

Paper # 4050: Point-to-Point Availability

Laura Finkelstein
VERITAS Software
1600 Plymouth Street
Mountain View, CA 94043
415-335-8409

Computers have become an integral part of doing business. As companies become more and more reliant on electronically stored information, the amount of storage needed to accommodate this transition continues to grow. Remaining competitive, therefore, requires the ability to quickly and continuously access information. As additional computing resources are introduced to accommodate increased business demand, the cost of downtime becomes much more expensive, and the challenge of providing highly available resources becomes more difficult.

The key to creating an availability solution is to determine the impact on operations of being unable to access computing resources on each server within an enterprise. Putting a dollar figure on downtime and quantifying its most prevalent causes will help determine which solutions to implement. With concrete information in hand, it is much easier to calculate the return on investment of the various availability solutions, and thereby justify the cost of the chosen solution.

This paper will highlight the main causes of downtime and recommend techniques that maximize availability, performance, and manageability of critical computing resources. The first section briefly discusses the types of problems that interrupt information availability and examines the likelihood of any one of these factors causing downtime. The subsequent sections provide solutions to the more common causes of downtime and suggestions to improve system performance. With an understanding of these concepts, businesses can formulate the appropriate approach to providing quick, continuous access to information.

Causes of Downtime

The most noticeable and adverse form of downtime is unplanned downtime. When information becomes inaccessible suddenly and for an indefinite period, the business cost is the highest. Unplanned downtime can translate into inability to book business, ship revenue, or react to changes in the financial marketplace. In some cases, short-term downtime has the long-term impact of losing current clientele.

The most common causes of unplanned downtime are disk drive, computer hardware, and computer software malfunction. Human error is also frequently cited as an information availability issue. The most devastating,

though relatively infrequent, causes of downtime are building/site catastrophes, such as fires, and natural disasters, such as a storms or earthquakes.

While unplanned software, server, and disk outages do represent a large portion of the overall availability problem, there are other significant issues. Current research indicates that system upgrades and maintenance are responsible for a large portion of downtime, so there is a growing trend to focus on planned downtime when evaluating system availability

The final area dealing with availability of information resources is system performance. Slow system response, which prevents quick access to information, can be very expensive to businesses in several ways. The businesses most severely impacted are those that need “real time” information updates to make split second decisions and take immediate reactive steps, for example finance and brokerage operations. To these companies, the cost of slow system performance can rival that of unplanned system downtime.

For any business whose success is based on quickly handling transactions and/or customer requirements, slow system performance negatively impacts competitiveness. Because each transaction that requires accessing computer information is much slower, more people must be hired to handle the load of requests. If the company chooses not to hire more people to adjust for slow system performance, the outcome will be longer and longer waiting times for customers, which will lead to customers taking their business elsewhere.

So far, this paper has examined the causes of downtime and the impact on competitiveness of not having quick, continuous access to information. Point-to-point availability is only achieved when each component of a system is optimized for availability. As the system components are optimized for availability, barriers to continuous data access become manageable, and mission critical application servers are able to perform the functions for which they are intended. The following sections proffer solutions that dramatically reduce the costs of downtime.

Guarding Against Disk Failures

Disk failures are the single largest cause of unplanned downtime. Luckily, stopping such failures from preventing access to vital information is probably the easiest to address. Therefore, it would be logical to assume that when striving for optimal availability, designing in a solution that prevents system downtime caused by disk drive failure would provide the “biggest bang for the buck”.

The most effective way of dealing with disk failure is to implement data redundancy. As part of maintaining data redundancy, it is important to have an automatic mechanism for returning information to a redundant state as soon as possible after a disk fails. RAID techniques are the solution generally proposed to achieve data redundancy. The most prevalent RAID redundancy techniques are RAID-1 (mirroring) and RAID-5 (data redundancy through use of parity). Striping (RAID-0) when used in conjunction mirroring can further improve performance by allowing the load of data requests to be spread across multiple disks.

When choosing one implementation over another, the key trade-offs are in performance, cost, and level of protection against disk failures. Additionally, RAID solutions can be provided by either hardware arrays with specialized controllers to offload RAID operations from the central CPU(s) or software products in conjunction with any storage. When making the choice between hardware and software RAID implementations, manageability is also added into the equation.

