Guide to GlassFish High Availability

White Paper
October, 2008

Abstract

Sun GlassFish Enterprise Server v2 includes advanced high availability features that enable enterprises to deploy GlassFish in business-critical environments. This white paper provides valuable insights into possible reference topologies that meet a variety of deployment needs from a high availability perspective in a virtualized environment.

Table of Contents

Introduction....1
1. Deployment Topology Overview....1
1.1 Simple Service Delivery Configuration....1
1.2 Service Availability Configuration....2
1.3 Service and Data Availability Configuration....2
1.4 Business-Critical Service Configuration....2
1.5 Deployment Topology Considerations....3
2. Topology 1: Light Weight (Light Load, No Fault Tolerance)....4
3. Topology 2: Data Availability With In-memory Replication....6
4. Topology 3: Data Availability With HADB Colocated Configuration....8
5. Topology 4: Non-colocated HADB Configuration....10
6. Additional Deployment Considerations....12
7. Summary....12
8. References....13
9. Acknowledgements....13

Introduction

Information Technology departments are bombarded with new requirements at an ever-increasing pace, while at the same time, Finance departments are working to reduce or restrict growth in IT budgets. In addition to maintaining existing business systems, lines of business are adding business-critical revenue-generating projects, expecting them to be in production in an ever-decreasing time frame. So how can these competing requirements be addressed?

Customers can **drive down cost** by significantly i**mproving utilization** of existing assets and by leveraging enterprise features of open source software.

With server virtualization and enterprise-ready open source software, customers can drive down costs by significantly improving utilization of existing assets and by leveraging enterprise features of open source software. In particular, Solaris 10 Containers are secure application environments with extremely low memory and CPU overhead. Sun GlassFish™ Enterprise Server (hereafter referred to as GlassFish) offers centralized administration, record-setting performance, and 99.999% availability. By combining the Solaris™ 10 operating system and GlassFish, customers can stay under budget and deliver business-critical services on time.

To facilitate enterprise deployment of GlassFish using Solaris 10 Containers, Sun has defined reference deployment configurations with multiple service availability requirements in mind. GlassFish high availability clustering addresses a wide range of market needs and suits various deployment plans. Clustering enables service availability, data availability, or both, in a scalable manner with varying performance and reliability combinations. The decision criteria for availability architectures primarily involve business risk, cost, availability needs, and performance.

1. Deployment Topology Overview

The reference configurations utilize:

- Sun GlassFish Enterprise Server
- Solaris 10 Containers
- Sun Fire T2000 servers

1.1 Simple Service Delivery Configuration

The simplest deployment option is to install and manage a single GlassFish runtime (instance). The advantage is simplicity itself—there is only one application server instance on the network to manage. The trade off, however, is that performance and data availability are constrained to one instance. If the instance fails, then the deployed service is no longer available. On the other hand, if the instance cannot perform well enough to meet demand, end user quality of service is impacted. A single-instance deployment of GlassFish is often sufficient for departmental services, but insufficient for external revenue-generating services.

1.2 Service Availability Configuration

When a single instance does not meet business requirements, a service availability configuration offers continuous access to business services. Service availability configurations are the foundation of scalability and ensure that business services are available when one or more instances in a cluster become unavailable. Core to service availability in GlassFish is a cluster, which groups multiple instances with a common configuration into a single logical managed unit. GlassFish enables a cluster distributed across multiple hosts to be easily and securely administered from a centralized management console. Service availability is the highest-performing option and is appropriate when data availability (session persistence) is not required. For example, a service availability configuration is appropriate for stateless services.

Sun Glassfish Enterprise Server enables a cluster distributed across multiple hosts to be easily and securely administered from a centralized management console.

1.3 Service and Data Availability Configuration

Service and data availability configurations enable business services and conversational state of user sessions to be available when one or more instances in a cluster become unavailable. GlassFish includes in-memory session state replication, which is a high-performance, reliable, scalable, high-availability solution suited for deployments where data availability is important, but 99.999% data availability is not required. In-memory session state replication should be sufficient for most high-availability deployments.

1.4 Business-Critical Service Configuration

For business-critical services where there is a very low tolerance for user session loss, availability is a higher priority than performance. GlassFish works in concert with a distributed persistent store, the GlassFish™ High Availability Database (HADB), to achieve 99.999% availability of user session data. HADB is not tied to GlassFish instances, and can be deployed to its own set of hosts to meet business-critical performance and reliability needs.

1.5 Deployment Topology Considerations

The reference configurations discussed below share numbers that are based on tests carried out in Sun's performance and quality engineering labs and should be considered a baseline. Actual production deployment numbers might vary depending on the application and load characteristics of the deployment involved.

Because this paper focuses on virtualized environments, these reference configurations primarily take into account the memory footprint requirements of GlassFish components. Additional business requirements, such as performance, will impact the number of virtualized environments per host.

