WebSpy: An architecture for monitoring web server availability in a multi-platform environment

Madhan Mohan Thirukonda

Shirley Ann Becker

WebSpy: An Architecture for Monitoring Web Server Availability in a Multi-Platform Environment

Madhan Mohan Thirukonda
Florida Institute of Technology
Melbourne, FL, USA
madhan@cosort.com

Shirley Ann Becker
Northern Arizona University
Flagstaff, AZ, USA
annie.becker@nau.edu

Abstract

For an electronic business (e-business), customer satisfaction can be the difference between long-term success and short-term failure. Customer satisfaction is highly impacted by Web server availability, as customers expect a Web site to be available twenty-four hours a day and seven days a week. Unfortunately, unscheduled Web server downtime is often beyond the control of the organization. What is needed is an effective means of identifying and recovering from Web server downtime in order to minimize the negative impact on the customer. An automated architecture, called WebSpy, has been developed to notify administration and to take immediate action when Web server downtime is detected. This paper describes the WebSpy architecture and differentiates it from other popular Web monitoring tools. The results of a case study are presented as a means of demonstrating WebSpy's effectiveness in monitoring Web server availability.

Keywords: Web Server, Web Server Management, Web Server Monitoring, Web Server Availability, Web Server Downtime, Electronic business, Electronic commerce, Enterprise Web Server.

The Need for an Effective Web Server Monitoring Tool

There is no question that the availability of a Web server plays a major role in the success of e-business. Ideally, a Web server should function as expected anytime, anywhere, and for any customer. The reality is that unscheduled Web server downtime happens, and often times it is due to factors beyond the organization's control (Schmidt, 2000). A disgruntled customer has a ripple effect on the use of a Web site in that many other customers are told about the negative experience. These customers, in turn, disseminate information to other customers causing long-term and perhaps irreparable damage to the organization. Proactive measures need to be taken to maintain a consistent, high-rate of Web server availability. An effective means of monitoring a Web server is needed such that downtime can be minimized.

An innovative Web monitoring architecture, called *WebSpy*, has been developed with this in mind. The automated tool, which supports this architecture, has special features including dynamic monitoring of a Web server, disseminating timely information, maintaining the status on availability, and unique reporting capability that provides the potential ongoing assessment of log data. It was built to satisfy the Web server monitoring requirements of a large e-business organization, which at that time couldn't be met by existing commercial tools.

The enterprise Web server of the organization, for which WebSpy was built, deploys server-side Java technology using third party application servers to serve its Web users. Many enterprise Web servers are increasingly using application servers in order to take advantage of dynamic applications that run on the server side of an e-business. An application server interacts with the database server to get the business data required by the end users. Due to the complexity of such an architecture, it is very important to ensure high availability of the Web server.

It is proposed in this paper that WebSpy be used to effectively minimize the unscheduled downtime of an enterprise Web server. Section 2 presents an overview of World Wide Web (Web) concepts. Section 3 describes the architecture of WebSpy tool from the point of view of an administrator. Section 4 describes the functionality of the tool and the protocol used to implement the architecture.

Editor: Mehdi Owrang

Table 1: Factors Affecting Web Server Availability	
Factor:	Explanation:
Server software failure	The inability of the Web server software to serve a request possibly due to its bad run time environment (e.g., memory leak).
Support server software failure	Supporting software (e.g., application or database server software) fails such that the Web server cannot respond to a HTTP request.
External factors	External factors (e.g., hacking attempt) impact the availability of the Web server.

Section 5 summarizes a case study used to evaluate the capabilities of the tool in meeting the requirements specified by the e-business. The paper concludes with future research directions.

World Wide Web (Web) Concepts

The Internet is a network infrastructure built on certain standards that provide the means for users to communicate with one another. The Web is one of the services offered by the Internet, and its architectural framework allows users to access geographically dispersed, linked documents (Tanenbaum, 1996). The Web also provides the capability for a running a multitude of applications using three simple mechanisms (Ciancarini, *et al.*, 1996):

1. URLs (Universal Resource Locators) provide information on the location of a document and the protocol needed to access it.
2. A client-server protocol, called Hyper Text Transfer Protocol (HTTP), provides the means to fetch a document from a Web server.
3. A simple markup language, Hyper Text Markup Language (HTML) allows content authors to describe the information and store them in a document. A HTML document is also referred to as a Web page.

A main component of the Web is a Web browser, which is a program that interprets and displays a Web page using HTML specifications. The most popular Web browsers are Netscape Communicator and Internet Explorer though there are others. The other main component is the Web server, which is a program that delivers Web pages upon a HTTP request from a Web browser. Fielding *et al.* (1999) define the HTTP protocol as a request/response protocol. A Web browser requests a Web page by sending the HTTP request message that contains the URL of the Web page to the Web server. The Web server responds with a HTTP response message followed by the Web page. E-business enterprises often use application servers to complement the Web server.

In general, availability is defined as the percentage of time during which a system is available for a given purpose (CAIDA Metrics Working Group, 2001). We define Web server availability as the percentage of a time interval during which the Web server is available to serve HTTP requests. Web servers may fail to serve HTTP requests due to a list of possible factors, as presented in Table 1. On an immediate basis, the e-business is focused on getting a Web server that is unavailable up and running as quickly as possible. In the long run, it has a goal of maintaining a consistently high rate of Web server availability.

Ideally, an e-business would want to establish an availability rate for each of its Web servers, and then gather statistical information about whether this rate is achieved. It would also want to track fluctuations in the availability rate so it can investigate the causes of availability problems. Thus, meaningful information needs to be captured at the time of failure in terms of down time, possible cause of failure, and recovery mechanisms. To support these goals, a process is needed to periodically check the availability of the Web server and take necessary action in the event of a problem. The process would alert a system administrator as soon as a problem is detected. In some cases, a corrective action is automatically taken as soon as a problem is detected, thus preventing human intervention. Such automatic recovery guarantees quick restoration of Web server availability.

