High-Availablility Infrastructure Architecture Web Hosting Transition

Jolanta Lapkiewicz
MetLife Corporation, New York
e--mail. J0036823@netzero.net

Kazimierz Frączkowski
Technical University of Wrocław Departament Computer Science,
e-mail. Fraczkowski@ci.pwr.wroc.pl

Abstrakt. Application high availability has been the subject of much discussion since the inception of the Hosting Project. It was first thought to be a straightforward extension of Sudna' highly reliable network and web page serving infrastructure to cover a few additional components — a simple matter of leveraging Sudna' experience with other financial customers and defining an enhanced service level agreement (SLA). However, after a great deal of work by a dedicated team, it became apparent that no simple agreement could be developed to cover all of the intricacies that an application running on these additional components could introduce.This analysis picks up from that initial SLA effort and examines the problem of delivering a highly reliable Internet application infrastructure from a perspective that acknowledges that infrastructure architecture is just one part of an integrated, total systems effort that must be focused on the goal of minimizing planned and unplanned downtime. A highly available application is the result of a purposeful application design that is engineered under strict quality control to properly utilize redundant infrastructure components. It must then be put into production and operated under stringent process controls that manage change without interruption and that can react to unplanned problems instantaneously.

1. Introduction

The Internet as a vehicle for doing business with anybody, anyplace, and at any time has been a major force driving the requirement that applications be available to as 24X7. In this “new” connected world, the cost of downtime is high, potentially resulting is lost prospects, lost customers, dissatisfied business partners, and a damaged reputation, not to mention lost productivity and even potential litigation.

In the past, companies have paid “significant premiums” for hardware and system software containing a high degree of engineering to achieve high availability (Donna Scott, Gartner Group, October 16 2000). While this premium has declined over the past few years, the advent of distributed, Internet applications have made high availability more and more difficult to achieve, because of the increasing complexity of these solutions.

The notion of high-availability involves two distinct ideas that are often confused in discussions of this topic. This paper will use the following definitions employed by Gartner analysts (Y. Natis and D. Scott, Sept. 29, 2000):

“A **highly available** application provides end-user access to applications and data during a high percentage (e.g., grater than 99 percent) of scheduled uptime, despite unscheduled incidents. High availability implies the ability to minimize unscheduled (also called unplanned) outages. Implied in high availability is application and data integrity as well as acceptable (as defined by the user) performance.”

“A **continuously operable** application provides end-user access to applications and data during expanded times – typically 24 hours a day, seven days a week (24X7). It implies the ability to minimize

scheduled (also called planed) outages (e.g., outages scheduled for maintenance or upgrade of hardware, networks, facilities, or software)."

"A **continuously available** application combines high availability and continuous operations to avoid or minimize both unplanned and planned downtime. It implies that the application is scheduled to be available 24X7 with a minimum of 99 percent end-to-end application availability (or less than 88 hours of unplanned downtime per year)."

This analysis will focus on approaches to implementing a highly available infrastructure, that is, one that minimizes unplanned outages caused by a failure of any infrastructure component within a single site or a failure of the entire site. For the purposes of this paper, infrastructure is defined as the underlying technology that supports an application. It will only deal with those application design issues that directly affect aspects of functionality related to high availability, e.g. database design issues related to dual site implementations and recovery.

While infrastructure is the focus of this effort, It must be noted that achieving highly available systems involves much more. According to a Gartner study, only 20% of unplanned downtime is caused by infrastructure failures associated with hardware, network, OS-level software, and environmental factors; the other 80% is caused by "people failures," e.g., application failures (caused by bugs, performance issues, or application changes) or operational errors. Achieving the highest levels of availability requires significant investment in making sure that IT processes are engineered to both facilitate change and minimize problems. This large task is outside the scope of this effort, but is being addressed through other initiatives like Change Management, other aspects of the Hosting project, development of IOG, and the efforts of QA.

