

APACHE WEB SERVER MONITORING

Saurabh Phaltane¹, Omkar Nimbalkar², Piyush Sonavle³, Prof. S.R.Vij⁴

^{1, 2, 3, 4} Department of computer,

Maharashtra Institute of Technology, Pune
Kothrud, Pune, Maharashtra, India

¹saaurabh.phaltane@gmail.com

²nimbalkaromkar@yahoo.com

³piyushssonavale@gmail.com

⁴sheetal.vij@mitpune.edu.in

Abstract— ‘Apache Web Server Monitoring’ offers webserver performance monitoring that allows administrator to track the overall functioning/performance of their server. The server monitoring software monitors the performance, health of the servers and the information can be obtained about its health from present and historic performance. It also simulates information about the visitors across multiple geographies, user agents used, top referrals to their websites/servers over Internet connections. It provides information about the number of visitors to a website and the number of page views. The uptime, Downtime, number of active processes, CPU utilization etc. are critical things in deciding the health of the server. The Error Log Information analysis and response time calculation help in predicting the probable cause of downtimes and debugging and tracking the cause of errors.

Keywords— logging, analysis, parsing, webserver performance monitoring, filtering

1. INTRODUCTION

‘Webserver monitoring’ allows businesses to simulate the overall functioning of their server. Central idea of project is log centralization and its analysis for relevant metrics deductions that can predict and analyse the performance of the server. Apache webserver monitoring monitors the performance of the servers and the information can be obtained about its health over the period of time. This tool mainly deals with the error log analysis along with the uptime, Downtime, number of active processes, CPU utilization etc. These all are critical things in deciding the health of the server.

This tool also simulates visitors across multiple geographies and servers Internet connections. It also gives the top browsers for particular server even top referrals along with IP address of the visitor. In short, web server monitoring is the collection and measurement of Internet data.

2. WEB SERVER MONITORING

This paper mainly deals with CPU load analysis, analysing error logs, analysing access logs and finding top browsers, top referral. The Error analysis serves an important tool to any to any server administrator to track the server error and hunt for the reasons for its low/degraded performance. The analysis of logs over a period of time and on a specific time slot can server as an important metrics to the server administrator. The analysis of score of warnings, severe error and mild errors can guide the administrator to take directional steps to rectify the issue. The proper planned utilization of server capacity can be availed with the judicious use of server error log analysis.

The metrics of Top browsers utilization can serve as a tool to server administrator to plan his extensions to his projects and also make alternative services available as per the requirements. The analysis of Downtime serve as an important tool to the server administrator the track the server uptime of his server also could monitor the server issues leading to downtime by the analysis of Error log over the period of time. The geographical access to the server can’t be monitored by tracking the IP that requests the server and determine their global location using the time stamps.

3. LOGFILE ANALYSIS

Logs are essentially the most important source of information about the server statistics and performance .The Apache server generate logs namely access logs and error logs ,which serve as a source of information that analysed over a period can predict the patterns of data usage, server performance, bandwidth utilization, peak utilization, varying loads over server.

For Logfile analysis you have to store and archive your own data, which often grows very large quickly. Although the cost of hardware to do this is minimal, the overhead for an IT department can be considerable.

In our approach we import logs from client’s server on our server and then analyse this logs. Initially few required modifications are made on these logs. This data is enriched

so that it can be more appropriate for further analysing the health of the server.

4. TYPES OF LOGGING

There are many types of logging but as per our need for analysing we select piped logging as it is more suitable for us. Given below are few loggings techniques:

A. Flat files: -

Is the simplest way to record hits, but they're impractical for long-term analysis. Its major drawback is that processing programs must read the records sequentially. Log writing affects web server performance due to the high number of disk I/O operations. Log rotation must occur regularly.

B. Database Logging:-

This technique is better than flat files but not suited for our purpose. `mod_log_mysql` can write logs directly to a networked database server. This provides considerable savings on local disk I/O operations. Also, one central database server can handle multiple Apache servers.

We can use SQL commands to extract only the fields and records that you want

Cons of Database Logging:-

The major drawbacks of this technique are that `mod_log_mysql` requires a rebuild of Apache and this module works only with a MySQL database. The database isn't anywhere close to fourth normal form. Database could fail or become unreachable.

C. Piped Logging:-

We will mainly concentrate on this technique and will be using same due to its superiority. These programs import the log files into a database and generate reports from the imported data. Report generation is usually much quick. Sometimes, it is advisable to delegate all the logging to specifically developed parsing engines or archiving utilities. When Apache is started, it runs the logging program and sends all the logging messages to it. This solution is valid in many situations.

For example:-

When you don't want to stop and restart your Apache server to compress your logs.

When we have many virtual hosts, if you use a different log file for each virtual host, Apache will need to open two file descriptors for every virtual domain, wasting some of the kernel's and processes' resources.

