Testing an AS/400 Clustering Solution: *A White Paper*

*Worldwide AS/400 Availability Team, May 2000*

By employing AS/400 clusters, your business has made a serious commitment to exceptional levels of availability. These levels are not only reliant on hardware and software, but can only be achieved with stringent problem and change management. Hence, identified problems must be replicated. If a change is put in place, this change must tested before further risk to production.

Defining and implementing an HA test environment - The highly available business in its truest sense should be an end-to-end entity. Therefore, when planning the test environment a holistic approach should be taken. This means that not only are the obvious things like hardware, network and applications tested, but also the less obvious components. Examples of these components are the business processes associated with the computer systems, the facilities the business uses, data and applications from external providers, and accurately documented job responsibilities.

One must also consider whether cluster environment is being planned or already exists. If a company was just starting to plan their HA solution, there is considerable implementation testing to be performed. If the HA environment exists it may be a matter of verifying that all recovery conditions have been documented and a test environment created. If this is a new solution, then much of the implementation testing can be carried forward to the problem, change management processes of the production systems.

Test Infrastructure - One of the first tasks to do is create the test infrastructure. This consists of small group of business and IT personnel. One member of the group should have overall responsibility for the test facility, and another very important member is the business test manager. The business test manager should be from a business unit, and have sufficient status to demand the attention of board members. Once these members have been appointed other members can be selected. For example, development test, hardware test, network test. External organizations should not be ignored. Make sure anyone who provides data, service or applications to your business or receives data or service from your business is involved in testing. These organizations may not be involved in all tests, or be required to attend meetings, but they must be considered. Define all roles, responsibilities and escalation paths within the test infrastructure.

The test infrastructure
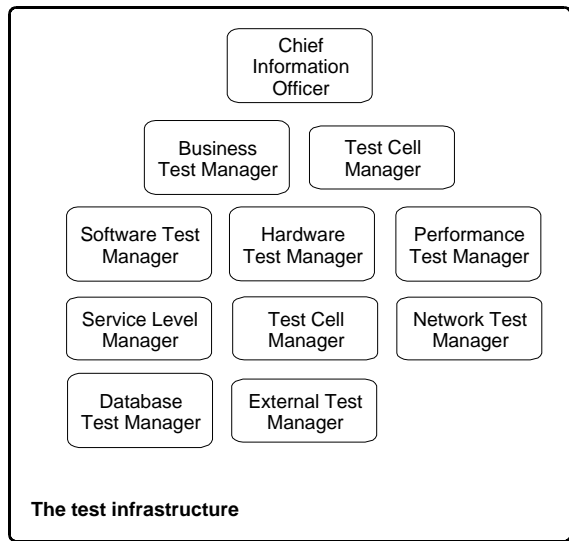
Environment Inventory - Once the infrastructure group is formed then tasks can be assigned.

Test Cell - A typical test cell should include replicants of the systems, and components that exist in production. These replicants do not necessarily need to be equivalent size, but should be able to perform the same functions. Volume testing is a separate topic that will be discussed later.

Start with the simple stuff, make sure there is a true representation of all applications on the systems. A copy of Gold code (e.g. final code version) must exist, to often changes are made to the application and these changes are not reflected in the gold code. This creates the sort of integrity problem that causes recovery to fail. A strict development cycle, with control over changes, movement into production, gold code update must be implemented.

With hardware, make sure that all critical and active components are replicated and if possible form part of the test cell. As new hardware is added, it too much be replicated into the test cell. In some cases if the component is very expense, a loan or rental may be available for testing.

Volume testing - Volume testing is another key element of pre-production testing. Making changes, such as adding a new application or some new hardware can create a potential for loss in availability through degraded performance. If the value of the availability is high, you may wan to considering a dedicated performance test facility like the Rochester Benchmark Center. There, applications can be tested on the latest hardware models, or after a significant application change or addition has occurred, verifying that the changes meets your agreed service levels.

Moving to an HA environment - This illustration depicts a business without an HA implementation. There are two sites with separate but dependent processing environments. Site A is a manufacturing site with a three-tier model, clients, application servers, and database servers. Site B is a distribution site. There is shop floor data collection input to the application. Customers and suppliers both have EDI links to both sites through the WAN, and information flows in both directions. As can be seen, this is already a fairly complex processing environment, and one that desperately needs to be highly available. Looking at this scenario, suppose there is a change to the application, and that change creates an unexpected problem for the end-user. The user is unable to perform the business function, and in
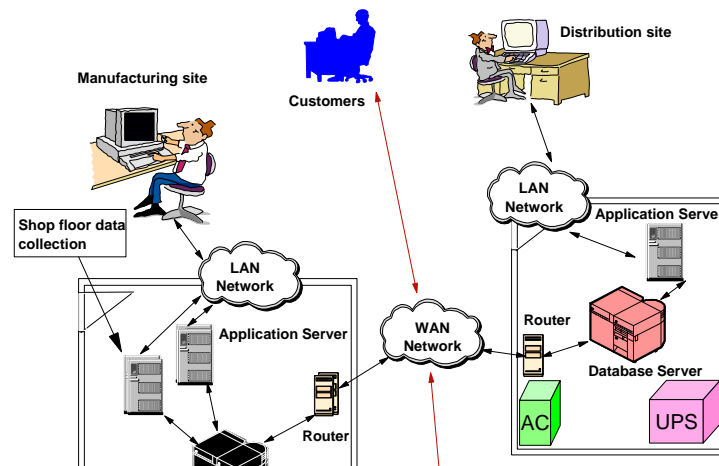
The next illustration shows the same scenario, but made highly available. Now we can see the true level of complexity of a very highly available environment as it applies to the users and hardware. It does not show the business process level. In this scenario the database servers have been given a switch-over/fail-over backup, plus they are also connected to the remote site for disaster recovery. Application servers and database servers are a four-node mutual takeover cluster. From a network perspective, three network providers connect the sites, suppliers and customers. The user population has been split across multiple LANs providing access for some users in the event that a LAN fails. This scenario contains many different components and multiple sites.

