IBM® Tivoli® Software

Assuring Your High Availability Solution for Tivoli Process Automation Engine Environments

Document version 1.0

Ana Biazetti, Senior Technical Staff Member, Architect
Daniel McConomy, Systems Configuration Specialist, Tpae Quality Assurance

CONTENTS

LIST OF FIGURES

REVISION HISTORY

Date	Version	Revised By	Comments
08/31/2012	1.0	acb	Initial version

1 Introduction

In today's global environment, more and more companies need to operate across time-zones and in 24x7 calendars. They need to reduce their downtime to the minimum possible and look for continuous availability of their systems.

Products based on Tivoli's process automation engine (Tpae), like Maximo Asset Management, Maximo Industry Solutions and Smart Cloud Control Desk, also play a role in such environments and thus also have continuous availability requirements. As part of that, it is important to understand the high availability (HA) capabilities of Tpae and how to assure that all the components of an HA solution are properly configured and tested to handle outages.

This paper provides an overview of HA concepts and then describes in detail the configuration and tests needed to fulfill the requirements for Tpae HA environments

1.1 Who should read this paper

The target audience for this paper is anyone interested in using Tpae based products in HA environments. Readers should be familiar with products like Maximo, Industry Solutions or Smart Cloud Control Desk and their deployment architecture.

2 Background and Terminology

2.1 Causes of Business Interruption

There are many causes of business interruption that can turn into outages for companies' IT environments, including:

- **Component Failure or Performance Degradation**
    - Caused by human error, software defects, disk failure, subsystem failure, hardware failure, power grid outage
    - Data is (generally) recoverable
    - But changes might be *stranded* until the failed component is recovered or the error corrected
- **Data Corruptions**
    - Caused by human errors (e.g., data deleted by mistake), software defects
    - Data is not recoverable
    - Requires 'un-doing' the mistake or going back to a reliable version of the data.
- **Planned Maintenance**

- System and application upgrades or reconfiguration
- Usually scheduled during 'off-hours'
- Requires careful planning and a swift execution. May need to be called off if there is a glitch

- **Disasters**
  - Flood, earthquake, fire, loss of a site
  - Once in a lifetime event
  - Data is not recoverable

In order to reduce or eliminate these outages, multiple techniques are available. Before discussing specific techniques, it is important to understand the different concepts related to availability as terminology can be confusing.

2.2 Availability Terminology

We use the definitions from the IBM HA Center of Competency [http://www-03.ibm.com/systems/services/labservices/solutions/hacoc.html], which include:

- **High availability (HA)–** The attribute of a system to provide service during defined periods at acceptable or agreed-upon levels and mask **UNPLANNED OUTAGES** from end-users. It employs fault tolerance; automated failure detection, recovery, bypass reconfiguration, testing, problem and change management
- **Continuous operations -** The attribute of a system to continuously operate and mask **PLANNED OUTAGES** from end-users. It employs non-disruptive hardware and software changes, non-disruptive configuration, software coexistence
- **Continuous availability –** The attribute of a system to deliver non-disruptive service to end users 7 days a week, 24 hours a day (there are no planned or unplanned outages).

The picture below shows how HA and continuous operations can be brought together to support continuous availability.

Figure 1 Terminology

In addition, another term that is often used in the industry is:

- **Disaster Recovery –**The process, policies and procedures related to preparing for recovery or continuation of technology infrastructure critical to an organization after a

natural or man-made emergency, or in the event of a disaster, political turmoil or criminal action.

- Disaster Recovery typically involves planned and unplanned site switch, and must be carefully integrated with business continuity processes and approvals.

- May or may not be combined with HA.

In this paper we focus on implementing and testing Tpae HA environments. Additional papers will focus on the other availability concepts.

2.3 Time concepts

Availability is usually expressed as a percentage of uptime in a given year. In this way, a given availability percentage corresponds to the amount of time a system would be unavailable per year, month, or week.

For example:

Availability %	Downtime per year
90% ("one nine")	36.5 days
99% ("two nines")	3.65 days
99.9% ("three nines")	8.76 hours
99.999% ("five nines")	5.26 minutes

Measuring the availability of a system is not a simple task, as you need to consider many variations of hardware, software and network capabilities. Also, it is important to consider that availability should not consider only the downtime due to unplanned outages (which is handled by HA), but also the downtime required for planned outages (which is handled by continuous operations).

Other commonly seen concepts in this domain include estimated time of repair (ETR) or recovery time objective (RTO), which mean the duration of time and a service level within which a business process be restored after a disruption in order to avoid unacceptable consequences associated with a break in business continuity. It can include the time for trying to fix the problem without a recovery, the recovery itself, testing, and communication to the users.

2.4 HA Classification

HA strategies are typically based on the concept of using redundant components to eliminate all single points of failure. Depending on the application characteristics (for instance, stateless, stateful) different HA strategies may be applied. The most common ones are:

Warm Standby: A single instance of the application uses a standby server to which the application can failover in the event of failure. The application on the backup node cannot be running at the same time as primary one, but the server is already up and running, thus considered a “warm” standby.

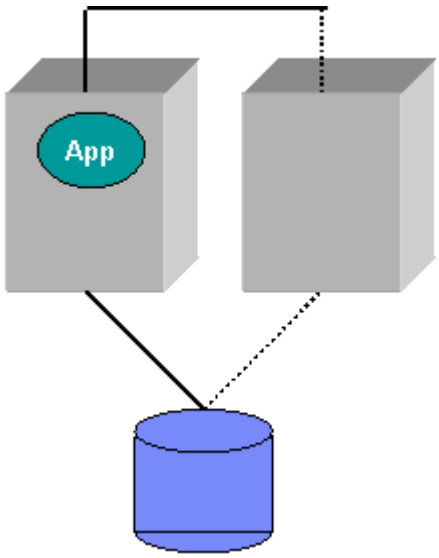


Figure 2 Warm Standby

Hot Standby: There are 2 instances of the application: a primary and a backup. The primary instance handles requests and replicates its state to the secondary instance. When a failure occurs, the roles are switched. It typically provides a faster failover than warm-standby.

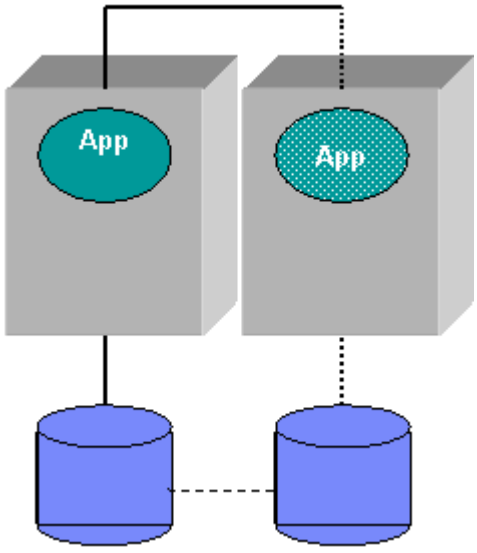


Figure 3 Hot Standby

Active-Active or Load Balanced: Several application instances process requests in parallel. If one fails, the requests can be routed to the other instances.