Solaris Containers (zones) offer:

- A secure application environment
- A unit of resource management

Solaris 10 Containers (zones) have a two-fold purpose within the reference configurations. First, as a secure application environment, zones ensure that an

A Sun Fire T2000 server is ideal for web environments, thanks to the combination of high throughput and low power consumption.

application in one zone cannot access data in any other zone. Second, Solaris resource management facilities can be applied to zones, ensuring that each zone is guaranteed a minimal level of CPU and memory resources to make sure that services meet business service level agreements. Because zones are extremely resource efficient, a large number of zones can be deployed to a single host, enabling a highly virtualized environment.

The reference configurations also utilize Sun Fire™ T2000 servers for their ability to deliver high throughput with low power consumption—an ideal solution for web environments. Each T2000 consists of 32 GB of RAM and 32 hardware threads on a single chip, with 8 CPU cores and 4 threads per core. Note that since these tests were run, Sun Microsystems has released the Sun Fire T5120, which doubles the RAM to 64 GB and hardware threads to 64, and the Sun Fire T5240, which again doubles the RAM to 128 GB and hardware threads to 128. These new servers respectively double and quadruple the maximum CPU and memory resources available to clusters in the reference configurations.

Figure 1: Two Clusters Spanning Two T2000 Hosts

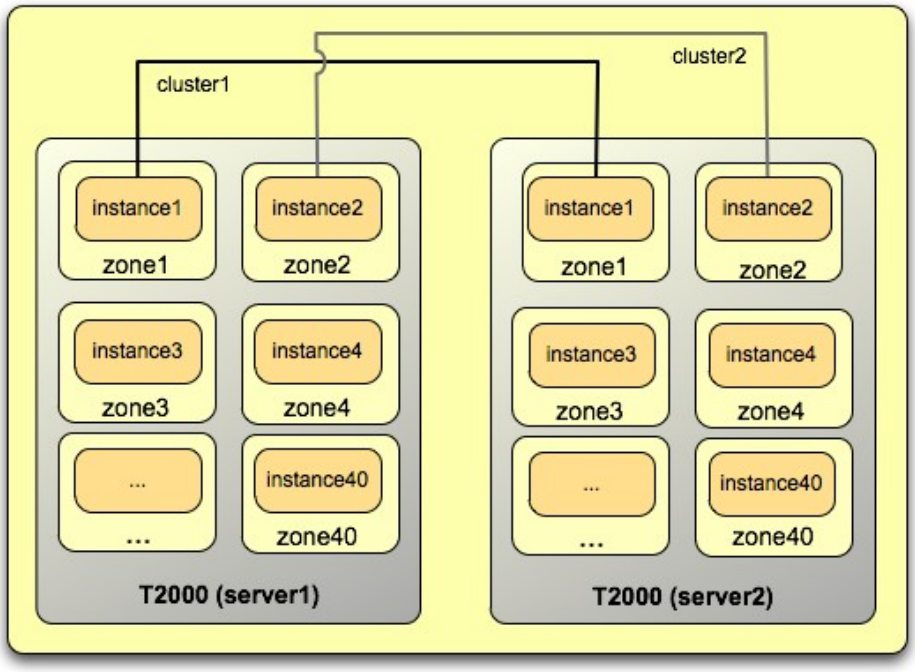

Figure 1 illustrates a base configuration with two T2000 servers running Solaris 10. Each T2000 contains up to 40 zones, each with an operating system instance. In this case, cluster1 is a 2-instance cluster consisting of instance1 of zone1 on each T2000 host. Similarly, cluster2 is a 2-instance cluster consisting of instance2

of zone2 on each T2000 host. Up to 40 clusters could be created in this manner before running out of resources.

2. Topology 1: Light Weight (Light Load, No Fault Tolerance)

In a basic service availability model, traffic is spread across multiple instances in a cluster by a load balancer, with no concern for data availability (session persistence). This configuration utilizes the GlassFish *cluster profile* with data availability disabled. The *cluster profile* enables centralized administration of a cluster and instances.

The 40-zone configuration can be increased significantly using Sun Fire T5120 and T5240 servers, which respectively double and quadruple the maximum available memory and processing threads.

The following table lists a reference configuration for a relatively simple web application with footprint calculations and assumptions:

Item	Configuration	Comments
T2000 server with 32GB RAM	40 zones	The number of zones for a deployment will be constrained by performance and application resource requirements.
Clusters	40 clusters with instances spread across 2 T2000 hosts	–
Domain Administration Server (DAS)	512 MB heap	Active only during administration tasks. A small number (1 - 3) of DAS processes are required to manage up to 40 clusters.
Sun GlassFish Enterprise Server Instance (cluster profile)	512 MB heap	Default setting for cluster profile instances.
Node Agent	256 MB, approx. 25 MB active footprint	Active only for instance lifecycle (creation, deletion) and watchdog function. Has very low active footprint. One node agent per zone/virtual host.