Table 2: Basic Features of Commercial Web Server Monitoring Tools

Feature:	Description:
Periodic monitoring	The Web server is monitored on a schedule based on its availability requirements. Basic timer settings include “sleeping time,” the interval between two monitoring cycles, and “slow response timeout interval,” the threshold time within which a response should be obtained from the Web server.
Alerts	An alert mechanism notifies individuals when the Web server is unavailable. An email note, for example, is sent to a designated administrator.
Log entry	A log entry is made when the Web server is unavailable in order to support the calculation of availability statistics.

Table 3. Additional Requirements of the E-business

Requirement:	Description:	Commercial Support:
Multiple Operating System Platforms	The monitoring service runs on multiple platforms to support production Web servers running on multiple platforms.	None
Start as NT Service	The monitoring tool is automatically started as a service in the Windows NT environment	Limited
Additional Timer Settings	The monitoring service starts monitoring only after the Web server has been started. An initial timing delay is needed to allow for the Web server to start up. This “initial sleep time” eliminates false alarms associated with start up when the Web server host is rebooted. The time between retry attempts (referred to as “retry time”) is specified by the administrator.	Limited
Web Interface for Viewing Log Data	A Web interface provides viewing capability of the log information for all Web servers.	Very Limited
Recovery Scripts	The monitoring tool is capable of executing a recovery script when the Web server becomes unavailable. The script has commands to restart the Web server, the database server, or the Web server host depending on the problem. The administrator configures certain commands in the recovery script that execute when a specific error occurs (the error message is displayed as part of the Web page log information).	Very Limited
Secondary Email Alerts	If the Web server is unavailable for a specified time, then a secondary email list is used for notification purposes. This provides the capability of alerting officials and managers of prolonged problems in Web server availability.	None
Availability Metrics	Web server recovery timestamps provide useful information in calculating availability metrics.	None

The WebSpy Architecture

Table 2 identifies the basic features of commercial Web server monitoring solutions that were assessed before the onset of this research effort. These solutions are the same ones cited in the Stanford Linear Accelerator Center (Cottrell, 2001), and include *Holistix Web Manager* by Holistix, Inc., *Argent Sentinel* by Argent Software, Inc., and *AI Monitor* by AI Tech. Though the basic requirements listed in Table 2 were met by all the commercial tools included in this study, none met the more complex monitoring requirements specified in Table 3. These additional monitoring requirements, typical of many of today's e-businesses, were the motivation for the development of WebSpy.

Figure 1 shows the WebSpy architecture comprised of two parts; the WebSpy station where both the Web server and WebSpy are installed, and the workstation where administrators receive alerts and view availability reports. WebSpy has two major components, the WebSpy client (hereafter referred to as client) and the WebSpy server (hereafter referred to as server). Each of these components is explained.