When you want to centralize your logging into one single host, the program specified in the configuration could send the log lines elsewhere instead of storing them locally, or for increased reliability, it could do both.

There are some disadvantages to using an external program. For example, if the program is too complex, it might consume too much CPU time and memory. In addition, if the external program has a small memory leak, it might eventually chew up all the system's memory. Finally, if the logging program blocks, there is a chance of causing a denial of service on the server.

4.1 Log

Formats:-

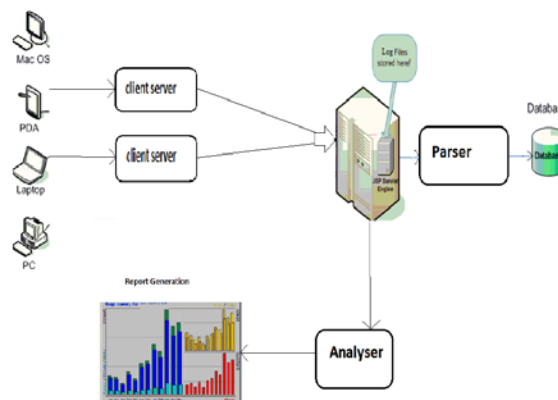
4.1.1 Access Logs:-

The server access log records all requests processed by the server.

Log Format "%h %l %u %t \"%r\" %> s %b \"%{Referer}i\" \"%{User-agent}i\" combined

4.1.2 Error Logs:-

The Error Log directive sets the name of the file to which the server will log any errors it encounters. If the file-path is not absolute then it is assumed to be relative to the Server Root.



5. MOD STATUS

The Status module allows a server administrator to find out how well their server is performing. A HTML page is presented that gives the current server statistics in an easily readable form. If required this page can be made to automatically refresh (given a compatible browser). Another page gives a simple machine-readable list of the current server state.

6. MATHEMATICAL MODEL SPECIFICATIONS

The different functionalities and use cases identified in our project are:

1. Logging the Access logs and Error logs of Apache.
2. Collecting monitoring parameters from Apache.
3. Analysis of logs, filtering of logs, reduction to useful parameters.
4. Generation of graphs, bar charts.

6.1 MATHEMATICAL MODEL FOR LOG COLLECTION:

Inputs:

Access logs.

Error logs.

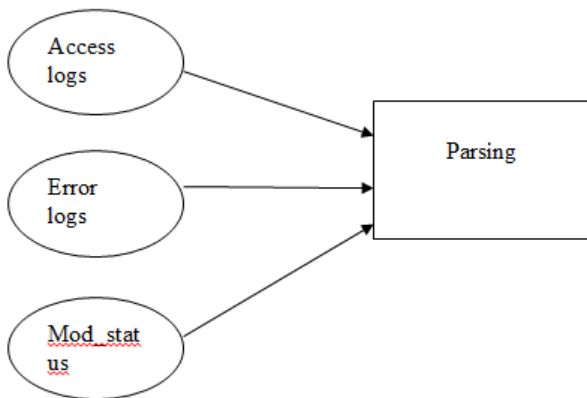
Server status information of apache.

Processing:

Piping Access and Error logs through an external script.

Collecting the server status/health information from apache by cron job.

Output: Parsed logs, separated tokens.



6.2 Mathematical Model for storing parsed tokens in database and formation of CSV:

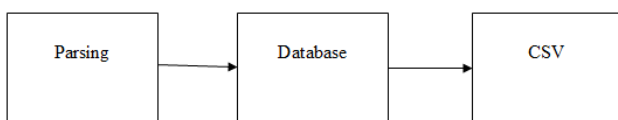
Input: Parsed tokens.

Processing:

Storing parsed tokens in mysql database.

Forming CSV from database.

Output: CSV.



6.3 Mathematical Model for log analytics:

Input: CSV.

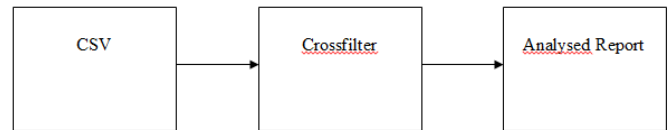
Processing:

Analysing the CSV.

Dimensioning and filtering the data.

Output: Analyzed reports in form of graphs and pie chart.

Functions: Top (), Group (), Group All (), Filter ().



6.3.1 Benefits of monitoring:-

- CPU load is analysed over a period of time and the peak hours are identified.
- The varying CPU load helps the server administrator to determine the peak hours on his server and accordingly make arrangements for scaling up his resources. The CPU load also acts as an indicator about the varying server stress and effective in predicting the failover regions and time periods.
- The CPU loads can be analysed over the range of time for a specified range of date.
- What relation response time and response code have server responds with the specific codes as per request. The analysis of the response code guides the administrator to track the server activity and response to various events of the day.
- The response code analysed in conjugation with response time, helps in tracking the issues in responses and changes in responses of the server due to specific loads over the server.
- Fraudulent activity over the server can be detected by guessing it by analysing the two matrices.