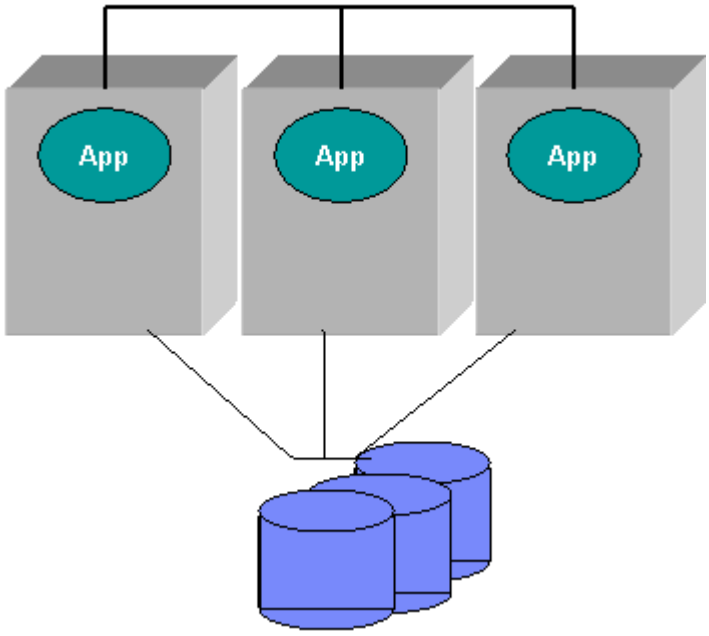


Figure 4 Active-Active

For more information see
http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/topic/com.ibm.mbs.doc/gp_highavail/c_cluster_config_models.html

2.5 Cluster Manager Concepts

HA implies redundancy that supports fail over. Multiple systems (nodes) are configured together as a cluster and can utilize a cluster manager to detect system status and make decisions based on this status. The cluster manager uses one or more resource groups with a defined set of policies to automate the fail over procedure when a failure is detected. Introducing a Service IP into the cluster configuration allows for transparency to the connecting applications and users and eliminates the need for configuration changes upon fail over. The cluster manager itself uses detection methods such as heartbeats, tie breakers and quorum to determine system status and to make decisions on how to act on this status.

- ***Cluster:*** group of connected systems (nodes) that work together as a single functional system in the perspective of the end user. Clustering allows servers to back each other up when failures occur by picking up the workload of the failed server.
- **Cluster member:** a single node that is defined within the cluster
- **Cluster manager:** an application or tool used to combine the nodes of a cluster and detect the status of the processes as defined by cluster resources and policies. The cluster manager drives the automated failover procedures and is highly configurable to the administrator. There are many cluster managers available that should function properly with TPAE but our examples all use Tivoli System Automation for Muliplatforms as the cluster management tool.

- **IP address takeover*:*** the ability to transfer an ethernet interface's IP address from one machine to another when a server goes down; to a client application, the two machines appear at different times to be the same server. Your cluster manager can be configured to automatically apply this service IP (or virtual IP) to the active node only. Upon failure, the cluster manager should remove the alias interface (and service IP along with it) from the primary node and apply it to the secondary node where services will be restored. Transactions and connections to the applications are always through the service IP address so the configuration of connecting components never has to change. In many cases, the startup of services is *dependent* on the application of the service IP.

- **Heartbeating**: a mechanism implemented by the cluster manager on each node to allow each system to detect whether other cluster members are alive or down. The nodes will send heartbeats to each other as a low-level system status.

- **Resources:** applications or pieces of hardware that can be defined to the cluster manager. These can be manually defined to the cluster manager, or some cluster managers such as Tivoli System Automation for Muliplatforms can use a concept of harvesting which will allow it to detect and define some resources on the system such as the ethernet interface.

- **Resource groups:** one or more defined resources lumped together as a functional group. This group serves a common purpose and the status of the resource group is related to the status of the members.

- **Policy:** a set of instructions defined to the resource group that dictates what procedure will occur when failures are detected

- **Relationships:** relationships defined within the resources of a cluster. An example of a relationship would be a dependency. When implementing a Service IP resource, other applications defined in the resource group may be dependant on the Service IP being active. This dependency ensures that services start in the correct order.

- **Quorum:** determines which node(s) are the active node and will process requests. There can be several types of quorums and they can function in specific ways depending on the number of nodes available in the cluster. To better understand the concept of a quorum, please follow the link at the end of this chapter for further investigation.

- **Tie Breaker**: determines which node has quorum and can access shared resources. When there is an even number of nodes within a cluster (commonly 2) the cluster will require a tie breaker to determine which node will have quorum. An example tie breaker is a network tie breaker. If 2 nodes are connected and sending heartbeats but the connection is disrupted, no heartbeats will make it to either system. From the perspective of each node, they do not know if it is the other node that has failed or if its own ethernet interface is down. The network tie breaker will attempt to ping a defined IP (you can have several defined) and the node that can ping the IP will win the tie breaker and have quorum.

  These were just a few of the many concepts related to cluster managers. For more information please review the End to End Automation with Tivoli System Automation for Muliplatforms RedBook: http://www.redbooks.ibm.com/abstracts/sg247117.html?Open

3 Tpae HA Architecture

There are a broad range of IBM products and applications built on top of the process automation engine, including Maximo Asset Management, various Maximo Industry Solutions and Add-Ons and Smart Cloud Control Desk. These applications enhance and add new components to TPAE infrastructure to achieve additional functionality

In general, these applications have similiar high availability characteristics as Tpae due to the same middleware software stack, so the HA solutions for the different Tpae based applications follow the same design principles.

3.1 Tpae Components

Tpae is composed of the following main components, each of which typically has multiple vendor support:

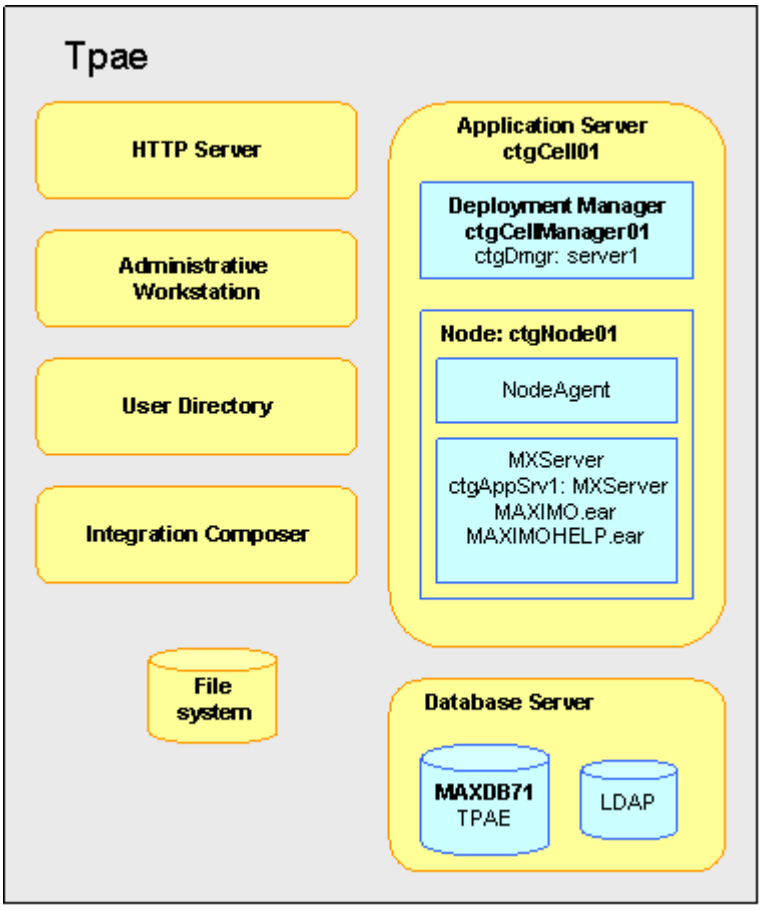


Figure 5 Tpae Components

- Database Server:
  - IBM DB2
  - Oracle Database

  - MS SQL Server
- User Directory:
  - IBM Tivoli Directory Server
  - Microsoft Active Directory
- Application Server:
  - IBM Websphere Application Server
  - Oracle Bea WebLogic Application Server
- HTTP Server
  - IBM HTTP Server
- Admin Workstation
  - Used for Tpae lifecycle operations, like installing, upgrading, installing fixpacks.
- Integration Composer
  - Used products to load actual/discovered data (e.g, computers) from environment.
- File System
  - Used for log files, attachments