Mirroring plus striping (RAID 1+0) provides the best performance and the highest level of protection against disk failures. Since mirroring requires twice the storage space of the non-redundant information, it is a more

costly solution than RAID-5 which requires only $1/n$ th (where n is the number of non-parity columns in a RAID-5 object) additional storage. RAID-5 provides data redundancy at the lowest cost, but sacrifices performance, particularly for write intensive applications. Multiple disk failures, at the same time, in a RAID-5 layout will require the system to be brought down and for information to be restored from backup. There is an additional impact that information stored since the backup will be lost unless the application has redo logs to reenter the lost information.

Mirroring performs faster than RAID-5 for write intensive applications because RAID-5 needs to calculate parity when new data is written. RAID-5 reads generally perform better than reads to data that is mirrored but not striped, because data seeks are spread across more disks. However, mirroring plus striping, using an equivalent number of non-parity stripes, will perform faster reads faster than RAID-5, if round robin mirroring (reading information from the optimal mirror) is implemented.

To attain a higher level of protection against multiple disk failure, a focus must be put on reducing additional points of failure. One such point of failure, which greatly affects availability, is the controller. If both copies of a mirror or two disks in a RAID-5 object are accessed through a single controller, the loss of that controller will make the data become unavailable. The key is to make sure that the mirror copies and each disk in a RAID-5 object are accessible by different controllers. Configuring storage so that a redundant object is not impacted by a controller failure reduces another potential cause of unplanned downtime.

After a disk failure(s), the key is to make the data redundant again quickly and seamlessly. The technologies of automatic hot sparing and of hot relocation help to achieve this. When a disk fails, the hot sparing process recreates all the redundant information of the failed disk on another disk(s) that has been designated as a spare. Hot relocation is a more evolved method of hot sparing because only the actual disk partitions that have failed are relocated; if no spare disks exist, free space is used to restore redundancy.

When deciding between hardware and software implementations of RAID techniques, the key decision points are price, level of protection against multiple disk failure, performance, and manageability. Generally, hardware implementations will perform faster and with a lower overhead on computer CPU resources. Software RAID solutions are much less expensive and more flexible than hardware RAID products. On-line configuration changes, performance tuning, and allocation of storage space are much simpler with software RAID. In fact, for some hardware RAID solutions, all configuration changes must be made by the supplier's own support staff. Software RAID solutions also provide more protection against single points of failure. Specifically, software RAID can be used to achieve data redundancy across multiple controllers and multiple storage arrays. Many hardware RAID solutions do not provide protection against a controller failure and none provide protection against an array failure.

If cost is not an issue, the optimal solution consists of both hardware and software RAID. The hardware RAID component can be used to improve performance, while the software RAID can be used to improve manageability and reduce potential points of failure. For example, if RAID 1+0 is desired, the HW RAID can be used to implement striping on two different controllers and the software RAID can ensure protection against controller and array failure by mirroring striped storage areas on two separate arrays. Additional performance can also be achieved by using software RAID to stripe across hardware array striped storage.

Being Prepared for Computer Hardware Failures

There are a variety of approaches for dealing with computer hardware failures. The key trade-offs being price and level of availability. Fault tolerant hardware, clustering products and fail-over software all provide solutions to this problem.

In the fault-tolerant hardware arena, the key focus is on redundant parts. Redundant power supplies, cooling supplies, and CPUs as well as uninterruptible power supplies provide a solid framework to deal with a variety of hardware failures. If the hardware failure is in one of these areas, no downtime costs will be incurred. Purchasing a hardware fault-tolerant solution is, of course, very expensive. Additionally, hardware failures outside the scope of the architectural redundancy will cause the system go off-line.

Clustering solutions are also rather expensive and may require a combination of customized hardware and software. Clustering solutions can be used for availability as well as performance. Additional performance can be obtained by balancing the application load across multiple servers. Clustering offerings are composed of multiple systems accessing shared memory. If one system in the cluster fails, another system automatically and immediately takes over the load of that system and responds to requests of users to the currently off-line system. All systems in a cluster continually access shared storage, so no downtime is incurred when one of the systems goes off-line.

Fail-over solutions (which actually fall under the umbrella of clustering) are probably the easiest and the least expensive approach to implement. On the flip side, the amount of time it takes to restore access to the information is roughly equivalent to the time it takes one system to go down and come back up. In a fail-over configuration, each system has detectors in place to determine if the other system has gone off-line. (Generally, it is advisable to have redundant detectors to remove the detector as a single point of failure.) When one system believes the other system has gone down, it forcibly imports the other system's storage and brings the storage to a consistent state. This generally means that a file system check (fsck) needs to be run on the imported file systems. To reduce the time it takes to fail-over, using a journaling file system is highly advised. With a journaling file system, only a replay of logged file system structural changes is required, thereby reducing a fail-over that might take hours, to one that takes less than a minute. The optimal fail-over solutions work with commodity hardware and can be configured quickly, without requiring outside consulting.

