End-to-End Availability for J2EE Applications using BEA WebLogic
Table of Contents

Executive Summary ......................................................................................................................................................... 1
Availability Challenges with the Service-Oriented Architecture .................................................................................. 2
The BEA and VERITAS Solution for End to End Availability .................................................................................... 5
Protecting the Session Environment with WebLogic Clustering .............................................................................. 7
   How It Works .................................................................................................................................................................. 7
      Three Reasons to Use WebLogic Server Clustering .......................................................................................... 10
         1. Building scalable applications ................................................................................................................ 10
         2. Simplifying management of distributed applications ................................................................................. 10
         3. Reducing client impact of application server failures .............................................................................. 10
Protecting Application Infrastructure with the VERITAS HA Solution ................................................................. 11
   VERITAS Solution Suite Components .................................................................................................................... 11
   How It Works .......................................................................................................................................................... 12
      Adding Global Availability ................................................................................................................................. 17
         Primary and secondary sites ............................................................................................................................ 17
         Volume Replication ........................................................................................................................................... 17
         Grouping Clusters into Sites .............................................................................................................................. 17
      Seven Reasons to Cluster the Application Infrastructure ................................................................................... 19
         1. Protecting interdependent applications ........................................................................................................ 19
         2. Protecting application infrastructure ............................................................................................................. 19
         3. Protecting single points of failure in the J2EE architecture ........................................................................ 19
         4. Managing change and maintenance proactively ......................................................................................... 19
         5. Restoring capacity within the WebLogic application in case of failure ....................................................... 19
         6. Reducing operational complexity ................................................................................................................. 20
         7. Implementing a global failover strategy for disaster recovery ..................................................................... 20
Conclusion ................................................................................................................................................................. 21
Executive Summary

Companies worldwide are using BEA WebLogic to implement a service-oriented architecture (SOA), integrating data and transactions across multiple back end systems into scalable customer-facing applications. These applications are now serving mission-critical functions. Customers and users expect dial-tone like reliability, and have little tolerance for outages or even slow-downs.

But maintaining the highest possible levels of availability and performance in a dynamic and multi-layered application environment is particularly challenging. BEA and VERITAS have worked together to help solve this problem – developing, testing, validating and supporting an integrated, end-to-end solution for application availability. By combining BEA WebLogic Server’s native clustering capabilities with the VERITAS HA Solution Suite for BEA WebLogic, enterprises can create highly scalable, highly available applications and manage them for optimal service levels in a complex IT environment. The combined solution even helps companies address disaster recovery and business continuity in the face of site-wide disruptions.

This paper describes the overall challenges of supporting highly available applications using the Service-Oriented Architecture. It then describes how to build availability throughout the application architecture and infrastructure by combining BEA WebLogic clustering and the WebLogic-specific HA solution from VERITAS.
Availability Challenges with the Service-Oriented Architecture

In today’s lightning-fast business environment, success requires a combination of responsiveness and efficiency. IT departments need to build, deploy, maintain and change applications to perform critical business functions, both quickly and efficiently. This can be challenging in an IT environment characterized by many heterogeneous business systems and data sources.

To address this situation, many companies are deploying a service-oriented architecture (SOA), linking vital applications together with middleware and building Java-based applications with reusable components. BEA’s WebLogic Platform provides a SOA-based environment for building, deploying, and managing enterprise applications and end-to-end business processes. It enables accelerated, standards-based application development while leveraging existing systems and protecting strategic investments. Developers rely on BEA’s unified, simplified and extensible platform to create and deploy mission critical applications that drive today’s integration efforts.

Gartner suggests that “Application integration and platform middleware technologies will provide the strategic backbone for the new stage of innovation in the business’ use of its information assets.”

Source: Gartner: Predicts 2004: Application Integration and Middleware [December 2003, Yefim V. Natis]

The availability mandate
The success of this service-oriented architecture presents IT and application developers alike with some new challenges – keeping these complex, n-tier systems online and operational around the clock. Availability is essential for applications running critical systems, such as customer-self-service, supply chain, provisioning and banking.

Application availability has become a business imperative. The cost of outages is significant; if a customer self-service application is down, the customer does not get service. And service-level expectations are very high; customers expect dial-tone reliability and speed-of-thought responsiveness.

Both application developers and operational IT staff face significant challenges in creating and maintaining highly available application services.