2. Framework of Alternative Scenarios

In order to help the reader analyze the many options that are available for achieving increasingly high levels of infrastructure availability, a framework is presented in Table 1 below that defines a baseline that is not highly available, and then shows four additional scenarios for achieving higher levels of availability. The first three build off the baseline, adding successively higher levels of availability at additional cost and complexity. The last scenario shows a less complex alternative for ensuring against the least likely scenario of a complete site failure. This analysis assumes that a redundant network infrastructure is in place.

Table 1. Framework that defines five scenarios that deal with increasing levels of high availability that can be achieved for two application types, read-only and transactional applications. (LB, load balancing.)

Scenario	Web Servers	Application Servers	Database Servers	Data
1. Baseline: No HA in a single site, all application types.	Single machine	Single machine	Single machine	Standard disks
2. HA in a single site, all application types.	LB cluster	Appl. cluster	Failover cluster	RAID-5 or 1

3. HA, multi-site redundancy, in active/active mode, read-only applications	Site 1: Active LB cluster	Site 1: Active Appl. Cluster	Site 1: Active failover cluster	Site 1: Active RAID 5 or 1; copied data
	Site 2: Active LB cluster	Site 2: Active Appl. cluster	Site 2: Active failover cluster	Site 2: Active RAID-5 or 1; copied data
4. HA, multi-site redundancy, active/active mode, transactional application	Site 1: Active LB cluster	Site 1: Active Appl. Cluster	Site 1: Active failover cluster	Site 1: Active RAID 5 or 1; remote replication
	Site 2: Active LB cluster	Site 2: Active Appl. cluster	Site 2: Active Failover cluster	Site 2: Active RAID-5 or 1; remote replication
5. HA, multi-site redundancy, active/passive mode, transactional applications	Site 1: Active LB cluster	Site 1: Active Appl. Cluster	Site 1: Active failover cluster	Site 1: Active RAID 5 or 1; remote replication
	Site 2: Passive LB cluster	Site 2: Passive Appl. cluster	Site 2: Passive Failover cluster	Site 2: Passive RAID-5 or 1

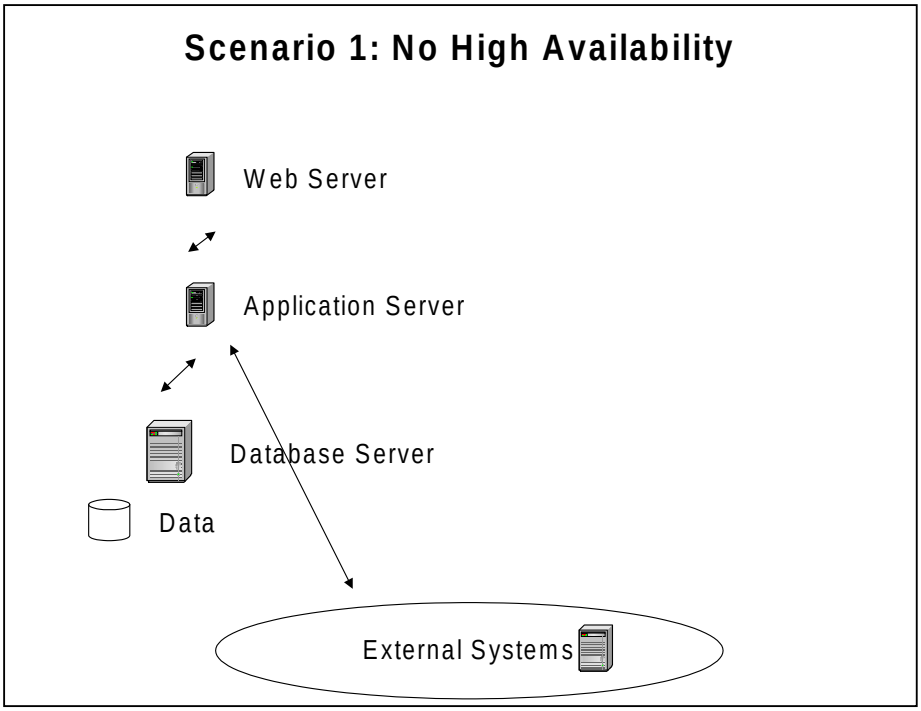

Figure 1. Scenario 1, baseline without high-availability.

