

Maximizing Web Server Availability

By John W. Graham

Analyzing mean time to failure (MTTF) and mean time to recovery (MTTR) ratios can help Web server administrators determine the causes of downtime and the obstacles to efficient recovery. Administrators can then implement preventive measures, such as failover plans or proactive system monitoring, to minimize future downtime.

Downtime is costly. Not only does it frustrate business users and end users who expect Web servers to be constantly available, but it also results in lost productivity and possible lost revenue. Damages are not limited to financial costs—downtime can harm company reputation, erode customer loyalty, and strain relationships with suppliers, channel partners, business partners, banks, and employees.

Downtime costs can be divided into three categories: hard costs, semi-hard costs, and soft costs. Hard costs are attributed to the hardware, software, and IT staff time and resources required to remedy an outage situation; semi-hard costs include lost business or transaction time during an outage; and soft costs include items such as lost end-user productivity, public relations damage control, public confidence, and lost business opportunities. Quantifying semi-hard costs and soft costs is difficult, but not doing so significantly understates the actual cost of downtime.

Web server administrators can minimize downtime by increasing system availability (see Figure 1). Systems achieve high availability when the number of failures and the time required to recover from failure decrease. To determine the best method for decreasing these factors, administrators should collect and analyze the availability data for their particular systems to determine what areas are most vulnerable to failure.

The costs associated with downtime should determine what level of availability is necessary for your business. For example, a five-person company might be able to tolerate some downtime and experience little cost from it. Conversely, downtime is unacceptable for a financial or online business and the higher the availability, the better.

Measuring availability and reliability

Although no de facto industry standard exists for calculating availability, mean time to failure (MTTF) and mean time to recovery (MTTR) are the metrics most commonly cited.

MTTF measures the average amount of time between failures, where failure is defined as a departure from acceptable service of an application, a computer system (or one of its components), or the network system connecting computer systems.

MTTR measures the average amount of time required to repair a failed system.

Availability is the ratio of the time a system is actually available to the time it should have been available. This ratio is calculated using both MTTF and MTTR:

$$\text{Availability} = \text{MTTF} / (\text{MTTF} + \text{MTTR})$$

For example, if a Web server fails once in six months and it takes 20 minutes to restore

The costs associated
with downtime
should determine what
level of availability
is necessary for your
business.

Level of availability	Availability (percent)	Downtime per year
Commercial or standard	99.5	43.8 hours
Highly available	99.9	8.75 hours
Fault resilient	99.99	53 minutes
Fault tolerant	99.999	5 minutes
Continuous	100	0 minutes

Figure 1. Levels of system availability

the service after the failure, then the Web server availability is:

$$6 \text{ months} / (6 \text{ months} + 20 \text{ minutes}) \times 100 = 99.92\%$$

Because a system with a nominal failure rate but zero recovery time is indistinguishable from one with zero failures, administrators can improve availability by increasing MTTF, reducing MTTR, or both.

Baselines and data analysis

When collecting data about system availability, administrators should study an adequate sample of servers over a sufficient and meaningful amount of time to ensure the integrity of the data. A reference point, or baseline, is also necessary to define the availability status quo and to demonstrate improvements or declines in availability.

After establishing an availability baseline, administrators should compare, categorize, and rank the data to determine current availability and downtime causes. Administrators can disregard planned downtime but should analyze unplanned downtime, categorizing it by failure source (such as software, network, or security) and calculating the total unplanned downtime for each category. The results should reveal which areas need the most attention in terms of increasing availability.

Downtime: Planned and unplanned

Planned downtime occurs when a system must be shut down for maintenance, repair, upgrade, or any other planned event. Unplanned downtime results from a failure, which may be one or more of the following:

- ▶ **Software.** A misconfigured or outdated operating system (OS), third-party drivers, system software, applications, or applications residing on Microsoft® Internet Information Services (IIS) such as HTML, Microsoft Active Server™ Pages (ASP), or

XML (Extensible Markup Language) can cause software failure. Subtle incompatibilities between applications, programming errors in applications, and programming errors in the underlying software of a database, file system, or OS also can cause software failure.

- ▶ **Hardware.** Hardware component failure can cause a server to become unavailable. In rare cases, a cascading effect occurs, causing other hardware components to fail and an outage to occur.
- ▶ **Security.** Security issues can cause server failure or denial of service.
- ▶ **Configuration.** Poor configuration in parts of the system, such as in the hardware or OS, can cause a server to become unavailable.
- ▶ **Network.** Any configuration, modification, bottleneck, or outage within the network can cause a server to become unavailable.
- ▶ **Operations.** Any operational action, such as a modification, or an omission, such as a failure to respond to an issue, can cause server downtime.
- ▶ **Environment.** Any natural disaster or external incident can cause loss of service.
- ▶ **Power.** Any instance of a power failure in which the uninterruptible power supply (UPS) does not function properly will likely result in downtime.
- ▶ **Unknown.** In some instances, it may not be possible to identify the root cause of an outage.