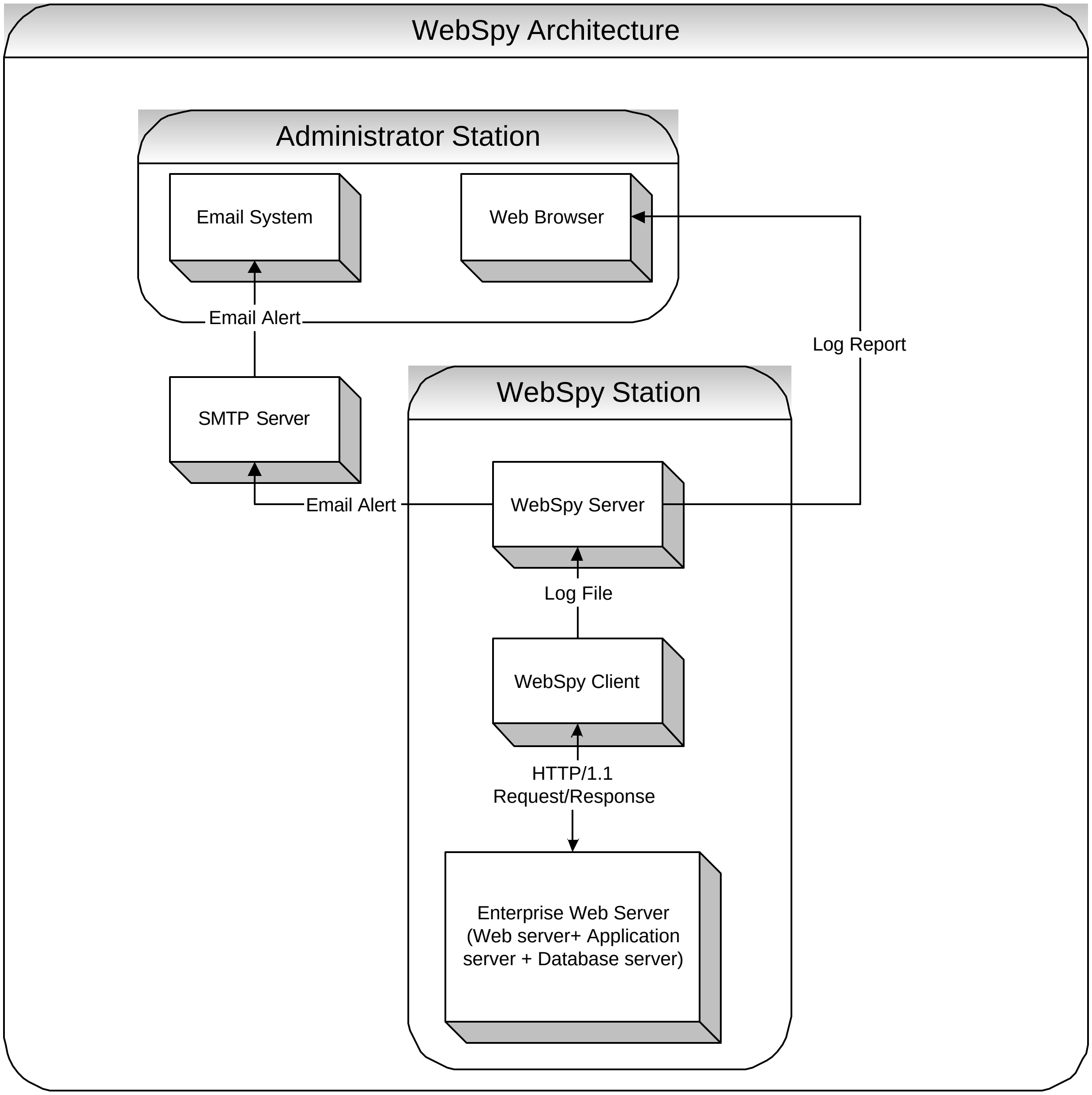


Figure 1. Enterprise Web Server with Monitoring Ability

Table 4: Settings for Web Monitoring

Properties:	Description:
URL	URL to be monitored (e.g., *http://cs.abc.edu/profiles.html* has components; *HTTP* protocol, host name *cs.abc.edu*, path for the resource *profiles.html*). One URL should be specified for each Web server installed on a Web server host.
Enable	The monitoring of a Web server is enabled using this property.
Init_time	An initial time is used to put the client to sleep in order to synchronize the startup of the Web server and the client.
Sleep_time	A sleep time is used as a time interval in which the Web server is not monitored. It needs to be reasonably short in order to detect availability problems in a timely manner. It also has to be reasonably long to minimize the impact on resources and overall server response time.
Slow_time	A slow time is used to specify a maximum time period for receiving a server response.
Retries	The client checks for Web server availability a specified number of times, after which a failure is declared.
Retry_Interval	The time in seconds between two clients retry attempts in determining server availability.
Text_failed	The administrator can customize text that becomes part of the email alert.
Script_on_failure	A recovery script that is automatically executed when a server is unavailable. The script may have several commands based on the command line arguments passed to it. WebSpy executes the script by passing the command line arguments depending on specific error message that appear on the Web page.
Email	Alert messages are sent to the comma-separated list of email addresses.
Log_transfer_interval	The log transfer interval specifies when the client sends the log file to the server.
Dynamic_Errors	A list of error messages of which one or more may appear in the Web page if there is a problem in dynamic Web page generation. Based on a particular error message, an appropriate command in the recovery script would be executed.
Escalation_Time	The time in minutes after which any problem in availability should be escalated.
Secondary_Email	Alert messages are sent to this comma-separated list of email addresses, after Escalation_Time. Email addresses of managers are typical for this category.

Client

The client is a program that periodically monitors the Web server availability based on settings specified by the administrator. The client is installed on the same computer as the Web server. Table 4 lists the settings that are the basis for monitoring. WebSpy allows these properties to be dynamically updated for effective monitoring. The client can concurrently monitor several Web servers installed on the same or different Web server hosts. For each Web server, a list of settings is specified.

The client assesses the availability of a Web server by monitoring whether the Web server is successful in serving a HTTP request. The HTTP request is based on a URL specified by the administrator. The Web page specified by

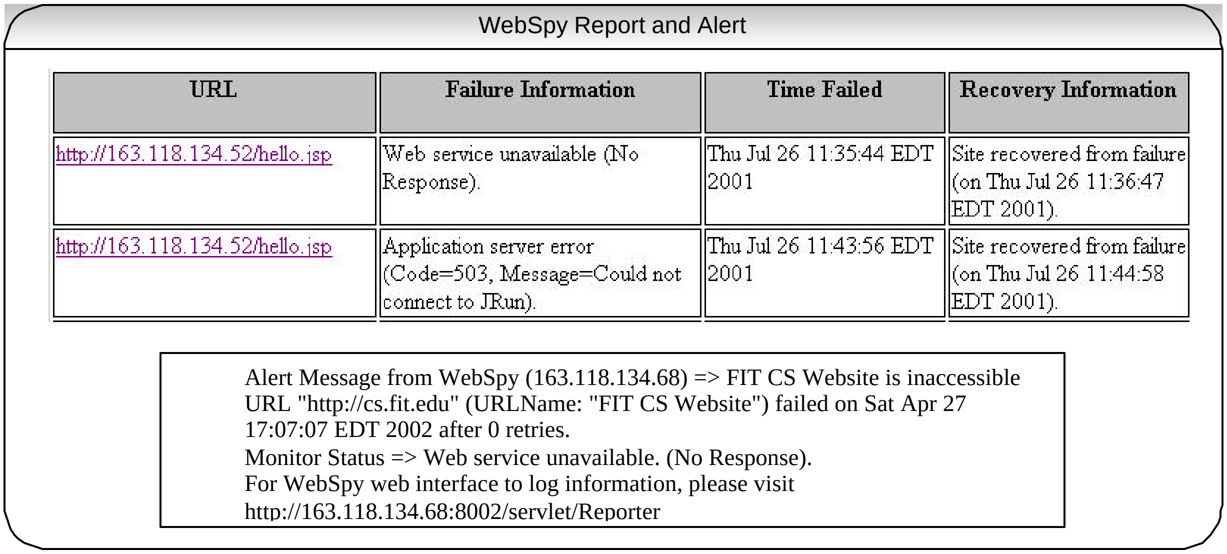
Table 5: Collective log file information

Parameter:	Description:
EventID	Each WebSpy event has a unique number associated with it. (WebSpy events are discussed in detail in Section 3.3)
URL	URL being monitored
Response	The HTTP response header information provided by the Web server
Fail Time	Local time when a problem in availability was detected
Recovery Status	Status of the recovery as executed by the client
Email Identifier	List of registered email addresses for receiving email alerts
Mail Status	Status of an email alert
Mail Content	Content of an email alert
Alert Message	Alert message formatted to be displayed in a Web browser
Host	IP address of the client
Server Name	Name of Web server software (includes version information)
Recovery Time	Local time of recovery

the URL may be static or dynamic, though production server Web pages are generally dynamic since the data is highly volatile and constantly changing. It is also important that the URL be valid and virtually free from programming errors.

