Web Log Analysis for Performance Troubleshooting

Authors: Rosario Alfano, Giuseppe Cassone, Danilo Gotta

Abstract

Web Server Logs contain powerful, but often hidden, information about the performance of your Web Applications. In this session, you'll learn how easy it is to enlarge your Web Performance Testing toolkit. Many commercial tools use log files to extract information about visitors' behavior. These tools calculate lots of statistics, often used in Load Testing to reproduce realistic workloads. But a log file contains other information. Web Performance Log Analysis is a new activity based on the performance information of Web Server Logs (elapsed time, bandwidth, number of hits, and more). Telecom Italia Lab shows you all the "performance" information you can extract from the log and how you can best use it describing a set of metrics and statistics that can help to highlight all the components affected by performance bottlenecks. For fast, easy access to this information, we have developed a tool that produces about ten graphs and tables that can be analyzed online (with a friendly GUI) or off-line (through an automatically generated Word document).

Introduction

All web servers save information about their activities in a log file. This file contains detailed information about each request received by the web server from site visitors' browsers. Each line recorded in the log file is called a "hit".

Many commercial tools use a log file to extract information about the behavior of site visitors, an activity known as Web Log Analysis (WLA) intended to characterize the behavior of the users of a web site. Commercial Web Log Analyzers calculate various types of statistics (for example, the main paths followed inside the site, the main entry pages to the site, the region in which users are located, etc.). These statistics can then be used for marketing or, in the case of performance testing, to reproduce a load that precisely reflects the effective load of the site during Load Testing.

However, many commercial tools overlook a field of the log files, i.e. the Time-Taken field. This parameter represents the time taken to process each hit on the web server, i.e. the time required by a web server to perform an HTTP request from a browser. The Time-Taken represents a good approximation of the time the user effectively waits and is therefore a useful yardstick for charting web application performance.

The fact that useful information for performance analysis can also be extracted from a log file has given rise to the activity of Web Performance Log Analysis and promoted development of the TILog Web Performance Log Analyzer program. The aim of this activity is to calculate certain statistics that provide an overview of performance and make it possible to pinpoint weak components of the site (for example, very heavy cgi scripts). The TILog program calculates these statistics using a web server log file and automatically generates a report in Microsoft Word format. This report contains all the graphs and tables obtained from the statistics, which promote fast identification of performance-critical parts of a site.

After a short introduction covering the method used by TILab for Web Performance Testing & Measurement, this document describes the activity supported by TILog, the reasons behind its development, the main characteristics of the program and various hints for subsequent development.

WEB Performance Testing & Measurement

The aims of Performance Evaluation are:

- Establish system, site or application current performance;
- Identify bottlenecks;
- Assess future performance;
- Provide useful advice regarding possible solutions and/or improvements.

Generally, a performance evaluation activity is based on:

End-to-end monitoring– This activity is directed towards charting performance as perceived by the end user who navigates on the site. This is important because users tend to use a site as long as the services available and performance remain satisfactory. For example, as soon as response times increase (see also the 8 seconds Rule) or if they cannot use site services, they have no hesitation in switching to competitor sites (each competitor is just ONE CLICK AWAY). End-to-End monitoring can be defined as such when it complies with two key factors: the measurement point is identical to that of the end user and the measurement tool is identical to that normally used by the end user Not all commercial monitoring tools or services (although often called End-to-End) permit effective assessment of the entire connection chain and should be defined more correctly as near End-to-End Monitoring tools. For End-to-End monitoring, TILab has developed a tool (BMPOP) based on real End-User Components (browsers, active-x, players etc.) and on interconnections to the network using the different access methods made available to the end user by an ISP, Corporate or Mobile Operators (dialup on PoP, also of different ISPs, distributed in a geographical area, ADSL, RAS or corporate LAN, VPN, GSM, GPRS, WAP). These distinctive characteristics promote checking and measurement (also in real time) of the performance effectively perceived by the user at periodic intervals and on a statistical basis, comparative performance assessments between various access services and monitoring of the typical quality parameters of the most advanced Service Level Agreements.

Near End-to-End Monitoring– Similarly to End-to-End monitoring, this activity is designed to establish performance as perceived by the end user who navigates on the site. Many different monitoring tools are available, some based on Ghost Transactions, others on Java Instrumentation, but not all comply with the two factors outlined above which characterize a real End-to-End measurement. Near End-to-End monitoring makes it possible to record estimates of the download times of the various pages of the site usually using fast line connected directly to the Internet backbone. These are useful, for example, for benchmarking sites but not for determining the absolute performance levels perceived by the user.

Load Testing - Load testing meets the need for advance checking of response in terms of load capacity of the site/application and of back-end systems (DB-server and Application server) as the number of users varies. It is possible to verify/forecast web site performance as the number of users connected increases. In this way, excessively high response times can be measured and the necessary corrective actions (for example, tuning the hardware and software platform) can be adopted well in advance in relation to the highest load conditions. This activity can be carried out directly in Operation, for checking of scalability and/or for tuning the current architecture, or in a Test Plant for sites not yet in production. Using a set of tests, it is possible to reproduce the behavior of a site in operation, stressing this with high “loads” as required and to test, for example, new site releases on a test-plant. To carry out these simulations, traffic, equal to the required number of users, is generated towards the site under test, precisely reproducing user behavior so as to obtain an overview of current performance of the site/application in terms of the response times of the pages (transactions). The delays measured in the response times, in relation to those expected, pinpoint the weak points of the system.

Web Log Analysis – Web Log Analysis (WLA) provides useful information about user behavior but, as regards performance, is able to support the creation of precise, realistic workloads. All web servers save a great deal of information in a log file. This file contains detailed information about the number of accesses to the site, the geographical classification of users, their daily distribution, the average navigation times for each user, pages visited, user paths between pages of the site, site entrance and exit pages, browsers used, the main keywords for searching for the site, Operating Systems, etc. Leveraging this information, load tests that provide a very precise view of user activities can be created (the users who navigate on the site are not all the same: they don't click on the same things and do not all have the same familiarity).

Resource monitoring – Resource monitoring makes it possible to measure systems engineering performance (in terms of CPU, Memory, Disk, etc. consumption) of the systems on which web applications run. However, resource monitoring must be directed towards correlating the information extracted and use of the site (application data). There is no point in knowing that a CPU is at 60% if you don't know what the application has produced with that consumption. Many of the commercial resource monitoring tools available now also permit concurrent extraction of information (tens of metrics) about the application platforms that run on these systems.

