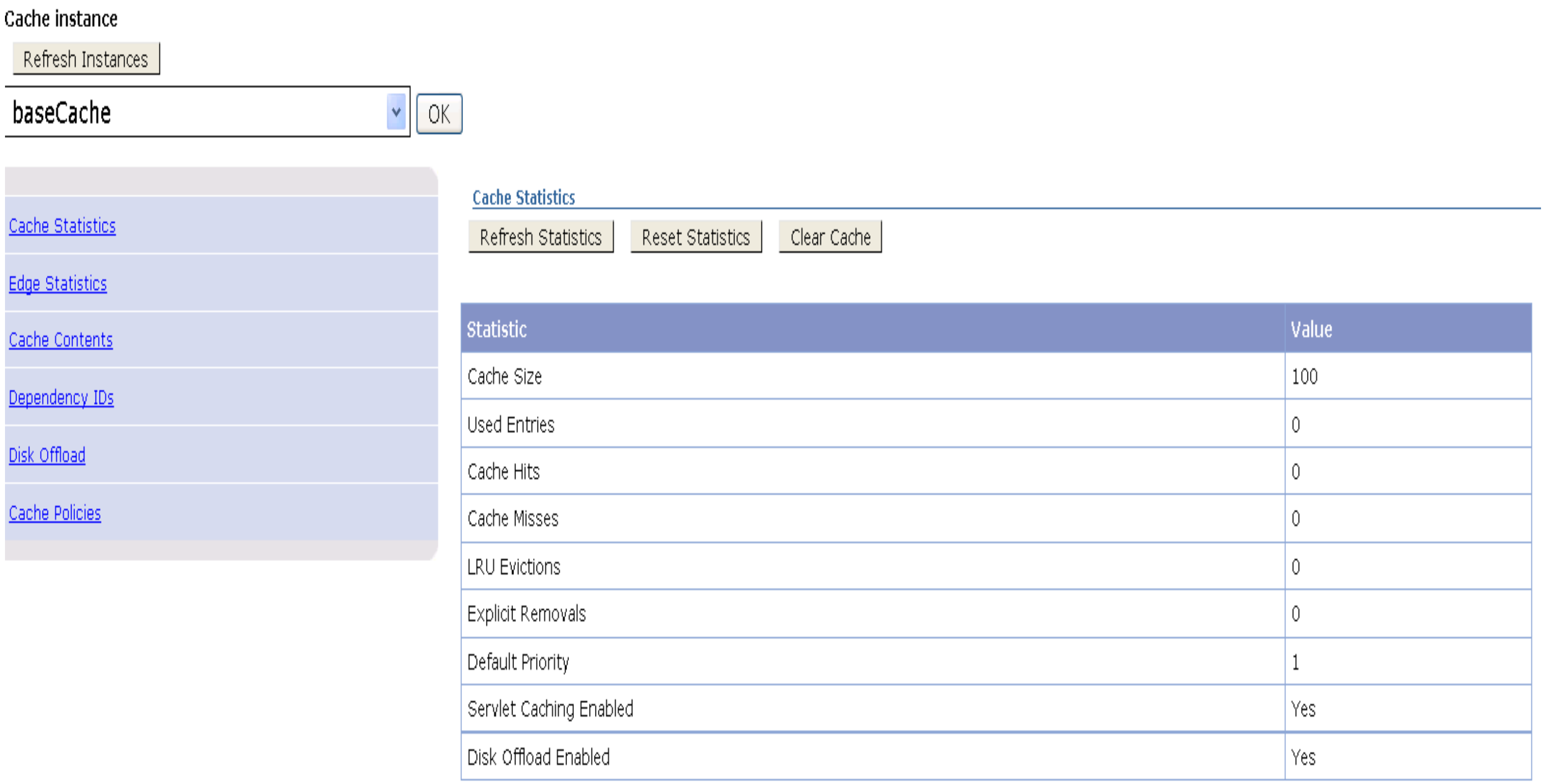
Low threshold: 70 %

Monitoring cache

Cachemonitor application

- Shipped with WebSphere application server install in the installableApps dir.
- Cachemonitor application is used to view -
    - Cache statistics
    - Cache policies
    - Existing state of dynamic cache
    - Configuration of dynamic cache
    - The data offloaded to the disk
    - Data in the edge cache
    - And manage the data in the cache
        - Remove an entry from a cache instance
        - Remove all entries for a certain dependency ID
        - Move an entry from the disk to the memory within a cache instance
        - Clear the entire contents of the cache instance
        - Clear the contents of the disk for the cache instance

Cachemonitor


Cache instance

Refresh Instances

baseCache OK

Cache Statistics

Edge Statistics

Cache Contents

Dependency IDs

Disk Offload

Cache Policies

Cache Statistics

Refresh Statistics | Reset Statistics | Clear Cache

Statistic	Value
Cache Size	100
Used Entries	0
Cache Hits	0
Cache Misses	0
LRU Evictions	0
Explicit Removals	0
Default Priority	1
Servlet Caching Enabled	Yes
Disk Offload Enabled	Yes

Extended cachemonitor application

- Used for monitoring the object cache
- Monitor cache statistics for cache instances across all members of the cluster.
- Available to download from developer works- http://www.ibm.com/developerworks/websphere/downloads/cache_monitor.html

Dynamic cache tuning

Dynamic cache tuning

- Change replication type of the replication domain to either NOT_SHARED or PUSH depending on your application requirement.