3.2 HA architecture for Tpae ecosystem

The HA architecture for the Tpae ecosystem follows basic principles of redundancy with failover, including:

- Eliminating all single points of failure.
- Application Server clustering.
- Redundant Shared storage (disk level replica/mirroring) for data kept in filesystem
- DBMS HA
  - DB2 HADR
  - DB2 PureScale
  - ORACLE RAC
- User Directory HA
  - Tivoli Directory Server Active-Active
  - Active Directory HA solutions

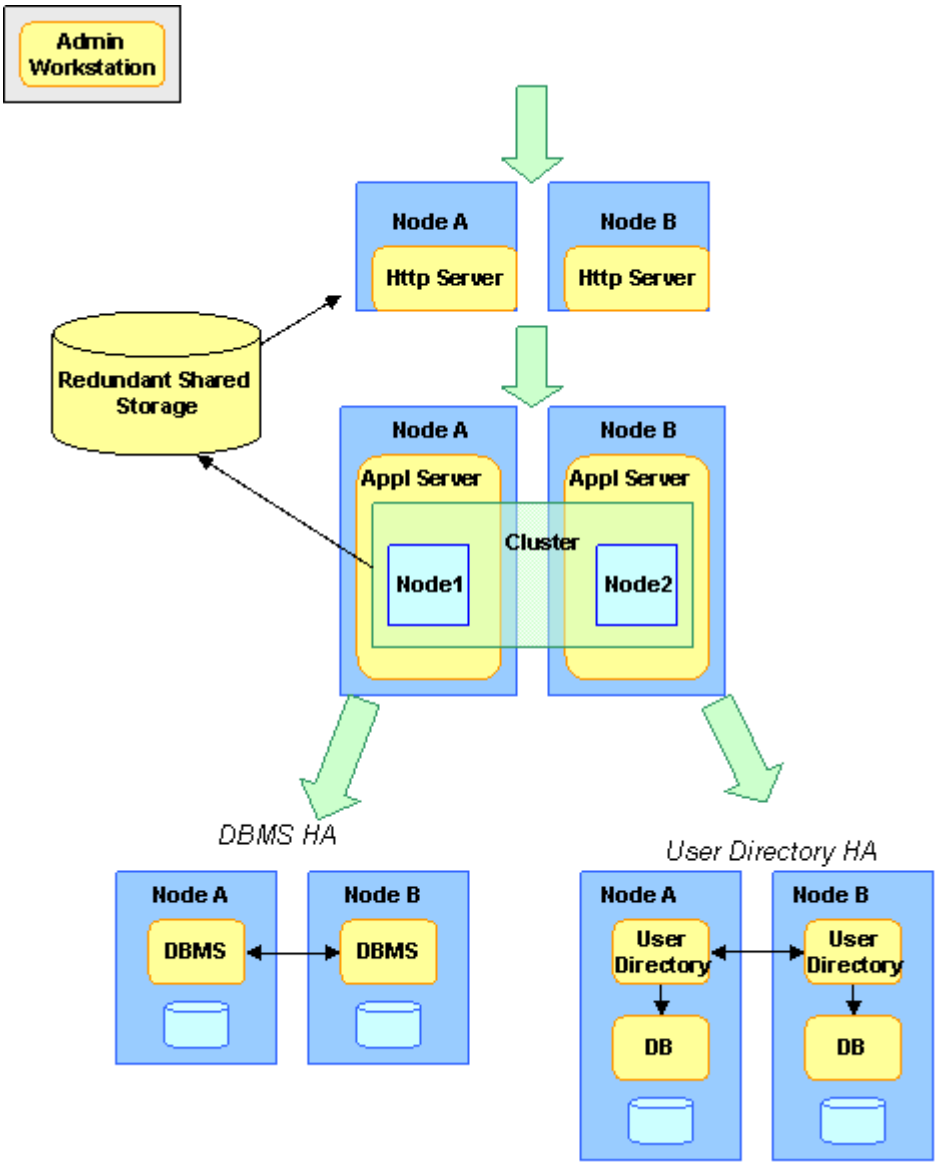


Figure 6 Tpae HA Architecture

3.3 What to expect from the Tpae HA environment

High Availability in your TPAE ecosystem will allow your TPAE application to detect and recover from many different failure scenarios. Standby systems can be brought online and services restored in very short periods of time. In many cases, failed services can be detected and restored in less than one minute.

While High Availability can drastically reduce system downtime, it is important to understand that it is not a symptomless solution and there may be a brief disruption in service during and immediately after the failover sequence has occurred. As the system

will utilize service IP addresses to allow for hostname/IP transparency to the users and connecting applications, most of the symptoms are the same as if the service was manually restarted. TPAE in most cases is resiliant enough to handle the component restart and should not require other middleware components to be restarted afterwards.

Currently, TPAE products cannot handle a UI session failover. What this means is if the specific application server JVM that a user is connected to happens to go down, this session cannot be pushed to another JVM to continue operations. In this scenario, a user will be redirected back to the TPAE login page and will have to re-authenticate. If the Application Server is configured in an active cluster, this redirection will occur immediately but the user may lose any unsaved work.

What you can expect from a Highly Available TPAE environment is the ability to recover from a system, process, network or other type of failure in a short amount of time. Having additional systems available and configured prior to failure will greatly reduce downtime, which can be quite costly and inconvenient. Implementing a cluster manager can automate these procedures, allowing for fast detection and failover to standby nodes. In most cases system downtime can be reduced to less than a minute with a properly configured HA cluster.

Highly available systems can also be used to minimize the impact of system maintenance. Often, systems require some downtime for hardware changes or operating system updates. If an active system requires maintenance, the standby system can be utilized to service application requests until maintenance is complete.

Please review the Symptoms of Component failure section of the TPAE HA documentation. This section describes some of the various UI symptoms when a failover procedure is initiated (or can simply occur when a specific middleware component is restarted while the environment is online):

http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/topic/com.ibm.mbs.doc/gp_highavail/r_symptoms_failover.html

4 Testing your Tpae HA environment

When you have completed your implementation of a HA topology with TPAE it is important to thoroughly test the HA configuration to ensure that failover is working as expected. A successful HA configuration can be determined by simulating failures of each individual middleware component in various ways. You can use commands from the cluster manager to assist with these simulations and takeovers but ultimately the best way to test is to physically take down a node as it may occur in a disaster scenario. Most of the following examples are using Tivoli System Automation for Multiplatforms on a Linux or UNIX operating system.

4.1 Application Server

4.1.1 WebSphere Application Server clustering

The most common application server TPAE HA solution is the WebSphere Active Cluster which can be configured across multiple physical servers (horizontal) with several JVMs per server (vertical). To ensure a highly available application server, vertical clustering alone would not be

sufficient. Having multiple physical servers would allow for a failure of an entire system while still keeping services available on one or more remaining systems.

An important consideration of application server clusters is how many systems could be allowed to fail before the user load would severely affect the system performance to the point where it is unusable. If the current system usage is already approaching maximum capacity and all JVMs are required to keep the environment stable, then this cluster can not be considered highly available. You will want to ensure that at least one entire system can fail and still have enough resources available to support the maximum user load. Some organizations opt to allow for multiple systems to fail before the performance is severely impacted. The amount of extra systems you have for high availability will affect the overall cost but should be weighted against the potential costs of system downtime.