A log file is used to store information generated by the client while monitoring the server. This information includes availability status, email alerts, and recovery status, among others, as shown in Table 5. This information is useful for calculating availability statistics for each Web server that is being monitored.

In addition to monitoring a Web server, the client is responsible for executing a script for restoring the availability of a Web service. The recovery script is executed with a



WebSpy Report and Alert

URL	Failure Information	Time Failed	Recovery Information
http://163.118.134.52/hello.jsp	Web service unavailable (No Response).	Thu Jul 26 11:35:44 EDT 2001	Site recovered from failure (on Thu Jul 26 11:36:47 EDT 2001).
http://163.118.134.52/hello.jsp	Application server error (Code=503, Message=Could not connect to JRun).	Thu Jul 26 11:43:56 EDT 2001	Site recovered from failure (on Thu Jul 26 11:44:58 EDT 2001).

Alert Message from WebSpy (163.118.134.68) => FIT CS Website is inaccessible
URL "http://cs.fit.edu" (URLName: "FIT CS Website") failed on Sat Apr 27 17:07:07 EDT 2002 after 0 retries.
Monitor Status => Web service unavailable. (No Response).
For WebSpy web interface to log information, please visit
http://163.118.134.68:8002/servlet/Reporter

Figure 2. WebSpy Report and Alert

command line argument based on the error message on the Web page. The recovery script should be configured to restart appropriate servers based on the command line argument. The output of the script is updated to the log file.

Server

The architecture of WebSpy supports multiple instances of the client in order to monitor different Web server hosts. Only one server is needed to manage these client instances, and it can reside on any of the Web server hosts or

Table 6. WebSpy Events

1) Failure After Retry Attempts - Event fired after the number of retry attempts is equal to the "retries" parameter value and the Web server is still unavailable.

Attribute	**Value**
EventID	1
Response	The HTTP response header information provided by the Web server, if received.
Server_Name	Web server name, if received.

2) Email Notification – Subsequent event that is fired if any email address is registered to receive email alerts.

Attribute	**Value**
EventID	2
Email_ID	Email address registered with the client.
Mail_Status	Null (The status of email alert is stored after its delivery).
Mail_Content	Email alert message.
Alert_Message	Email alert message formatted to be displayed in a Web browser.

3) Execution of Recovery Command – Event that is fired after executing the recovery script.

Attribute	**Value**
EventID	3
Recovery_Status	The output and the return code of the command execution.

4) Web Server Recovery – Event that is fired when the Web server is recovered after a failure has occurred.

Attribute	**Value**
EventID	4
Recovery_Time	Local time of the client when the Web server is recovered.

5) Email Notification: Escalated – Event that is fired when the Web server is still unavailable after Escalation_Time.

Attribute	**Value**
EventID	5
Email_ID	Email address registered with the client.
Mail_Status	Null (The status of email alert is stored after its delivery).
Mail_Content	Email alert message.
Alert_Message	Email alert message formatted to be displayed in a Web browser.

on a separate workstation.

The clients periodically communicate with the server to send it the log information. The server collects the log information of each client instance and stores it in a cumulative log file. The server uses an internal Java servlet engine to periodically generate a Web report from the monitoring results stored in the cumulative log file. The report, shown in Figure 2, is listed as a URL in the email alert sent to the administrator. The server requires access to a Simple Mail Transfer Protocol (SMTP) server to forward email alerts.

Events

During the monitoring process, one of four events could occur that require an action taken by the client. For each event that occurs, the information describing it is written to the log file. Table 6 lists each event and the information stored for it.

WebSpy Functionality

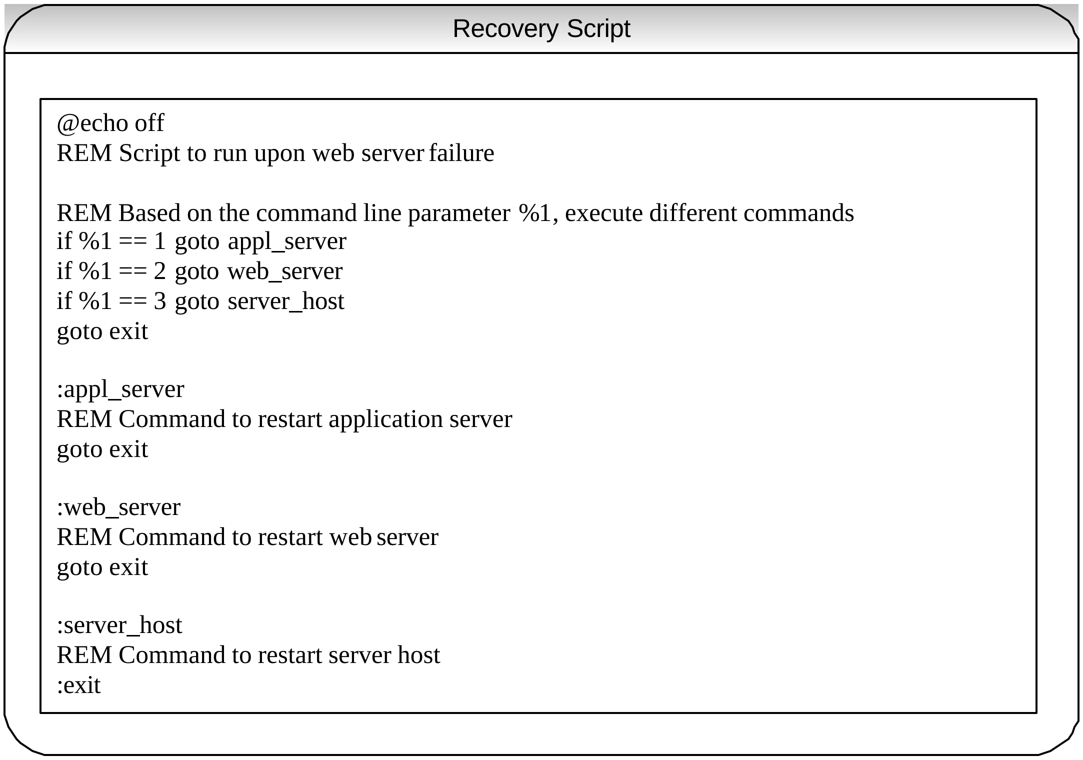
The main functionality of the client is to monitor the Web server and take appropriate actions when the Web server is not available. The main functionality of the server is to act as a coordinator in collecting the monitoring reports from each client, sending email alerts, and making the cumulative report available online.