The Complete Approach to Web Performance

To achieve the objectives of a Performance Evaluation activity, this must be based on an approach supported by sound, valid underpinnings. TILab has defined its "Complete Approach to Web Performance Testing and Measurement" [1], which is based on integration of all the components described above and on the use of suitable tools. TILAB uses the best commercial tools available and where these do not completely cover the requirements that have emerged, it develops ad-hoc tools (BMPOP, TILOG).

TILab is convinced that an integrated approach to the problem, from several points of view, returns the best results and assures higher value added compared with isolated end-to-end monitoring, load testing, web log analysis and resource monitoring activities.

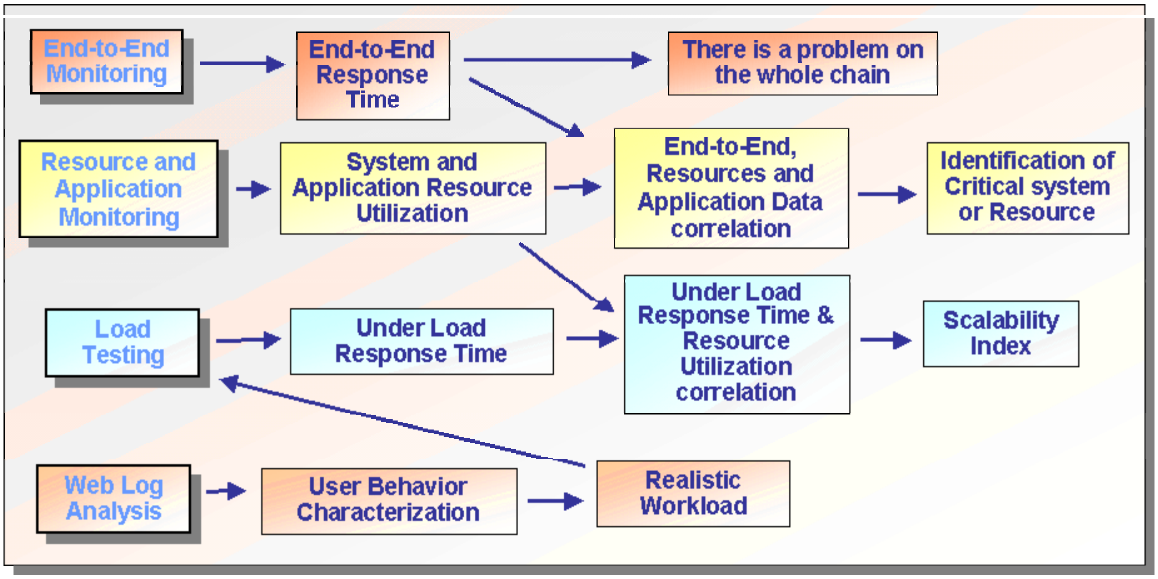

End-to-End Monitoring
End-to-End Response Time
There is a problem on the whole chain
Resource and Application Monitoring
System and Application Resource Utilization
End-to-End, Resources and Application Data correlation
Identification of Critical system or Resource
Load Testing
Under Load Response Time
Under Load Response Time & Resource Utilization correlation
Scalability Index
Web Log Analysis
User Behavior Characterization
Realistic Workload

Figure 1: Complete Approach Schema

As can be seen in the figure above, the underlying idea of TILAB's complete approach is integrated use of the various components as follows. Initial End-to-End monitoring activities permit fast charting of performance as perceived by the end user. In the case of downgrading, activate resource and application program monitoring, correlating the results with the application data in order to pinpoint critical elements (both application and systems engineering). If high increases in load are expected (for example, following an advertising campaign), critical situations can be avoided by carrying out load tests to check site scalability. As the more realistic the workloads used, the more significant the load test, the current load in operation must be established correctly and then reproduced according to certain multiplication indexes. Web Log Analysis tools can be used for correct workload definition. These tools, intended originally for marketing purposes (to verify user behavior and therefore offer increasingly attractive, productive sites), can be used for correct identification of operating load and therefore to construct realistic workloads.

Web Performance Log Analysis

The new Web Performance Log Analysis (WPLA) activity makes it possible to extract a suite of useful information for performance analyses from the log file of a web server. Web server logs can be configured to record an optional value that is very useful for performance analyses, i.e. the Time-Taken field which measures the time taken by the web server to process each object of the page. Analyzing this value, it is possible to expand the Performance Evaluator's toolkit, integrating this with other components of the complete approach.

The log files of a web server (how these are generated, what they contain, how they are customized), how performance information can be extracted, which metrics are to be produced and how, are discussed below. Using the TILOG tool, this activity is very simple and automatic.

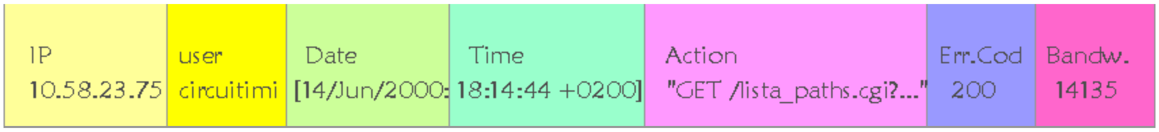
Web Server and Log File

Web Servers are applications able to interpret and respond to HTTP requests from web browsers. The HTTP protocol, which defines the rules for exchange of web pages and contents, is client/server transaction oriented (the client is usually called Browser and the server is defined as Web Server). A typical HTTP transaction is performed as follows:

- The user types the URL of the site he wants to visit or the page to be displayed in the browser.
- The browser establishes a TCP connection with the machine on which the Web Server, containing the page identified by the URL, is running.
- The browser sends a request to the Web Server in HTTP format in which the object required is usually specified.
- The Web Server interprets the HTTP request, searches for the object required (for example, static pages) if necessary querying a Database or an Application server in order to construct the object (for example, dynamic pages).
- The Web Server constructs a reply in HTTP format containing a numeric response code (for example, 200 means that the request has been served successfully) and the object requested and sends this to the client.
- The TCP connection is closed.
- Some of the data relating to the transaction are recorded in the log file.

Log file format

A certain amount of information regarding each navigation action performed by the user of the browser on the site is saved in the log file. The information is recorded according to single hit, i.e. for each single object of the web page. For example, there will be the hit of the html file, of all the gif or jpg images, of the cgi, of the applets, etc. The date, time, origin, user name, path of the object, etc. are usually recorded for each hit.