When testing the application server cluster, you can simulate system failures by taking an entire system offline. As TPAE does not currently support session failover, any users logged into the specific application JVMs will be redirected to an online JVM and will have to re-login. Single Sign On users will be redirected back to the start center. This is just a simple test to see if the application failover is functioning, but it would not be sufficient to determine if the remaining JVMs will be able to handle a peak user load. It is recommended that you use a performance validation tool such as Rational Performance Tester (RPT) or HP Loadrunner to simulate a specified number of concurrent users on the system.

A manual restart of the failed services is required in this scenario.

More information can be found at:
http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/index.jsp?topic=%2Fcom.ibm.mbs.doc%2Fgp_highavail%2Fc_ctr_high_availability.html

4.1.2 WebSphere Application Server clustering with TSA-MP for node restart

In addition to your WebSphere clustering, you can also introduce a cluster manager to assist with the detection and automatic restart of services. We recommend Tivoli System Automation for Multiplatforms (TSA-MP) to serve this purpose. TSA-MP uses a set of Resources configured with specified policies to perform this automation. Dependancies can be set such as the Application JVMs depend on the node agent having an online status before they will start up.

Below is an image of a simple two-node WebSphere Horizonal cluster. We use the lssam command with TSA-MP to view the status of the cluster:

```
[root@tpaeha07 ~]# lssam
Online IBM.ResourceGroup:SA-was-as1-rg Nominal=Online
        '- Online IBM.Application:SA-was-as1:tpaeha07
Online IBM.ResourceGroup:SA-was-as2-rg Nominal=Online
        '- Online IBM.Application:SA-was-as2:tpaeha08
Online IBM.ResourceGroup:SA-was-na1-rg Nominal=Online
        '- Online IBM.Application:SA-was-na1:tpaeha07
Online IBM.ResourceGroup:SA-was-na2-rg Nominal=Online
        '- Online IBM.Application:SA-was-na2:tpaeha08
```

Here we see the resource group SA-was-as1-rg with a nominal status of Online. This means TSA-MP will always try to bring the system to an online status in case of failure. This resource group has four actual resources; as1, na1, as2, na2. We can tell that as1 and na1 are both on the same node TPAEHA07 while as2 and na2 are both on TPAEHA08. as1 and as2 are both the application server JVMs while na1 and na2 are the node agents for each system. There is a dependency set that the application servers will not attempt to start until the node agents are online.

To test the automation we can simulate a total system failure in many ways. We could power off a system entirely, or remove the network interface connection from one of the systems. It is important to simulate as many possible types of failures as you can to ensure that the automation software can properly detect these failures.

Below is an image of the lssam command after TPAEHA08 has been powered off:

```
[root@tpaeha07 ~]# lssam
Online IBM.ResourceGroup:SA-was-as1-rg Nominal=Online
        '- Online IBM.Application:SA-was-as1:tpaeha07
Failed offline IBM.ResourceGroup:SA-was-as2-rg Control=MemberInProblemState Nominal=Online
        '- Failed offline IBM.Application:SA-was-as2:tpaeha08 Node=Offline Binding=Unbindable
Online IBM.ResourceGroup:SA-was-na1-rg Nominal=Online
        '- Online IBM.Application:SA-was-na1:tpaeha07
Failed offline IBM.ResourceGroup:SA-was-na2-rg Control=MemberInProblemState Nominal=Online
        '- Failed offline IBM.Application:SA-was-na2:tpaeha08 Node=Offline Binding=Unbindable
```

TSA-MP uses heartbeat and monitor scripts to determine that the node agent and application server are both offline on TPAEHA08. Any users connected to TPAEHA08 will be redirected by the HTTP Server to TPAEHA07 (Note: We cover high availability for the HTTP server in this document as well.)

The final important test you should run is the automatic startup of the WebSphere applications when the system comes back online. After restoring power to TPAEHA08 we see the TSA-MP status as:

```
[root@tpaeha07 ~]# lssam
Online IBM.ResourceGroup:SA-was-as1-rg Nominal=Online
        '- Online IBM.Application:SA-was-as1:tpaeha07
Pending online IBM.ResourceGroup:SA-was-as2-rg Nominal=Online
        '- Pending online IBM.Application:SA-was-as2:tpaeha08
Online IBM.ResourceGroup:SA-was-na1-rg Nominal=Online
        '- Online IBM.Application:SA-was-na1:tpaeha07
Pending online IBM.ResourceGroup:SA-was-na2-rg Nominal=Online
        '- Pending online IBM.Application:SA-was-na2:tpaeha08
```

This shows that TSA-MP has detected that TPAEHA08 is back online and is attempting to bring the services back online. Finally, we should see the full online status as depicted in the first image of this section.

More details about WebSphere clustering with automatic restart can be found at:

http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/index.jsp?topic=%2Fcom.ibm.mbs.doc%2Fgp_highavail%2Fc_ctr_high_availability.html

4.1.3 Warm Standby WebSphere Application Server shared disk through TSA-MP

Some environments require a highly available application server that may not span multiple machines. In this scenario we can use a hot standby configuration for WebSphere. WebSphere is installed to a mountable shared drive and is mounted on a single machine at a time. If failure occurs the drive can be mounted on a standby machine and the WebSphere services started there.

As this configuration utilizes a shared storage system it is important that this shared disk is not a single point of failure. If connection to this shared storage is lost altogether, then neither node will be able to mount and recover the services. TSA-MP or other cluster managers can be used to automate the mount and startup of the required services on the active node.

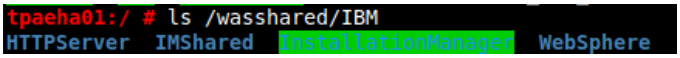
The below image shows an example of a policy configured for WebSphere in Hot Standby mode:

```
tpaeha01:/ # lssam
Online IBM.ResourceGroup:was Nominal=Online
        |- Online IBM.Application:SA-was-as
                |- Online IBM.Application:SA-was-as:tpaeha01
                '- Offline IBM.Application:SA-was-as:tpaeha02
        |- Online IBM.Application:SA-was-na
                |- Online IBM.Application:SA-was-na:tpaeha01
                '- Offline IBM.Application:SA-was-na:tpaeha02
        |- Online IBM.Application:ihs-rs
                |- Online IBM.Application:ihs-rs:tpaeha01
                '- Offline IBM.Application:ihs-rs:tpaeha02
        |- Online IBM.Application:wasshard-rs
                |- Online IBM.Application:wasshard-rs:tpaeha01
                '- Offline IBM.Application:wasshard-rs:tpaeha02
        '- Online IBM.ServiceIP:wasip Request=Online
                |- Online IBM.ServiceIP:wasip:tpaeha01
                '- Offline IBM.ServiceIP:wasip:tpaeha02
Online IBM.Equivalency:was_nieq
        |- Online IBM.NetworkInterface:eth0:tpaeha01
        '- Online IBM.NetworkInterface:eth0:tpaeha02
```

This particular example has several resources; the application server (SA-was-as), the node agent (SA-was-na), IBM HTTP Server (ihs-rs), the shared disk (wasshard-rs) and the service IP (wasip).

The resource group and policies must be configured with dependencies in the correct order. It is important that the shared disk be mounted before starting any of the WebSphere services. The shared disk should only be mounted on the active node and NEVER on multiple nodes at the same time as this can cause data issues. Once the shared disk is mounted and detectable by the cluster manager, then you can go ahead and apply the service IP as well as starting the node agent followed by the application server. IBM HTTP Server is optional but commonly used,. In this case it can be installed on the same shared drive as WebSphere and started after the disk is mounted.

On the active node we can verify that the shared drive path contains all the WebSphere files:



Listing the same directory on the standby node should show that the files do not exist. When triggering a failover, ensure that this drive properly dismounts from the original primary and then becomes mounted on the standby system.

