



GLASSFISH V2 CLUSTERING AND LOAD BALANCING

Satyajit Tripathi

ISV-Engineering, Sun Microsystems



OPEN



Project GlassFish

GlassFish V2 UR2

Open for Business – Deploy and Redistribute for free

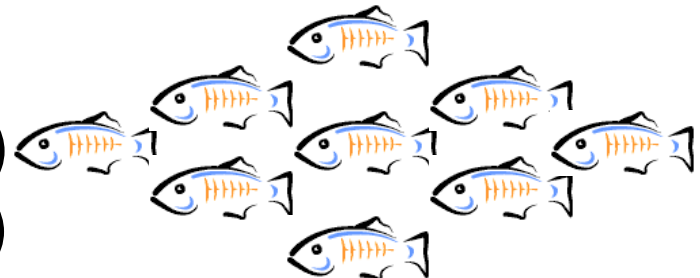
- Latest release in August 2008 ([Download](#)). 3-Months cycle
- Production Quality. First Application Server with Java EE 5
- Code base as Sun Java System Application Server 9.1 U2
- Platform Of Choice: Solaris, MS Windows, Linux, MacOS, AIX
- Open Source. OSI License CDDL or GPL-v2, with CPE
- Advanced features supported
 - > Clustering (HA and Scalability). In-Memory Replication (**new feature!**)
 - > Centralized Administration and Monitoring – CLI and GUI based
 - > Web Service Interoperability and JBI (JSR 208) compliant runtime
 - > Improved Self Management capabilities & best-in-class Performance
- Attractive Support Pricing [subscription](#) based



GlassFish V2 Clusters

Centralized Administration, High Availability, Scalability

- GlassFish V2 supports three different Administration profiles
 - > Developer : No clustering features or NSS keystore
 - > Cluster : Clustering features without HADB and NSS keystore
 - > Enterprise : HADB or Network Security Services (NSS)
- Setup different profiles using setup.xml or setup-cluster.xml command `${glassfish.home}/lib/ant/bin/ant -f setup-cluster.xml`
- Profile attribute `AS_ADMIN_PROFILE` in the file `${glassfish.home}/config/asadminenv.conf`
- Enhanced Cluster capabilities
 - > Memory Replication or HADB (99.999%)
 - > Dynamic Java Clustering ([Project Shoal](#))



GlassFish Cluster Administration

asadmin>

Home | Version
Logout | Help

User: admin | Domain: domain1 | Server: localhost

Sun Java™ System Application Server Admin Console

Common Tasks

- Registration
- Domain
- Applications
- Web Services
- JB1
- Custom MBeans
- Resources
- Clusters
 - cluster1
 - instance-ONE
 - instance-TWO
- Stand-Alone Instances
- HTTP Load Balancers
- Node Agents
 - node1
- Configurations

Clusters > cluster1

General | Applications | Instances | Resources | JB1

Clustered Server Instances Save

Before a server instance can be started or stopped, its node agent must be running. Refer to the online help for more information.

Server Instances (2)

| New... | Delete | Start | Stop

	Name	Weight	Configuration	Node Agent	Status
<input type="checkbox"/>	instance-TWO	<input type="text" value="100"/>	cluster1-config	node1	✔ Running
<input type="checkbox"/>	instance-ONE	<input type="text" value="100"/>	cluster1-config	node1	✔ Running