Applications must be able to respond rapidly to increases in demand.
IT organizations must be able to add capacity to applications promptly in response to changing demand. For example, an online banking application may start by serving a small, trial group of users but grow to serve a large community, with variable demand peaks.

Application architects need to be able to design applications to run on any number of servers, and add capacity in response to changing demands. They need to build for scalability without introducing multiple points of failure or increasing management complexity.

Application components lack dependency awareness.
Enterprise applications built using a service-oriented architecture contain a collection of independently constructed and deployed application components. Each component offers a reusable, easily discoverable service. A component may also be a client in this loosely coupled collection of application services. The advantage is self-evident to the J2EE application architect. He or she can design, develop and implement a core set of unique applications services without regard to their usage pattern and independent of the overall architecture. The goal is to create modular, shared service components that work collaboratively within an application ecosystem.

The availability challenge rises from the unpredictable usage patterns that emerge from the resulting interconnected services as they run in a production environment. Because we can’t predict application component usage nor can we foresee application service dependencies, it becomes essential to keep all of these
application components continuously available in order to protect the integrity of the resulting application economy.

**The operational environment is complex and heterogeneous.**
Several components reside between the application and the end user, including web servers, load balancers, and the logical IP address infrastructure. Because application delivery depends on the total environment, the operations team must be able to manage and maintain application servers, operating systems, storage systems, network infrastructure, and databases. Most often these skills are distributed through a number of different people, possibly in different organizational units. This situation leads to confusion and the “blame game” when problems occur.

To reduce administrative costs while improving availability, IT organizations need a single interface for managing all components of the application environment, across different platforms and databases.

**N-tier applications have many potential points of failure.**
To access the application, the user needs the *entire* application infrastructure to be working. The more tiers and components supporting an application, the more potential points of failure need to be protected.

J2EE applications depend not only on the other application components (such as the database and application servers), but also on the infrastructure components, such as the disks/volumes, NFS shared mounts, virtual IP addressing, network cards, and so on. Building redundancy at a component level at each layer increases the complexity of the IT environment – and hence the cost of managing that environment.

**The application infrastructure is in constant change.**
The application infrastructure is a moving target, many up of many different components that require ongoing maintenance, patches, updates and upgrades.
These changes are necessary and usually good. However, every change introduces potential for error and system instability.

In an inherently volatile environment, IT organizations need the ability to upgrade, test, maintain and deploy infrastructure components non-disruptively. With global operations, the few hours that operations staff once had in the middle of the night are business hours somewhere around the globe. They need the time to maintain applications and correct problems as they occur, without interrupting application access.

Applications must be part of the business continuity plan.
In addition to the moment-to-moment and day-to-day availability concerns, IT organizations also have to figure out how to provide continued access to critical applications in the case of a disaster. For the most critical applications, this means having the ability to run at an alternate location geographically separated from the production environment.

Alternate data centers are costly, however, and the money spent on disaster recovery facilities comes from a fixed pool of resources available for IT and infrastructure. For many companies, disaster recovery planning entails identifying a “cold” recovery site that requires an initial investment and ongoing maintenance to remain current. In the case of disaster, recovery may require significant time and manual intervention. By taking advantage of existing resources and multi-purposing equipment, it’s possible for companies to build “hot” recovery sites that can provide nearly instant recovery without manual intervention, and that minimize the impact of disaster recovery planning on the IT budget.

IT operations staff and application architects must work together to solve these availability challenges.
The BEA and VERITAS Solution for End to End Availability

Availability isn’t a problem that can be solved in a single location; IT organizations must address potential problems at all levels:

- At the session level, what happens to a user’s session when a single application server fails in the middle of a transaction?
- At the application level, what happens when the back end database fails?
- At the infrastructure level, what happens if a network interface card fails on an application server?
- At the business level, what happens if lightning strikes the production data center?

VERITAS and BEA offer an integrated, out-of-the-box solution for BEA WebLogic Server-based applications. The integrated BEA/VERITAS solution suite reduces both application and operational complexity, and addresses the HA challenges from the application session level to the data center level.

The integrated solution for BEA WebLogic Server relies on different level of clustering, both within the WebLogic Server itself, within the broader data center, and across the enterprise. Clustering is the process of grouping multiple resources into a single, relocatable logical unit to eliminate single points of failure within an application or a data center infrastructure.

