STUDIA UNIV. BABEŞ-BOLYAI, INFORMATICA, VOLUME XLII, NUMBER 2, 1997

EVALUATING THE WEB SERVERS CAPABILITIES FOR MULTIMEDIA REQUESTS

SERGE MIGUET AND VASILE-MARIAN SCUTURICI*

Abstract: The Web has known these last few years an exponential development. This is due, among other causes, to the improvement of the performance of web servers. These performances are tested using benchmarking tools that evaluate several factors. The most studied performance factors for Web servers are: the maximum number of requests per second delivered by a server, and the quantity of data per second (throughput). In this paper we study the distribution in time of the data arrival. We also define two unavailability factors, that can be used for benchmarking purposes, and that gives a better measurement of the quality of a server response, in the case of multimedia requests. We present the unavailability factors obtained as results in a series of experiments with large MPEG files delivered by an Apache server on the Windows NT platform.

1. Introduction

The Web has known these last few years an exponential development resulting in an ever increasing solicitation of Web servers. These servers have to be highly efficient to be able to answer quickly to thousands of requests and to deliver megabytes of data at every second. In this framework is essential to be able to measure the absolute performance of Web servers that are tested using benchmarking tools that evaluate several factors. The most studied performance factors for Web servers are: the maximum number of requests per second delivered by a server and the quantity of data per second (throughput).

Also, the information found on the web is of very different natures: from the small gif images or HTML pages to large multimedia files. If the distribution in time of data arriving to the client is not so important for the small files, for the multimedia files this may be important, especially if the client wants to process the data as long as it arrives to its level.

In this paper we study the distribution in time of the data arrival and we define two *unavailability factors,* that can be used for benchmarking purposes, and that gives a better measurement of the quality of a server response, in the case of multimedia requests. We will consider the data that arrive as an answer to a multimedia file request.

---

* This work is supported in part by CS Technologies Informatiques
Received by the editors: November 21, 1998.
1991 *Mathematics Subject Classifications*: 68M20, 68P20.
CR Categories and Subject Descriptors: H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems - *evaluation/ methodology.*
Additional Key Words and Phrases: Web Servers and Services, Multimedia on the Internet

Beginning on the definition of the *unavailability factors*, we will study the performances of some Web servers for large multimedia files requests.

2. Web Servers Testing Methodology

The evaluation of Web server's performances may use operational data or laboratory testing (benchmarking). Analysis of operational data include analysis of active servers logs(e. g. [Arlitt-97]) and network monitoring, server operating system and server software (e.g. [McGrath-96a], [Almeida-96], [Mogul-95a]). The operational data analysis results are dependent upon too many variables to make comparisons between different configurations reliable.

Benchmarking is a laboratory testing procedure. It uses a predefined data set and measures the results returned by the system under examination. Actual benchmarking tools study the servers capacity in almost real conditions. There are four metrics most often used to measure the capacity of Web servers. These are (after [McGrath-96a], [Rubarth-96]):

1. *The connections served* or *requests made per second* is a measure of how many HTTP requests a server can manage within a given length of time;
2. *The throughput in bytes per second* is the measure of the maximum amount of data the server can send through all open connections during a given amount of time, the total number of bytes transferred per time unit;
3. *The response time* is a measure of how long it takes the server to serve a client request.;
4. *The number of errors per second* is the measure of how many HTTP requests were lost or not handled by a server.

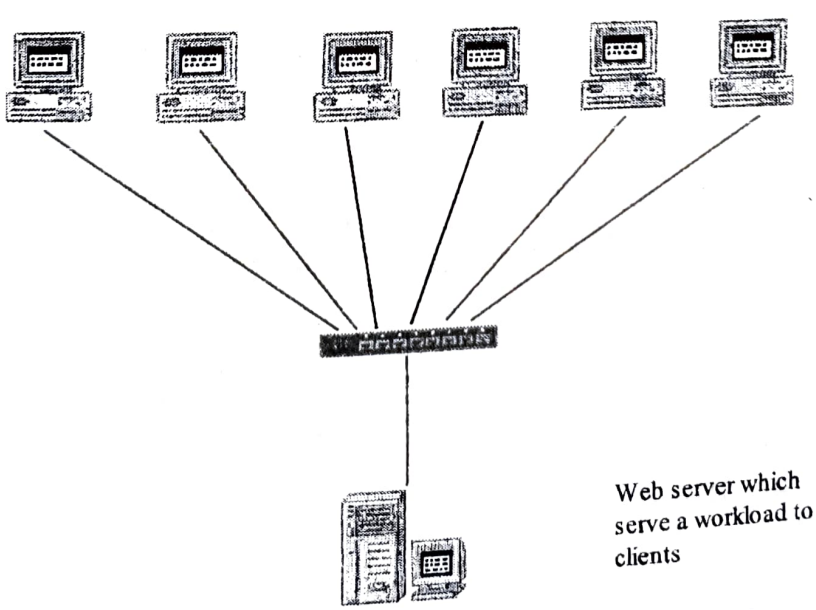

Figure 1 A simple benchmark configuration

A benchmarking instrument assumes, in general, that the Web server is installed on a computer to which one or more clients are connected. On the clients computers run processes part of the benchmark tool. These clients simulate Web browsers. The HTTP server has a predefined data set (named workload) ready to be served to the clients.

This data set is installed by the benchmarking tool using a statistically calculated file size distribution having as goal the correct representation of data found on the Web.
Client processes send requests to the server. The requests are either for static files found in the workload or a combination of static and dynamically created files (the latter are produced by CGI applications that run to produce the data that the server supplies to the client). When the server replies to a client request, the client records information such as how long it took for the server to respond and how much data it has returned and then sends a new request.
Client processes try to reach the maximal solicitation of the server. When the test ends, the benchmark tool calculates the overall server scores on the used hardware. The most used benchmarking tools are SPECWeb96, WebBench and WebStone. These tools all measure the maximum number of requests per second, but the last two additionally evaluate the maximum throughput in bytes per second (see [SPECWeb96], [WebBench], [WebStone]).

3. The classical factors in the case of multimedia requests

When a client requests a large file (e.g. a MPEG file) it might process the data during its reception. If the file has to be viewed at a certain theoretical throughput, we may expect that the throughput of the flow sent by the server to be the same or greater than the desired throughput (if the network allows it), to provide a good quality of visualization.

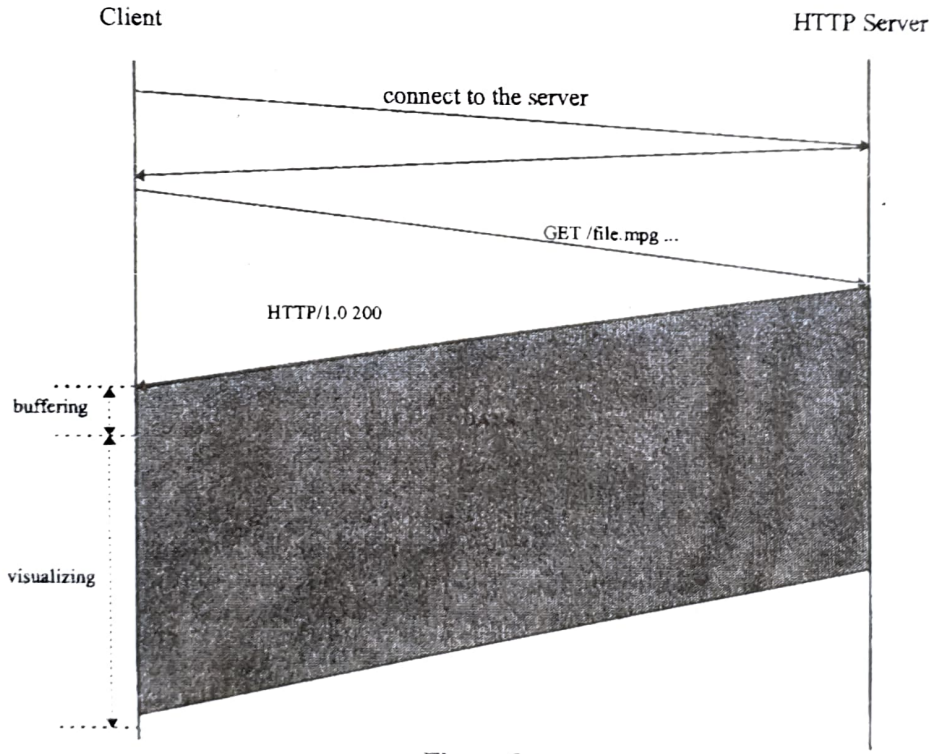

Figure 2

Assuming that the client wants to view the file "on the fly", this process will be like in Figure 2. For the best viewing quality (without interruptions or missing frames) the client must always have all the data necessary for the display process. Practically, the received data has to be available when it is needed to be displayed.

In the Figure 3 we have the representation of such a request. The continuous line represents the desired throughput expected by the client, and the dashed line represents the real throughput, obtained as a test result.

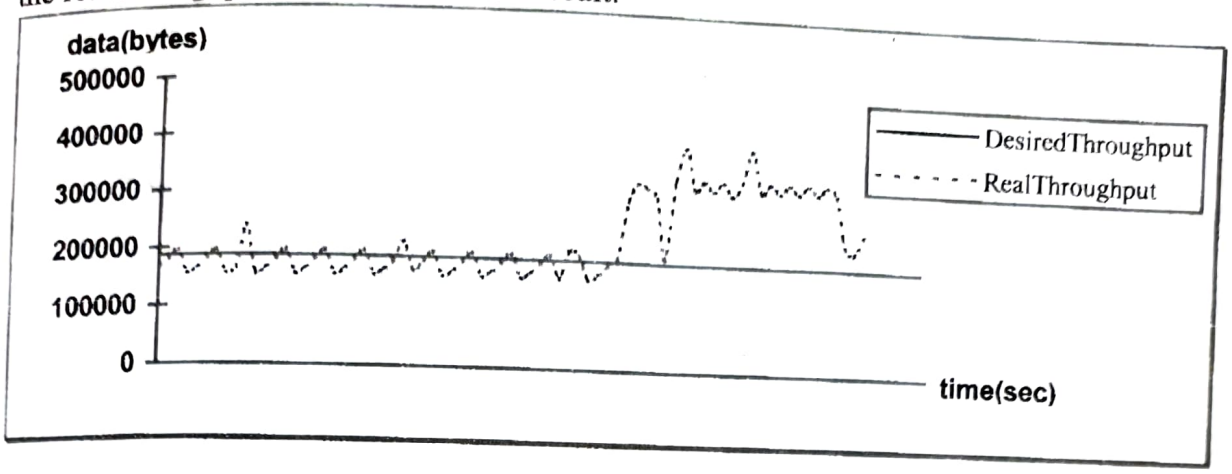

Figure 3 A multimedia file transfer example

In this example, assuming that the buffering time is smaller than the total transfer time, the data arrives at due time, globally, but, it is practically impossible to view in time the first part of the file, because the first approximately 50% of data arrives too late to be used on the fly.

With the presented metrics used in the benchmarking technology, we can not detect this problem. In the following section, we present two factors that measure the data availability quality for a multimedia request for a HTTP server.

4. Factors for measuring the data unavailability

We raise the problem of building a function that will approximate the throughput quality of a file sent by a server with respect to a constant throughput. This function will normally depend on a reference throughput and on the data that arrives late, as compared to the moment when it is needed (it will be called an unavailable data).
We will present two factors that give supplementary information about the file transfer process:

- *the average quantitative unavailability factor (quf)*: for a file, this is the percentage of bytes that arrive late (wrt the moment when they are needed)
- *the average temporal unavailability factor (tuf)*: for a file, this is a measure for the average delay of the bytes that arrive late (wrt the moment when they are needed)

A file F is a byte stream that has to be sent from the server to a client. Every byte θ arrives at the client at a moment t_θ, but it should be used at moment t'_θ. If $t_\theta \leq t'_\theta$ then the byte is not late and it has a null unavailability contribution. If $t'_\theta > t_\theta$, then the byte is late and we say that it has an unavailability contribution $t_\theta - t'_\theta$ (in seconds).

We can consider the D (the desired throughput) and R (the real throughput) as continue functions, $R,D:[0, size(F)] \rightarrow [0, +\infty)$.

Starting from this, we may define the average quantitative unavailability factor (*quf*) as:

$$quf(F,R,D) = card(\{\theta \in F,\ byte\ \theta\ arrived\ in\ time\ (R(\theta) \le D(\theta))\}) / size(F)$$

(Definition 1)

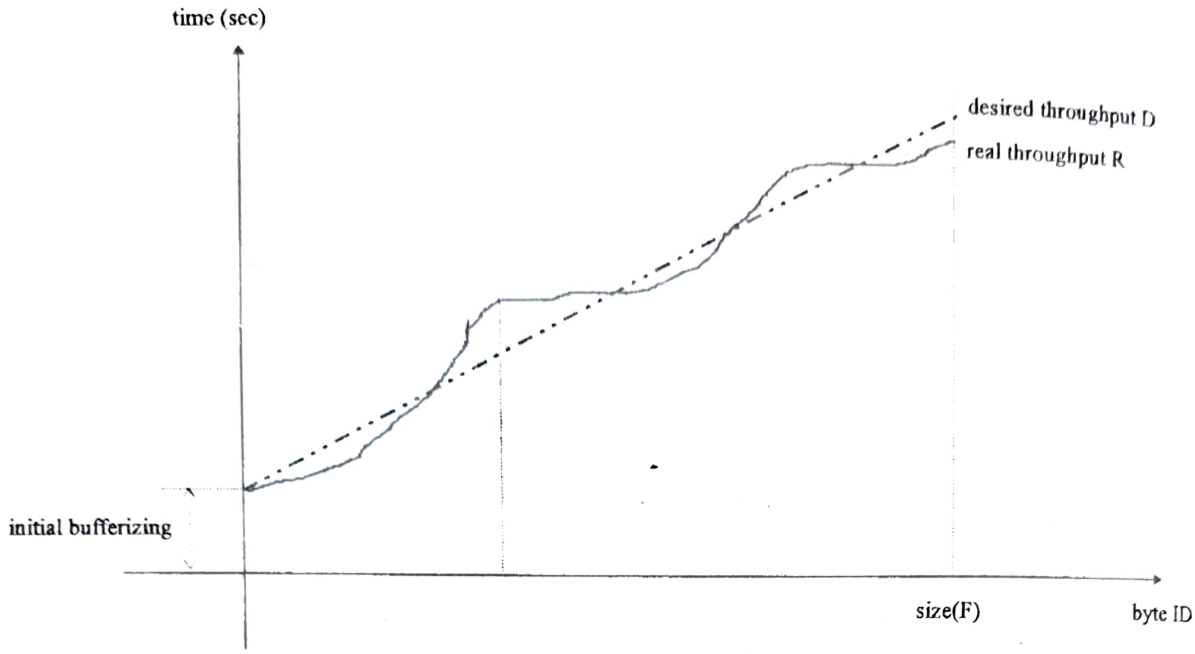

Figure 4 A multimedia file transfer example

and the average temporal unavailability factor(*tuf*):

$$tuf(F,R,D) = \left(\int_0^{size(F)} (R(\theta) - D(\theta)) * sign(R(\theta) - D(\theta)) d\theta \right) / (size(F))$$

(Definition 2)

where

$$sign(x) = \begin{cases} 0, if & x \le 0 \\ 1, if & x > 0 \end{cases}$$

Properties:

1. If *F1* and *F2* are two files to be transferred with the throughputs *D1* and *D2* respectively, then $tuf(F_1,R_1,D_1) < tuf(F_2,R_2,D_2)$ doesn't imply $quf(F_1,R_1,D_1) < quf(F_2,R_2,D_2)$. In other words, there are situations where $tuf(F_1,R_1,D_1) < tuf(F_2,R_2,D_2)$ and $quf(F_1,R_1,D_1) > quf(F_2,R_2,D_2)$ or $tuf(F_1,R_1,D_1) > tuf(F_2,R_2,D_2)$ and $quf(F_1,R_1,D_1) < quf(F_2,R_2,D_2)$ (see figure 8 and figure 9).
2. *quf* may be calculated at the moment when, theoretically, the file transfer has to finish. *tuf* will be exactly calculated only at the moment when the file transfer really ends. (Figure 5).
3. $quf(F,R,D) \in [0,1]$, $tuf(F,R,D) \in [0,\infty)$
4. $quf(F,R,D) = 0 \Leftrightarrow tuf(F,R,D) = 0$

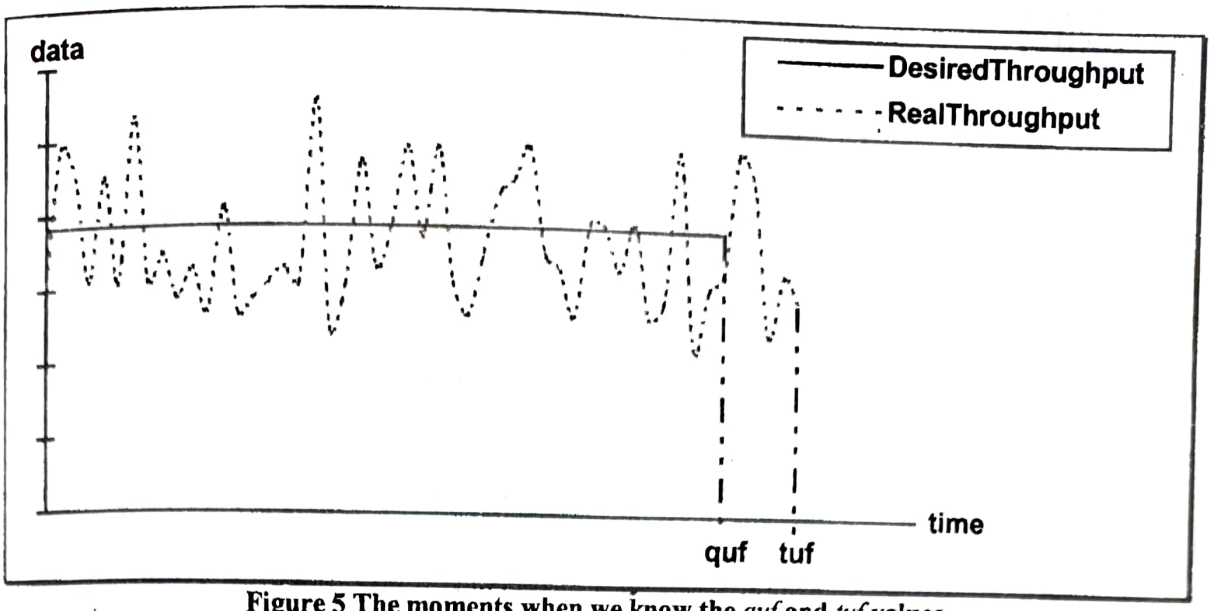

Figure 5 The moments when we know the *quf* and *tuf* values

Normally, the closer to zero these factors are, the better data availability will be. If *quf* = 0 (⇔ *tuf* = *0*, see obs.), there were no bytes that arrived late, so the data is received in time, when it is needed or faster. The bigger these factors are, the more bytes are late and the smaller the data availability will be.

Considering the results obtained for different files (see the next section), we will assert that a response to a multimedia request has an acceptable quality if *tuf* ∈ *[0, 0.1]* and *quf* ∈ *[0, 0.1]*.

5. Experimental results

We present the availability factors obtained as results in a series of experiments with large mpeg files delivered by an Apache server on the Windows NT platform.

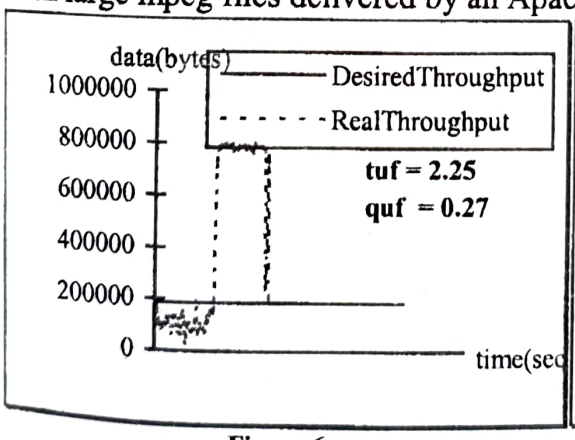

Figure 6

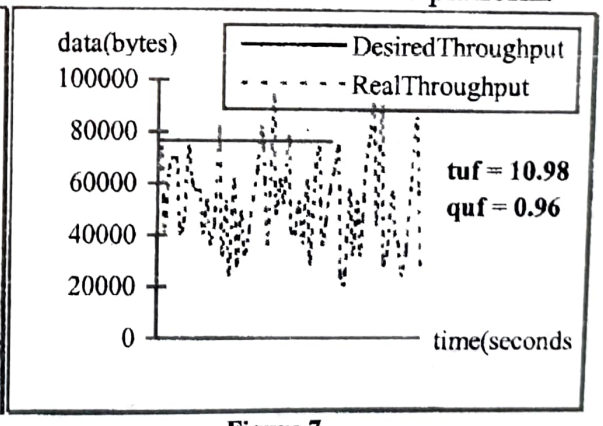

Figure 7

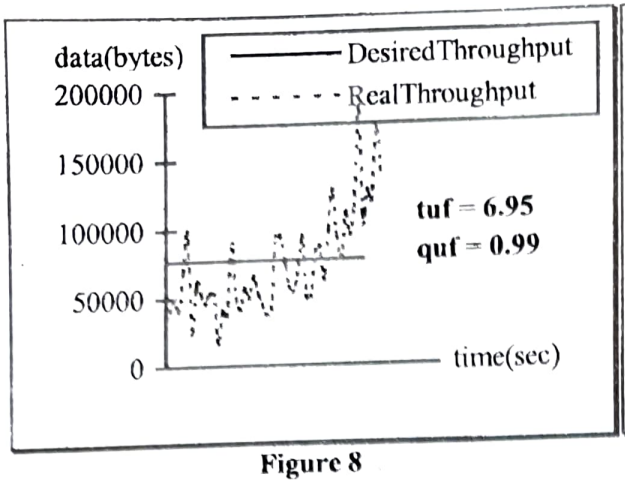

Figure 8

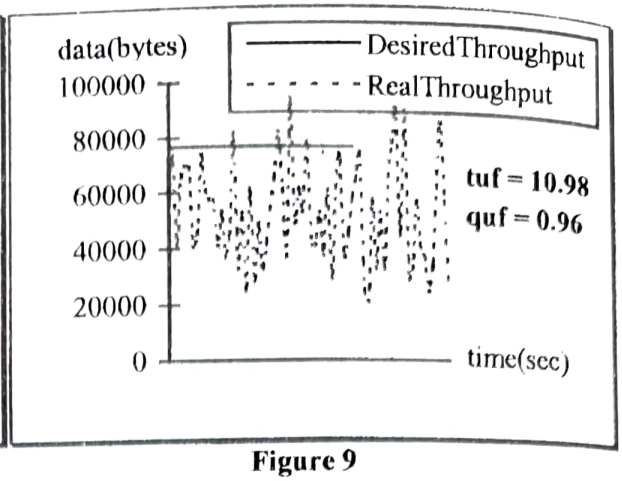

Figure 9

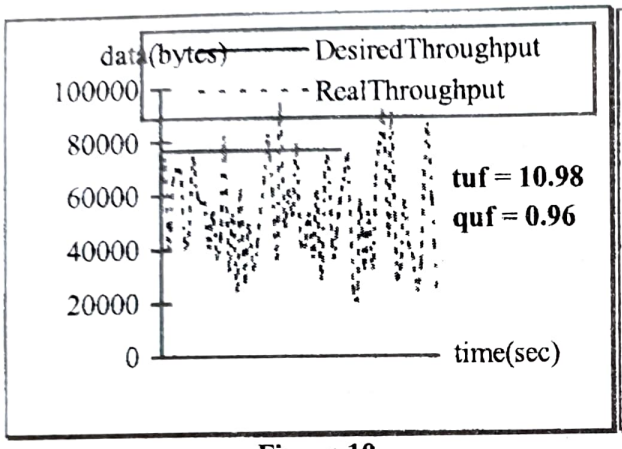

Figure 10

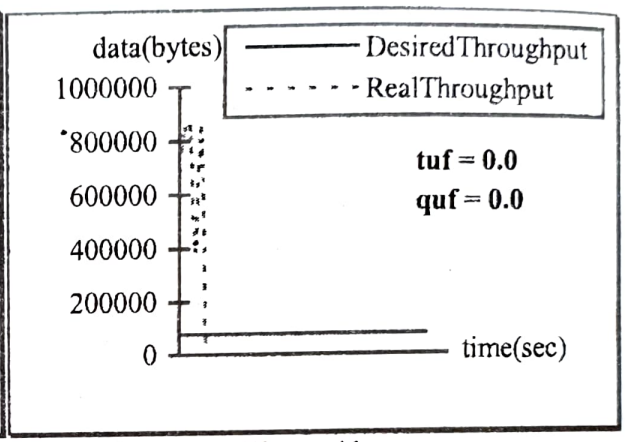

Figure 11

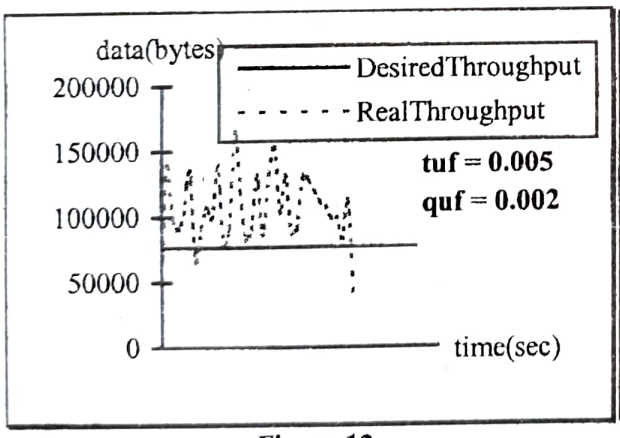

Figure 12

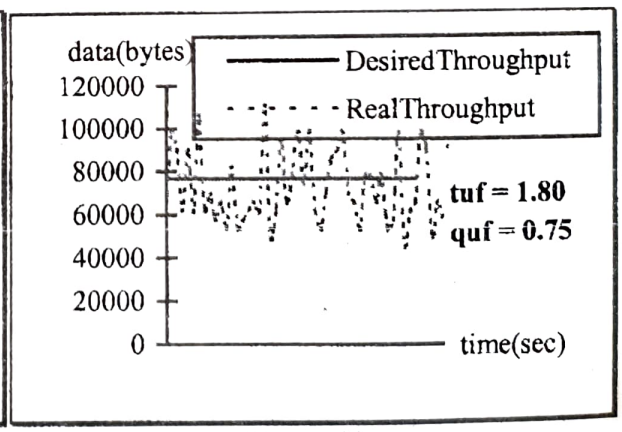

Figure 13

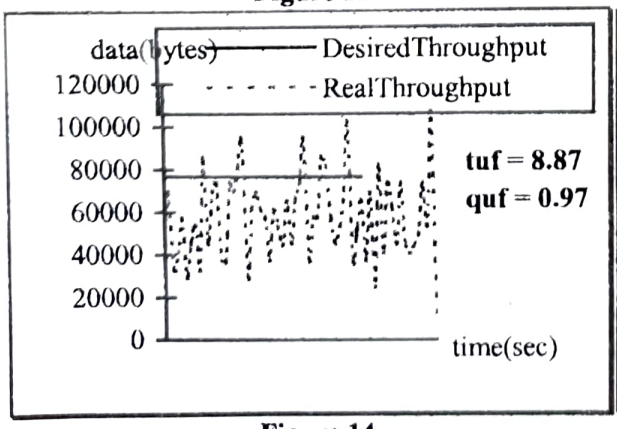

Figure 14

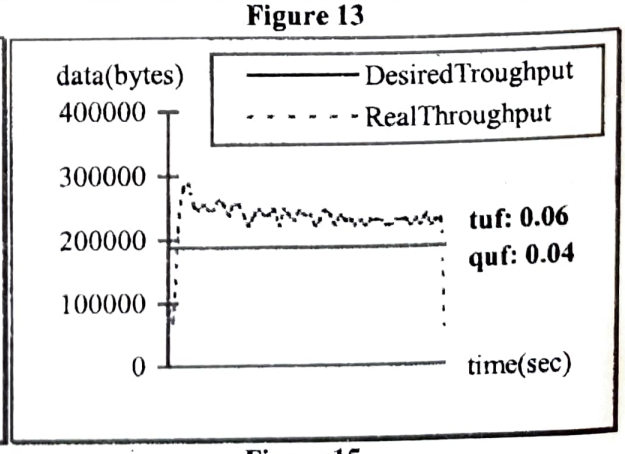

Figure 15

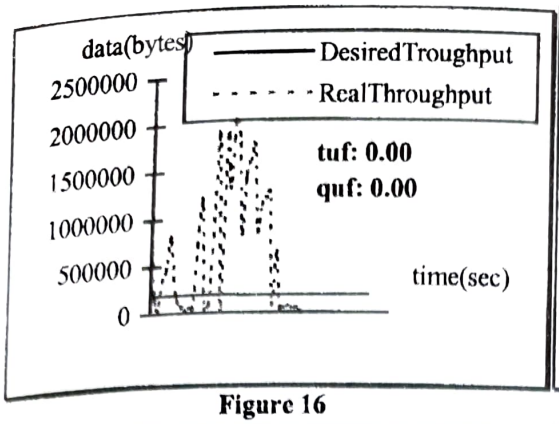

Figure 16

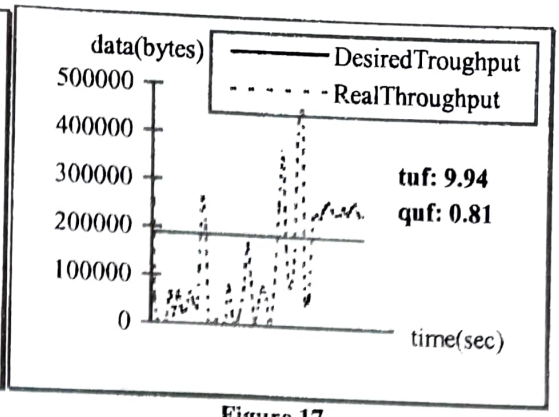

Figure 17

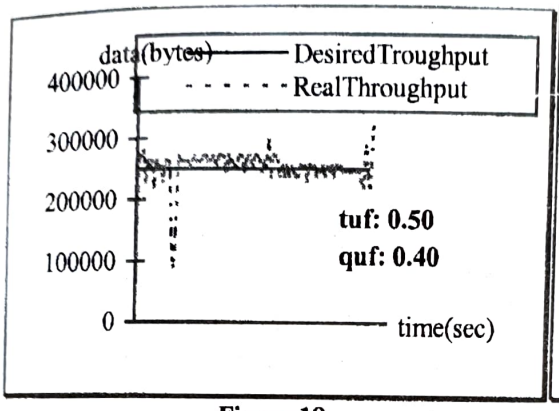

Figure 18

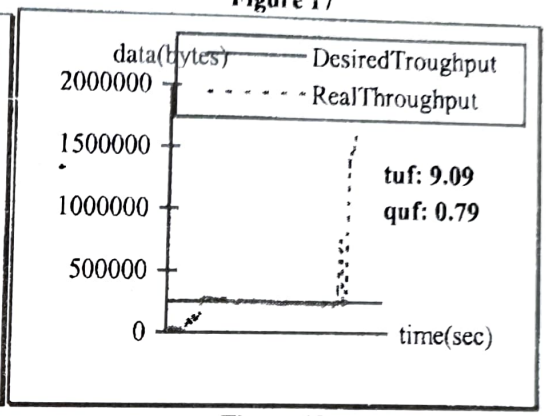

Figure 19

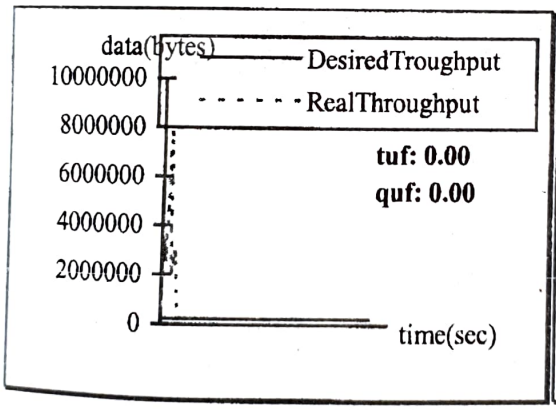

Figure 20

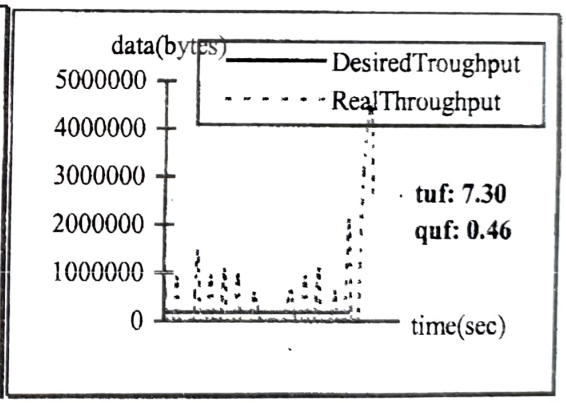

Figure 21

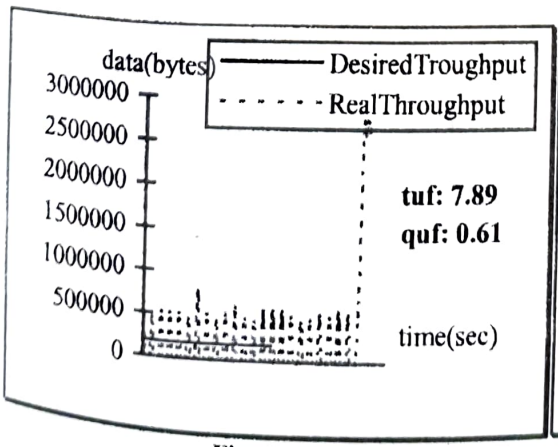

Figure 22

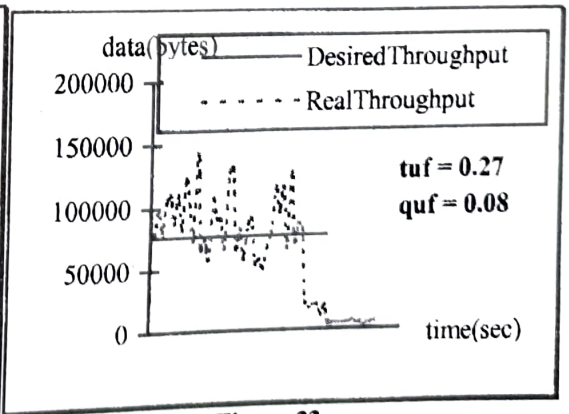

Figure 23

From the client's point of view, we consider as acceptable the following requests: Figure 11, Figure 12, Figure 15, Figure 16 and Figure 20.

The other requests (Figure 6, Figure 7, Figure 8, Figure 9, Figure 10, Figure 13, Figure 14, Figure 17, Figure 18, Figure 19, Figure 21, Figure 22, Figure 23) are not acceptable because many more data are not received in time (verified by *quf*) or the last part was transferred with problems (verified by *tuf*: see Figure 23).

6. Conclusion

Having this data availability quality defined before, the Web server testing may be extended taking in account the increasing requirements concerning them. So, a measure of the quality of these servers may be:

- maximum number of connections of a certain quality, served over the time unit;
- maximum quantity of data served over the time unit, to achieve a certain quality for the connections.

We presented the problems that a client may have when he wants to receive and process at the same time a multimedia file sent by a HTTP server. The fluctuations in throughput, as received by the client, are due to many factors: network, the concurrency management mechanism of the HTTP server, the performances of the server (hardware, SO, ...).

As a conclusion, we consider as useful the possibility of specifying at the server level a special category of requests (e. g. those that require a certain throughput). Such requests are the multimedia files requests for the HTTP servers on Internet or Intranet.

REFERENCES

[Almeida-96] Virgilio Almeida, Jussara M. Almeida, David D. Yates. Measuring the Behavior of a World-Wide Web Server. 1996, Technical Report CS 96-025

[Arlitt-97] Martin A. Arlitt, Carey L. Wiliamson, Internet Web Servers: Workload Characterization and Performance Implications. IEEE/ACM Transactions on Networking, vol. 5, no. 5, October 1997

[http1.1] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Berners-Lee, T., Hypertext Transfer Protocol-- HTTP/1.1, RFC-2068, available to ftp://ds.internic.net/rfc/ rfc2068.txt

[Hu-97] James Hu, Irfan Pyarali, and Douglas C. Schmidt. Measuring the Impact of Event Dispatching and Concurrency Models on Web Server Performance over High-speed Networks. Proceedings of the 2nd Global Internet Conference, IEEE, November 1997

[Hu-98] James Hu, Sumedh Mungee, and Douglas C. Schmidt. Principles for Developing and Measuring High-performance Web Servers over High Speed Networks. INFOCOM '98 Proceedings

[Kwan-95] Thomas T. Kwan, Robert E. McGrath, and Daniel A Reed, NCSA's World Wide Web Server: Design and Performance IEEE Computer , Volume 28, Number 11, November, 1995

[McGrath-94] Robert E. McGrath. Diagnosing Web Server Performance Problems. 1994. available to http://www.ncsa.uiuc.edu/InformationServers/Performance/Platforms/checklist.html

[McGrath-95a] Robert E. McGrath. Improving Performance of HTTP on Solaris Server Platforms. November 1995. available to http://www.ncsa.uiuc.edu/InformationServers/Performance/Solaris/rep3.html

[McGrath-96a] Robert E. McGrath. Measuring the Performance of HTTP Daemons. 5 February 1996. available to http://www.ncsa.uiuc.edu/InformationServers/Performance/Benchmarking/bench.html

[McGrath-96b] Robert E. McGrath. Performance of Several Web Server Platforms. 22 January 1996. http://www.ncsa.uiuc.edu/InformationServers/Performance/Platforms/report.html

[Mogul-95a] Mogul, J., Network Behavior of a Busy Web Server and its Clients Digital Equipment Corporation Western Research Lab Technical Report DEC WRL RR 95.5.

[Mogul-95b] Mogul, J., The Case for Persistent Connection HTTP. Proceedings of the 1995 SIGCOMM '95 Conference on Communications Architectures and Protocols.

[Nielsen -97] Nielsen, H., Gettys, J., Baird-Smith, A., Prud'hommeaux, E., Lie, H., Lilley, C., Network Performance Effects of HTTP/1.1, CSS1, and PNG. W3 Consortium. Note available at http:/ /www.w3.org/pub/WWW/Protocols/ HTTP/Performance/Pipeline.html

[Rubarth-96] James Rubarth-Lay. Keeping the 400lb. Gorilla at Bay: Optimizing Web Performance. 1996, available to http://eunuch.ddg.com/LIS/CyberHornsS96/j.rubarth-lay/PAPER.html

[Slothouber-95] Louis P. Slothouber. A Model of Web Server Performance. 1995. available to http://louvx.biap.com/webperformance/modelpaperhead.html

[SpecWeb96] SPECweb96 Benchmark, available to http://www.spec.org/osg/web96/

[Spero] Simon E Spero. Analysis of HTTP Performance problems. available to http://sunsite.unc.edu/mdma-release/http-prob.html

[Touch-96] Joe Touch, John Heidemann, and Katia Obraczka. Analysis of HTTP Performance. August 16, 1996, available to http://www.isi.edu/lsam/publications/http-perf/

[WebBench] WebBench benchmark program, available at http://www1.zdnet.com/zdbop/webbench/

[Yates-97] David J. Yates, Virgilio Almeida, Jussara M. Almeida. On the Interaction Between an Operating System and Web Server. Technical Report CS 97-012

[Yeager-96] Nancy J. Yeager and Robert E. McGrath. Web Server Technology: The advanced Guide for World Wide Web information Providers. Morgan Kaufmann, 1996

[WebStone] WebStone: The Standard Web Server Benchmark, available to http://www.mindcraft.com/webstone/

Laboratoire E.R.I.C., Universite Lyon 2, Lyon, France.
e-mail: miguet@univ-lyon2.fr, vscuturi@csti.fr