IP	user	Date	Time	Action	Err.Cod	Bandw.
10.58.23.75	circuitimi	[14/Jun/2000:	18:14:44 +0200]	"GET /lista_paths.cgi?..."	200	14135

Various commercial Web Servers are available. The most famous and widely used are Apache and Microsoft IIS. Most Web Servers make it possible to save the log files in “Common Log Format”, a standard defined by the W3C consortium and supported by major Web Servers and most Web Log Analyzers.

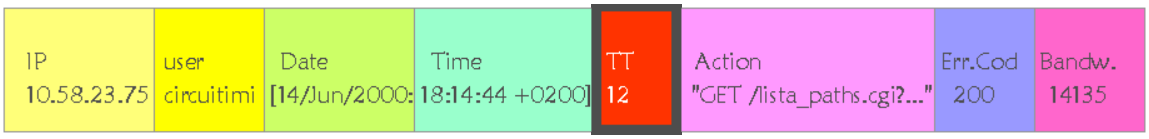
Commercial Web Servers also make it possible to define one or more proprietary formats that can be configured as required by the user. Configuration methods vary according to the Web Server: Apache provides a directive in its textual configuration file, while IIS provides a graphic window for selection of the fields to be included in the log file. The Time-Taken field, used for

Web Performance Log Analysis, is not present in the Common Log Format and in other pre-defined formats. Therefore, the site administrator must modify the format of the log file.

The Time-Taken field

The data saved in a standard log file provide an accurate, detailed view of web site users' behavior. This information, which is also very useful to Marketing for commercial purposes (for example, advertising banner management), can also be used effectively to generate realistic workloads for load tests. But a log file can provide much more: the Time-Taken field.

The Time-Taken can be considered a new element for performance analyses. This field can be inserted in each entry of the log file (and therefore for each hit recorded).



IP	user	Date	Time	TT	Action	Err.Cod	Bandw.
10.58.23.75	circuitimi	[14/Jun/2000:	18:14:44 +0200]	12	"GET /lista_paths.cgi?..."	200	14135

The Time-Taken measures the time that passes between reception of the GET by the server to reception, by the server, of the ACK referring to the last packet, i.e. from the moment in which the server starts to manage the request until reception of the last byte is confirmed by the client.

The DNS resolution, the start of the TCP connection and (partially) client-side processing are excluded from the measurement. However, with HTTP 1.1, the start of the TCP connection is performed only for the first resource requested which means that the time measured is very close to the time the user effectively waits.

In the following cases, the time-taken field considerably underestimates user waiting time:

- Java or Flash applets that considerably increase client-side processing on slow clients;
- Advertising banners on external sites, possibly with exchange of cookies (many TCP connections are opened and closed, with the related initial DNS connections and handshakes; also, the TT of the banner which is not on the server concerned is not known);
- Pages with many reduced size resources in the case of HTTP 1.0 connections;
- Particular conditions on the server that cause a delay between the moment in which a packet is received to when reception is communicated to the HTTP demon/service. There are 2 error points of this type with opposite signs (a delay in communicating the GET reduces TT, a delay in communicating the ACK increases this). Therefore, except in the case of major variations, these offset each other and must in any case remain within some tens of milliseconds.

It should be noted that IIS records the value of the Time-Taken field in milliseconds, therefore with maximum precision. Apache, on the other hand, measures it in seconds and performs a “strange” rounding off. In fact, rounding off (through truncation) is carried out not on the Time-Taken effectively measured but on the start time and end time (the Time-Taken is measured as the difference between the 2). Apache measures the times with the *time()* function of the C which measures the seconds starting from a certain date. Therefore, for example, a 999ms transaction that starts at 9:00:00.000 will be measured as 0 seconds and a 1001ms transaction that starts at 9:00:00.999 will be measured as 2 seconds. The maximum error should therefore be 1s.

Generally speaking, a hit (that is not a "download" such as a pdf or a zip etc.) with high TT is always a problem while a hit with low TT may also represent a problem due to faulty design (many small images, external banners, etc.).

Tests carried out at TILAB have shown that this time is a good approximation of the time each user waits for execution of each hit. The element missing in relation to the time the end user actually waits is the rendering time on the browser and the time of the GET of the data. This is however a very minor request in terms of size, comparable with a simple PING, which is normally very fast.

Why Time-Taken is useful

Having established that Time-Taken is, in the most of cases, a good approximation of the time each user waits for each object of the web page, the next problem is the usefulness of this information and, in particular, how this activity can be integrated in the complete approach described above.

Global vs. Targeted Monitoring

End-to-End monitoring makes it possible to assess the end user's perception of site performance. Many companies use services of this type to track how their customers rate the performance of their site. These services are based on Ghost Transactions or on Java Instrumentation. In the first case, the agents must be configured initially to monitor particular pages/transactions and, usually, the response times and availability of some of the most important and/or critical pages are measured. In the second case based on Java Instrumentation, reference is usually made to the applets that are downloaded on the users' PCs and which are used to make the measurements.

Monitoring based on End-User Components (remote access, web browser, active-x, player, etc, i.e. the same tools that are used by the end users) reproduces users' behavior patterns and therefore records real performance. This technology is able to highlight performance problems for particular configurations of end users that are, otherwise, difficult to quantify (e.g. network accesses with limited band or with high latency, non-optimal hardware configurations etc.), to reveal the existence of performance problems also in complex situations (e.g. when the individual components that contribute to providing a service are working correctly but are affected by interaction and/or configuration problems). Near End-to-End and End-to-End monitoring measure performance on a statistical basis as they carry out tests with a certain sampling interval on an incomplete set of web pages, not by all the users but via agents that are in any case configured on real end-user stations. The significance of the measurement depends to a considerable extent on the technology used, on the statistical sample defined and therefore on the configuration of the actual measurement system. Performance analyses based on the Time-Taken field refer to all navigation on the site by all users who have accessed this. They provide precise feedback regarding the download time of the individual objects that form the web pages. As mentioned above, if this measurement is closely related to the download time of a web page because it does not take into account certain times (e.g. DNS resolution, processing of the web browsers, cache mechanisms of the web browsers, content delivery, advertising mechanisms, etc.), the measurement returned is in fact a good approximation in most cases. Therefore, Time-Taken analyses are essential during site tuning and advanced troubleshooting in order to establish if and which objects have excessively long download times and return complete results both in terms of transactions and users because they are obtained using log files generated by the navigation of all users on the entire site.