The integrated WebLogic HA solution includes the following levels of clustering:

- WebLogic Server clustering, an integral part of WebLogic Server, enables the development and deployment of highly scalable and available applications by distributing and checkpointing application resources and session information across multiple server instances in a WebLogic Server cluster.

- VERITAS local clustering groups all of the resources required for a WebLogic application, including network infrastructure, file servers, and back end systems, and understands the dependencies between resources. It monitors these resources and automates corrective action in case of failures, migrating application components or services as necessary between servers.

- VERITAS global clustering enables an entire data center or application environment to be moved between geographically-distributed data centers at the flick of a switch, with automated data replication between sites for fast and accurate failover.
### Primary Focus

**Application Focus:** Focus is on the development, deployment and run-time performance of enterprise Java applications.

**Operational Focus:** Focus is on the protection of application relevant infrastructure to include front end http services, middleware components, database management servers, network infrastructure and storage components that support an enterprise Java application.

### Primary Goal

The goal is to provide a scalable and highly available environment for the J2EE application and the underlying set of J2EE application infrastructure services.

The goal is to provide a robust monitoring, alerting and fail-over mechanism designed to ensure that dependent application infrastructure is always on and always available.

### Primary function

Provide object/memory replication, checkpointing, and relocation

Provide automated server and process failover, application monitoring/alerting, and data replication

### Reasons to cluster

- Simplify application deployment complexity
- Design for application scalability
- Eliminate client interruptions
- Provide object/memory replication

- Simplify operational complexity
- Protect single points of failure
- Protect entire application infrastructure and user data
- Provide data center disaster recovery

### Target audience

J2EE architect

Data center operations staff

Individually, each clustering solution enhances the availability of the overall application environment. Taken together, however, they complement each other, offering an end-to-end solution for addressing application and operational complexity and offering the highest possible service levels for critical applications. For example,

- If a physical server within a WebLogic cluster fails, the WebLogic Clustering solution automatically redistributes the running sessions on other available WLS servers, while the VERITAS solution automatically restarts the failed server’s processes on a different server and restores capacity to the WebLogic application as a whole.

- While WebLogic Server clusters distribute applications over multiple instances, VERITAS clustering protects any single points of failure (pinned services) within the WebLogic application infrastructure.

- VERITAS clustering protects the availability of the critical back end systems accessed by WebLogic applications, such as enterprise databases, ERP systems, etc.

The rest of this paper looks at each of these clustering solutions in more detail, and outlines the reasons for using clustering at each level in an integrated HA environment.
Protecting the Session Environment with WebLogic Clustering

WebLogic Server offers integrated clustering capabilities that enhance the scalability and availability of applications in the J2EE environment. WebLogic Server clustering support focuses on application objects such as Enterprise Java Beans, Servlets, JMS messaging components and JDBC connections, as well as maintaining session state across the cluster of WebLogic Server instances.

How It Works

The WebLogic Server is an application server that acts as a container for applications and components implementing the Java 2 Enterprise Edition (J2EE) specification. In the J2EE development model, WebLogic Server uses little direct storage as part of its ongoing operations. Application storage and persistency are delegated to a database or messaging tier outside of the WebLogic Server environment. WebLogic Server clustering focuses primarily on protecting and distributing in-memory state (session state) and processing capacity.

Using WebLogic Server clustering, multiple WebLogic server instances run simultaneously, appearing to users as a single "super" server. These instances, together with any related resources (network channels, machine definitions, startup classes) are grouped in a WebLogic domain, managed by a single Administration Server. The Administration Server configures, manages, and monitors all other server instances and resources in the domain. All other servers in the domain are referred to as Managed Servers.

A WebLogic Domain may include multiple clusters, as well as additional non-clustered servers, and a single Administration Server.

WebLogic Administration Server distributes the application processing load among the different server instances automatically, providing superior application scalability. WebLogic clustering also distributes session information in the cluster so that if a single server instance fails, another instance can load the session state information and continue the user's interaction or transactions without restarting.

Session protection: Although WebLogic delegates storage and persistency to external tiers, there is a certain amount of state within the server used to tie a series of client requests together. By replicating and failing over this state information, WebLogic protects clients from interruptions if a failure occurs. The client will only notice the...
failure if the event occurs while a request is being processed, a window of a few hundred milliseconds or less. Even if this does happen, the client can retry the request immediately without having to replay the entire sequence.

