Reliability and High Availability

Businesses have come to expect very reliable computer systems today, even as they spend less for servers that are more powerful than ever before. But reliable components do not always yield a highly available system. This paper examines some of the terms associated with reliability and availability, as well as defining a general spectrum of system availability. This spectrum starts with standard reliability and progresses through high availability to fault tolerance.

Reliability

In its simplest terms, reliability is the probability that a component or system will perform a task for a specified period of time. It is often measured as a function of the time between system failures using the term MTBF (Mean time Between Failure). Computer and component builders usually set MTBF goals through extensive testing and quote MTBF in thousands of hours. Of course, some components are inherently more reliable than others, and a system can be said to be only as reliable as its weakest component.

There are many features found in PCs and entry-level servers that enhance their reliability. One such feature is the memory self-test at boot-time in which the system detects and isolates bad memory blocks, allowing the system to ignore the bad blocks and operate normally. Others include ECC (Error Checking & Correcting) memory, and automatic retries of intermittent failures.

High Availability

The more a business depends upon a computer system, the more it needs the continuous availability of its systems. Availability is measured as the percentage of time that a system is functioning and usable. For instance, a system that provides 99% availability on a 24-hours-per-day, seven-days-per-week basis would actually experience the loss of 88 processing hours a year, which is unacceptable to many users. However, a 99.999% level of availability translates to 5.25 minutes of unscheduled downtime per year, but this level of availability may be far too costly to achieve. Users must look at their mission critical applications (those whose downtime would prove to be extremely costly or detrimental to the business's core services) and the cost of each hour of downtime of those applications. Users can then determine which combination of high availability features to purchase that will get them to an acceptable level of availability. High availability features help keep a system operational even when one or more components fail. Of course, the preferred method of ensuring availability is to avoid any unscheduled downtime before a failure takes place. Predictive failure and avoidance features help predict potential component problems, thus avoiding failures and associated service disruptions. If failure can be predicted, an operator or automated management subsystem can then take the component out of service in a controlled manner, possibly while the system remains on-line. Other features fall into the "recoverability" realm; these features help a system recover quickly after a failure.

Examples of high availability features are:

- RAID (Redundant Array of Independent Disks) which offer data redundancy through disk mirroring.
- Server management tools which monitor hardware and software, and allow system administrators to proactively prevent surprises such as full-disk situations. Various thresholds can be tuned and configured so that, when exceeded, the system administrator can be paged or emailed.
- UPS (Uninterruptible Power Supply) which can keep a system running during a power failure, at least long enough to shut down gracefully.
- Powerfail Recovery which allows the system to save the contents of memory to disk before shutting down, then restores memory after power returns so that applications are restored to their prior states.
- Redundant components such as power supplies, fans, and other components that are likely to fail at some point.

- Hot pluggable (or hot swappable) drives, fans and power supplies which can be replaced while the system is on line. (When combined with RAID, a failed disk can be hot-swapped and the data rebuilt on-line).
- System or application failover which allows two or more servers to be clustered so that in the event of hardware or software failure, applications can be automatically restarted on another system without human intervention. Communications resources and users are also switched over so that users experience only a few seconds or minutes of downtime.

It is important to recognize that system availability is influenced by four factors: hardware, software, people and environment. High availability features like those in the above list can reduce the likelihood of hardware and software failures, but they can only attempt to minimize the effects of human and environmental problems.

Fault Tolerance

The highest level of system availability is delivered by fault tolerant systems, which incorporate redundancy in virtually every aspect of the system. The tradeoff for the continuous availability provided by true fault tolerant systems is that they tend to be very costly, and until recently were associated only with proprietary hardware and software. In addition to high cost, performance generally takes a back seat to system availability. However, users of applications that require nothing less than 100% availability, such as air traffic control systems, understand that uptime is more critical than lightning speed.

The clustering solutions offered by high availability systems, on the other hand, are built on standard building blocks (i.e. traditional commercial servers) that offer a single operational view. When one server fails, the failover process is seldom immediate or transparent; clients will likely see error messages for up to several minutes until the failover is complete. Thus, there is an obvious tradeoff where a small amount of downtime is tolerated to avoid the high cost of incremental uptime.

Key Points to Remember

- Four factors influence system availability: Hardware, Software, People, and Environment.
- Common High Availability/Reliability features include ECC memory, RAID, UPS and hot swappable components.

- Businesses need to determine their cost of downtime to help identify what level of availability they can justifiably afford.
- Even if it were possible to create hardware and software that never failed, human and environmental factors still create the need for redundancy and failover functions.