1

5

3

4

2



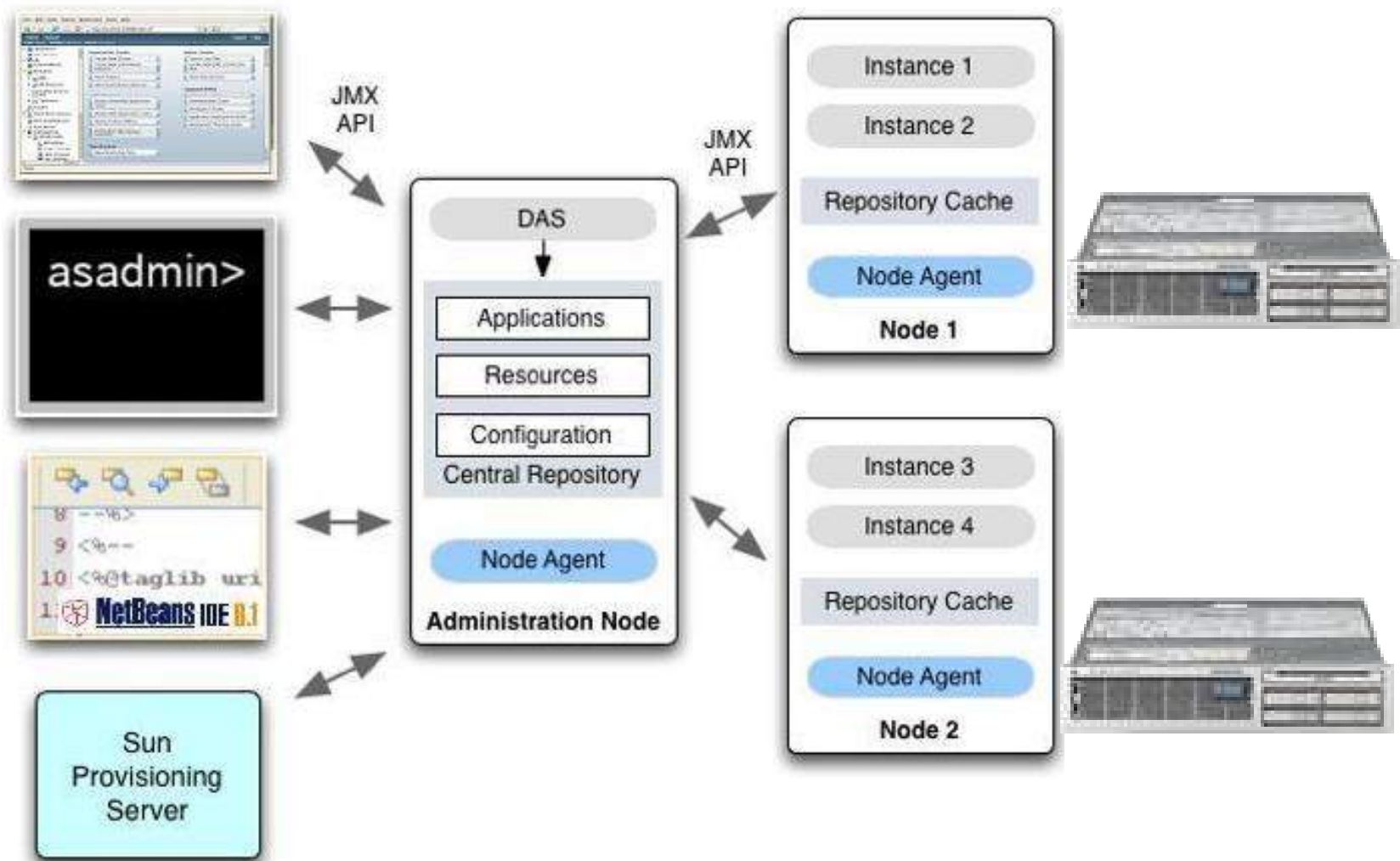
Cluster Components

Architecture

- Domain Administration Server (DAS)
- Node Agent
- Cluster
- Server Instance
- Application
- Group Management Service (GMS)

Domain Administration Architecture

Centralized Administration





Domain Administration Server

Based on Java Management Extensions (JMX) Technology

- At runtime the process manages itself, server instances, clusters, applications, and resources. Comprise of set of configuration files.
- Domain Administration Server (DAS) must run on a physical host
- DAS maintains Central Server Repository Cache
- Delegates management of server instances to the Node Agent
- By itself can serve as a fully compliant Java EE server
- Command tool `asadmin`>
 - > `create-domain [--adminport][--profile][--domaindir] <domain-name>`
 - > `start-domain <domain-name>`
 - > `list-domains [--host][--port][--user][--password]`
- JMX (JSR3 and JSR160) compliant Managed Bean (MBean) supports dynamic resource and configuration instrumentation



Node Agent

Manages Server Instances on Hosting system

- Controls the life cycle of server instance(s) on the host
- Need to run on every physical host running server instance(s)
- Delegated by the DAS to manage the server instances
- Maintains local Instance Repository Cache
- Synchronizes Instances with Central DAS Repository Cache
- Command tool `asadmin>` (*default master password: changeit*)
 - > `create-node-agent <node-agent-name>`
 - > `start-node-agent [--syncinstances=true|false] <node-agent-name>`
 - > `list-node-agents`
- Monitors (“watchdog”) the server instance(s) and notifies DAS