Load balancing: The J2EE specification defines standards for application and component delivery that enable WebLogic Server to load balance and manage applications across clusters without custom code. WebLogic Server provides several mechanisms for balancing the load across the available instances, including simple DNS load balancing, hardware load balancers, using a dedicated load-balancing WebLogic Server as a proxy server, or by plugging in into some of the common Web servers available on the market.

For example, the following cluster uses a hardware load balancer to direct traffic to the different server instances in the cluster. The client session illustrated is using Server A, but Server B is designated as the secondary server for the session.

When Server A fails, WebLogic automatically replicates the state information from Server B to Server C, and makes C the primary server:
The user continues to work with the application without interruption, as the application objects and user state information are replicated on another server in the cluster.

All of these capabilities are implemented within the bounds of the Java environment, without any direct operating system access. This allows WebLogic Servers, and the applications hosted within, to be easily deployed and managed across heterogeneous environments. Since Java Virtual Machines are widely available on nearly all platforms, WebLogic can be deployed into almost any setting.
Three Reasons to Use WebLogic Server Clustering

Availability is just one of the benefits of using the inherent clustering capabilities of WebLogic Server. Clustering also enables applications to scale out to large numbers of servers, and reduces the task of managing and configuring applications across many servers.

1. Building scalable applications

WebLogic Server Clustering gives application architects the ability to build highly scalable applications, and to add capacity to an application as needed, responding dynamically to changing demands on the application.

For example, a financial institution might create a cluster with 12 WebLogic Server instances on 4 physical machines to handle a new customer service application. After a direct mail campaign dramatically increases the application usage, the application developer can add new instances dynamically, without interrupting service.

2. Simplifying management of distributed applications

WebLogic clustering simplifies the process of configuring and managing applications that require multiple server instances.

A large application running on 10 different servers might use thousands of files and objects, leaving the application architect the job of deploying, tracking and managing application components across multiple servers. Using WebLogic Server clustering, the centralized Administration Server deploys an application, and all of its essential components, automatically across all servers in a cluster, maintaining version control and ensuring consistency throughout the environment.

3. Reducing client impact of application server failures

By running applications on multiple instances and replicating session state information, WebLogic Server clustering reduces or eliminates the client impact of a server instance failure.

If a server instance fails in a WebLogic Server Cluster, the cluster directs its session to another instance in the cluster. Generally the user will not even notice the failure.

Note that the WebLogic Cluster cannot automatically relocate and restart the failed instance. It can restart the server instance on the same machine if the machine is still intact and a simple WebLogic Server restart will solve the problem. However, if the underlying server is compromised (i.e. Unix panic) the cluster would continue running with reduced capacity until the failed machine can be repaired and brought back online. Integrating the VERITAS Cluster Server support with the WebLogic cluster ensures that the failed instance is restarted on other resources and capacity is immediately restored.
Protecting Application Infrastructure with the VERITAS HA Solution

The VERITAS HA solution suite for BEA WebLogic operates at a different level within the application infrastructure from the BEA WebLogic clustering. Using VERITAS clustering together with WebLogic Server clustering might be seen as “clustering the cluster.” Although this sounds redundant, it is essential to address the broader issues of operational risk and complexity in the entire application infrastructure.

The VERITAS HA solution suite creates an HA environment that monitors, manages and protects the entire application infrastructure, including:

- The WebLogic cluster
- Enterprise applications
- File servers
- Back end databases
- Network resources
- Web servers

Adding replication and global clustering, you gain a high measure of disaster protection for critical applications, ensuring application availability in the event of data center-wise disasters or problems.