- Set the following as JVM custom properties or as properties on your servlet/object cache instance to reduce redundant DRS traffic, improve performance & force creation of cache instances

```
com.ibm.ws.cache.CacheConfig.filterLRUInvalidation = true
com.ibm.ws.cache.CacheConfig.filterTimeOutInvalidation = true
com.ibm.ws.cache.CacheConfig.filterInactivityInvalidation= true
com.ibm.ws.cache.CacheConfig.ignoreValueInInvalidationEvent = true
com.ibm.ws.cache.CacheConfig.useServerClassLoader = true  -   For Portal
com.ibm.ws.cache.CacheConfig.createCacheAtServerStartup = true
com.ibm.ws.cache.CacheConfig.cacheEntryWindow= 50
com.ibm.ws.cache.CacheConfig.cacheInvalidateEntryWindow= 50
com.ibm.ws.cache.CacheConfig.cachePercentageWindow - 2%
com.ibm.ws.cache.CacheConfig.batchUpdateMilliseconds - 1 sec (default)
com.ibm.ws.cache.CacheConfig.cacheInvalidatePercentWindow -2%
```

Dynacache Tuning

- Read the following technote on High Availability Manager (HAManager) & Data Consistency Services (DCS) Tuning
  http://www-01.ibm.com/support/docview.wss?uid=swg21251873

- Read Data Replication Service (DRS) Tuning
  Configure DRS to use its own threadpool *default since 7.0.0.15
  http://www-01.ibm.com/support/docview.wss?uid=swg1PM16027

- Read the following Technote for Portal Server Cache Replication issues
  http://www-01.ibm.com/support/docview.wss?uid=swg21322640

- Read the following Technote for Commerce server replication issues
  http://www-01.ibm.com/support/docview.wss?uid=swg21358672

- Read the following document for cache replication tuning
  http://www-01.ibm.com/support/docview.wss?uid=swg27006431

Problem determination

Objects are not replicated

- Make sure the replication instance is launched
- Make sure you are using PUSH mode.
- If PUSH_PULL mode is used, only cache Id will be pushed.
- The following messages in SystemOut.log file tells Dynamic cache instance is initialized and replication service is launched. There is a delay between initialization and replication instance launching. If application is looking for replicated data during the delay, it may not find it.

```
[2008-07-10 18:56:01:828 EST] 0000002c CacheServiceI I DYNA1001I: WebSphere
Dynamic Cache instance named /cache/instance_one initialized successfully.
[2008-07-10 18:56:02:828 EST] 00000042 DRSBootstrapM A CWWDR0001I:
Replication instance launched : /cache/instance_one
```

- Introduce small delay in application code.
- Please refer the following technote:

http://www-01.ibm.com/support/docview.wss?uid=swg21313480

Dynamic cache and DRS messages

- Dynamic cache messages start with “DYNA” and DRS messages start with “CWWDR”.

- **Positive messages**
  - WSVR0049I: Binding services/cache/diskoffload as services/cache/diskoffload
  - DYNA0053I: Offload to disk is enabled
  - DYNA0048I: WebSphere Dynamic Cache initialized successfully.
  - DYNA0062I: Successfully loaded cache-instances from configuration file
  - DYNA1001I: WebSphere Dynamic Cache instance named /cache/instance_one initialized successfully.
  - CWWDR0001I: Replication instance launched : /cache/instance_one

- **Problematic messages**
  - DYNA0045E: XML parsing error: The markup in the document following the root element must be well-formed.
  - DYNA0037E: Unrecognized element property while processing element <property>.
  - CWWDR0010E: Replication instance /cache/instance_one caught exception when sending/receiving messages

Dynacache Congestion

- when ever there is lot of data stored in cache and a member joins a HAManager view Or if there are lot of invalidation events across servers, the Congestion occurs.
- Congestion can also occur in Push mode if individual cached objects are large.
- The following messages indicates that the Congestion occurs:
  - DCSV1051W/DCSV1052W: DCS Stack DefaultCoreGroup.MyCluster at Member CustomServer1: Raised a high severity congestion event for outgoing messages.
  - HMGR0152W: CPU Starvation detected. Current thread scheduling delay is 149 seconds.
- Use the NOT_SHARED or PUSH_PULL replication mode.

References

References

WebSphere Applicaton Server information center
http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp

WebSphere scalability and performance handbook
http://w3.itso.ibm.com/abstracts/sg246392.html?Open

IBM journal
http://www.research.ibm.com/journal/sj/432/bakalova.html

Caching WebSphere commerce pages.
http://www.ibm.com/developerworks/websphere/library/techarticles/0405_caching/0405_caching.html

More about high volume websites
http://www.redbooks.ibm.com/abstracts/SG246562.html

Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at: http://www.ibm.com/software/websphere/support/supp_tech.html
- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at: http://www.ibm.com/developerworks/websphere/community/
- Join the Global WebSphere Community: http://www.websphereusergroup.org
- Access key product show-me demos and tutorials by visiting IBM Education Assistant: http://www.ibm.com/software/info/education/assistant
- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically: http://www.ibm.com/software/websphere/support/d2w.html
- Sign up to receive weekly technical My Notifications emails: http://www.ibm.com/software/support/einfo.html

Connect with us!

1. **Get notified on upcoming webcasts**
   Send an e-mail to wsehelp@us.ibm.com with subject line “wste subscribe” to get a list of mailing lists and to subscribe

2. **Tell us what you want to learn**
   Send us suggestions for future topics or improvements about our webcasts to wsehelp@us.ibm.com

3. **Be connected!**
   Connect with us on Facebook
   Connect with us on Twitter

Questions and Answers