TPAE users will access this environment through the service IP. It is important to note that failover on a hot standby will take longer and is less transparent to the end user than an Active/Active WebSphere cluster. An easy way to test is to disconnect the ethernet interface from the primary node. The cluster manager should detect the network failure and begin to gracefully stop all WebSphere services, remove the Service IP, and dismount the shared drive. The standby node should now become active and the cluster manager will start all services in the correct order after the shared disk is mounted.

Other failure scenarios such as system shutdown, shared disk connection failure, process failures, and so on should be tested as well. From an end user perspective, there should be only a brief moment of disruption until the system is completely online on the standby node. Once the standby is active, users should be able to access the TPAE system and log back into the application. As there is no session replication support, users will be bounced back to the login page (or start center with SSO). You can also issue the rgreq command to move the resource group to the standby system: # greq -o move was

For more information on the warm standby WebSphere configuration please visit:

http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/index.jsp?topic=%2Fcom.ibm.mbs.doc%2Fgp_highavail%2Fc_was_networkdeployment_shared_disk.html

4.2 Database

Your database is one of the most critical components to the TPAE application. It is important that the application servers always have access to the database with minimal disruptions. There are several different types of database servers with many different high availability configurations. It is recommended you review the TPAE High Availability documentation to choose the best solution for your topology. As bringing your database down can cause expensive outages, it is advised that you test your Database HA solution extensively before going live in a production environment.

http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/topic/com.ibm.mbs.doc/gp_highavail/c_ctr_ha_for_databases.html

4.2.1 DB2 High Availability Disaster Recovery (HADR)

DB2 HADR is commonly configured in a two-node cluster with one primary and one standby database. The databases are both online and synchronized through transaction logs that are shipped from the primary to the standby. If communication breaks between the primary and the standby, all transactions on the database are stored in the logs until the systems are both online and replicating. For this reason it is important to make sure there is plenty of extra disk space on both machines incase there is a failure and logs must be generated.

Tivoli System Automation for Multiplatforms can be configured alongside DB2 HADR to assist with detection and automatic failover of the primary during a system failure. Below is a common two-node HADR configuration with TSA-MP:

```
[root@tpaeha03 ~]# lssam
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_MAXDB75-rg Nominal=Online
        |- Online IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs
                |- Online IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs:tpaeha03
                '- Offline IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs:tpaeha04
        '- Online IBM.ServiceIP:db2ip_9_23_7_86-rs
                |- Online IBM.ServiceIP:db2ip_9_23_7_86-rs:tpaeha03
                '- Offline IBM.ServiceIP:db2ip_9_23_7_86-rs:tpaeha04
Online IBM.ResourceGroup:db2_db2inst1_tpaeha03_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_tpaeha03_0-rs
                '- Online IBM.Application:db2_db2inst1_tpaeha03_0-rs:tpaeha03
Online IBM.ResourceGroup:db2_db2inst1_tpaeha04_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_tpaeha04_0-rs
                '- Online IBM.Application:db2_db2inst1_tpaeha04_0-rs:tpaeha04
Online IBM.Equivalency:db2_db2inst1_db2inst1_MAXDB75-rg_group-equ
        |- Online IBM.PeerNode:tpaeha03:tpaeha03
        '- Online IBM.PeerNode:tpaeha04:tpaeha04
Online IBM.Equivalency:db2_db2inst1_tpaeha03_0-rg_group-equ
        '- Online IBM.PeerNode:tpaeha03:tpaeha03
Online IBM.Equivalency:db2_db2inst1_tpaeha04_0-rg_group-equ
        '- Online IBM.PeerNode:tpaeha04:tpaeha04
Online IBM.Equivalency:db2_public_network_0
        |- Online IBM.NetworkInterface:eth0:tpaeha03
        '- Online IBM.NetworkInterface:eth0:tpaeha04
```

The policy is configured using the db2haicu utility shipped with DB2. We can see that TPAEHA03 is the Active node by the application resource and ServiceIP being online on this node only. The DB2 instance and all other resources are online on both nodes. The Service IP is automatically applied to the active node using TSA-MP which allows the database connection URL to stay the same regardless of which node is servicing the database requests. When configuring TPAE with HADR, you will use the service IP hostname in your system properties.

By running an ifconfig you can see that there is an alias interface created with the service IP applied:

eth0:0 Link encap:Ethernet HWaddr 00:xx:xx:xx:xx:xx

inet addr:9.x.x.x Bcast:9.x.x.x Mask:255.x.x.x

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

This alias should only exist on the current primary node. When a failure of the primary occurs, a TAKEOVER command is issued which converts the standby database to the primary. This TAKEOVER command can be manually issued but when using TSA-MP this happens automatically. There will be a brief moment of downtime as the cluster manager detects the failure and switches over to the standby machine. Logs will now be stored on the new primary machine until connectivity is restored to the original primary and can now be synchronized.

DB2 HADR can handle many types of failures such as complete system failure and network interface or connectivity disruptions. A network quorum can be configured on the policy which will allow both nodes to constantly ping a specified IP or several IP addresses (such as the gateway for example). This quorum allows the nodes to detect if they are the ones who are offline due to a network disruption. Heartbeats are sent between the nodes to determine if communication is possible. It is best to simulate many types of failures when testing your HADR configuration with TPAE products.

Below is an example of what the TSA-MP policy looks like when the primary node goes offline:

```
[root@tpaeha04 ~]# lssam
Online IBM.ResourceGroup:db2_db2inst1_db2inst1_MAXDB75-rg Request=Lock Nominal=Online
        |- Online IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs Control=SuspendedPropagated
                |- Failed offline IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs:tpaeha03 Node=Offline
                '- Online IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs:tpaeha04
        '- Online IBM.ServiceIP:db2ip_9_23_7_86-rs Control=SuspendedPropagated
                |- Failed offline IBM.ServiceIP:db2ip_9_23_7_86-rs:tpaeha03 Node=Offline
                '- Online IBM.ServiceIP:db2ip_9_23_7_86-rs:tpaeha04
Failed offline IBM.ResourceGroup:db2_db2inst1_tpaeha03_0-rg Nominal=Online
        '- Failed offline IBM.Application:db2_db2inst1_tpaeha03_0-rs
                '- Failed offline IBM.Application:db2_db2inst1_tpaeha03_0-rs:tpaeha03 Node=Offline
Online IBM.ResourceGroup:db2_db2inst1_tpaeha04_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_tpaeha04_0-rs
                '- Online IBM.Application:db2_db2inst1_tpaeha04_0-rs:tpaeha04
Online IBM.Equivalency:db2_db2inst1_db2inst1_MAXDB75-rg_group-equ
        |- Offline IBM.PeerNode:tpaeha03:tpaeha03 Node=Offline
        '- Online IBM.PeerNode:tpaeha04:tpaeha04
Online IBM.Equivalency:db2_db2inst1_tpaeha03_0-rg_group-equ
        '- Offline IBM.PeerNode:tpaeha03:tpaeha03 Node=Offline
Online IBM.Equivalency:db2_db2inst1_tpaeha04_0-rg_group-equ
        '- Online IBM.PeerNode:tpaeha04:tpaeha04
Online IBM.Equivalency:db2_public_network_0
        |- Offline IBM.NetworkInterface:eth0:tpaeha03 Node=Offline
        '- Online IBM.NetworkInterface:eth0:tpaeha04
```