Scenario 1. Baseline: no special infrastructure for high availability. The baseline is a typical application with no special effort made to add redundancy and recovery capabilities in the event of a component failure. An off site disaster backup and recovery plan may be in place, but it is assumed that site recovery is greater than a few hours.

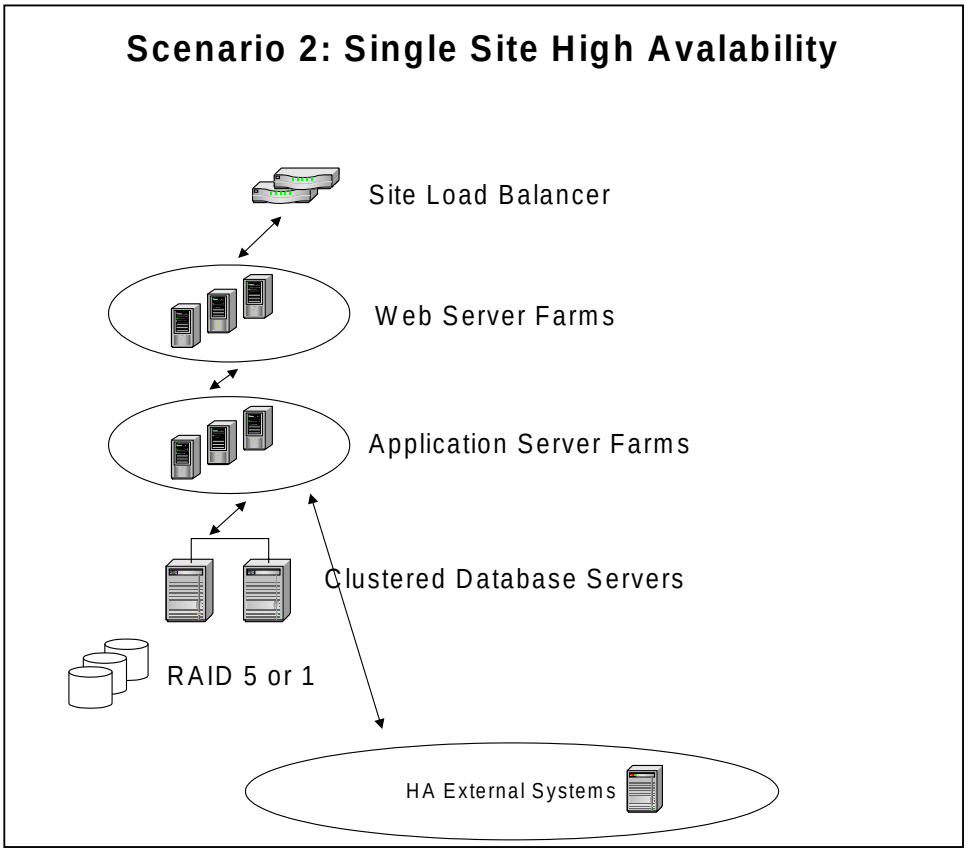

Figure 2. Scenario 2, high-availability within a single site.

Scenario 2. High availability in a single site, transactional applications. Failure of whole site is the least likely scenario so a major focus of any effort that involves a transactional application should be on making a single site highly available. This is important, because the most difficult aspect of running redundantly in two locations involves maintaining two synchronized copies of the database.

Key features include web and application server farms, a database failover cluster, and RAID-5 or 1 redundant storage.

The incremental cost for adding single site high availability to Scenario 1:

- Load balancers, primary and backup. If the baseline application is such that multiple web servers are needed for scalability reasons, then load balancers will already be in place.
- One additional Web Server beyond what is needed to meet capacity.
- One additional Application Server beyond what is needed to meet capacity requirements.
- Additional DBMS server with same capacity as the primary server to handle the entire workload in case of failure.
- RAID 5 or 1 array, depending upon performance requirements. RAID 5 will require 12% more DASD than non-RAID environment, while RAID 1 requires 100% more DASD. The RAID subsystem itself is expensive, but becomes a smaller percentage of the total cost as the amount of data increases.