Client functionality

Once activated by an administrator, the client waits the specified *Init_Time* time period, and then sends the Web server a HTTP request using the specified URL. The client waits for a HTTP response from the Web server. If a response header is not obtained within the specified *Sleep_Time* time period, then the client repeatedly sends the HTTP request until the number of retries is equal to the *Retries* parameter value. Each retry attempt occurs after the specified *Retry_Interval* time period has lapsed. If no server response is obtained during or after the specified *Retries* attempts, the Web server is considered not available and the log file is updated.

When a HTTP response is received, the client extracts the response code and message from the header. The *2xx* response code represents a successful transaction, which is an indication that the Web server is available. The Web server is considered unavailable if there is no response or if a *5xx* response code is received. *5xx* response code means that the Web server was unable to serve the HTTP request because of a server-side problem.

To illustrate this, we use an example from our Web monitoring experiences. A recurring problem encountered during the analysis stage of the selected architecture was denoted by a *503* error code. This is defined in the HTTP specification as a *Service Unavailable* problem (Fielding *et al.* 1999). This problem may occur when the Web server



```
@echo off
REM Script to run upon web server failure

REM Based on the command line parameter %1, execute different commands
if %1 == 1 goto appl_server
if %1 == 2 goto web_server
if %1 == 3 goto server_host
goto exit

:appl_server
REM Command to restart application server
goto exit

:web_server
REM Command to restart web server
goto exit

:server_host
REM Command to restart server host
:exit
```

Figure 3. Recovery Script

is unable to receive services of another external application. For our Web architecture, JRun 3.1 was the application server. The client received an error code *503 JRun closed connection* when the JRun application server was not purposely stopped.

The final determination of Web server unavailability occurs only after executing the specified retry attempts. In the case of a *5xx* response from the Web server, the response message is compared with the list of error messages called *Dynamic_Errors*. In this list, the administrator would include messages that he suspects appear on the response message due to a problem during dynamic page generation (in our example, an error message in the e-business Web server was 'JRun closed connection').

When a match is found in the list, the failure is reported as an *Application server error* with the response code and message, and then the client executes the recovery script. The command line argument for the recovery is calculated based on the position of the error message in the list. The recovery script is written to execute the appropriate commands based on the command line argument as shown in Figure 3.

When there is no matching code or no response from the server, it is reported in the log as *Web service unavailable.* WebSpy events, presented in Table 6, are fired and the log file is updated. The client continuously monitors the URL until the Web server becomes available. The log files are frequently sent to the server based on the parameter *Log_transfer_interval* (as presented in Table 4).

Server functionality

The server listens to a specified TCP/IP port for client connection. It receives periodic connections from the clients and gets the log information. On receiving the log file from the client, the server looks for the *Email notification event* record (event 2 in Table 6). If it is found, then an email is sent based on the information and the email alert status is stored in the cumulative log file. Other event information is also stored in the cumulative log file, which is later used to produce the cumulative report. The server has a Java servlet engine to dynamically generate a report based on the information in the log file. The Java servlet that is used to generate the report can be invoked by using the URL, which is contained in the alert sent to the administrator.

Performance impact

The HTTP transactions, due to WebSpy monitoring, have an impact on Web server performance because they simulate user requests. Reducing the number of transactions assessing availability minimizes the negative impact on performance. The parameter *Sleep_Time* defines the time during which the client would wait to send a HTTP request. *Sleep_Time*, as noted in Table 4, should be reasonably set to detect availability problems as quickly as possible without causing a noticeable performance impact on the Web server.

A worst-case analysis is described in terms of the performance impact of WebSpy monitoring the Web server. If the administrator enters sixty seconds (minimum allowed limit) as the value for the parameter *Sleep_Time*, then the total time required for a monitoring cycle is calculated as follows:

$$T_{mt} = T_{et} + T_{st}$$

Where: T_{mt} is the total time for the monitoring cycle, T_{et} is the execution time of a HTTP transaction for a URL request, and T_{st} is the sleep time required for the monitoring cycle.

For illustration purposes, let's assume that T_{et} takes a minimal amount of time. Thus, $T_{mt} \sim T_{st.}$ This means that there is approximately one HTTP request per minute. For a twenty-four hour time period, there are approximately 24 * 60 = 1440 HTTP requests from the WebSpy monitoring tool. In most cases, this is an acceptable number of HTTP requests because the number of transactions can be handled by the Web server without a noticeable impact on performance.