Detailed Analysis of Site critical objects

Complementary to real-time End-to-End monitoring, which is essential to record the user's real perception, with different results according to the method of interconnection to the site, analysis based on the Time-taken recorded in the log files of the web servers is intended to permit precise identification of the individual critical objects (graphic files, applets, cgi, queries) on the entire site.

No downgrading is introduced

After describing the reasons why this value has been chosen and before illustrating how it is used, it is important to note that insertion of the Time-Taken in the log file does not slow the web server. Tests on various Web Servers (both IIS and Apache) have not revealed any downgrading of response times and no increase in load on processing resources. The only impact is on the size of the log file, which increases by around 2%.

WPLA in the Complete Approach

As mentioned above, Web Performance Log Analysis integrates seamlessly with the activities described in the complete approach. The link between the activities is, in fact, defined by the end-to-end monitoring, resource monitoring, log file Time-Taken field analysis sequence. The figure that represents the complete approach is revealed obtaining the following layout.

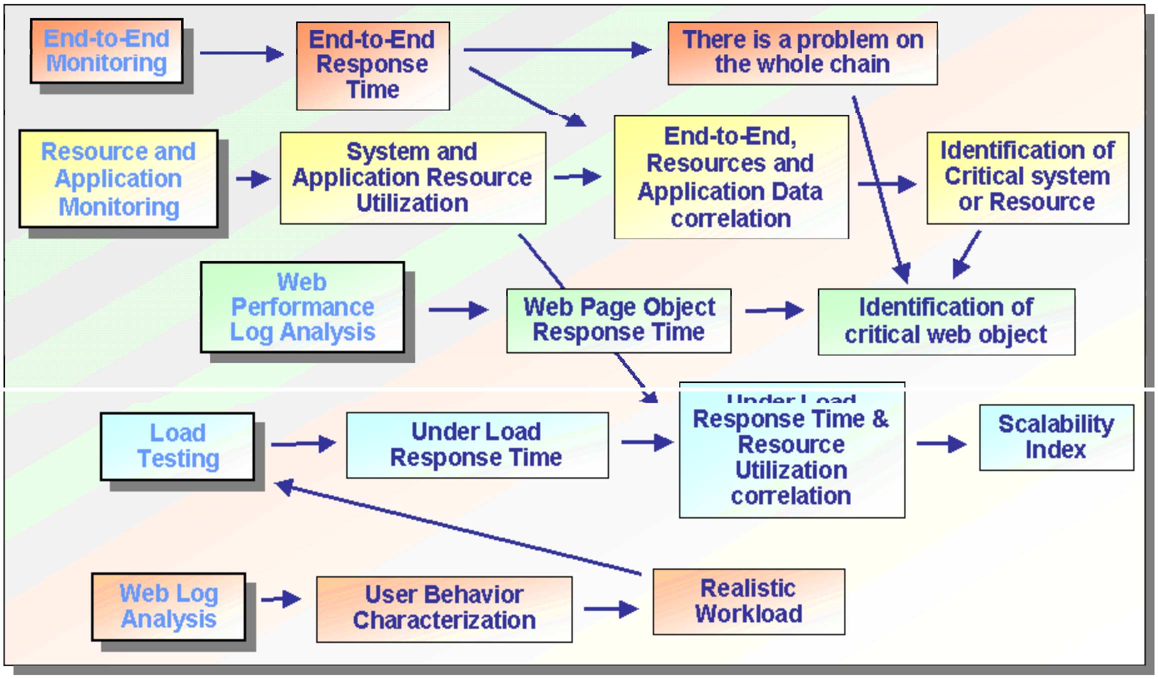

End-to-End Monitoring
End-to-End Response Time
There is a problem on the whole chain
Resource and Application Monitoring
System and Application Resource Utilization
End-to-End, Resources and Application Data correlation
Identification of Critical system or Resource
Web Performance Log Analysis
Web Page Object Response Time
Identification of critical web object
Load Testing
Under Load Response Time
Under Load Response Time & Resource Utilization correlation
Scalability Index
Web Log Analysis
User Behavior Characterization
Realistic Workload

Figure 2: WPLA e Approccio Completo

TILOG – Telecom Italia Lab Log Analyzer

TILAB considers that very important results can be obtained from performance analyses based not only on Load Testing, End-to-End Monitoring, Resource Monitoring but also on the information saved in web server log files. Commercial Web Log Analysis tools usually consider other fields of the log file (the standard fields) without analyzing the Time-Taken field. This can be ascribed partly to the fact that these tools are designed for other purposes (to check user behavior). TILAB decided to develop TILog in order to fill this gap.

TILog is a very simple to use as all that is needed is the log file. It automatically recognizes the format of the log file but it is always possible to instruct the tool manually so that it recognizes non-standard formats. TILog generates analyses (graphs and tables) on-line but also generates a report in Microsoft Word format.

TILog has proved to be a very useful tool in supporting TILAB's Web Performance Evaluation activities and is presented below in this paper in order to describe the various types of information produced using the log file and, in particular, the Time-Taken field.

Information Produced by TILog

TILog is a tool for Log file Analysis for performance purposes and is not therefore intended to replace, as regards use, WLA tools. It does not attempt to provide information about users, their behavior, their preferences. The TILog tool completes the Performance Evaluator's toolkit and, as such, simply analyzes the information of a log file able to provide site performance measurements and therefore integrate end-to-end, load testing and resource monitoring information.

The underlying idea of the output generated by TILog is to produce a set of reports that help to identify problems. Our initial efforts in this direction have led us to a number of basic considerations regarding statistics.

> *The object of statistics is to discover methods of condensing information concerning large groups of allied facts into brief and compendious expressions suitable for discussions.*
>
> *Francis Galton*

The above quotation is a satisfactory summary of the way in which TILog uses statistics. The program calculates a set of summary statistics regarding web site behavior so as to highlight parts that need to be tuned. The statistics have been selected so that the user is able to identify immediately the weakest components of the site as regards performance. The tables and graphs are always sorted according to indexes that highlight relevant aspects of site performance (for example, objects that tend to slow down most of the operations of the site). This is the first version of the tool so further releases may be useful in order to improve and integrate the metrics produced at the moment.