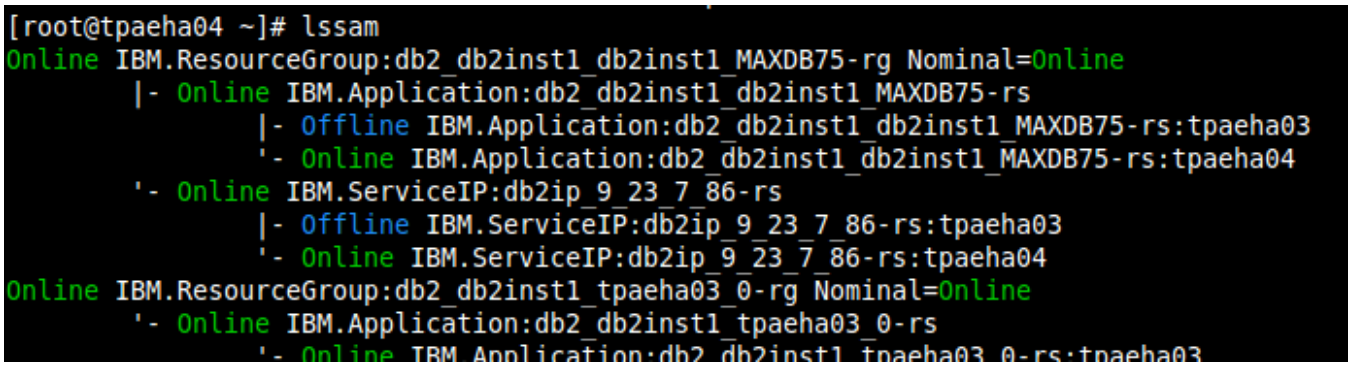
We can see that the interfaces and all other services on TPAEHA03 have gone offline. TSA-MP has detected that TPAEHA03 has gone down and issued the TAKEOVER command on TPAEHA04 to force it to become the primary. The application has started servicing requests on TPAEHA04 and TSA-MP automatically applies the ServiceIP to an alias interface on TPAEHA04. The control mode is in SuspendedPropagated because it is not connected to the original peer so it is locked until the original node comes back online with communication and replication re-established.

While failover is being detected and the TAKEOVER process is underway, there will be a brief disruption in the TPAE UI for users who are currently accessing the system. These UI symptoms are outlined here:

http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/index.jsp?topic=%2Fcom.ibm.mbs.doc%2Fgp_highavail%2Fr_symptoms_failover.html

Once the failover process is complete, users should be able to continue using the application as they would normally. The Service IP allows the takeover of the primary to be transparent to the application.

When the failed node comes back online, the lssam command should show an output similar to the original picture but TPAEHA04 will remain the new primary system until you force it back to the original:



Now if the original node is the desirable primary, you can issue a command to move the resources back to the original machine. In this example we issue the command

rgreq -o move db2_db2inst1_db2inst1_MAXDB75-rg

lssam shows the output:

```
[root@tpaeha04 ~]# rgreq -o move db2_db2inst1_db2inst1_MAXDB75-rg
Action on resource group "db2_db2inst1_db2inst1_MAXDB75-rg" returned Token "0xa1551cf2906ca02e4fd91b5088a40b00" .
[root@tpaeha04 ~]# lssam
Pending online IBM.ResourceGroup:db2_db2inst1_db2inst1_MAXDB75-rg Request=Move Nominal=Online
        |- Pending online IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs
                |- Pending online IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs:tpaeha03
                '- Offline IBM.Application:db2_db2inst1_db2inst1_MAXDB75-rs:tpaeha04
        '- Online IBM.ServiceIP:db2ip_9_23_7_86-rs
                |- Online IBM.ServiceIP:db2ip_9_23_7_86-rs:tpaeha03
                '- Offline IBM.ServiceIP:db2ip_9_23_7_86-rs:tpaeha04
Online IBM.ResourceGroup:db2_db2inst1_tpaeha03_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_tpaeha03_0-rs
                '- Online IBM.Application:db2_db2inst1_tpaeha03_0-rs:tpaeha03
Online IBM.ResourceGroup:db2_db2inst1_tpaeha04_0-rg Nominal=Online
        '- Online IBM.Application:db2_db2inst1_tpaeha04_0-rs
                '- Online IBM.Application:db2_db2inst1_tpaeha04_0-rs:tpaeha04
Online IBM.Equivalency:db2_db2inst1_db2inst1_MAXDB75-rg_group-equ
        |- Online IBM.PeerNode:tpaeha03:tpaeha03
        '- Online IBM.PeerNode:tpaeha04:tpaeha04
Online IBM.Equivalency:db2_db2inst1_tpaeha03_0-rg_group-equ
        '- Online IBM.PeerNode:tpaeha03:tpaeha03
Online IBM.Equivalency:db2_db2inst1_tpaeha04_0-rg_group-equ
        '- Online IBM.PeerNode:tpaeha04:tpaeha04
Online IBM.Equivalency:db2_public_network_0
        |- Online IBM.NetworkInterface:eth0:tpaeha03
        '- Online IBM.NetworkInterface:eth0:tpaeha04
```

So here we see that the primary is being switched back to tpaeha03 as it was originally.

For more information on DB2 HADR with TPAE please visit:

http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/index.jsp?topic=%2Fcom.ibm.mbs.doc%2Fgp_highavail%2Fc_ha_db2_hadr.html

4.2.2 Warm Standby DB2 shared disk with TSA-MP

The DB2 shared disk configuration utilizes a shared disk to store database and instance files. Similar to the WebSphere warm standby scenario, the shared disk is only mounted on the active database instance and can be dismounted and mounted on the standby node in the event of a failure. As this configuration relies on a shared file system, it is crucial this shared disk is highly available and always accessible. A cluster manager such as TSA-MP is used to automate the mounting of the shared disk as well as the application of the Service IP and startup of services.

Below is an example TSA-MP resource group and policy configured for a shared disk DB2 environment:

```
tpaeha07:~ # lssam
Online IBM.ResourceGroup:db2_db2inst1_0-rg Nominal=Online
        |- Online IBM.Application:db2_db2inst1_0-rs
                |- Offline IBM.Application:db2_db2inst1_0-rs:tpaeha07
                '- Online IBM.Application:db2_db2inst1_0-rs:tpaeha08
        |- Online IBM.Application:db2mnt-db2shared-rs
                |- Offline IBM.Application:db2mnt-db2shared-rs:tpaeha07
                '- Online IBM.Application:db2mnt-db2shared-rs:tpaeha08
        '- Online IBM.ServiceIP:db2ip_9_32_163_159-rs
                |- Offline IBM.ServiceIP:db2ip_9_32_163_159-rs:tpaeha07
                '- Online IBM.ServiceIP:db2ip_9_32_163_159-rs:tpaeha08
Online IBM.Equivalency:db2_db2inst1_0-rg_group-equ
        |- Online IBM.PeerNode:tpaeha07:tpaeha07
        '- Online IBM.PeerNode:tpaeha08:tpaeha08
Online IBM.Equivalency:db2_public_network_0
        |- Online IBM.NetworkInterface:eth0:tpaeha07
        '- Online IBM.NetworkInterface:eth0:tpaeha08
```

The DB2 application is dependent on the mounted disk and the service IP being active on that particular node. TSA-MP can detect a failure and dismount the shared disk and apply it to the standby node during failover. Utilizing a Service IP in this policy allows the TPAE application to automatically reconnect to the database after failover occurs without a need to change the database URL in the system properties.

Another consideration when automating the failover of the DB2 services is the db2nodes.cfg file. When you install DB2 and create an instance, it utilizes the db2nodes.cfg file to determine the current hostname. This file does not list the service IP and must be the actual hostname of the active instance. As we have created the DB2 instance on a shared disk, this configuration file must also be changed during failover When using TSA-MP and the db2haicu configuration tool (bundled with DB2) this policy will be configured automatically and the db2nodes.cfg will be changed during automated failover.