Outstanding risks:

- Load balancers and failover equipment do not catch all possible software failures.
- Application design must account for session failover in case an Application Server node fails.
- Database clustering failover will result in an outage of service varying from 3 to 20 minutes.
- RAID will not prevent corruption of the database. Normal database backups are still necessary.
- No high availability in case of site failure (including degradation of performance) or environmental problem affecting user's access to the site.

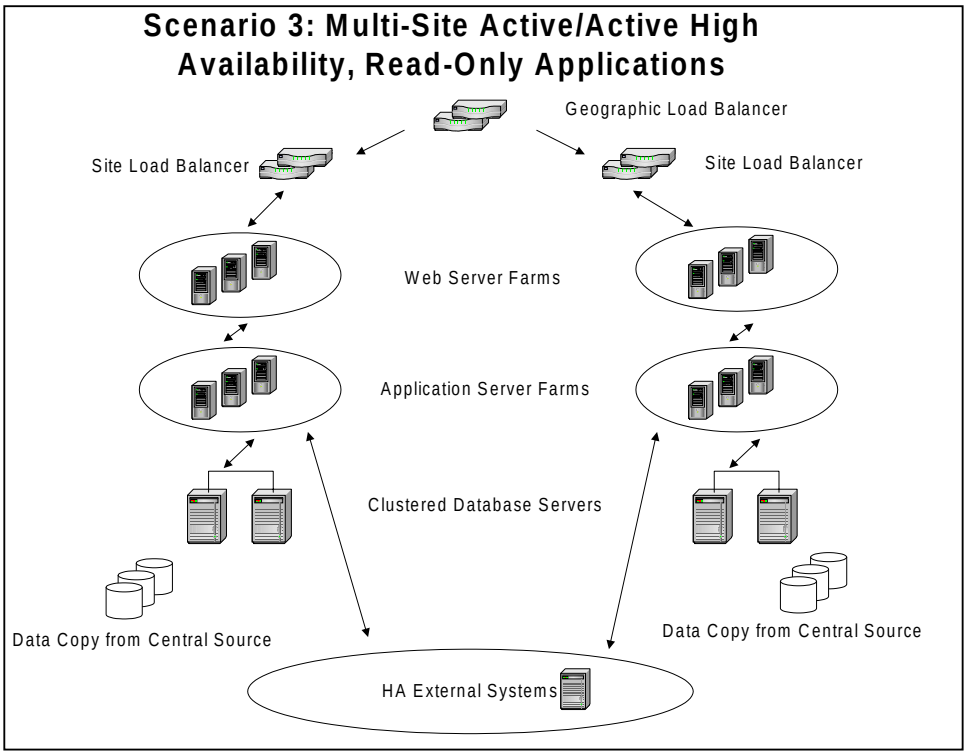

Figure 3. Scenario 3,multi-site high availability for read-only applications.

Scenario 3. HA, multi-site redundancy, in active/active mode, read-only applications. This scenario includes a copy of the entire environment at a geographically remote site. Since the data is read only, both sites can simultaneously process user queries and a geographic load balancer is used to distribute requests to the least-loaded site. In the event of a major site failure, all traffic is automatically routed to the available site. Batch updates to refresh the read-only content are applied simultaneously to the two sites. This scenario is often employed for ensuring high-availability of static web sites, e.g., the existing IXC public web sites.

The incremental cost for adding multi-site high availability to Scenario 2, for read-only applications:

- Cost of building out a second site, facilities and network.
- Geographic load balancers, primary and backup.
- Double the number of web servers required for a single site. If degraded performance can be tolerated in the event of a site failure, then less than 2X the number of servers are needed. E.g.,

for a site that requires 10 servers in a single location; the multi-site scenario may require only 14 servers, 7 in each location. It may be acceptable to the business owners that in the event of a site failure, only 70% of capacity will be available. A major benefit is that in this scenario the extra capacity is always on line and able to handle unplanned spikes in demand during normal operations.