VERITAS Solution Suite Components

The VERITAS High Availability Solution Suite for BEA WebLogic includes the following integrated components:

<table>
<thead>
<tr>
<th>Local availability</th>
<th>VERITAS Storage Foundation</th>
<th>VERITAS Cluster Server (VCS)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Combines VERITAS' file system and volume management solutions for a highly available, robust foundation for the database and file systems accessed, directly or indirectly, by the J2EE environment. The Storage Foundation provides database-specific optimizations for Oracle, DB/2, Sybase, Microsoft, and Oracle RAC databases.</td>
<td>Eliminates planned and unplanned downtime by clustering critical applications and the resources they require. Monitors and manages critical components using specific agents for: BEA WebLogic BEA Tuxedo Critical enterprise applications (SAP, Siebel, etc.) Back end database systems File servers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Global Availability</th>
<th>VERITAS Volume Replicator</th>
<th>VCS Global Cluster Option</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Delivers reliable, storage-independent replication over any IP network. Replicates data at the logical volume level, and ensures data integrity and reliability.</td>
<td>Monitors and controls multiple, geographically-distributed VCS clusters and replication between them. Migrates entire data centers, including data and applications, with a single click.</td>
</tr>
</tbody>
</table>

Working together, these software components help IT organizations improve application availability on a daily basis while offering significant protection from the loss of service and data in the case of a disaster or regional disruptions.
You can augment the HA solutions with additional, integrated and validated products for managing and optimizing J2EE applications, such as:

- Performance management using VERITAS Indepth™ for J2EE
- Server provisioning using VERITAS OpForce™

**How It Works**

VERITAS Cluster Server (VCS) monitors each infrastructure component with application-specific agents, and maintains “heartbeat” connections between cluster nodes to determine system availability. A VCS agent for WebLogic Server monitors the WebLogic component.

For a critical J2EE application, you might want to include the following application components in the VCS cluster:

- The WebLogic Administration Server and Managed Servers
- Web servers
- Enterprise applications (ERP systems, etc)
- Back end databases
- File servers containing essential data

The cluster must have spare capacity to handle the failover of any application component. For example, if the back end database fails, another server must have sufficient capacity to run the database instance, potentially in conjunction with other, already-running applications. VCS supports flexible, “active-active” clustering configurations.

VCS provides application failover by encapsulating the resources required for each application into a service group – creating a virtualized application that can be moved between cluster nodes. Operations staff can operate on the cluster itself, on the service group (starting, stopping, switching over, etc.), or on the specific resources within the service group.

Each service group contains a set of dependent resources – the lower-level components that an application requires to operate successfully. Resources include disk groups, disk volumes, file systems, IP addresses, NIC cards, and dependent application processes.

VCS starts, stops, monitors and migrates service groups on any server in the cluster in response to server or resource faults. In addition, an administrator can proactively move a service group between cluster nodes to perform preventative maintenance or apply patches. The service group includes logic about the dependencies between application components.

For example, the diagram below illustrates resources required to support a node in WebLogic cluster - WebLogic Managed Server:
Each service group requires a dedicated file system, volume and disk group to store the service group’s data and programs. By placing managed server binaries, configuration files and application data on shared storage, VERITAS Cluster Server can quickly and efficiently relocate a WebLogic Server instance from a failed cluster node to another node in the cluster maintaining all disk mounts, configuration files and application binaries.

In addition to disk groups, applications also require specific network resources, such as Network Interface Cards (NICs) and IP addresses. VCS automatically migrates the Virtual IP address of failed applications so that user can continue connecting to the relocated application transparently.

A typical WebLogic installation uses one VCS service group to protect the WebLogic Administration Server, and additional server groups for managed servers in the domain. Although operations in the cluster continue if the Administration Server fails, most organizations want to protect this resource to provide continuity in application deployment and configuration. Please note that VCS will take over responsibility for starting, monitoring and stopping WebLogic Server instances so WebLogic’s Node Manager must be turned off on each managed server node.

For example, consider a VCS cluster that contains two WebLogic Managed servers, a PeopleSoft application server accessed by the WebLogic application, and the back end Oracle database instance. All of these components are required for successful application delivery. All reside on servers managed in a VCS cluster, accessing shared storage.
Now assume that some sort of catastrophic event (power disruption, Unix kernel panic or network hub failure) occurs in the datacenter completely disabling cluster nodes 2 and 3. Upon initial detection of the cluster node outage, VCS automatically initiates service group migration for the managed WebLogic Server and for the running Oracle Database instance. VCS will ensure that all existing server processes are shut down before initiating service group fail-over within the cluster. Service group fail-over is governed by the VCS rules engine that takes into account best fit and overall cluster load to determine the best place to relocate affected services.