The indexes and First Tables

When describing a set of data, it is useful to select an index that is representative of most of the values of the sample analyzed. These indexes are called central trend indexes. A frequent error is that of thinking that the arithmetic mean is always the most suitable central trend index to represent a set of data. The response times, on which TILog calculates the statistics, are a perfect example of how the arithmetic mean can result in errors of interpretation.

According to an analysis of real cases, the distribution of the response times, obtained from the log file, is generally characterized by a very high number of values with very low response times and a few values with very high response times, as shown in the figure below.

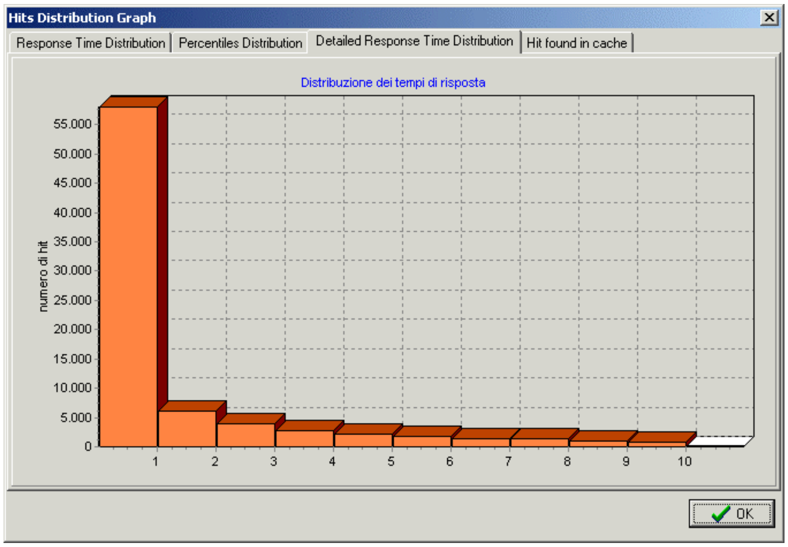
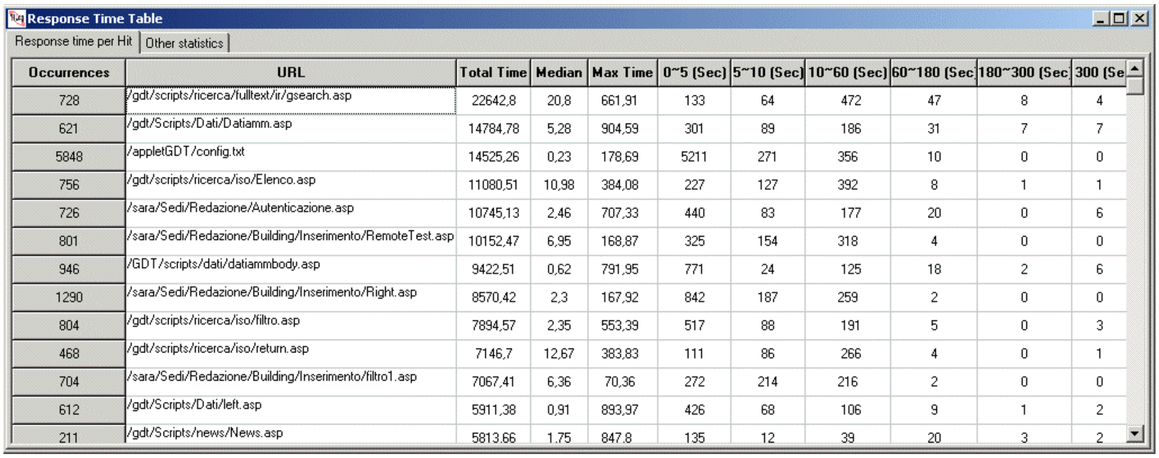


Figure 3: Detailed distribution of response times from 0 to 30 seconds

The mean is not the most suitable index for distributions of this type since, as it is an index affected by the peak values of the samples, it could result in errors of assessment regarding site performance. In these cases, the most suitable central trend index is the median.

The main table produced by the program indicates the values of this index together with other statistics that make it possible to appreciate site behavior.