With this configuration, the heap size configurations are commensurate with the application involved. Larger applications will require larger heap sizes depending on the number of concurrent users and the size of user sessions. As a result, memory footprint for each application server instance and corresponding zone will

be higher and hence reduce the number of zones and clusters per host. For horizontal scalability, a sizing exercise should be conducted to achieve performance goals, either by giving each cluster more resources (by adjusting zone resources) or by adding more T2000 servers. Note that Sun™ xVM Ops Center can facilitate the provisioning of Solaris 10 and zones to multiple hardware architectures.

Another strategy would be to consolidate two or more small, lightweight applications into a single cluster while using the remaining resources for larger applications over multiple clusters. This can produce considerable resource savings and improve manageability.

Up to 40 secure clusters can be deployed across two Sun Fire T2000 servers with service and data availability enabled.

3. Topology 2: Data Availability With In-memory Replication

In this deployment scenario, both service availability and data (user session) availability are addressed. The sample web application in this configuration is deployed with availability enabled, taking advantage of GlassFish in-memory replication for high availability. This solution is available in both the *cluster profile* and *enterprise profile* of Sun GlassFish Enterprise Server. In this configuration, since the application is a web application, HttpSession objects are persisted. Objects in the session scope should implement the java.io.Serializable interface. This replication solution involves each instance autonomously picking a replica partner instance in the cluster to act as a backing store for its replication artifacts.

Process Details	Configuration	Comments
Zones per T2000	20	Because of the need for each instance to contain a replica of user session data, each cluster instances consumes more memory. This reduces the total number of zones by half.
Clusters	20 clusters with instances spread across 2 T2000 hosts	–

Process Details	Configuration	Comments
DAS	512 MB heap	Active only during administration tasks. A small number (1 - 3) of DAS processes are required to manage up to 20 clusters.
Sun GlassFish Enterprise Server instance	1024 MB heap	Requires setting this heap in the domain configuration repository through administration console.
Node Agent	256 MB	Active only for instance lifecycle (creation/deletion) and watchdog function. Has very low active footprint.

Heap size settings are commensurate with the application involved. The sizing of the heap involves calculating the number of concurrent user requests multiplied by average session size, doubling it for in-memory replication's replica cache. Additional head room is factored in to compensate for the load normally handled by a failed instance. As session sizes and number of concurrent user requests increase, a corresponding increase in heap space is required.

In the above tested configurations, a conservative session state persistence strategy was employed (save the entire session, after every request). Better performance can be realized by trying other supported persistence scopes.

Up to 20 secure clusters can be deployed across two Sun Fire T2000 servers with business-critical service and data availability enabled.

For example:

- Enable session persistence only for very important and relevant artifacts. For example, don't persist static images that go with each session.
- Specify persistence scope for only modified sessions. Replication occurs only when a session is modified.
- Specify persistence only for modified attributes. Be cautious regarding multiple threads that operate on the same field, for example, one with an update, and another with a delete operation.

The optimizations described above can potentially deliver improved performance for well-written applications.

4. Topology 3: Data Availability With HADB Colocated Configuration

In this configuration, the High Availability Database (HADB) is used for persisting an application's conversational state. HADB is a distributed persistence store used exclusively for this purpose, and not a general purpose relational database used, for example, to catalog data in an e-commerce application.

In this configuration, HADB is co-located on the same hosts or zones as the application server instances. HADB provides proven 99.999% availability—very high reliability with extremely low potential for session loss. This reliability comes at the cost of lower performance than an in-memory replication solution. It is well suited for business-critical deployments with very low tolerance for session loss, that is, session state availability is more important than performance. With this configuration, performance is expected to be better than when HADB is on a separate host cluster.

Process Details	Configuration	Comments
Zones per T2000	20	–
Number of clusters	20 clusters spread across 2 T2000 hosts	–
DAS	512 MB	Active only during administration tasks. A small number (1 - 4) of DAS processes are required to manage up to 20 clusters.
Sun GlassFish Enterprise Server Instance	512 MB	–
HADB	512 MB	Two HADB processes are needed for optimal performance. One process is sufficient for lighter session data needs.
Node agent	256 MB	Active only for instance lifecycle (creation/deletion) and watchdog function. Has very low active footprint.

As seen above, the heap size requirement per application server instance is

similar to Topology 1, because memory requirements for session persistence are on the HADB layer.

For improved performance, HADB can be deployed to it's own hardware tier.

HADB requires shared memory to be set up in the operating system. On each zone, 512 MB of memory space must be set aside for this purpose.

Each HADB node requires 1 GB of disk space for installation and data store. For lightly loaded applications, one HADB node per zone is sufficient. For applications that are more heavily used, two HADB nodes might be needed per zone. Appropriate capacity planning should be done prior to deployment.