7. IMPORTING DATABASE TO CSV

CSV servers as a compact format of data storage and data transfer over the network. The compatibility of the CSV with the cross filter is the important reason for its use. The CSV is loaded at the server end and is analysed at the client end with tools such as cross filter.

7.1 Error Logs analysis:

The Error analysis serves an important tool to any to any server administrator to track the server error and hunt for the reasons for its low/degraded performance.

The analysis of logs over a period of time on a specific time slot can serve as an important metrics to the server administrator.

The analysis of score of warnings, severe error, and mild errors can guide the administrator to take directional steps to rectify the issue.

The proper planned utilization of server capacity can be availed with the judicious use of server error log analysis. Information about other matrices and how they convey information to user

7.1.1 TOP Browsers:

The metrics of Top browsers utilization can serve as a tool to server administrator to plan his extensions to his projects and also make alternative services available as per the requirements.

7.1.2 Analysis of Downtime:

The analysis of Downtime serve as an important tool to the server administrator the track the server uptime of his server also could monitor the server issues leading to downtime by the analysis of Error log over the period of time.

The geographical access to the server can't be monitored by tracking the IP that requests the server and determine their global location using the time stamps.

1. Crossfilter:

Crossfilter is a JavaScript library for exploring large multivariate database. Crossfilter supports extremely fast (<30ms) interaction and provides views.it can be used with datasets containing a million or more records.

With crossfilter you can easily create a dimension for each attribute and create a dimension accordingly with called filter method .You can even filter this new dimension again and again if required.

It provides many library methods which can be called directly and create a dimension with the corresponding filter criteria you indicated

1.1 CROSSFILTER ([RECORDS]): It constructs a new crossfilter. If record is specified, simultaneously adds the specified records.

1. Crossfilter.add (records): Adds the specified records to this crossfilter.
2. crossfilter.size (): Returns the number of records in the crossfilter, independent of any filters.
3. crossfilter.groupAll (): grouping all records and reducing to a single value.

1.2 CROSSFILTER.DIMENSION (VALUE): It constructs a new dimension using the specified value access or

function. The function must return naturally-ordered values, i.e. <, <=, >= and > operators.

1. dimension.filter (value): Filters records according to dimension's value and returns this dimension. The specified value may be null, in which case is equivalent to ALL filter.
2. dimension.filterRange (range): Filters records according to dimension's value are greater than or equal to range [0], and less than range [1], returning this dimension.
3. dimension.filterFunction (function): Filters records such that the specified function returns truth y when called with this dimension's value, and returns this dimension. For example:
4. dimension.filterAll (): Clears any filters on this dimension, selecting all records and returning this dimension.
5. dimension.top (k): Returns a new array containing the top k records, according to the natural order.
6. dimension.bottom (k): Returns a new array containing the bottom k records, according to the natural order.
7. dimension.remove (): Removes this dimension Group (Map-Reduce)

1.3 DIMENSION.GROUP ([GROUPVALUE]): Constructs a new grouping for the given dimension, according to the specified group Value function, which takes a dimension value as input and returns the corresponding rounded value.

1. group.size (): Returns the number of distinct values in the group, independent of any filters.
2. group.reduce (add, remove, initial): Specifies the reduce functions for this grouping, and returns this grouping.
3. group.reduceCount (): method for setting the reduce functions to count records;
4. group.reduceSum (value): A method for setting the reduce functions to sum records using the specified value access or function.
5. group.order (order Value): Specifies the order value for computing the top-K groups.

Scalar vector graphics (svg): Used to display data in form of reports and graphs to the user.

Scalar vector graphics is an XML-based vector image format for two-dimensional graphics. It has support for interactivity and animation. The SVG specification is an open standard developed by the World Wide Web Consortium (W3C).

SVG images and their behaviours are defined in XML text files.

SVG has following advantages

1. Images can be searched, indexed, scripted, and, if need be, compressed.
2. Images are scalable and are zoomable
3. SVG files are pure XML.

The different elements in SVG.

<svg>: The SVG code begins with this element

</svg>: the end of svg code.

<rect>: rectangle

<circle>: circle

<line>: line

8. CONCLUSIONS

The paper presents the architectural view of the web server monitoring tool. The web server monitoring tool works in the phases of logging, parsing & analyzing the logged data of the server. The web server monitoring tools provides a utility to monitor the web server with the time based statistics of the server performance.

ACKNOWLEDGMENT

We express our true sense of gratitude towards our project guides Prof. Mrs S.R.Vij & Mr Ashish Gupta, who at every discrete step in the study of this project, contributed with her valuable guidance and provided with perfect solutions for every problem that arose.