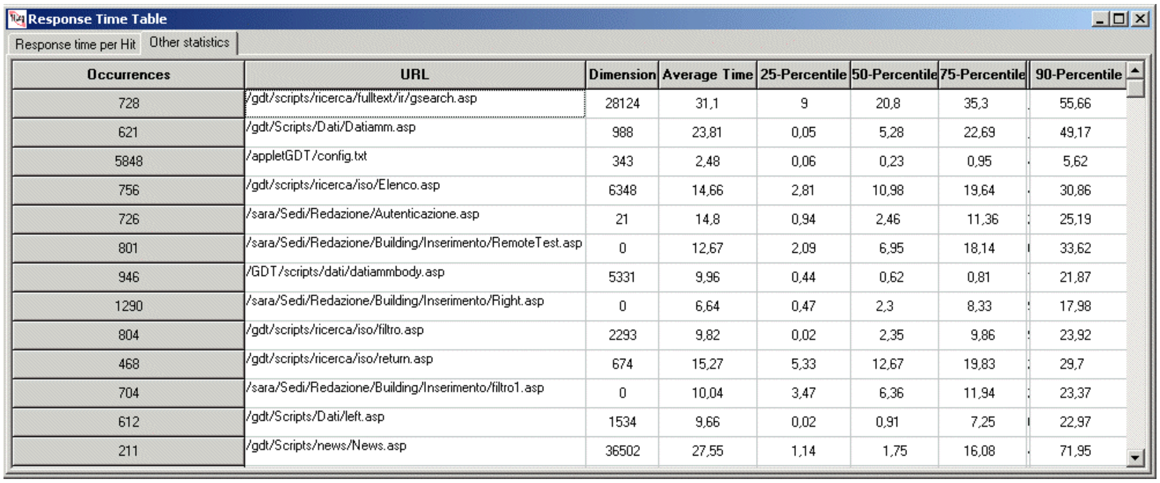


Occurrences	URL	Total Time	Median	Max Time	0~5 (Sec)	5~10 (Sec)	10~60 (Sec)	60~180 (Sec	180~300 (Sec	300 (Se
728	/gdt/scripts/ricerca/fulltext/ir/gsearch.asp	22642,8	20,8	661,91	133	64	472	47	8	4
621	/gdt/Scripts/Dati/Datiamm.asp	14784,78	5,28	904,59	301	89	186	31	7	7
5848	/appletGDT/config.txt	14525,26	0,23	178,69	5211	271	356	10	0	0
756	/gdt/scripts/ricerca/iso/Elenco.asp	11080,51	10,98	384,08	227	127	392	8	1	1
726	/sara/Sedi/Redazione/Autenticazione.asp	10745,13	2,46	707,33	440	83	177	20	0	6
801	/sara/Sedi/Redazione/Building/Inserimento/RemoteTest.asp	10152,47	6,95	168,87	325	154	318	4	0	0
946	/GDT/scripts/dati/datiammbody.asp	9422,51	0,62	791,95	771	24	125	18	2	6
1290	/sara/Sedi/Redazione/Building/Inserimento/Right.asp	8570,42	2,3	167,92	842	187	259	2	0	0
804	/gdt/scripts/ricerca/iso/filtro.asp	7894,57	2,35	553,39	517	88	191	5	0	3
468	/gdt/scripts/ricerca/iso/return.asp	7146,7	12,67	383,83	111	86	266	4	0	1
704	/sara/Sedi/Redazione/Building/Inserimento/filtro1.asp	7067,41	6,36	70,36	272	214	216	2	0	0
612	/gdt/Scripts/Dati/left.asp	5911,38	0,91	893,97	426	68	106	9	1	2
211	/gdt/Scripts/news/News.asp	5813,66	1,75	847,8	135	12	39	20	3	2

Other indexes have also been considered in addition to the median: Total Time and Percentiles.

The Total Time is the time taken by the web server to respond to all the requests for a specific object. It is, therefore, an index that takes into account both the slowness of the object and the number of times the object is requested from the web server during site navigation. Sorting the hits according to this index is intended to facilitate identification of the objects to be tuned, thus maximizing the results.

Using a single value to summarize a set of data is rather restrictive. Usually, another index is used that makes it possible to appreciate the variability of the data in relation to the central trend index adopted. This value is called the dispersion index. Obviously, the form of distribution has also influenced selection of the dispersion index. The percentiles are values that, in these cases, provide an excellent indication of data dispersion. The second table indicates the values of the first, second and third quartile for each object of the site, providing information regarding data dispersion. This value is another index used frequently in the graphs of the report.



Occurrences	URL	Dimension	Average Time	25-Percentile	50-Percentile	75-Percentile	90-Percentile
728	/gdt/scripts/ricerca/fulltext/ir/gsearch.asp	28124	31,1	9	20,8	35,3	55,66
621	/gdt/Scripts/Dati/Datiamm.asp	988	23,81	0,05	5,28	22,69	49,17
5848	/appletGDT/config.txt	343	2,48	0,06	0,23	0,95	5,62
756	/gdt/scripts/ricerca/iso/Elenco.asp	6348	14,66	2,81	10,98	19,64	30,86
726	/sara/Sedi/Redazione/Autenticazione.asp	21	14,8	0,94	2,46	11,36	25,19
801	/sara/Sedi/Redazione/Building/Inserimento/RemoteTest.asp	0	12,67	2,09	6,95	18,14	33,62
946	/GDT/scripts/dati/datiammbody.asp	5331	9,96	0,44	0,62	0,81	21,87
1290	/sara/Sedi/Redazione/Building/Inserimento/Right.asp	0	6,64	0,47	2,3	8,33	17,98
804	/gdt/scripts/ricerca/iso/filtro.asp	2293	9,82	0,02	2,35	9,86	23,92
468	/gdt/scripts/ricerca/iso/return.asp	674	15,27	5,33	12,67	19,83	29,7
704	/sara/Sedi/Redazione/Building/Inserimento/filtro1.asp	0	10,04	3,47	6,36	11,94	23,37
612	/gdt/Scripts/Dati/left.asp	1534	9,66	0,02	0,91	7,25	22,97
211	/gdt/Scripts/news/News.asp	36502	27,55	1,14	1,75	16,08	71,95

The 90th percentile is the time below which the web server has responded to 90% of the requests for the object. Unlike the total time, this value does not always make it possible to identify the objects that affect all-round performance of the site. Some objects at the top of the table may have high times but may have been requested only a few times. However, this value is important to identify objects that, as they are structurally slow, could delay some site operations. Improving this value for an object assures that 90% of the requests will be answered in a reasonable time and, in most cases, the absence of objects that are too slow. The 90th percentile also has the advantage of cutting the limit values of the sample. Working on the 90th percentile prevents errors of interpretation due to time peaks for some replies.

Other Graphs Produced

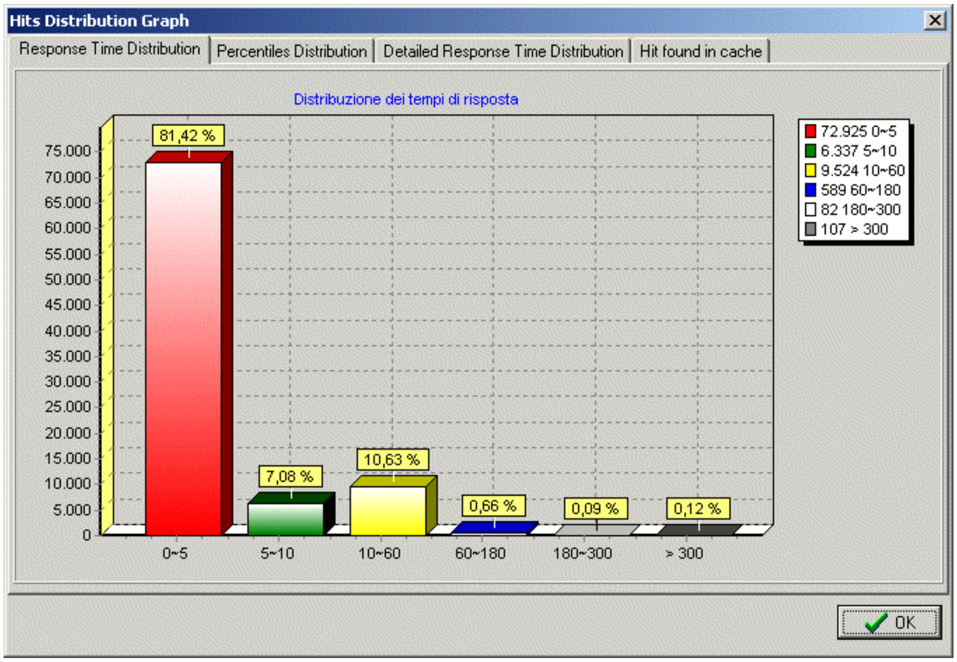
In addition to the 2 tables shown in the previous paragraph, TILog generates other graphs that are very useful for problem analysis:

- Response Time distribution (0-5, 5-10, 10-60, ...>300);
- Response Time distribution details (0,1,2,3,4,...10);
- Distribution of the Percentiles;
- Hit/hour distribution;
- Median Trend and 90th percentile;
- 90th percentile of the Top 10 sorted according to Average Time;
- 90th percentile of the Top 10 sorted according to Total Time;
- Percentage Hits found in cache.