VCS will automatically restart both the managed WLS Server and the Oracle Database. VCS understands that the managed WebLogic Servers are dependent upon the database so it first restarts the Oracle instance choosing Server 4 because this cluster node has the greatest amount of capacity to serve the needs of the database. As part of this process, the virtual IP address is moved, file systems are remounted on the target server, and the database instance is started.
Next VCS restarts the WebLogic Server instance on Server 5. VCS has a built-in sense of cluster load and can relocate application components based on available capacity augmented with a rules based policy engine.

At this point, the administrator can make appropriate repairs to Servers 2 and 3. When server repairs are complete, nodes 2 & 3 rejoin the cluster and are available to run application load as required. The administrator can, with a push of a button, migrate application service groups back to these servers to restore the original server deployment mix.
On the other hand, the administrator may choose to change the application mix to leverage specific machine capacities and minimize the disruption to the running WebLogic Server instance now running on Server 5. The administrator could:

- Move the Oracle Database instance to Server 2 to take advantage of Server 2’s high speed fiber network interface.
- Move the PeopleSoft Scheduler to the now free Server 3 in order to free capacity on Server 4 for the PeopleSoft Application Server.
- Move the PeopleSoft Application Server to Server 4 because Server 4 has over 4 GB’s of physical memory to handle additional application concurrency.
- Continue to run the second managed WebLogic Instance on Server 5.

Clustering application components like the database server, middleware servers and WebLogic Server provide the operations team with many configuration options while providing a robust and highly available environment.
Adding Global Availability
By adding data replication and global clustering to the highly available WebLogic environment, you gain the ability to switch over an entire application site between geographically-distributed data centers, quickly and accurately.

Primary and secondary sites
Global clustering depends on the existence of an alternate site. The cost of the alternate site is a major concern for many organizations – how many can afford to have a duplicate data center, somewhere distant from the primary data center, ready to accept production processing at a moment’s notice in case of a disaster?

Using VERITAS Volume Replicator and VCS Global Cluster Option, enterprises can reduce the cost of having a secondary site by leveraging existing equipment and using the alternate site for other processes:

- The secondary site doesn’t need to be identical in terms of computing resources to the primary site. It only requires sufficient capacity to run the application.
- The secondary site doesn’t need the same kind of storage as the primary site, as long as the overall capacity is sufficient. VERITAS Storage Foundation and Volume Replicator do not require the same disk array on both sides of the replication link.
- The secondary site could be running other, non-critical processes (such as development and QA) that could be stopped in the case of a failover or switchover.

Volume Replication
VERITAS Volume Replicator replicates the contents of each volume across a wide area network to the secondary site. It is completely transparent to the application components. Unlike traditional block-based approaches, it replicates I/O instead of disk tracks to ensure the data is always replicated in a consistent fashion, guaranteeing the recoverability of the application.

Grouping Clusters into Sites
Global Cluster Option extends VERITAS Cluster Server’s cluster management and failover capabilities across multiple clusters. You can manage multiple VERITAS Cluster Server clusters (regardless of platform) from a single console, and switchover or failover a single application, multiple clusters, or an entire site between geographically-distributed locations.

In the event of a site-wide disaster, the integration solution automatically performs the following steps:

1. Stop replication: In a switchover, the first thing that occurs is that replication between the primary and secondary sites stops. (If the primary site has experienced a dramatic failure, then replication has already stopped.)
2. Reconfigure the network: The Global Cluster Option reconfigures the logical network, migrating the virtual IP addresses for the application to the secondary site’s systems.
3. Promote the replicated data: The Global Cluster Option promotes the secondary data to primary status. If a secondary site is still available, VERITAS Volume Replicator can replicate to another site to maintain data availability.
4. Restart services: Global Cluster Option starts the Cluster Server service groups, bringing the entire WebLogic application infrastructure back online.
The following graphic illustrates the basic steps undertaken in a migration between two data centers, with one in Dallas and the other in Charlotte.

The same process applies when it’s time to migrate services back to the primary site.
Seven Reasons to Cluster the Application Infrastructure

By implementing the entire VERITAS HA solution suite for BEA WebLogic Server in conjunction with the WebLogic Clustering, you can create a highly available application environment for critical WebLogic applications, addressing a number of significant business and operational needs.

1. Protecting interdependent applications

J2EE applications depend on a number of components outside of the application environment itself, including the web servers, back end databases, and enterprise applications such as PeopleSoft, Siebel, SAP, etc. The failure of any of these application components may disrupt application service. WebLogic Server provides clustering within the J2EE application layer, while VCS provides clustering across application layers.