With fail-over software there are trade-offs available to reduce costs by giving up some application performance. The most cost-effective fail-over solution is to have two production systems acting as stand-bys for each other (symmetric fail-over). In this configuration, when one system fails, the other responds to the user requests required of both systems. This reduces the speed at which requests are executed if the additional load exceeds the bandwidth of the surviving system. In some cases, no appreciable degradation in performance is detected. A more expensive approach, that guarantees consistent performance, is a fail-over configuration with a standby machine (asymmetric fail-over). The standby machine is inactive until a fail-over from the primary machine occurs. A common approach to reducing costs while maintaining performance is to have a development/test machine functioning as a stand-by for a production machine. When the production machine fails, developers stop work on the stand-by machine. In this way, the "stand-by" hardware is being optimally utilized.

Preparing for Software Failures

Many software application vendors, such as SAP, incorporate automatic load balancing of user requests among multiple application servers. These applications protect against downtime caused by the failure of an application server(s). It must be remembered that the database server that these applications are accessing requires a fault-tolerant hardware, clustering, or fail-over solution to prevent hardware failure of the database server from being the single point of failure.

An application can stop working even though nothing is actually wrong with the software. In these cases, simply restarting the application fixes the problem. In other situations, the application has been damaged and

a new image of the software is required. Fail-over software coupled with application monitoring agents addresses both of these situations. The agents will monitor critical application daemons. If an agent detects a failed daemon, it will try restarting the daemon. If after multiple tries at restarting the application, it continues to fail, the agent will initiate a fail-over to another system with a working copy of the application. Tunable agents are available from fail-over software vendors for a variety of applications. Additionally, many fail-over products provide toolkits to build “homegrown” agents.

Alleviating Human Errors

Human errors are very difficult to protect against. There are two areas to focus on here: reducing human errors and quickly recovering from those that occur. The best way to reduce human errors is have administration tools that make it very easy to manage the system, and to develop procedures that make it less likely that a mistake will occur.

Tools that make system administration easier generally “up level” tasks. That is to say, an administrator should be able to easily visualize storage resources and direct distribution of critical information resources to optimal storage locations. Additionally, an administrator needs to be able to make changes to existing configurations, reacting to changing business requirements without impacting productivity.

When a human error causes the deletion of critical information, the solution is easily restorable backup. For applications, that do not have redo logs, the key is a frequent incremental backup to reduce the cost of lost information. The ability to perform on-line backups provides for reliable, up-to-date backups without any disruption to users, to most effectively guard against human error. Even for applications with redo logs, easily administrated backup is critical for fast recovery of information, to restore access to production information resources as quickly as possible.

Planning for Disaster Recovery

Up until now, all the solutions proposed have been implemented at a single site. Another threat to availability is building, site, or metropolitan disasters. These occurrences, while being much less frequent, are the most devastating of failures. When one of these disasters occurs, access to information could be delayed for days. In fact, if off-site backup of information resources is not part of the backup strategy, the information may never be recovered. The solutions proposed for reacting to large-scale disasters revolve around replicating information to remote sites. Implementing any of these solutions is relatively expensive.

Many database vendors provide replication products to percolate changes at one site to all others. This is also used to reduce the load of and increase the performance for customer requests by geographically distributing databases. The downside of these database replication products is that only database information is replicated, not any of the other information on the system that may be needed.

For replicating all server information to a backup site, there are several solutions in the market today. Most solutions offer two modes of replication: synchronous and asynchronous. Asynchronous transfer doesn’t require acknowledgement of data receipt to continue on with operations. Therefore, it is a quicker form of replication, with the downside that when the disaster occurs, some transactions will be committed to users, but not committed on the backup site server. Synchronous transfer requires commitment from the remote server before a transaction is committed. To prevent synchronous confirmation from crippling operations, high-speed networking interconnects need to be included as part of the solution.

Managing Planned Downtime

Planned downtime is by definition easier to plan for. There are many reasons why a system would need to be brought down or information taken off-line. The optimal approach allows as many of these operations as possible to be performed on-line without interruption to users. Many of tools previously mentioned in the unplanned downtime sections can be used to address planned downtime requirements.

Probably the most important regular maintenance activity required is system backups. The goal here should be to eliminate downtime required to do backups. There are several tools available to do online backup so that no interruption to users needs to be planned for this activity.