- Double the number of application servers required for a single site. The implications for employing less are the same as for web servers.
- Duplicate DBMS cluster. As with web and application servers, each of the database machines do not need to have the ability to support the full workload if the business can afford to operate in degraded mode for the duration of a site outage.
- Duplicate data storage facility, plus a development effort to handle updating the two databases and making sure they are exact copies of each other. Depending on the size of data and the amount of batch updates, this can be a small or very large effort.

Risks:

- Session management across both sites is not possible, so if session state is important, the application must employ a sticky bit in the geographic load balancers to ensure a user is routed back to the same site for the duration of an entire session.
- The geographic load balancers must be able to monitor events that define a site failure, e.g., unpingable local load balancers. Any other event must be handled by the high availability local infrastructure.
- Introducing two copies of the data increases the risk that an operational error will result in their being out of sync and providing users inconsistent results.

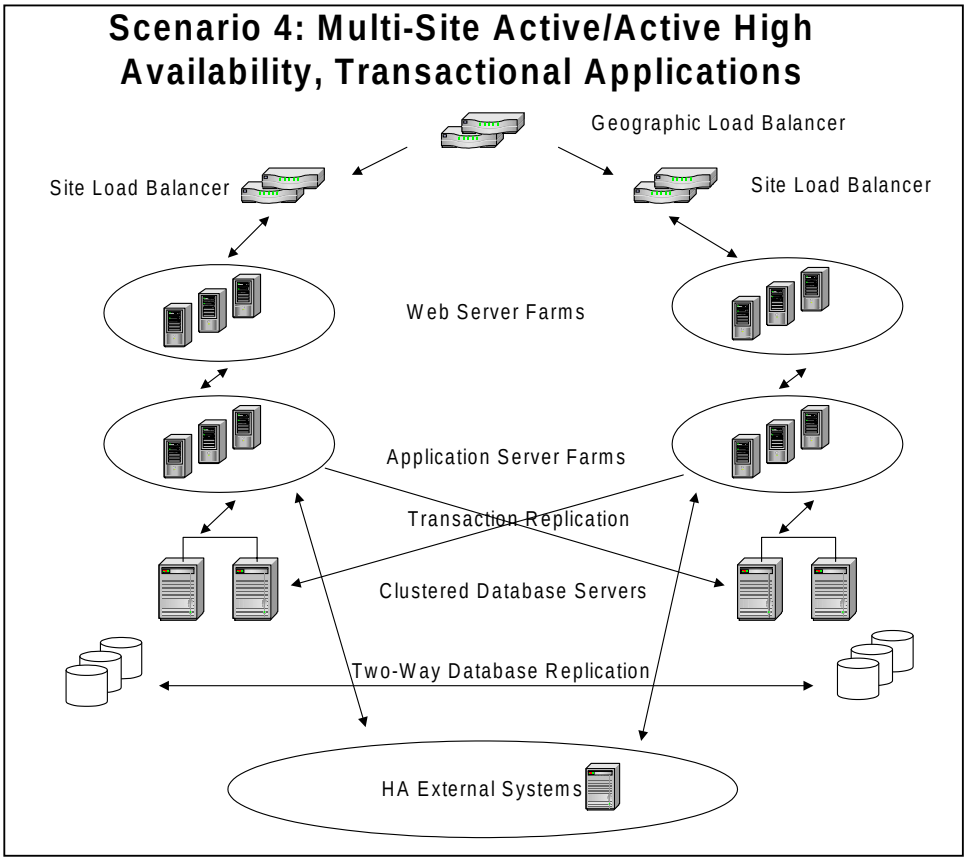

Figure 4. Scenario 4, multi-site high availability for transactional applications with either application-level transaction replication or database-level data

