WEB SERVER MONITORING

Sorin POPA, Associate Professor, PhD.
University of Craiova

Keywords: monitoring systems, web server, monitoring process

Abstract. This paper introduces web-server monitoring, explaining its importance and describing various monitoring concepts and types. A set of reasons of web-server monitoring are enumerated. Then, in a monitoring strategy, various types of monitor are presented along with a comparison between deep and shallow monitors. Monitoring process meaning and elements are described, revealing that the monitoring process is largely dictated by the monitoring software package in use. A comparison between internal and external monitoring methods, benefits and limitations of each of them, is done. The managing problems are discussed, and the importance of maintain highest possible website uptime and availability is revealed. Finally, we argue that an immediate notification of problems will significantly increase the chance that the problem can be fixed without affecting customers or users.

In today's new Internet economy, a well-designed and smoothly operating Web site provides a distinct competitive advantage: the ability to reach customers around the world 24x7. Ensuring that all of the elements of a Web site are functioning properly is critical to maximizing the company's Web investment Therefore, system and network monitoring is an important and critical element of any Web presence, as a tool for reliability, planning, and analysis.

By "monitoring" we mean the automated process of testing, tracking and reporting on the availability and condition of the systems, services, and networks that make up a Web presence.

The main purpose of this activity is to ensure that site visitors are able to access their online applications, websites, and perform actions such as searching, online shopping, checking an account balance, or simply researching and reading. The aim is to avoid and minimize downtime and keep the server and online applications running.

1. Reasons of Web Server Monitoring

Because of the distributed nature of the Internet, failures can occur at many points, some of which are outside of a Web administrator's control. Being able to quickly ascertain which components are functioning and which have failed can dramatically reduce the time it takes to diagnose the problem. Once the Web site is back up and functioning, hard data can be used to inform customers and management of the causes of a given failure.

Monitoring can prevent problems by revealing patterns of resource usage and performance that might otherwise go undetected. For example, full disks typically cause a host of problems for applications and operating systems. Simply knowing that a disk is nearing capacity may save Web administrator hours of time fixing the problems caused by a full disk.

We can summarize and state that are several reasons why web server monitoring is necessary:

24/7 Availability: Web sites are expected to be available at all hours of the day, and even short periods of unavailability are often considered unacceptable. Proper monitoring can identify (and sometimes correct) problems and potential problems as effectively as possible, even without someone actively on duty.

Proliferation of Servers: These days, services are often implemented across "arrays" or "farms" of smaller, commodity servers, and those servers are often single purpose servers, used solely for running a single component of a Web site or service. Today, it is far less likely that a system administrator will notice a given failure of a system or service soon after it happens.

Proliferation of Services: The growth of the Web has meant that there are far more externally visible services than ever before. As customers and business partners come to rely on those services, proper monitoring becomes more and more important. The sheer number of different sites and services means that manual monitoring and problem identification are impractical.

Remote Locations: With today's extensive use of hosting and co-location services, system administrators need to rely on remote monitoring systems to keep track of their systems and services.

Remote Users: When all the users of a service are in the same location as the system administrators, it's often easy to monitor the availability of a server or service by listening to what the users are saying — many outages have come to the attention of the system administrators when users start asking each other if they are having problems. Remote users make this notification process much less effective.

2. Monitoring Strategy

A "monitor" is a test, typically a diagnostic command or emulation of actual use of the system. Its results are recorded so that they can be stored and/or acted upon. Monitoring systems or processes check services and components on a periodic basis, frequently enough to catch failures in a timely manner, but not so often as to significantly impact system resources. Monitoring systems typically consist of a set of monitors, mechanisms for alerting administrators if failures occur, and a historical log of data collected by the monitors.

Such system provides three types of information:

Exceptions - are those events or situations that indicate a problem or issue needs attention (network availability, service availability, server health, network health, security related information);

Trends - provide information on how usage and activity are changing with time, and are most often used to plan the timing and extent of upgrades and expansions (bandwidth utilization, server disk utilization and capacity, processor, memory, and i/o utilization on servers, activity counts);

History data - is used to track and report on outages, failures, and activity levels for such things as service level reporting, problem tracking, and resource or activity charging.

The focus of various monitor types covers a broad spectrum, from "deep" monitors that simulate user actions and that test many components of a Web site simultaneously, to "shallow" monitors that measure a single aspect of a single component.