Cluster

High Availability (HA) and Session Replication

- A logical entity, binds multiple homogeneous server instances
- Performs fail over between server instances transparent to LB
- Heartbeat should be Enabled (GMS described later)
- Command tool asadmin>
 - > create-cluster [--host][--port] <cluster-name>
 - > start-cluster <cluster-name> (**Starts all Server Instances*)
 - > list-clusters
- HA and Session Replication can be achieved either through
 - > Memory Replication OR
 - > HADB
- **JXTA**TM technology for inter process communication



Server Instance

Java EE 5 Compliant Application Server

- Server instance must run on a physical host and requires a JVM
- Runs as Java EE server process and hosts Java EE application
- Instances within the cluster have homogeneous configuration
- Command tool `asadmin`>
 - > `create-instance [--host][--port][--cluster][--nodeagent] <instance-name>`
 - > `start-instance <instance-name>` (**Node agent must be running*)
 - > `list-instances`
- Server Instances listen through
 - > HTTP : Listener Port and SSL_Listener Port
 - > IIOP : Listener Port, SSL_Listener Port and SSL_MutualAuth Port
 - > JMS Provider Port
 - > JMX System_Connector Port



Group Management Service

Enable for Dynamic Clustering and In-Memory Replication

- GMS provides dynamic membership information about cluster and its member instances
- Provides Notification event model for runtime shape change events in the cluster and Management Rules
- Core is based on open source **JXTA**TM (Juxtapose) technology
- Provides Service Provider Interface (SPI) for plug-able group communication with cluster(s)
- GMS in GlassFish Application Server monitors cluster health
- GMS provides support to memory replication modules



Application

Configuring for High Availability

- Web modules to be marked as `<distributable />` in `web.xml`
- Should support replication of session state, e.g. Http session, Stateful session bean, and Single sign-on object(s)
- Should implement Serializable interface for session replication
- Group Management Service (GMS) should be enabled
- Configure Availability of individual application or EJB module
`asadmin> deploy --target <cluster-name> --availabilityenabled=true <application>`
- Availability of EJB Container `asadmin>`
 - > `set availability-service.ejb-container-availability.availability-enabled="true"`
 - > `set availability-service.ejb-container-availability.sfsb-persistence-type="file"`
 - > `set availability-service.ejb-container-availability.ha-persistence-type="ha"`
- Individual Bean in file `sun-ejb-jar.xml` add `<ejb availability-enabled="true">`



Availability Service

domain.xml

Home | Version
Logout | Help

User: admin | Domain: domain1 | Server: localhost

Sun Java™ System Application Server Admin Console

Common Tasks

- Registration
- Domain
- Applications
- Web Services
- JBI
- Custom MBeans
- Resources
- Clusters
 - cluster1
- Stand-Alone Instances
- HTTP Load Balancers
- Node Agents
- Configurations
 - cluster1-config
 - JVM Settings
 - Logger Settings
 - Web Container
 - EJB Container
 - Java Message Service
 - Security
 - Availability Service
 - Transaction Service
 - HTTP Service
 - ORB

Configurations > cluster1-config > Availability Service

Availability Service

Web Container Availability

EJB Container Availability

JMS Availability

Web Container Availability Save

Availability for the web container

[Load Defaults](#)

Availability Service: **Enabled**
Enable instance-level availability service; use the Web/EJB/JMS tabs to override default HADB availability settings.

Persistence Type: replicated
HTTP session persistence mechanism

Persistence Frequency: web-method
Frequency at which the HTTP session is stored

Persistence Scope: session
Scope of HTTP session changes required for storage to occur

Single-Sign-On State: **Enabled**
Controls whether the single sign-on state is available for failover

HTTP Session Store: jdbc/hastore
JNDI name for the JDBC resource of the high-availability database; must contain only alphanumeric, underscore, dash, or dot characters

Persistence Store Health Check: **Enabled**

Additional Properties (0)

Add Property
Delete Properties

Name	Value
No items found.	