Scenario 4. High availability, multi-site redundancy, active/active mode, transactional application. This is the most difficult scenario to implement, because two independent databases must be kept in sync as users at either live site submit transactions that write to their respective local databases. This solution is not widely employed and involves a large development effort, with an exhaustive test cycle to ensure that the resulting very complex environment is correctly implemented. Storage-level replication scenarios are not appropriate in this situation, because both databases must be on line and processing transactions. Two programming approaches are available; one involves transaction-level replication and the other database replication. Only transaction-level replication, where both databases are simultaneously updated in a single, two-phase commit transaction, ensures complete real-time synchronicity of the two environments. However, this cure may be worse than the disease, because the most likely failure would not be a site failure, but a failure related to the complexity of the environment that must be implemented to support the distributed update. Programming to accommodate this failure scenario in complex, involving queuing and restoring the failed transactions, e.g., queuing up failed transactions in a redundant queuing environment like IBM's MQ Series product. Performance may also be an issue, since each transaction would include an additional remote write. An alternate approach would be to employ asynchronous remote updates, either at the transaction level by writing programs to queue the remote updates (again MQ can be employed) or by defining two-way database replication schemes. The later has been employed by Bank of America, where Oracle has reported that they have achieved a transaction rate of 20 transactions per second in a financial application. This approach introduces a time interval where the two databases are

not in sync (before the remote asynchronous updates have been applied) and the application team must address this.

The incremental cost for adding transactional capability to Scenario 3:

- Cost of developing and testing a complex program modification.
- Cost of developing database consistency checks.
- Cost for a Queuing subsystem if the application development approach is used. Database replication is included in the cost of the database management system.

Risks:

- The complexity of the solution introduces additional points where the system can fail. The fact that this scenario is not widely implemented indicates that this is not a trivial development effort.
- With some approaches, there will be an interval, which cannot be exactly determined, where the databases will be out of sync.
- The additional overhead of a wide area write can adversely affect the performance of high-transaction applications.

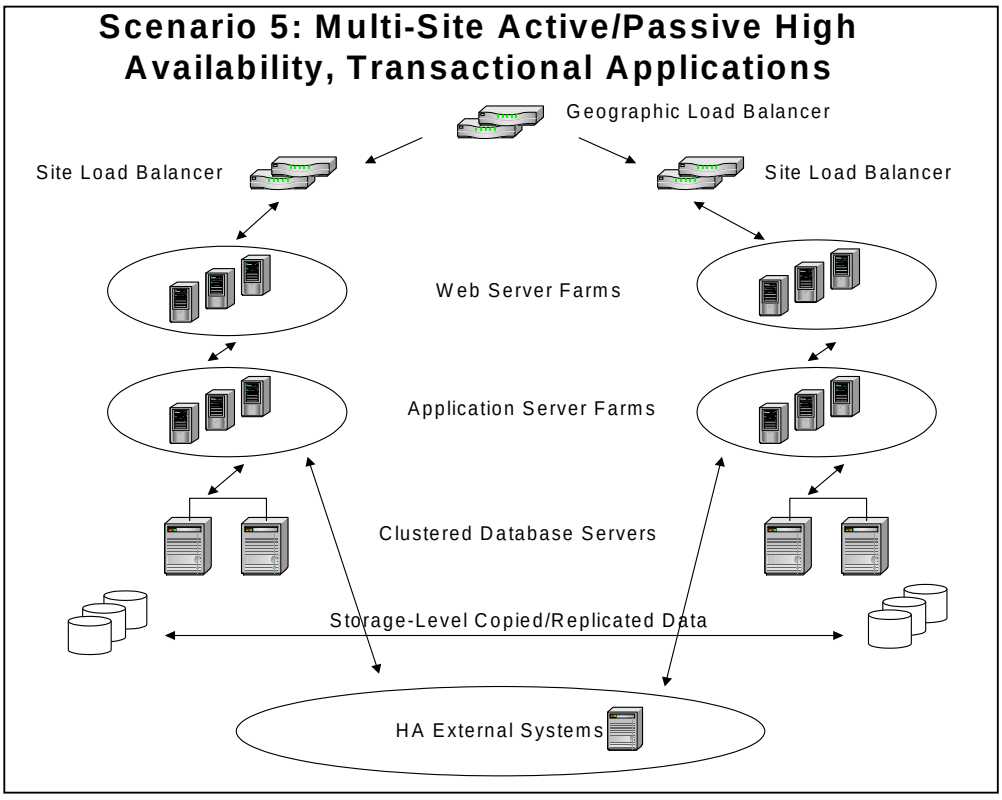

Figure 5. Scenario 5, multi-site failover to a passive site for rapid recovery in case of site failure.