Case Study

A case study was conducted to determine the effectiveness of WebSpy in monitoring the availability of a Web server given the e-business requirements. The Web server architecture was comprised of Internet Information Server (IIS) as the Web server and JRun as the application server.

The initial sleeping time *Init_Time* was set to 20 seconds, the interval between two monitoring cycles *Sleep_Time* was set to 60 seconds, and slow response time out interval *Slow_Time* was set to a maximum of 45 seconds. These parameter settings are typical of the e-business' real-world setting for minimizing false alarms (Web server is available but reported have failed) and maximizing availability by reporting problems quickly after they occur (Loren Buhle, 1996). The client was set to transfer the log file to the server periodically with an interval of 30 seconds. A recovery script was written such that it would restart the Web server for a '1' command line argument, restart the application server for a '2' command line argument, and restart the Web server host for a '3' command line argument.

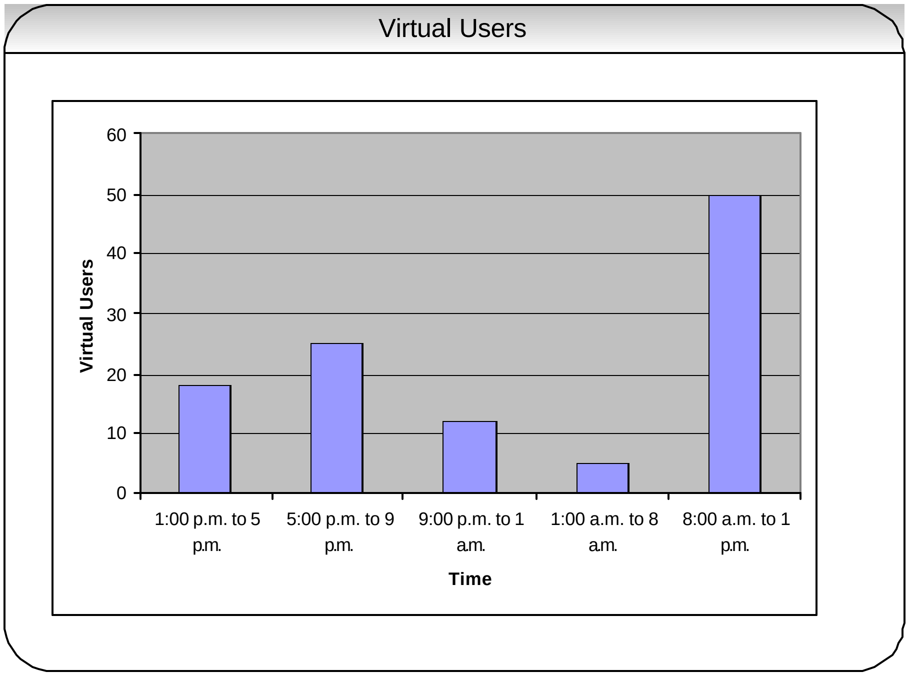


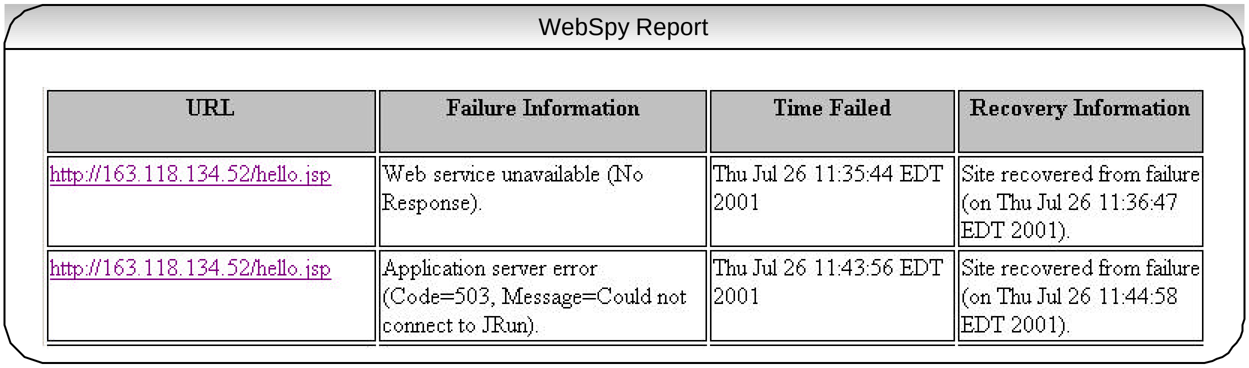
Figure 4. Simulation of Users Requesting HTTP Services

A program was written to simulate users requesting HTTP service of the Web server. The Web server had twenty URLs, ten of which were static HTML pages ranging from 17 Kbytes to 234 Kbytes in size. The other ten URLs were dynamically produced resulting in a HTML page ranging from 1Kbyte to 350 Kbytes in size. Simulated (virtual) users were set to randomly download any one of these 20 Web pages (every random time between 1 and 5 minutes to simulate reading).

The maximum number of user requests (clients) for the Web server was restricted to ten so that the maximum limit was reached thus ensuring the Web server was busy. Figure 4 shows the number of users per time interval in the simulated, twenty-four hour time period. The number of users peaked at fifty during the morning hours, as is generally the case for many e-businesses. To simulate the Web server's inability to serve a HTTP request, we stopped the application server and the Web server once for a several minutes.