Memory Replication

High Availability Techniques

- Available with GlassFish V2 Cluster or Enterprise profile
- JXTA technology used can handle high volume and throughput
- Module uses Grizzly NIO for high I/O performance
- GMS should be enabled
- Properties in domain.xml to be set as
 - `persistence-type=replicated`
 - `persistence-frequency=web-method | time-based`
 - `persistence-scope=session | modified-session | modified-attribute`
- Replication relies on version of HTTP-Session, container managed cookies JSESSIONID and JSESSIONIDVERSION



HADB

High Availability Techniques

- HADB software is supplied with SJS AS 9.1 distribution
- HADB available only in the Enterprise profile and with IPv4 + IPv6
- Network should be configured as UDP multicast
- Command tool `asadmin> create-cluster`
`[--host][--port][--hosts][--haagentport][--haadminpassword] <cluster-name>`
- Command tool `asadmin> configure-ha-cluster`
`[--host][--port][--haagentport][--haadminpassword][--autohadb] <cluster-name>`
- Command tool `asadmin> configure-ha-persistence`
`[--host][--port][--scope][--store=<jndi>][--type=memory|file|ha] <cluster-name>`
- In `domain.xml` override Session-Manager specific sub properties
`manager-properties`, `store-properties`, `session-properties`



Clustering Considerations

Pros and Cons

- Provides options for High Availability and Scalability, but has performance overheads
- HADB is highly reliable, but has relatively higher cost, than Memory Replication, for Implementation and Maintenance
- Memory replication is faster, but needs proper Capacity Planning to Session failover considerations
- Application modifications may be required to make Session objects Serializable to enable replication across multiple JVM
- Multi-tier distributed application architecture may require special considerations and configurations
- Currently clustering is possible only within single Sub-net



Load Balancer

Architectural Choices

- Software Load Balancer
 - > Sun HTTP LB plug-in* (on SJS Web Server, Apache, Microsoft IIS)
 - > Apache mod_jk ([How to setup](#))
- Hardware Load Balancer
 - > Sun Secure Application Switch ([General info](#))
 - > F5 BIG-IP v4.5 application traffic management device ([How to setup](#))
- Dynamic Load Balancing
 - > MBean monitors or change specific attributes using GlassFish Self Management JMX (JSR 3 and JSR 160), and
 - > AMX, **A**pplication server **M**anagement **eX**tension, APIs (JSR 77) to change Application Server configurations



Sun HTTP LB Plugin

Features

- Sticky RR Load Balancing
- Support for multiple clusters
- Rapid HTTP request failover (< 30 ms)
- Configurable Health checking to re-enlist server
- Check and reload the dynamic LB configuration changes
- Support for Quiescence, enables rolling web service upgrade
- Auto-retry of failed requests for idempotent URL
- Configurable Error pages



Load Balancer Configuration

Install LB Plugin

- Install a supported Web server on respective OS ([matrix](#))
- Installation of LB Plugin
 - > Create a directory lbplugin under `${glassfish.home}/lib`
 - > Download the Sun HTTP LB Plugin jar file ([aslb-9.1](#))
 - > Unjar the file in the directory `${glassfish.home}/lib/lbplugin`
 - > Unzip the files `SUNWaslb.zip` and `SUNWasp.x.zip`
 - > Ensure permission 755 to directory `${glassfish.home}/lib/lbplugin/lib`
- Administration Console to setup Load Balancer
 - > Create Load Balancer node and select Automatically Apply Changes
 - > Select Targets either as Cluster or Standalone instances
 - > Enable Cluster for load balancing
 - > Enable Application for load balancing and failover



Load Balancer Configuration

Administration Console

Home Version Logout Help

User: admin Domain: domain1 Server: localhost

Sun Java™ System Application Server Admin Console

Common Tasks

- Registration
- Domain
- Applications
- Web Services
- JB1
- Custom MBeans
- Resources
- Clusters
 - cluster1
- Stand-Alone Instances
 - server (Admin Server)
- HTTP Load Balancers
- Node Agents
- Configurations

HTTP Load Balancers

New HTTP Load Balancer OK Cancel

Application server can automatically manage the software load balancer plugin for the Java Enterprise System Web Server. Provide the device details below to enable changes to be applied directly to the plugin.

Name: *

Automatically Apply Changes: **Enabled**
If enabled, immediately push changes to lb config to the physical load balancer

All Instances: **Enabled**
Load Balance all instances of the selected target(s)

All Applications: **Enabled**
Load Balance all applications deployed to the selected target(s)

Device Host: *
Host name or IP address for the device

Device Admin Port: *
Device administration port number

Proxy Host:
Proxy host used for outbound HTTP

Proxy Port:
Proxy port used for outbound HTTP

Targets

Selected targets must all be either cluster or standalone instances. A combination of both is not supported.