Tests performed by deep monitors involve a large number of components, and often simulate a typical user transaction; results are measured in "user units," such as the number of seconds to complete a common task.

Deep monitors indicate whether a large set of components that provide a given service is functioning properly (in which case the performance data represents what a user would see) or if something is wrong with at least one of the components (although a precise diagnosis may be difficult). One common example of a deep monitor is a monitor that periodically attempts to retrieve a URL, recording any errors that may occur and the amount of time the retrieval took. This procedure can be performed manually, using a browser to retrieve the page, or automatically, using a monitoring system.

In contrast, shallow monitors test one or more aspects of a single component of the system.

Shallow monitors indicate precisely when a given component has problems, but often lack the context of real system use.

Measurements from these monitors are typically in the units of the component to be monitored; for example, bytes per second or CPU utilization percentage. The following are examples of shallow monitors:

Ping monitor: tests whether a machine is reachable over the network and whether the target machine is functioning well enough to send a simple reply

Domain Name Server (DNS) monitor: tests whether a machine's name can be mapped to a network address

Process monitor: ensures that a process is still active and using an acceptable amount of system resources

CPU monitor: measures the utilization of the CPU and flags chronic overloading

Memory monitor: measures memory use and paging activity

In practice, a combination of deep and shallow monitors gives the most effective and understandable picture of a Web site's current state, and allows the quickest diagnosis of problems.

There are several ways to implement a Web-monitoring strategy. Most simply, a person or group of people can manually check various aspects of the Web site on a periodic basis. The results of manual monitoring are rarely recorded in a form that permits historical and quantitative analysis. Information gathered by hand also does not generally get broadly disseminated, so only a few people know the status of the site.

Automated monitoring tools run these tests 24x7. These tools exist in a large number of forms: commercial products, shareware, freeware scripts and solutions. A number of commercial remote-monitoring services also exist. They all share the common abilities of periodically measuring some aspects of the system and recording this data. Most have a variety of shallow monitors, some also allow deeper levels of monitoring. By keeping historical records, trends that will lead to problems can be proactively managed.

3. Monitoring Process

Monitoring a Web Server means that the server owner always knows if one or all of his services go down. When monitoring a web server for potential problems, an external web monitoring service checks a number of parameters. The commonly used benchmarks or measured metrics are response time, availability, consistency and reliability. Depending on service, web site monitoring can check HTTP, HTTPS, FTP, SMTP, POP3, IMAP, DNS, Telnet, SSL, TCP and a range of other ports and protocols.

First of all, it monitors for a proper HTTP return code. By HTTP specifications RFC 2616, any web server returns several HTTP codes. Analysis of the HTTP codes is the fastest way to determine the current status of the monitored web server.

Table 1

Web Server HTTP Return Codes

Status code	Meaning
200	OK
201	Created
202	Accepted
204	No Content
301	Moved Permanently
302	Moved Temporarily
304	Not Modified
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable

The monitoring process is largely dictated by the monitoring software package in use, but there are certain elements that are likely to be found in most non-trivial monitoring systems.

Polling - monitoring systems are typically configured to “poll” every device, value, and service every few minutes to ensure availability, to identify errors or exceptions, and to collect data points to be logged. In any situation, it is necessary to have multiple “pollers” running in parallel, both for reasons of monitoring capacity and for the ability to deal with failures in the monitoring system itself. Pollers typically report their data to a central database and dispatch system.

Traps and alerts - monitoring systems typically provide “trap handlers” to catch and handle asynchronously generated events and traps, outside the normal polling process.

Hierarchy of monitored elements - the configuration and reporting components of sophisticated monitoring systems implement hierarchies and dependencies in the devices and elements being monitored. This is important so that a failure in network connectivity or a single device does not generate an error message for all the devices and elements behind the failed device, thereby obscuring the real fault.

Aggregation and de-duplication of data - monitoring systems will often have more than one probe or poller monitoring each device to guard against certain failures in the monitoring system or its connectivity. This can result in more than one copy of each data point being reported to the central monitoring server(s) — any such duplicates need to be eliminated before being recorded or dispatched in the central systems. As historical data is accumulated in the central database, there needs to be some mechanism for aggregating and summarizing data to retain the trends while lowering the total amount of data kept online.