Scenario 5. High availability, multi-site redundancy, active/passive mode, transactional application. Because of the difficulty of implementing Scenario 4, this scenario is introduced as a simpler, less costly alternative for transactional applications. This scenario is based upon the assumption that a complete site failure is the least likely unplanned outage, and in the unusual

circumstance where a disaster occurs, it is acceptable to the business to tolerate a brief outage as the passive (inactive) failover facility is brought on line.

This approach builds upon Scenario 3, but the second site is not active and storage-level replication is used to keep the remote database up to date. Tools for this have been previously discussed and are available from IBM, EMC and Veritas. If the DBMS is Oracle, facilities are available to keep the remote database on line in a standby mode, which will reduce recovery time in the event of a failure.

When a site failure is detected all systems can be automatically started via remote clustering support provided by either IBM or Veritas.

The incremental cost for adding transactional capability to Scenario 3:

- Remote replication software and/or hardware that can support it.
- Geographic load balancing is not needed for this scenario.

Risks:

- Careful planning is necessary for a remote recovery: fist the remote site has to be brought on line and then the DNS routers have to be re-pointed to the secondary site.
- Detecting appropriate events that signal site failure requires careful implementation and testing.
- Remote clustering tools are not mature.
- Cost of maintaining spare equipment for redundant site.
- The additional overhead of a wide area write can adversely affect the performance of high-transaction applications.

3. High-Availability Decision Tree

The decision tree provided in Figure 6 below can be used to help determine the most appropriate approach for any specific application.

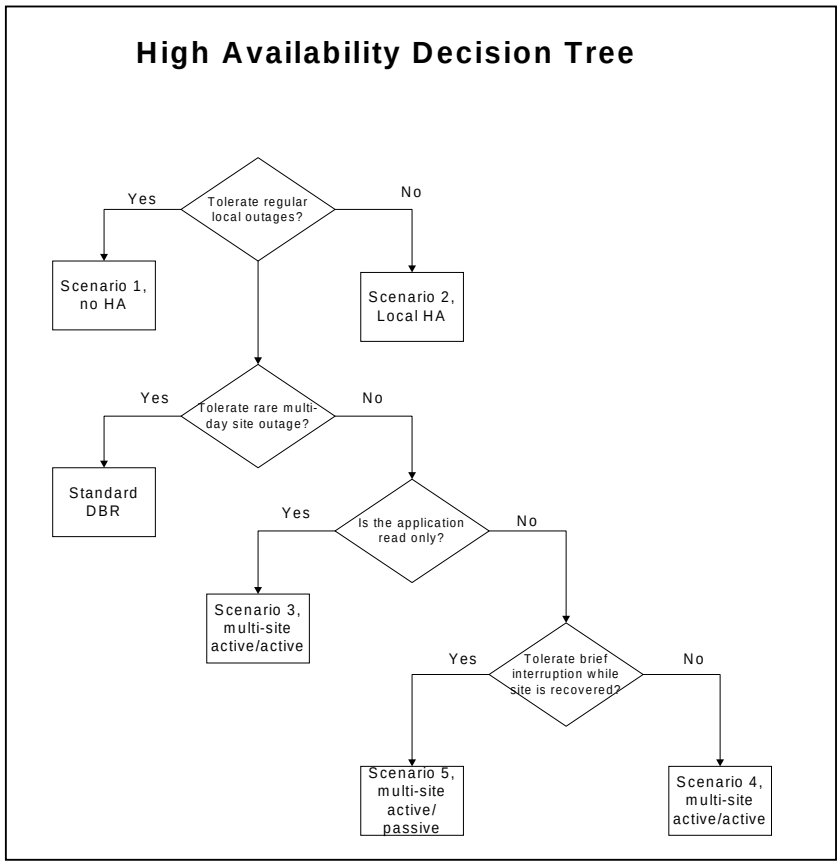

Figure 6. Decision tree for analyzing high-availability option.

The first decision addresses local availability.