Growing a file system is another frequent administrative function that generally is delegated to the late night hours. Some file systems that have quickly increasing data requirements or have not been sufficiently monitored can run out of space at any time and may require immediate maintenance during the business day. When this happens, planned downtime cost is really equivalent to unplanned downtime cost. Traditional file systems require bringing down the system, backing up the file system, creating a larger storage partition and building a new, larger file system. Acquiring a file system that has on-line resizing capabilities allows file system changes to be made any time of the day. Coupled with on-line growth of storage devices allows quick, easy configuration changes alleviating potential human errors.

Clustering and fail-over products mentioned in the Hardware Failure section can be of great use when performing software upgrades. When desiring to upgrade the operating system or applications on a system, that system's storage can be failed over to the server while the upgrade is taking place. By using this approach, users are not impacted by software upgrades.

Adding these tools to a business-critical system simplifies system administration, enabling planned downtime windows to be greatly reduced or eliminated. The result is that the cost of planned downtime is reduced and that system administrators can perform maintenance tasks more conveniently during the business day.

Improving Performance

As mentioned in the Causes of Downtime section, performance bottlenecks can translate into increased costs or reduced revenues for companies. To aid in combating this problem there are many tools available.

At the most basic level, the choice of underlying storage has a great impact on application performance. Striping of data across multiple disks can greatly improve response times. If certain disks are overloaded and therefore reducing overall performance, moving data to less busy disks can improve performance.

Purchasing an extent-based file system can also provide further performance gains by allowing files to be stored contiguously. Extent-based file systems are particularly useful for large I/O transfers as time consuming disk head movements are reduced. A file system that can be easily tuned to the underlying storage layout for optimal performance enhances the investment in data optimization tools.

In the database arena, administrators have typically turned to raw devices to achieve the best performance since traditional file systems degrade performance an average of 30%. Unfortunately, managing and backing up raw devices are more difficult than file system, thereby requiring more administration man-hours and potentially causing more human errors. File system technology is now available to allow applications to run on a file system at raw disk speeds.

Evaluating Availability Solutions

Figuring out the total cost of unplanned downtime can be very difficult. The two factors that need to be addressed are cost of an outage and frequency of outages. The cost may vary depending on if the outage occurs at a peak or off-peak business period. The frequency, of course, is only a prediction.

Historical outage information is very useful in projecting future costs and frequencies. Internal downtime information as well as industry benchmarking can form the basis for an hourly outage cost as well as for the average duration of outages and number per year. Sometimes the cost of a single outage can justify the purchase of an availability solution.

Metropolitan and site disasters need to be addressed differently from other outages. In these cases, if a recovery solution is not in place, critical information may be lost and business grinds to a halt. The cost of these solutions are generally more expensive and are more complex than the “on-site” availability solutions. In addition, the frequency of these outages is very low.

Determining the cost of planned system downtime involves not only understanding the costs of lost business when taking a system down during the optimal (least busy) times, but also the staffing costs of doing system maintenance during off hours. Some businesses have really no good window for downtime. For these companies, minimizing or eliminating planned downtime is almost as important as addressing unplanned downtime. Additionally, when system administration staff are unable to make system changes during the business day that might prevent unplanned downtime or address system performance bottlenecks, these issues may then need to be addressed at a time when changes are much more costly.

Planned downtime costs are a little easier to manage and predict. System and software upgrades can be planned for times when no business is transacting. Monitoring of growing storage requirements can reduce emergency system maintenance during the business day. Operations that have no windows of inactivity need to use the cost of downtime during “off-peak” periods as an estimate for planned downtime expense.

The cost of slow performance can be measured by determining how long a user is waiting for a system response instead of doing work. In businesses where the longer waiting time translates to more personnel being required or business being lost, a cost can be associated with quick access to information. This analysis can be used to do a return on investment for system performance enhancement options.

In doing a return on investment analysis, it is important to include cost of ownership estimates for the availability and performance solutions. Product quality, customer support responsiveness, and product integration are some of the pieces that impact the cost of implementing a business solution.

Summary

This paper has attempted to explain why designing availability into information resources is currently a topic of great concern to data dependent businesses. Not being able to access critical enterprise information can be severely damaging to a company’s reputation and its bottom line. Fortunately, there are many solutions available to alleviate most data accessibility problems.

Understanding vendor offerings, technology roadmaps, and customer references will help businesses evaluate a variety of solutions. Testing solutions in small, trial situations reinforces appropriate selection, and allows

problems to be ironed out before full-scale implementation. With the right approach, business applications become more available, more manageable, and more competitive.

© 1997 VERITAS® Software Corporation. All rights reserved.