We also extend my sincere thanks to all staff members who extended their kind support and encouragement during the preparatory steps of this project.

We would also like to express my appreciation and thanks to all my friends who knowingly or unknowingly, assisted us with their valuable suggestion and comments and we are very grateful for their assistance.

REFERENCES

- [1] Y. Hu, A. Nanda, and Q. Yang, "Measurement, analysis and performance improvement of the Apache Web server," in Proceedings of the 18th IEEE International Performance, Computing and Communications Conference, (Phoenix/Scottsdale, Arizona), pp.261-267, Feb. 1999.
- [2] T. Robinson, E. Merenda, and S. Curtis, "IBM RS/6000 Web server sizing guide." <http://www.rs6000.ibm.com/resource/technology/sizing.html>, Apr. 1996.
- [3] E. D. Katz, M. Butler, and R. McGrath, "A scalable web server: The NCSA prototype, in WWW'94 Conference Proceedings, 1994.
- [4] A. Bestavros, R. L. Carter, M. E. Crovella, C. R. Cunha, A. Heddaya, and S. A. Mirdad, "Application-level document caching in the internet," in Proceedings of the Second Intl. Workshop on Services in Distributed and Networked Environments (SDNE'95), 1995.
- [5] A. Luotonen and K. Altis, "World-Wide Web proxies," in WWW'94 Conference Proceedings, 1994.
- [6] M. Abrams, C. R. Standridge, G. Abdulla, S. Williams, and E. A. Fox, "Caching proxies: Limitations and potentials," in Proceedings of the Fourth International Conference on the WWW, (Boston, MA), Dec. 1995.
- [7] C. Maltzahn, K. J. Richardson, and D. Grunwald, "Performance issues of enterprise level web proxies," in Proceedings of the 1997 SIGMETRICS Conference on Measurement and Modeling of Computer Systems, June 1997.
- [8] P. Cao and S. Irani, "Cost-aware WWW proxy caching algorithms," in USENIX Symposium on Internet Technologies and Systems (USITS), Dec. 1997.
- [9] J. Gwertzman and M. Seltzer, "The case for geographical pushcaching," in Proceedings of the 1995 Workshop on Hot Operating Systems, 1995.
- [10] S. Glassman, "A caching relay for the World Wide Web," in WWW'94 Conference Proceedings, 1994.
- [11] A. Chankhunthod, P. B. Danzig, C. Neerdaels, M. F. Schwartz, and K. J. Worrell, "A hierarchical internet object cache," in Proceedings of the 1996 USENIX Technical Conference, (San Diego, CA), Jan. 1996.
- [12] A. Cockcroft, "Watching your Web server." <http://www.sun.com/sunworldonline/swol-03-1996/swol-03-perf.html>, Mar. 1996.
- [13] E. P. Markatos, "Main memory caching of web documents," in Fifth International WWW Conference, May 1996.
- [14] J. Rubarth-Lay, "Keeping the 400lb. gorilla at bay: Optimizing web performance." <http://eunuch.ddg.com/LIS/CyberHornsS96/j.rubarth-lay/PAPER.html>.
- [15] The Apache Team, "Apache HTTP server project." <http://www.apache.org/>.
- [16] M. F. Arlitt and C. L. Williamson, "Web server workload characterization: The search for invariants," in Proceedings of the 1996 SIGMETRICS Conference on Measurement and Modeling of Computer Systems, May 1996.
- [17] V. Almeida, A. Bestavros, M. Crovella, and A. de Oliveira, "Characterizing reference locality in the WWW," in Proceedings of the 1996 IEEE Conference on Parallel and Distributed Information Systems, Dec. 1996.
- [18] M. E. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: Evidence and possible causes," in Proceedings of the 1996 SIGMETRICS Conference on Measurement and Modeling of Computer Systems, (Philadelphia, PA), May 1996.
- [19] The Standard Performance Evaluation Corporation, "SPECweb96 benchmark." <http://open.specbench.org/osg/web96/>.
- [20] SGI, "WebStone World Wide Web server benchmarking." <http://www.sgi.com/Products/WebFORCE/WebStone/>.
- [21] D. A. Helly, AIX/6000 Internals and Architecture. New York, NY 10020: McGraw-Hill, 1996.
- [22] L. McVoy and C. Staelin, "Imbench: Portable tools for performance analysis," in Proceedings of the 1996 USENIX Conference, 1996.
- [23] J. M. Almeida, V. Almeida, and D. J. Yates, "Measuring the behavior of a world-wide web server," in Seventh Conference on High Performance Networking (HPN), (White Plains, NY), pp. 57-72, Apr. 1997.
- [24] R. E. McGrath, "Performance of several web server platforms." <http://www.ncsa.uiuc.edu/InformationServers/Performance/Platforms/report.html>.