Methods for reducing unplanned downtime

To reduce unplanned downtime, administrators must improve availability. Sufficient failover and planned escalation processes can decrease MTTR; proper testing, configuration, and monitoring can minimize the chance of failure and maximize MTTF.

Tested and certified configurations. Software and OS failures often occur because of improperly configured systems, failure to adhere to standard configurations, or use of third-party drivers and software. To avoid such failures, Web server administrators should use tested, certified configurations for their systems.

Load balancing and traffic management. Microsoft Network Load Balancing (NLB) or traffic management devices such as the Dell® PowerApp.BIG-IP help administrators manage high-availability Web sites. These load-balancing devices can significantly increase the overall uptime of a site by routing users to available Web servers. Administrators can remove failed or unavailable servers from the cluster so that the

Administrators
can improve availability
by increasing MTTF,
reducing MTTR,
or both.

Web site remains highly available even when individual servers are not.

Hot spares. Keeping hot or warm spares in a convenient location—so administrators can quickly swap them with the failed server—dramatically reduces MTTR. Administrators can then analyze the failed server in a controlled, detailed manner to determine the root cause of failure.

Proactive hardware management. Because hardware follows a predictable life cycle, administrators can identify when particular hardware components are about to enter “fail mode.” Scheduling planned downtime to replace parts prior to failure helps to minimize MTTR. Administrators also can avert the cascade effect, in which one hardware problem leads to another, through aggressive, well-defined responses to monitoring alerts.

Limiting system modifications. Web server environments present increased risks for downtime because Web pages and content are often updated multiple times a day. The number of modifications made to a stable, working production system is directly proportional to the probability that the system will fail. Limiting the number of system changes ensures fewer failures and higher availability.

Service Level Agreement. A Service Level Agreement (SLA) between a product vendor and a company can minimize MTTR. The company chooses which level of support is required for its system and the vendor guarantees swift repair and recovery within a specific time period. Larger IT departments in some businesses often provide SLAs to other departments within the company. When IT departments offer SLAs, they are likely to have contingency plans and escalation procedures in place.

Formal change and control processes. Administrators can help to prevent failure through tight operational procedures including regular, complete backups and avoidance of unnecessary changes to systems, applications, and network configurations. Stringent processes, such as documentation and trouble tickets, also provide a record of previous problems so that administrators can predict future problems and recovery times.


Running ASP applications out-of-process (IIS 5.0). Microsoft IIS 5.0 has added out-of-process application pooling to allow the configuration of several ASP applications outside the Inetinfo.exe process space but within a single Microsoft Transaction Server (MTS)/ASP subsystem. (In IIS 4.0, each out-of-process application required its own MTS and ASP subsystem running—one for each instance.)

Web server
environments present
increased risks for
downtime because
Web pages and content
are often updated
multiple times a day.

IIS 5.0 can run ASP applications out of process with reduced memory and reduced processor usage while still retaining crash protection for IIS. For example, if a poorly written, out-of-process ASP application causes a failure, the Inetinfo.exe process will keep running and IIS will not crash. This crash protection allows the server to maintain high availability.

Minimizing downtime to maximize performance

Reliability and availability data provide a map to determine where a system might fail and how failure responses help or hinder recovery.

Building recovery and availability plans based on this data can prevent Web server downtime from ruining site performance and business relationships. 

John W. Graham (john_graham@dell.com), a product marketing manager in the Enterprise Systems Group at Dell, has worked on projects such as the PowerApp.web product line and Application Center 2000. Prior to Dell, he worked at Microsoft on the Windows® 2000 Advanced Server. John has degrees in Computer Information Science and Law and Society.

FOR MORE INFORMATION

Contingency Planning Research, Inc.:
<http://www.contingencyplanningresearch.com>

DataPro. “High Availability: An overview.” 1998

“Deploying Microsoft Windows NT Server for High Availability.” Redmond, WA: Microsoft Corporation, 1998

EverGreen Data Continuity, Inc.:
<http://www.evergreen-data.com>

Garzia, Mario. “Microsoft IT Showcase High Availability SQL Server Platform Engineering.” Redmond, WA: Microsoft Corporation, 1999

Tippet, Peter. “Security for the CXO.” *Information Security*. March 2001
http://www.infosecuritymag.com/articles/march01/columns_executive_view.shtml