2. Protecting application infrastructure

Applications depend on a number of infrastructure components, such as storage, IP addresses, network interface cards, etc. VCS offers a centralized interface for managing and maintaining availability across the entire, heterogeneous application infrastructure, from storage to web servers.

3. Protecting single points of failure in the J2EE architecture

Even when using WebLogic Server clustering, it’s possible to have application components that represent a single point of failure. For example, WebLogic uses the concept of pinned services, which can be active on only one server at a time and cannot be migrated automatically in the event of a failure. Possible pinned services include JMS destinations and the JTA transaction recovery service. Likewise, custom Java code may use native calls or named pipes to non-Java resources. These resources must be relocated together with the non-Java resources.

VCS offers automated monitoring and policy-based failover of these pinned services, eliminating this potential single point of failure from the application infrastructure.

4. Managing change and maintenance proactively

The application infrastructure is in constant change, yet operations staff must operate with little or no planned downtime on critical systems. Simply managing patches and equipment upgrades is difficult on critical applications.

Using VCS, operations staff can smoothly move critical application resources to other servers in the cluster while they perform maintenance on individual components. VCS takes care of smoothly shutting down and restarting applications.

5. Restoring capacity within the WebLogic application in case of failure
If one of the servers fails within a WebLogic cluster, WebLogic automatically transfers sessions and session state information to other servers in the cluster. Although the session remains running, users may experience performance degradation because the overall capacity is reduced.

For example, if a customer-facing application uses five servers and one of the servers hangs, then the application as a whole is running at only 80% of its capacity, which can slow performance in a heavily-loaded application. The load balancer may introduce further delays while waiting for the hung server.

VCS will detect the server problem automatically and attempt a clean shutdown and restart. If the entire server is unavailable, it will restart the stalled process on another server in the cluster – restoring capacity and performance to the application quickly and effectively.

6. Reducing operational complexity
The application infrastructure itself may be quite complex, including servers on different operating systems, with hardware from multiple vendors. Managing the entire infrastructure effectively requires a wide range of IT skills within operational staff.

Using the VERITAS HA solution significantly simplifies the operational environment, reducing the potential for error and bridging a potential gap in operational skills. VCS provides the following:

- A single interface can start, stop, monitor and manage all infrastructure components – across platforms and applications.
- A point-and-click interface performs complex operations automatically, such as migrating running applications to spare capacity for maintenance purposes.
- Integrated dependency information eliminates the need for operational staff to understand the order in which to start or stop various components.

7. Implementing a global failover strategy for disaster recovery
Using the global clustering components of the HA solution suite (VERITAS Volume Replicator and the VCS Global Cluster Option) you can protect critical applications from site-wide problems by failing over, within minutes, to an alternate site.
If data is replicated between sites, VCS Global Cluster Option can automatically migrate the entire application infrastructure between sites, moving IP addresses and restarting applications appropriately on the secondary site using replicated data.

The centralized, single-click operation is in marked contrast to the processes companies follow today for moving applications to alternate sites, which include:

- Maintaining the DR site at the same update and patch levels as the primary site.
- In a disaster, moving tapes and staff to the DR site.
- Rebuilding servers, loading operation systems and backup software while tracking and maintaining what’s available at each site.
- Relying on experts to bring up the application appropriately.

The VERITAS solution eliminates the change control problems, as all changes to the primary application (and its subsidiary components) are replicated to the DR site automatically. The failover is automated and nearly instantaneous, reducing risk while improving application availability.

Conclusion

Both BEA and VERITAS are committed to helping enterprises simplify complexity, adapt rapidly to changing environments, leverage heterogeneous equipment and data, and put IT in service of business objectives. The companies are working together not only to improve the availability, manageability and performance of web-based applications, but to promote the broader vision of utility computing.

The integrated, end-to-end high availability solution described in this paper is just one of fruits of a strategic partnership. The companies are cooperating on other integrated offerings that will help enterprises build the infrastructure for utility computing. Customers can implement those pieces they need in phases. By supporting a ‘building block’ approach to infrastructure, BEA and VERITAS give customers a way to start achieving the benefits of utility computing today while building an infrastructure that will support flexible, adaptive computing for years to come.