### End-to-End testing in production

In this environment testing is both important and complicated. If a change or problem occurs on any component it must be tested before being brought into production. For example, if the IT group decides to upgrade a router. The a pair of the new routers are installed in the test cell. They are tested for control and application function within the simulated environment. Then they are load tested to ensure that they meet the perform required to support the service level agreements. Once these tested have been carried out the new routers are installed in pairs. As there are three routers in the network, resiliency is only partially degraded. These routers will then be monitored for a couple of weeks to finally check that there are no unexpected problems. Once the IT group has accepted these new components, the other pairs of routers are swopped out.

The database and application server is another critical resource that needs careful handling when making changes. Lets look at a database server, and a change needs to be made to the database. There is a definite requirement to test the change before implementing in production. But when the change is made

Testing Examples - The following are examples of testing activities that could be carried out in a highly available business.

Business process testing
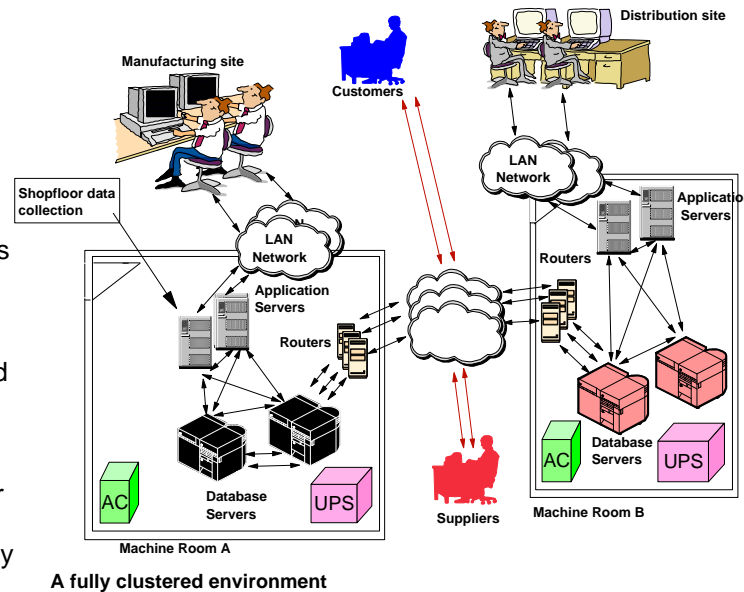
- Staff access (physical access to local and remote locations). If there is a security system in place, does it allow access during a system failure. When the remote location is many miles away, what is the method for moving staff to the remote location and accommodating them.

- Customer/supplier access (network and telephony) Alternative numbers and system access point should be published to customers and suppliers.

- Emergency power - many systems will have UPS support to the machine rooms. However this is no good if there is no power to office space. Client UPS or limited generated UPS to some areas should be considered.

Integration testing

- Hardware components (new and upgrade) - as new hardware or upgrades are added. These must be tested. Some functions will be able to be tested in the test cell, others may require testing during the upgrade process.

- Operating systems upgrade (servers and clients) - most of these tests can be carried out in the test cell.

- Application upgrade - after an application upgrade, new functions should be tested within the business structure. The performance under load conditions should be tested. These load conditions should be representative of peak business periods and for systems that are expected to have the ability to accept workload from other systems during switch/fail over.

- Database integrity (disaster, upgrade, switchover) - if there is a switch/fail over the business must be able to test the integrity of the database. In planning for these tests, establish key business indicators that can be reviewed on restart. Develop a plan for managing the business if the integrity can not established in a very short period of time.

- Interactive job (restart and performance) - interactive jobs loose much of their individual environment during a failure. Test the recovery of these jobs after switch/fail over. After a change to the application, hardware, network ensure the performance of interactive work meets expectations.

- Batch (restart and performance) - batch is normally more complicated than interactive. Jobs can be single or multithread, short or very long running. To test these jobs the test cell will be vital. Allow the jobs to run to completion and check their results. Test performance on similar hardware to the proposed solution.

- Backup subsystems (recovery and performance) - backup management during change and testing is critical. It can be very easy to get out of sync. Normal cyclic backups may have to be supplemented to retain similar levels of backup redundancy during major changes. This is particularly pertinent to staged or phased upgrades and where the tape backup window is very small.

- Access to systems internally and externally - there should be sufficient redundancy to allow changes to the network where old and new components can coexist in product for a short period after the change.

Appendix

The Worldwide AS/400 Availability Team is:

Sue Baker, Steve Finnes, Nick Harris, Eric Hess, Paul Kirkdale, Sue Nee, and Valerie Smith