Using the environment in the above screenshot, we can take a look at what is currently configured in the db2nodes.cfg:

```
tpaeha08:/db2shared/db2inst1/sqllib # cat db2nodes.cfg
0 tpaeha08 0 tpaeha08
```

The active node is tpaeha08 and is what is listed in the db2nodes.cfg. If we were to simulate a failover to tpaeha07, then the db2nodes.cfg must also be changed to tpaeha07 after it is mounted there.

Testing this scenario would involve simulating as many types of failures as possible such as network, hardware, shutdown, process failures, and so on. Issuing the rgreq command through TSA-MP can trigger a fail over to the standby node. Example: rgreq -o move db2_db2inst1_0-rg. Your TPAE application should be able to reconnect to the database when it is available on the standby node and should not require a restart of the application servers. A good test for this is to log in to the TPAE application and then simulate a failure. There should be a brief disruption in the UI and when the database is back online on the standby node the application should continue to function as normal.

For more information on this configuration please visit:

http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/index.jsp?topic=%2Fcom.ibm.mbs.doc%2Fgp_highavail%2Fc_was_networkdeployment_shared_disk.html

4.2.3 DB2 PureScale

DB2 PureScale is considered an Active/Active database solution. This provides the benefits of a highly available database with the scalability and performance improvements of an active cluster.

There are additional configuration options with TPAE when using PureScale that are outlined here:

http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/topic/com.ibm.mbs.doc/gp_highavail/c_ha_db2_purescale.html

As the TPAE application can utilize something called seamless failover, there should be very minimal disruptions in the application UI when a database node failure occurs. PureScale can detect the failed node and route the database transactions to another active node in the cluster instantly. Testing this configuration can be as simple as simulating a node failure at a system level or a network disruption.

Similar to a WebSphere Active/Active cluster configuration, it is important that your environment has enough nodes in the active cluster to allow for a specified number of nodes to fail before performance degrades significantly. Load testing tools such as Rational Performance Tester can be used to simulate a certain amount of users on the system. As a system administrator, you will need to determine the maximum amount of users that your environment must support and how many nodes it needs perform at an acceptable level. For example, if your DB2 PureScale configuration is working at maximum capacity with a full user load, then a failure of one or more nodes may slow the system down considerably. It is best to allow for at least one node in the cluster to fail while still serving the peak user load at an acceptable level. Some organizations may have a requirement that multiple nodes in the PureScale cluster can fail while still maintaining appropriate TPAE performance levels.

4.2.4 ORACLE Real Application Clusters

Oracle RAC is a similar solution to DB2 PureScale in that it is an active cluster database that can be scaled across many nodes. Oracle RAC requires a special configuration in the maximo.properties when specifying the Database URL. Please review the documentation at:

http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/index.jsp?topic=%2Fcom.ibm.mbs.doc%2Fgp_highavail%2Fc_oracle_rac.html

An important thing to note when utilizing Oracle RAC is that you need to specify all the database nodes in the database URL for TPAE. For this reason, if you decide to scale up your RAC environment and add more RAC nodes you must reconfigure the application and redeploy. Another option is to implement Oracle Single Client Access Name (SCAN) which creates a SCAN listener that you can connect to which will allow for scalability that is transparent to the TPAE application. It is best to research the performance implications to using SCAN to see if it is right for your topology. Also, having one SCAN listener can create a point of failure to the whole RAC environment.

Load testing is an important test for RAC with TPAE. It is best to determine the maximum user load that the organization would require and implement a RAC environment that can allow for one or more nodes to fail before performance degrades to an unacceptable level. Tools such as Rational Performance Tester can be used to simulate users on the system.

If users are in the TPAE UI during the time of a RAC node failure, the failure should be virtually transparent to the end user. Users may experience brief symptoms outlined at the following URL:

http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/index.jsp?topic=%2Fcom.ibm.mbs.doc%2Fgp_highavail%2Fr_symptoms_failover.html

Consult Oracle's website or documentation for more information on Oracle RAC and SCAN.

4.3 User Directory

When utilizing an LDAP server such as Tivoli Directory Server for user authentication with your TPAE product, it is important that this user directory is highly available or users will not be able to log in if connection to the directory is lost

4.3.1 IBM Tivoli Directory Server with Peer to Peer replication

Tivoli Directory Server can be configured with multiple servers connected by a replication method such as Peer to Peer replication. When utilizing multiple servers, only one would be active at a time and connected to the WebSphere Application server using a Service IP. Implementing a cluster manager such as TSA-MP can help automate the failover of the primary.

When testing your Tivoli Directory Server configuration with peer-to-peer replication, it is logical to start with the user directory itself to ensure that the replication mechanism is functioning as expected. The easiest way is to add one or more users and groups manually to the directory on the primary server, then query the secondary servers to check whether the users exist there as well. This can be done either through the Web Administration Tool or through a command line interface. Consult the ITDS documentation for more information on these utilities.

The next part you would want to test is the TSA-MP policy itself. Below is an example policy and resource group configured for ITDS with peer-to-peer replication:

```
[root@tpaeha05 ~]# lssam
Online IBM.ResourceGroup:SA-itds-peer-ip-rg Nominal=Online
        '- Online IBM.ServiceIP:SA-itds-peer-ip-1
                |- Online IBM.ServiceIP:SA-itds-peer-ip-1:tpaeha05
                '- Offline IBM.ServiceIP:SA-itds-peer-ip-1:tpaeha06
Online IBM.ResourceGroup:SA-itds-peer1-rg Nominal=Online
        |- Online IBM.Application:SA-itds-peer1-db:tpaeha05
        '- Online IBM.Application:SA-itds-peer1-server:tpaeha05
Online IBM.ResourceGroup:SA-itds-peer2-rg Nominal=Online
        |- Online IBM.Application:SA-itds-peer2-db:tpaeha06
        '- Online IBM.Application:SA-itds-peer2-server:tpaeha06
Online IBM.Equivalency:SA-itds-peer-equ
        |- Online IBM.Application:SA-itds-peer1-server-shadow:tpaeha05
        '- Online IBM.Application:SA-itds-peer2-server-shadow:tpaeha06
Online IBM.Equivalency:SA-itds-peer-nieq-1
        |- Online IBM.NetworkInterface:eth0:tpaeha05
        '- Online IBM.NetworkInterface:eth0:tpaeha06
```

As with other products the resource group is configured to keep both ITDS servers online at a time. This allows the peer-to-peer replication mechanism to function properly and keep the user directories in sync. Utilizing a service IP we can tell which node is currently the primary. In this screenshot it is tpaeha05. To test failover, simply disconnect the ethernet interface on the primary, kill some of the Tivoli Directory Server services processes, or shut down the machine. This will

force TSA-MP to detect the failure and move the service IP to the standby node. Issuing the rgreq command will have a similar outcome.

What you want to make sure after failover is complete, is that the service IP is applied to the secondary server and TPAE users still exist. There are several ways to check this. First, try to login to the TPAE application itself. Alternatively, you could log into WebSphere Administration Console and check the Users and Groups to make sure they still exist.

Failover of this resource group should be fairly quick (approximately 30-60 seconds). During this time users will not be able to log in to the application but users who are already authenticated before failover should still be able to use the application normally. Side effects of user directory failover are usually minimal as long as the failover process is quick.