If applications result in extensive disk writes, separate external disks will deliver better performance. For the same reason, HADB Data devices should be kept on separate disks if possible.

Finally, the actual number of zones in a deployment might vary based on customer application load and session persistence requirements. The result of a typical application simulation is described in the next section.

5. Topology 4: Non-colocated HADB Configuration

In this topology, the HADB is deployed on a separate host cluster tier that is distinct from the application server cluster tier to improve performance.

Process	Configuration	Comments
Zones per T2000	28	
DAS	512 MB	Active only during administration tasks. A small number (1 - 4) of DAS processes are required to manage up to 28 clusters.
Sun GlassFish Enterprise Server Instance	512 MB	
Node agent	256 MB	

Process	Configuration	Comments
HADB (on remote Intel or AMD x64-based hosts)	512 MB	Per node. Each 2xDualCore host will run 12 HADB nodes.
HADB Nodes per Intel or AMD x64-based hosts	14	
Number of Intel or AMD x64-based hosts per T2000	2	2 x64-based hosts are recommended.

HADB nodes are spread across multiple x64-based hosts for improved availability and performance. Additionally, network bandwidth could be a bottleneck if the HADB nodes on a host are not spread out across network interfaces. There are up to 4 Gigabit Ethernet ports in Sun x64-based hosts. Throughput can be improved by carefully spreading the 14 HADB nodes evenly across Gigabit ports. An additional means to achieve doubling of bandwidth is through IP Bonding and Link Aggregation, which effectively combines two or more network interfaces into one bonded virtual interface at the router/switch end. Link aggregation provides automatic NIC card level failover and load balancing and generally provides higher throughput.

General Comments

1. Internal measurements with standard Web and GlassFish workloads (without HADB) show that T2000 is roughly 5-6 times the throughput of a 3.2 GHz Intel/ Xeon. This might serve as a useful but rough guideline as you consider consolidating a customer's web workloads onto T2000 hosts.
2. Sun testing utilized 8 core, 32 GB RAM T2000 hosts. Each core is capable of running 4 processes or threads, where 32 different threads can execute simultaneously. Solaris Resource Management was used to create CPU and Memory utilization quotas to zones (and applications). The more important zones (or applications) might get a sizable reservation, while the remaining zones can share the unreserved amount. This should be explored as a step in capacity planning before production.

6. Additional Deployment Considerations

Using Shared Configurations for Clusters

The memory footprint of the DAS is directly proportional to the number of different configurations defined. One way to ease application server cluster management is to create a shareable configuration that is applied to all the clusters. The shared configuration will specify the behavior of containers. Applications, ports, and other elements can be different for each cluster.

How Many DAS Instances (Domains) Should Be Created?

Ideally, you should create a domain for each distinct portfolio of applications that need to be administered separately. Set up no more than 6–8 clusters per domain, with 2–4 instances per cluster.

7. Summary

IT departments are asked to deliver a continuous stream of new business services into a constrained, if not shrinking, IT budget. GlassFish, together with Solaris 10 Containers and Sun Fire T2000 servers, can enable IT departments to deliver these services under budget. GlassFish offers the cost-efficiency and productivity of open source products, while Solaris 10 Containers on Sun Fire T2000 servers minimize the number of manage nodes.

While cost reduction and manageability are important, a deployment configuration must be able to meet the availability requirements of the business. The reference configurations presented here demonstrate how a customer can deploy business services to meet various availability requirements, from the highly scalable service availability configuration to the business-critical, 99.999% service-and-data availability configuration. The reference configurations are flexible, and should be modified to meet an organization's availability and requirements.

8. References

1. *Sun Java System Application Server 9.1 Installation Guide:* http://docs.sun.com/app/docs/doc/819-3670
2. *Sun Java System Application Server 9.1 Quick Start Guide:* http://docs.sun.com/app/docs/doc/819-3193
3. *Sun Java System Application Server 9.1 Administration Guide:* http://docs.sun.com/app/docs/doc/819-3671
4. *Sun Java System Application Server 9.1 Deployment Planning Guide:* http://docs.sun.com/app/docs/doc/819-3680
5. *Sun Java System Application Server 9.1 Performance Tuning Guide:* http://docs.sun.com/app/docs/doc/819-3681
6. *Sun Java System Application Server 9.1 High Availability Administration Guide:* http://docs.sun.com/app/docs/doc/819-3679
7. Managing Solaris Containers with Sun xVM Ops Center: http://www.sun.com/software/products/xvmopscenter/

9. Acknowledgements

Contributors: Suveen Nadipalli, Gopal Jorapur, Shreedhar Ganapathy, and John Clingan. We greatly appreciate the review of this article by Sreeram Duvuru, Larry White, and Kedar Mhaswade.