- If the business can tolerate outages due to component failures (infrequent, but which regularly occur) or systems maintenance, then Scenario 1, no special precautions, is acceptable.
- If regular outages cannot be tolerated, consider implementing Scenario 2, hardening a single site.

The next decision point revolves around off-site recovery in the event of a disaster.

- If the business can tolerate an outage of several days, then a typical DBR (disaster backup and recovery) plan can be implemented at a second site, utilizing existing equipment deployed for other purposes and tape backups.

Multi-site recovery options depend upon whether an application is read only or transactional.

- For a read-only application, if the business cannot tolerate an extended recovery, no matter how Infrequently it may occur, then Scenario 3, which features two simultaneously active sites, should be considered. This deploys resources most efficiently.

Because of the complexity of multi-site operations for transactional application, it must be decided whether a brief interruption in service can be tolerated in the event of a site disaster.

- If a brief interruption of service cannot be tolerated, then Scenario 4, featuring two sites on-line, with transactions writing to duplicate databases at either site, is optimal. However, this is very difficult to implement successfully.

- If a brief interruption of service can be tolerated in the event of a site disaster, then Scenario 5, which features a passive second site, is appropriate. This is the most expensive to implement from a hardware/software cost perspective, but more readily achieved than Scenario 4.

4. Conclusions

Achieving a level of availability that is "good enough" to meet the needs of an e-commerce application can be a daunting task, the options are many and the tradeoffs and payback difficult to assess. A framework was presented based upon five scenarios to help application developers and their business partners structure their analysis and determine an optimal appropriate. Below is a summary of the key ideas and recommendations developed in this report:

- Don't consider a complex application as a monolithic entity, where all parts need the same level of availability. For instance, while a home page must be continuously available, it may be acceptable to handle some subsystem outages by notifying users that requests will be handled at a later time and then storing information in temporary files or, alternatively, notifying users that a particular transaction is temporarily unavailable and they should try again later.
- Re-engineer and automate as much systems management as possible to minimize the opportunities for "people failures," which Gartner Group studies indicate account for about 80% of unplanned outages.
- Focus on the most likely potential problem areas first and incrementally add additional measures only after a through cost-benefit analysis. Use the framework and decision tree to develop detailed plans for each scenario.
- Redundant components are the key to high availability. The database management system is the one component that cannot easily be made redundant, so care should be taken to choose a platform that is least likely to fail.
- Design the applications for high availability and push ISVs to make their applications highly available. This extends to making sure that the applications are release tolerant, i.e., they can be upgraded while the system is available to users.
- Test and re-test the system to ensure that failover mechanisms actually work. Consider using planned failovers as part of the normal change management process.

Appendix: Research References

“24X7 E-Commerce Availability”, Donna Scott, Gartner Symposium/ITXPO 2000.

“Building Continuous Availability Into E-Applications,” Y. Natis and D. Scott, Sept. 29, 2000.

Private communication with Donna Scott, Gartner Group, October 31, 2000.

Private communication with Charlie Garry, Meta Group, October 6, 2000.

“Building Continuous Availability Into E-Applications,” Y. Natis and D. Scott, Gartner Advisory, 2000.

IBM Storage Solutions: http://www.storage.ibm.com

IBM Clustering Solutions:

http://www-1.ibm.com/servers/aix/products/ibmsw/high_avail_network/

IBM Data Replication: http://www-4.ibm.com/software/data/dpropr/

Private communication regarding IBM high-availability solutions, with Rich Hughes, IBM; Richard Zamier, IBM; Craig Bickerstaff, Articulent; Peter Sjoberg, Articulent; October 18, 2000.

Veritas clustering and replication products: http://www.veritas.com/us/products/#cluster

"No Data Loss Standby Database," EMC White Paper.

EMC storage and replication solutions, http://www.emc.com/products

Oracle database information: http://www.oracle.com/ip/deploy/database/index.html?content.html

Private communication regarding Oracle high availability alternatives, Kevin Martyn, October 17, 2000.

"Unraveling the Mysteries of Clustering," Ron Anderson, Mike Lee, and Steve J. Chapin, Network Computing, Vol. 11, No. 19, October 2, 2000.