Available Targets:

server

Selected Targets:

cluster1

Add >

Add All >>

< Remove

<< Remove All



Load Balancer Configuration

Setup LB Plugin

- Generate Load Balancer configuration loadbalancer.xml
 - > `asadmin create-http-lb-config --target cluster1 lb-config`
 - > `asadmin enable-http-lb-server cluster1`
 - > `asadmin enable-http-lb-application --name clusterjsp cluster1`
 - > `asadmin export-http-lb-config --config lb-config loadbalancer.xml`
- OR a single command
 - > `asadmin configure-http-lb-config --target lb-config loadbalancer.xml`
- Copy `libpassthrough.so` to `${ws.home}/plugins/lbplugin/bin/`.
- Copy `./errorpages` to `${ws.home}/plugins/lbplugin/errorpages/`.
- Copy `*.res` to `${ws.home}/plugins/lbplugin/resource/`.
- Copy `loadbalancer.xml` to `${ws.instance}/config/`.
- Copy `sun-loadbalancer_1_2.dtd` `${ws.instance}/config/`.



Load Balancer Configuration

loadbalancer.xml

- Generate Load Balancer configuration loadbalancer.xml

```
<!DOCTYPE loadbalancer PUBLIC "-//Sun Microsystems Inc.//DTD Sun One Application Server  
8.1//EN" "sun-loadbalancer_1_2.dtd">
```

```
<loadbalancer>
```

```
  <cluster name="cluster1" policy="round-robin">
```

```
    <instance name="instance1" enabled="true" disable-timeout-in-minutes="60"  
      listeners="http://host:port1" weight="100"/>
```

```
    <instance name="instance2" enabled="true" disable-timeout-in-minutes="60"  
      listeners="http://host:port2" weight="100"/>
```

```
    <web-module context-root="clusterjsp" enabled="true" disable-timeout-in-minutes="60"  
      error-url="sun-http-lberror.html" />
```

```
    <health-checker url="/" interval-in-seconds="10" timeout-in-seconds="30" />
```

```
  </cluster>
```

```
  <property name="reload-poll-interval-in-seconds" value="60"/>
```

```
  <property name="response-timeout-in-seconds" value="30"/>
```

```
  <property name="https-routing" value="true"/>
```

```
  <property name="require-monitor-data" value="true"/>
```

```
  <property name="active-healthcheck-enabled" value="false"/>
```

```
  <property name="number-healthcheck-retries" value="3"/>
```

```
  <property name="rewrite-location" value="true"/>
```

```
</loadbalancer>
```



Load Balancer Configuration

Web Server Configuration

- Modify `${ws.instance}/config/magnus.conf`

```
##BEGIN EE LB Plugin Parameters
```

```
  Init fn="load-modules" shlib="${ws.home}/plugins/lbplugin/bin/libpassthrough.so"  
  funcs="init-passthrough,service-passthrough,name-trans-passthrough" Thread="no"  
  Init fn="init-passthrough"
```

```
##END EE LB Plugin Parameters
```

- Modify `${ws.instance}/config/obj.conf`

```
NameTrans fn="name-trans-passthrough" name="lbplugin" config-file=  
  "${ws.instance}/config/loadbalancer.xml"
```

And

```
<Object name="lbplugin">
```

```
  ObjectType fn="force-type" type="magnus-internal/lbplugin"
```

```
  PathCheck fn="deny-existence" path="*/WEB-INF/*"
```

```
  Service type="magnus-internal/lbplugin" fn="service-passthrough"
```

```
  Error reason="Bad Gateway" fn="send-error" uri="$docroot/badgateway.html"
```

```
</Object>
```

GlassFish V2 Resources

- GlassFish V2 Cluster : [Developer Technical Article](#)
- [Training and Certification](#) On Java EE technologies
- GlassFish Project : <http://glassfish.dev.java.net>
 - > Download: previews, current and past releases, plug-ins
 - > Learning: documentation, tutorials, technical articles, flash demos
 - > Community Aquarium: latest news, forums, events, mailing lists
- GlassFish Wiki : <http://wiki.glassfish.java.net>
- Bug list : Priority [P1](#) and [P2](#)
- Users Alias : [<users@glassfish.dev.java.net>](mailto:users@glassfish.dev.java.net)



GLASSFISH V2 CLUSTERING AND LOAD BALANCING

Satyajit Tripathi

<http://blogs.sun.com/stripathi>