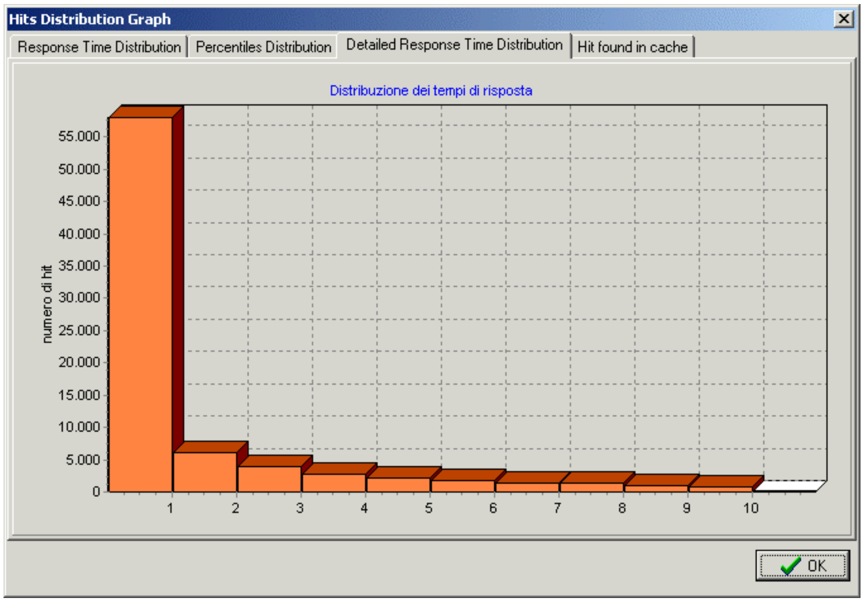
We will now have a closer look at these.

Response Time Distribution

The response time distribution graph makes it possible to establish the outline trend of the site. As the measurements are made on individual objects and not on entire pages, it can be expected that most of the response times of the objects are in the first range (0-5 seconds). Higher response times for a single object are to be considered critical (Note: the 8 seconds rule refers to the entire page !!!) except in the case of specific, normally heavy objects such as search or query functions which generate extraction of a large bulk of data from a DB.

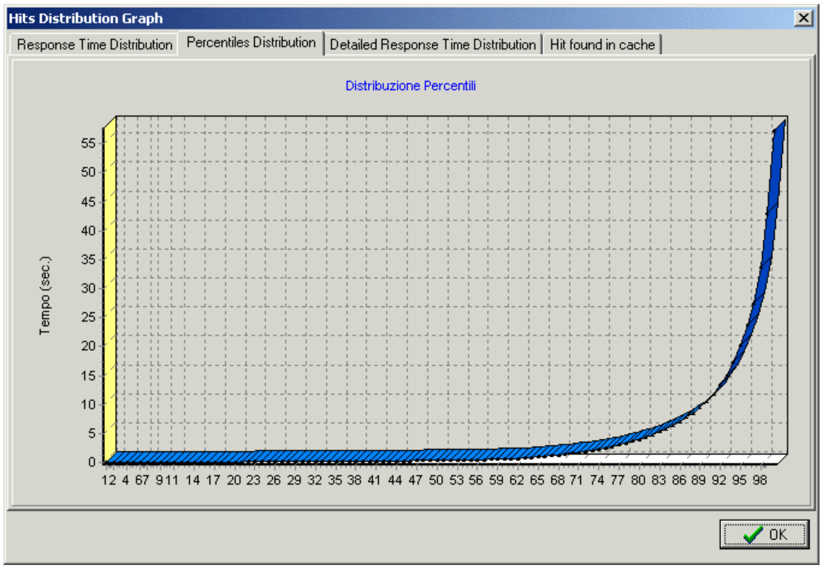


In optimal situations, almost all the response times are within the first range. In this case, the following graph, which illustrates the precise distribution of the response times of the 0-10 seconds range, is very useful for further analyses.



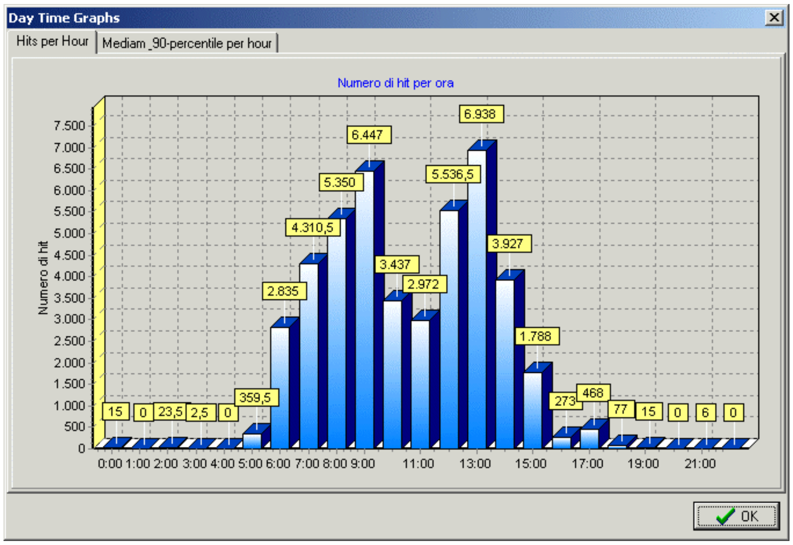
Distribution of the Percentiles

The graph that illustrates the distribution of the percentiles makes it possible to verify the response time trend of each object in relation to the total population. After identifying the "worst" objects using the "total times" table, the graph of the percentiles can be generated using a number of filters provided in the software. The graph of the percentiles highlights the percentage users with good response times. Sharp increases in the graph indicate critical points in scaling the performance of the object. The optimal situation is characterized by low, almost flat percentile graphs. This means that all the users "perceive" the same low waiting time. Also, according to the trend of the percentiles, a quality assessment can be made of the effectiveness of the Average, Median and 90th percentile metrics.

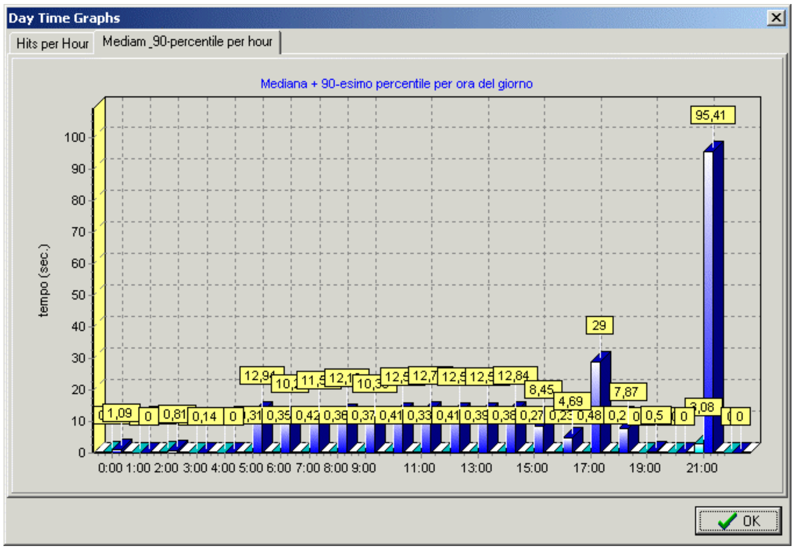


Hourly Trend (Hits, Medians and 90th Percentile)

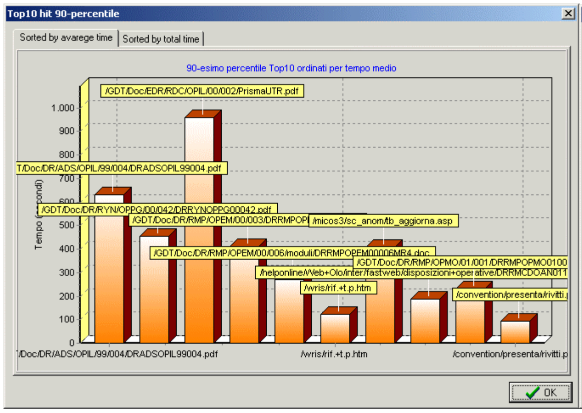
The following graphs provide an hourly view of the response times. In particular, they show how the hits are distributed during a period of 24 hours (a filter is also provided to restrict calculation to different time bands).

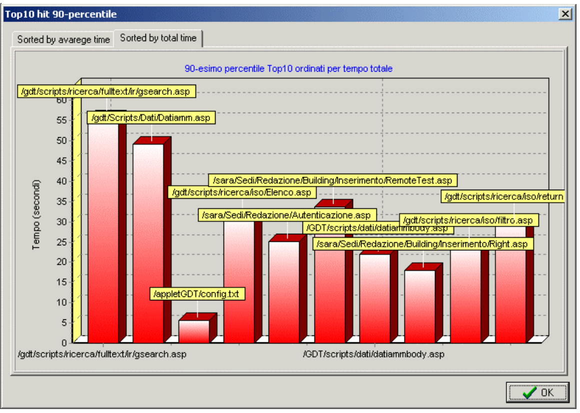


This graph can be correlated with that provided below which describes, also referring to the 0-23 time band, the trend of the medians and of the 90th percentile.



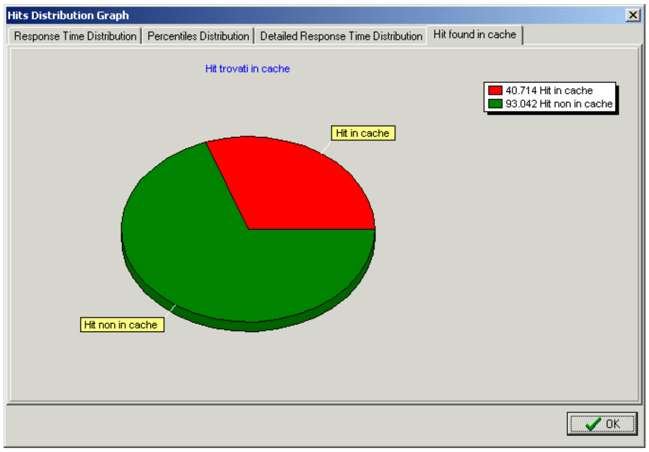
The following 2 graphs show the 90th percentile of the “Top 10” objects, sorting these both according to Average Time and by Total Time.





Percentage Hits found in the cache

The last graph shows the number and percentage of objects found in the cache. Using the filters on the objects, it is possible to verify whether the caching mechanism is working properly or whether action is required.



Conclusions

Web platforms continue to evolve, providing new capabilities, often to the detriment of performance. Success in e-business is based, amongst others, on performance and knowing how to manage, measure and forecast this is very important. Load testing, end-to-end monitoring, HW resource monitoring are the bases, the underpinnings of those who deal with performance. Performance problems can be solved, decreeing therefore the success of an e-business company, only through effective integration of these activities. Web Performance Log Analysis is a new activity, to be integrated with the others, for performance assessment, reinforcing the underlying idea of the complete approach, i.e. it is possible to provide effective, winning analyses only by integrating various information.

Credits

All the work presented in this paper is one of the results of the Telecom Italia Lab “Web Check-Up” program. Our thanks go to the Project Managers Enzo Corda and Gabriele Elia and all colleagues who, during the project, contributed to the paper with advice, suggestions, hints and reviews. A particular thank-you to Massimo Grando, of Telecom Italia, who contributed actively to defining the TILOG metrics.

Bibliography

[1] G. Cassone, G. Elia, D. Gotta, F. Mola, A. Pinnola – “Web Performance Testing and Measurement: a Complete Approach” – CMG ITALIA 2001 and CMG USA 2001 conference proceedings.

[2] R. Jain - “The Art of Computer System Performance Analysis”, John Wiley & Sons, 1991