Notification and reporting - Monitoring systems usually provide a number of standard notification mechanisms for use in the event of a failure or exception. These typically include sending messages to pagers, e-mail, fax, and some form of on-screen display, along with some API or other interface to allow the use of custom notification mechanisms. For trend and historical reporting, there should be graphical and other interfaces to query and summarize the data.

Web Server Monitoring may be internal, i.e. web server software checks its status and notifies the owner if some services go down, and external, i.e. some Web Server Monitoring companies check the services status with a certain frequency. External monitoring is much more reliable, as it keeps on working when the server completely goes down.

External monitoring refers to tests done by machines that are outside a site's internal network. Internal monitoring occurs on a Web server itself, or from a machine on the internal network.

The advantages of internal monitoring are that it:

- is closer to the source of addressable problems: internal monitoring keeps confounding factors such as ISP or backbone failures from obscuring site-specific problems. Because of the finer granularity of measurements possible, more precise problem diagnosis can be achieved. Administrators at the site can generally correct problems detected by internal monitoring because they control the machines and the networks.
- allows automatic corrective actions to be taken: internal monitors, especially those running on the servers they are monitoring, have the direct access required to take automatic actions. External monitors usually don't have the security access required to initiate these actions.
- is more reliable: internal monitoring doesn't depend on other networks to monitor and deliver data.
- is easier to administer: internal monitoring software and configuration are on directly accessible machines.

External monitoring has its own set of benefits. It:

- provides truer access-time measurements: access-time measurements taken remotely are a truer reflection of the end user's experience than measurements taken from internal machines.
- detects configuration errors that affect external users: configuration errors in Web servers, firewalls, proxy servers and routers may permit access from internal machines to internal sites, but may prevent legitimate external users from reaching a Web site.
- detects problems with ISP and backbone links: testing connectivity from sites out on the Internet can also help detect failures in ISP or backbone links that may be affecting users' ability to access the Web site.
- serves as a backup monitoring system: a catastrophic event could crash and/or disable all of the machines at a site. Without some monitoring from the outside, this failure would not be detected.

4. Managing Problems

It’s very important for a website, and hence the server where the the web site is hosted to experience minimal or no downtime. Website and server uptime is important to ensure there is no lost revenue or profit, beside to ensure that interested viewers can access the website without any downtime, network failure, system outage or connection failure so in the hope that they will return to visit the site in the future. Users or visitors most probably won’t return again if the site is always offline and inaccessible, and it will cause great annoyance and lose of trust.

In order to minimize server downtime or maintain highest possible website uptime and availability, it’s important that once there is any server outage, network disconnection or website downtime, the problem or the error have to be remedied immediately to bring back online the web server and website. In order to that, the persons responsible for the equipment or function in question should be notified. The faster the notification, the greater the chance that the problem can be fixed without affecting customers or users. When implementing a notification scheme, if the scheme can be disabled by the same problems that are being monitored, notification may not succeed.

Notification is important; automatically correcting problems is even better. A hung Web-server process can be automatically restarted, a nearly full disk can have temporary files automatically cleaned off of it, or a problematic machine can be automatically rebooted. Automatically fixing as many problems as possible ensures minimum down time, and reduces the need for human intervention.

Conclusion

Proper Web server monitoring is an essential element of any Web presence. Monitoring can protect that investment from the inevitable failures and performance problems that accompany the complex array of software, hardware and network connections that comprise a Web servers on the Internet.

Implementing and maintaining an effective monitoring system is an often complicated and expensive undertaking, most often best left to the professionals.

Regardless of the quality and effectiveness of the monitoring system that we rely on, without a prompt and professional response to any alerts or exceptions that get generated by the monitoring system, much of the effort and value expended will be for naught.

REFERENCES

1. Berthold, M. ,Brandner, R. (1993) - *Systems and network management in distributed environments*, Research Triangle Park: International Business Machines;

2. Nemeth E., Snyder G., Seebass S., Hein T.R. (1995) – *UNIX System Administration Handbook.*, Prentice Hall;

3. Thirukonda, M.M., Becker, S. A. (2002) – "WebSpy: An Architecture for Monitoring Web Server Availability In a Multi-Platform Environment", *Technical Report CS-2002-07*, Computer Science Department, Florida Institute of Technology;

4. Welter, P. (1999) – "Web server monitoring white paper", http://www.summitonline.com/apps-databases/papers/fhesh-man.html.