WebSpy behavior

WebSpy detected the availability problem that occurred during the case study and restored the Web service as expected. As soon as a problem in Web server availability was detected, WebSpy sent an email to the specified administrators. The *Web Interface* component of the WebSpy displayed detailed information including failure detection time, Web server name, status of the restoration command executed, recovery time, and other pertinent information. For each failure, WebSpy detected it and restored Web



WebSpy Report

URL	Failure Information	Time Failed	Recovery Information
http://163.118.134.52/hello.jsp	Web service unavailable (No Response).	Thu Jul 26 11:35:44 EDT 2001	Site recovered from failure (on Thu Jul 26 11:36:47 EDT 2001).
http://163.118.134.52/hello.jsp	Application server error (Code=503, Message=Could not connect to JRun).	Thu Jul 26 11:43:56 EDT 2001	Site recovered from failure (on Thu Jul 26 11:44:58 EDT 2001).

Figure 5. WebSpy Report

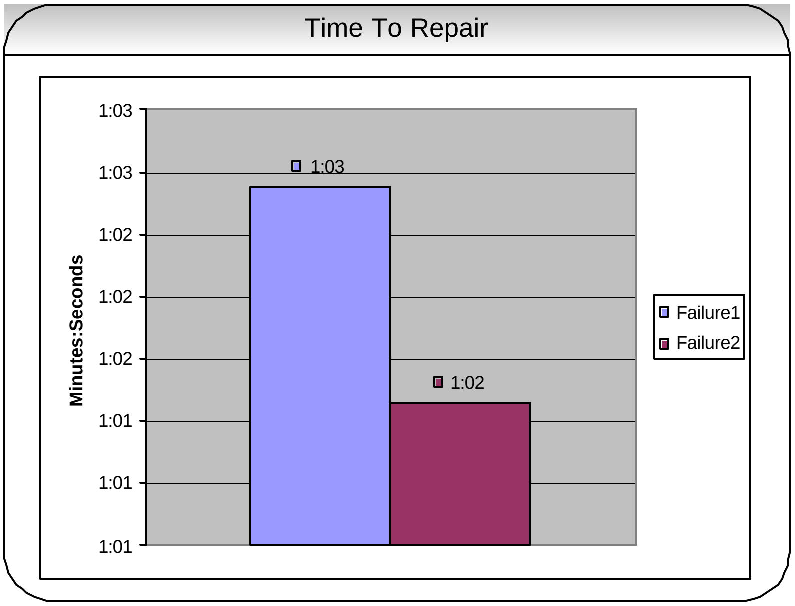


Figure 6. Time To Repair

service immediately as evidenced in Figure 5. The *Web Interface* provides the information obtained from the log file. Each hyperlink in the *Web Interface* is linked to another Web page that provides detailed information on the output of the recovery script executed, status of the email alert, and a copy of the email alert message.

Availability metrics

There are two metrics that provide useful information in terms of Web server availability. The *Mean Time to Repair* (MTTR) provides the average time taken to restore normal service after a loss of availability. This metric provides ongoing feedback as to the downtime associated with a server, and its potential impact on customer satisfaction

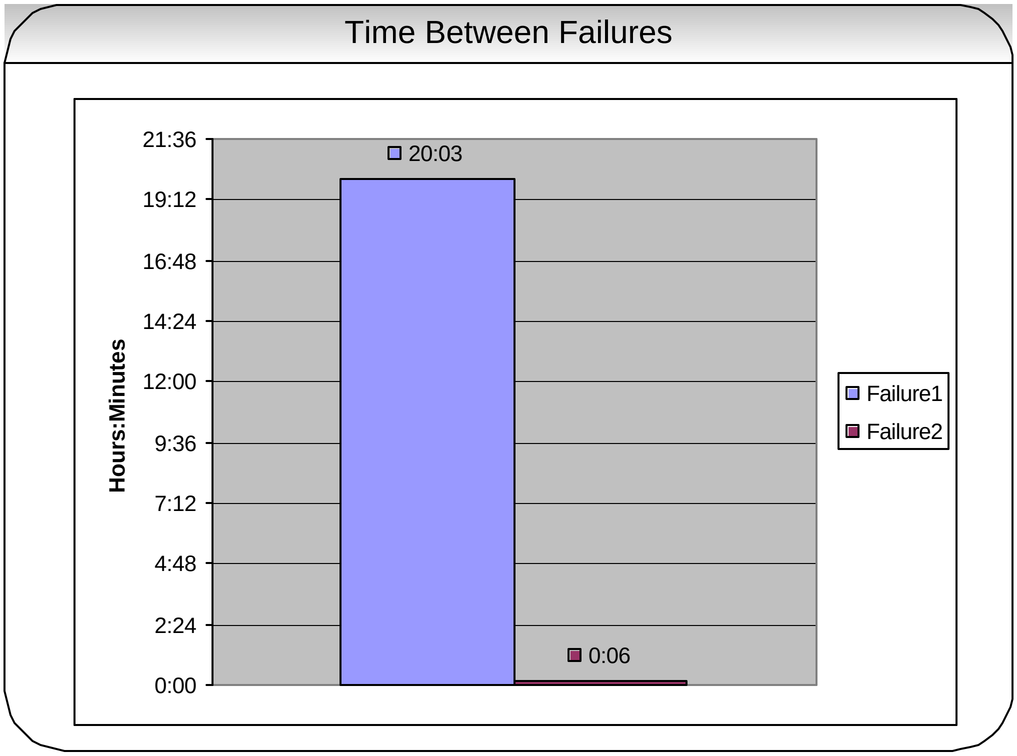


Figure 7. Time Between Failures

and retention. The other important metric is *Mean Time Between Failures* (MTBF), which provides the average time between the beginning of normal service and the next loss of availability. This metric also provides insight into the long-term impact of server failures on an organization's customer base. The availability metrics are useful from the administrator point of view in terms of finding the stability of the Web server and the application server. Figures 6 and 7 show the *Time To Repair* statistics and the *Time Between Failures* statistics for the Web server.

The *MTTR* (~1 second) maintained a low value meaning that the Web service was restored very quickly. The MTBF (~10 hours) was quite high, which means that the Web server was stable and not failing at a high rate. For the Web server architecture, used in this case study, we can define availability in terms of these two metrics (Blanchard, 1997). We start defining uptime (MTBF) and downtime (MTTR):

Uptime = S Time Between Failures

Downtime = S Time To Repair

The uptime for the Web server was 86,275 seconds and downtime was 125 seconds. Availability is calculated as:

Availability = Uptime / (Uptime+Downtime).

Where: Availability = 86275 / 86400 = 99.85%

What we can conclude from this data is that the Web server was maintaining a high rate of availability during the case study, as recovery scripts were executed quickly after a problem was detected. The Web server was automatically recovered quickly, which means that unscheduled downtime was kept to a minimum.

Overall Findings and Future Research

The WebSpy provided several features, as described in Table 7, that were not fully supported by other commercial tools. The email alert capability is an important contribution, as it provides additional information in a Web-based environment for tracking Web server failures. It also runs in multi-platform environments, unlike any of the other commercial tools. The WebSpy has an automatic recovery capability, which is necessary for fast recovery from certain types of failures.

One of the important contributions of this research is WebSpy's Web-based report generation capability in terms of monitoring performance issues in a real-time mode. Though not shown in this paper, the log data is readily imported into a relational database. Thus, the log data provides an opportunity to be mined for trends in performance, recovery, availability, and security. These can be computed over a long period of time in order to optimize the tool's settings.

In our case study, we set the Sleep_Time to 60 seconds to minimize the impact of this timing constraint on performance. However, the log data in a relational form can be queried for performance trends when the Sleep_Time is set equal to, greater, or less than 60 seconds. Thus, an organi-

Table 7. WebSpy Features

Features	Description	Commercial Tool
Multiple Platform Support	Webspy is built in Java, which virtually runs on all operating systems.	None
Web Interface to Log Information	A single report presents information about the availability of the Web servers, which is viewed via a browser.	Very Limited Support
Customizable Automatic Recovery	Recovery scripts are automatically executed thus minimizing human intervention and time delays. Recovery scripts can be customized for different possible problems regarding Web server availability.	Very Limited Support
Secondary Email Alerts	Email alerts are sent to designated personnel when there is prolonged unavailability. This is in addition to the primary email list that is notified as soon as a problem is detected.	None
Availability Metrics	Web server recovery timestamps provide useful information in calculating availability metrics.	None

zation would be able to identify the optimal setting to meet its performance needs. In terms of security, there is a potential that the tool can be misused to create DOS (Denial of Service) attacks (e.g., when Sleep_Time is set to a low number). The data in the logs provides an opportunity to identify security breaches such that the tools is actually being used to hack into another system.

Future research would identify additional data that could be gathered in order to identify long-term trends in Web monitoring. It would be very useful, for example, to maintain statistics on Web page load times, average number of retries attempted for Web page access, and the average throughput given environmental constraints. This information would prove useful in not only monitoring but also in predicting availability problems. In this way, downtime would be minimized as certain types of problems could be prevented.

WebSpy offers features that none of the other tools completely supported, which makes it very useful for e-businesses working in the global, online marketplace. On a final note, the e-business that funded this research has been using WebSpy successfully to monitor various Web servers in a geographically dispersed environment. This is a reflection of the capability provided by WebSpy in meeting the needs of a global e-business enterprise.

References

Blanchard, B. S. (1997). *System Engineering Management*, 2nd Edition, Wiley-Interscience, http://www3.interscience.wiley.com/, pp.120-128.

Buhle, L. (1996). *Webmaster's Professional Reference*, New Riders Publishing, Indianapolis, IN, pp. 468-497.

CAIDA (Cooperative Association for Internet Data Analysis) Metrics Working Group (2001). "Network Measurement FAQ," Super Computer Center, University of California, San Diego, CA, http://www.caida.org/outreach/metricswg

Ciancarini, P., Tolksdorf, R., and Vitali, F. (1996). "Weaving the Web in a PageSpace Using Coordination," Technical Report, January, ftp://ftp.cs.unibo.it/pub/cianca/pagespace.ps.gz .

Cottrell, L. (2001). "Network Monitoring Tools," Stanford Linear Accelerator Center, Stanford University, CA, http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html.

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T. (1999). *Hypertext Transfer Protocol – HTTP/1.1*. The Internet Engineering Task Force, http://www.ietf.org/rfc/rfc2616.txt .

Schmidt, J. (2000), "To Be Up or Not To Be Up - Analysis of Web Server Downtimes", *Magazine for Computer Technology*, August, Eva Wolfram, English translation, www.heise.se/ct/english/00/08/174, January 2, 2002.

Tanenbaum, A. S. (1996). *Computer Networks*, 3rd Edition, Prentice Hall, Saddle River, NJ.

Biographies

Mr. Madhan Mohan Thirukonda is a software engineer of CoSORT with his master's degree in computer science from Florida Insitute of Technology, Melbourne, Florida. He earned his bachelor's degree in computer science and engineering from Thiagarajar College of Engineering, Madurai, India. His research interests are in distributed computing, e-commerce, and high-performance data processing in open systems. He is a member of ACM and IEEE computer society.

Dr. Shirley Ann Becker is a computer information systems professor at Northern Arizona University. Dr. Becker's research interests include Web accessibility and literacy, global Web usability, e-commerce, and data quality in legacy systems. She has published over 50 articles in these and related areas, and has received research grants from IBM, Texas Instruments, NASA, and NSF. Dr. Becker has an MS and PhD in information systems from the University of Maryland, College Park. She is a member of IEEE, the ACM, and the Information Resource Management Association.