For more information on configuring ITDS peer to peer replication for TPAE please visit:

http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/index.jsp?topic=%2Fcom.ibm.mbs.doc%2Fgp_highavail%2Fc_peer2peer_replication.html

4.3.2 IBM Tivoli Directory Server with Proxy Server

To assist with high availability and load balancing you can introduce a Tivoli Directory Server proxy server to the mix. The proxy server accepts authentication requests and connects to WebSphere but forwards the requests to one or more Tivoli Directory Servers. This allows for load balancing if there is a lot of stress on the user directory. Tivoli Directory Server servers can be configured similarly to the previous example “IBM Tivoli Directory Server with peer-to-peer replication” without implementing a service IP into the policy. You can then connect the proxy server to both of the Tivoli Directory Server servers. When implementing proxy servers, it is important that this is not a single point of failure. For this reason we suggest having a standby proxy server that can take over if the primary fails. In this case, Tivoli Directory Server proxy server is installed and configured on a highly available shared file system that is mounted on the active node only. The WebSphere Application Server connects to the active proxy server through a Service IP. Using TSA-MP can help automate this failover procedure.

Below is an example of a highly available Tivoli Directory Server Proxy server that connects to multiple Tivoli Directory Server servers:

```
tpaeha05:~ # lssam
Online IBM.ResourceGroup:db2_dsrdbm01_0-rg Nominal=Online
        |- Online IBM.Application:db2_dsrdbm01_0-rs
                |- Online IBM.Application:db2_dsrdbm01_0-rs:tpaeha05
                '- Offline IBM.Application:db2_dsrdbm01_0-rs:tpaeha06
        |- Online IBM.Application:db2mnt-itdsshared-rs
                |- Online IBM.Application:db2mnt-itdsshared-rs:tpaeha05
                '- Offline IBM.Application:db2mnt-itdsshared-rs:tpaeha06
        '- Online IBM.ServiceIP:db2ip_9_32_163_158-rs
                |- Online IBM.ServiceIP:db2ip_9_32_163_158-rs:tpaeha05
                '- Offline IBM.ServiceIP:db2ip_9_32_163_158-rs:tpaeha06
Online IBM.ResourceGroup:itds-rg2 Nominal=Online
        '- Online IBM.Application:itds-rs-2
                |- Online IBM.Application:itds-rs-2:tpaeha05
                '- Offline IBM.Application:itds-rs-2:tpaeha06
Online IBM.Equivalency:db2_dsrdbm01_0-rg_group-equ
        |- Online IBM.PeerNode:tpaeha05:tpaeha05
        '- Online IBM.PeerNode:tpaeha06:tpaeha06
Online IBM.Equivalency:db2_public_network_0
        |- Online IBM.NetworkInterface:eth0:tpaeha05
        '- Online IBM.NetworkInterface:eth0:tpaeha06
```

This policy and resource group is configured to mount the shared storage to the active node, and only bring the proxy server online in one node. The service IP allows for a transparent connection from WebSphere. This proxy server connects to two Tivoli Directory Server servers that are replicating using peer-to-peer replication.

The first thing to test would be to bring down one of the Tivoli Directory Server servers that this proxy connects to. This enables you to ensure that there is no disruption to the user directory. The proxy server will route all requests to the active Tivoli Directory Server instance so you should still be able to log in to your TPAE application with no symptom. If there are a lot of active users on the system, this could decrease authentication performance and it may take longer to log in.

The next important test would be to simulate a failure of the active Proxy server. This could be by removing the network cable, killing one or all of the proxy processes, shutting down the server entirely or issuing the rgreq command. You should see that the file system is no longer mounted to the primary and is now mounted to the standby system. Once all services are restored and the service IP is applied to the standby, users should be able to authenticate to TPAE. Failover may take 60 seconds or more in this configuration as it relies on a shared storage that must be dismounted and remounted. During this time, users who are trying to log in to the application will get an error and will not be able to log in. In most cases, the failover procedure will have completed upon the users second time trying to log in so the effects are minimal. Users who have already authenticated before the failover should not notice any symptoms in the application.

More details for Tivoli Directory Server proxy servers with TPAE can be found at:

http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/index.jsp?topic=%2Fcom.ibm.mbs.doc%2Fgp_highavail%2Fc_itds_proxyserver.html

4.4 HTTP Server

The HTTP Server builds the URL that users will connect to their TPAE application from. In an Active WebSphere cluster environment, the HTTP Server also acts as a load balancer distributing the sessions across all the active application JVMs. For this reason, it is important to consider the HTTP server when implementing high availability.

4.4.1 IBM HTTP Server with TSA-MP Standby

The IBM HTTP Server can be configured for high availability by utilizing a shared filesystem with the HTTP server installation or by having two or more HTTP Servers installed with an identical configuration. In both cases, a Service IP is used to allow users to connect to the same URL no matter which server the HTTP services are currently running on.

If utilizing a shared storage for HTTP Server, it is important that this shared storage is only mounted on one HTTP Server node at a time. The HTTP server will be started on the active node and can be brought up on the standby in the event of a failure. If the HTTP server configuration files ever change, there is no need to make this change on the standby server as well as the files are shared between both nodes.

If your HTTP server configuration rarely changes, it may be more reliable to install each HTTP server separately and manually synchronize the configuration on each server. This is usually the desirable approach as it does not rely on a shared file system. This also speeds up the failover procedure as there is no mounting required. In either case, TSA-MP can be used to automate the failure detection and failover of the services to the standby node.

Below is an example IBM HTTP Server with TSA-MP policy where the HTTP server is installed separately on two nodes:

```
[root@tpaeha01 ~]# lssam
Online IBM.ResourceGroup:httpd_rg Nominal=Online
        |- Online IBM.Application:tpaehttp
                |- Online IBM.Application:tpaehttp:tpaeha01
                '- Offline IBM.Application:tpaehttp:tpaeha02
        '- Online IBM.ServiceIP:httpd_SIP
                |- Online IBM.ServiceIP:httpd_SIP:tpaeha01
                '- Offline IBM.ServiceIP:httpd_SIP:tpaeha02
Online IBM.Equivalency:httpd_nieq
        |- Online IBM.NetworkInterface:eth0:tpaeha01
        '- Online IBM.NetworkInterface:eth0:tpaeha02
```

Here is the configuration in the WebSphere Application server:

Select	Name	Web server Type	Node	Host Name	Version	Status
You can administer the following resources:						
☐	webserver1	IBM HTTP Server	tpaeha01-node	tpaeha01	Not applicable	
☐	webserver2	IBM HTTP Server	tpaeha02-node	tpaeha02	Not applicable	
Total 2						

You will notice that two web servers are listed but only one is online at a time. Both web servers have an identical plugin configuration.

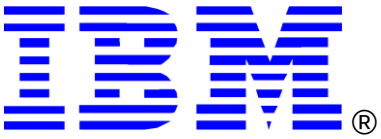
It is a very simple policy and should not take more than 30 seconds to fail over to the standby node. During this time, users will see errors in the TPAE application if they are already logged in, or a failure to load the page if they are attempting to connect for the first time. Some of these symptoms are outlined at the following page:

http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/index.jsp?topic=%2Fcom.ibm.mbs.doc%2Fgp_highavail%2Fc_itds_proxyserver.html

When services are online on the secondary node and the service IP successfully applied, the TPAE application should continue functioning as normal with minimal disruption to the users.

To test this configuration, you need to simulate such failures as system shutdown, process failures, and network disruption, and ensure that the failover mechanism is functioning correctly. Once the failover procedure is complete, ensure that the TPAE application is still available on the same URL.

More details on this configuration can be found at:
http://pic.dhe.ibm.com/infocenter/tivihelp/v49r1/index.jsp?topic=%2Fcom.ibm.mbs.doc%2Fgp_highavail%2Fc_ha_http_server.html



IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PAPER “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes may be made periodically to the information herein; these changes may be incorporated in subsequent versions of the paper. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this paper at any time without notice.

Any references in this document to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
4205 South Miami Boulevard
Research Triangle Park, NC 27709 U.S.A.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks