XII konferencja „Sieci i Systemy Informatyczne”
Łódź, październik 2004

GRZEGORZ HRYŃ
ZBIGNIEW JERZAK
Politechnika Śląska w Gliwicach, Instytut Informatyki
Dresden University of Technology, Systems Engineering Group

ADAPTIVE BANDWIDTH DRIVEN CONTENT DELIVERY FOR WWW CLIENTS

Summary

The user-perceived web server performance has a strong bearing on popularity of a portal, site or service as users may choose not to return to a particular web page if it takes a long time to retrieve its content. We present a new approach to how a server can identify incoming clients that suffer from a poor performance and how can it react in order to improve a user-perceived connection quality based on that identification.

1 Introduction

Most Web site administrators are trying to retain users by reducing the ”time to glass” factor – the delay between a request sent by a user and the subsequent delivery of the content and displaying in a user’s browser. Important fact is that from a user’s point of view it is irrelevant whether the high values of ”time-to-glass” are caused by a low bandwidth, high latency, network congestion or the delay on the path between the client and server. As performance issues may originate from different factors, the server cannot define them, and what is more important, eliminate them. Although these factors are not under the direct control of the server the service provider has a strong need to improve a method of a content delivery in order to deliver the most suitable content the quickest way possible to the given user.

This paper presents a new approach to how a server can identify incoming clients that suffer from a poor performance and how can it react in order to improve a user-perceived performance based on that identification. We continue our previous work [9] where we proposed a proxy server forwarding client’s request to a server storing a content that is appropriate to client’s connection capabilities, like bandwidth or delay. In this follow up we present the design and an implementation of a working prototype of a proxy server. Our prototype allows for an estimation of a client perceived performance by means of the following factors: the RTT (*round trip time*) and bandwidth estimations passively obtained at a server side.

The idea of our system is presented on Figure 1: Clients send their requests to a proxy server as they would do to a regular WWW server. Our system (consisting of *VisProxy* proxy server and *Network Monitor* module) performs client characterization and forwards

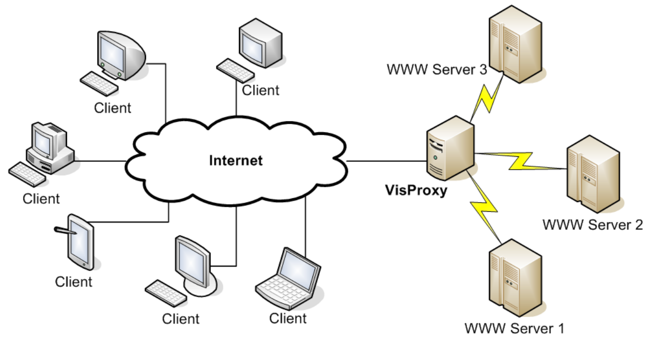

Fig. 1: VisProxy System Architecture - an overview

a client's request to a WWW server that stores content matching client's link capabilities in terms of volume. Web server's responses are in turn forwarded back, so that VisProxy appears transparent to a client requesting a resource.

2 Background and Related Work

Thee need for adaptive content delivery has been early recognized and diagnosed as an important and desired functionality of the WWW services [6]. The need to differentiate originates from the heterogeneous character of the Internet where many clients with different hardware and software characteristics use common service provider – in our case a WWW server. One of the most common approaches towards client characterization is offline bandwidth estimation based on the WWW server logs [10]. Other methods utilize clients piggybacking their connection and hardware characteristics with each request [18] sent to server. Such characteristics can be included in HTTP headers, or alternatively clients could store this information in a profile that is made available to a proxy server. Other approaches use explicit client self-identification by means of XML based templates [8]. There are also numerous attempts at estimating the link characteristics [5], [3], [2] using different methods (RTT, TTL), though they usually take advantage of active measurement [1]. There are also examples of passive estimation [7], however, they are aimed at the current bandwidth utilization instead of total available bandwidth for the given client. Finally there are also attempts at calculating the client response time as perceived by the server [14].

Our approach is novel in that it utilizes passive, server-side measurement methods which are transparent for the client and do not interfere with already existing traffic. We have implemented a dynamic link library in C++ which estimates the available bandwidth for a given connection by combined means of RTT and throughput calculation. Our solution for adaptive content delivery differs from the one proposed in [11] in that we utilize the online passive bandwidth estimation which allows us to take appropriate actions *before* client is served the first answer from the server. Such an approach allows us to adjust the server response to the very first request – thus the client always receives the content that matches his link capabilities. Moreover, due to the fact that the estimation is performed continuously we are able to react to sudden changes in the available client bandwidth which are quite common and unpredictable, if one considers for example the *Slashdot Effect* [19]. We are able to detect and react to bottlenecks that occur both on the client and the server side.

3 Adaptive Content Delivery

In order to evaluate the proposed approach we have implemented the VisProxy server along with the Network Monitor (NM) module. The VisProxy server has been implemented using C# whilst the NM module has been implemented as a dynamic link library using the C++ language. For NM module implementation we use the open source *WinPcap* libraries from Politecnico di Torino [4], [15]. We present system architecture on Figure 2. VisProxy similarly to a typical proxy server listens for incoming connections. NM module taps the incoming connections and gathers statistics for all the clients accessing the desired VisProxy port. Clients are recognized based on their IP address, which is obtained from the IP protocol header of the packets they send to the server.

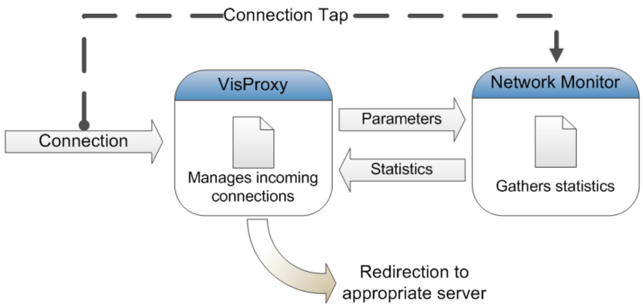

Fig. 2: VisProxy and Analyzer module interaction

Upon receiving a request VisProxy queries the NM in order to obtain the client's connection parameters. NM operation basically involves continuous gathering of the bandwidth and RTT samples. The clients that have not been classified yet (i.e. not yet seen by the NM) the first coarse characteristics is obtained by means of calculating the RTT from the three way handshake used in the TCP protocol. As the connection than progresses more accurate estimation based on bandwidth is possible. We compute the bandwidth as the median value of last 201 samples. Such a value proved not to be prone to temporary spikes and it reacted promptly to permanent changes in the bandwidth. Bandwidth sample is obtained as a product of division of the packet size(s) and the time that was needed for the client to confirm the reception of the data. This time is calculated as an interval between the sending of the first data packet[1] and the reception of the ACK packet from

[1] *Due to the possibility of occurrence of delayed ACKs we need to store the arrival time of the first data packet and sum of sizes of all received data packets that has not been confirmed yet. Upon reception of ACK packet we calculate the total*

the client confirming that it has correctly received the data. VisProxy server upon reception of new requests performs a query to the NM asking for the bandwidth value for the given client. If the client has not been previously classified by the NM module (there is no bandwidth estimation so far) the returned bandwidth value equals zero. In such case VisProxy queries the NM module again asking for the RTT based bandwidth estimation. Upon reception of the approximated value VisProxy assigns client to appropriate class and redirects his requests to a server containing content that matches client's capabilities in terms of volume. If the client has been already classified VisProxy receives the most actual bandwidth estimation and proceeds as described above. RTT estimation is based on the observation of arrival time of the ACK SYN acknowledgement sent by the server (in reply to the open session request) and the ACK sent back by a client in a final stage of three way handshake. As mentioned before this approach allows us to measure the RTT at the very beginning stage of client connection, which in turn lets redirect a client that has not been yet classified to an appropriate resource. This coarse measurement is further refined during the lifetime of connection (i.e. data exchange between server and client) by calculating the effective available bandwidth.

4 Evaluation and Testing

In order to evaluate our system we have performed two types of tests. First type were performance tests, which were responsible for measuring general performance of the system and overhead imposed by VisProxy. Second type were redirection tests which evaluated accuracy of classification under different network conditions.

4.1 Performance Tests

Performance tests aimed at stressing VisProxy with requests for resources of different sizes sent to a server using various connection rates in order to find out a maximum number of connections VisProxy can serve.

The first part of performance tests was conducted using *httperf* tool [13]. The test environment consisted of five test computers (load generators), one manager computer (which does not take part in the traffic generation) and VisProxy server connected to the WWW server. VisProxy was installed on a Dell Optiplex GX110 machine with PIII 866MHz processor and 384MB of RAM. The WWW server was installed on a Dell Optiplex GX270 with PIV 2,26GHz equipped with 256MB RAM. We have decided to split load generation in order to avoid the situation when a load generator becomes a bottleneck during the test. In order to automate the tests using *httperf* we have used the *autobench* program [12] that allows for simultaneous testing from multiple computers. We have also used *Perfmon*, a performance monitor tool, to monitor processor, memory and network related resources utilization during tests in order to identify origin of any bottlenecks occuring. Figure 3 demonstrates response rates and times for httperf-based tests, measured at the client side.

---

size of acknowledged data packets and knowing the arrival time of the first data packet and the ACK packet we can easily compute bandwidth.

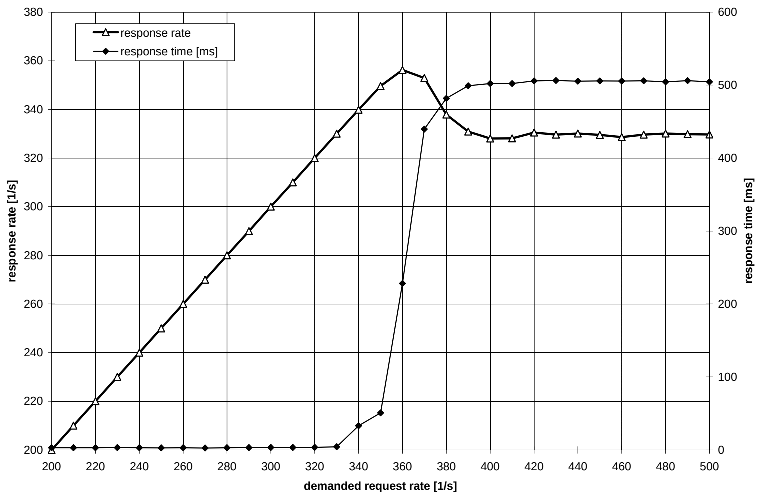

Fig. 3: Response rate and time for 10kB objects (10 requests per session)

In order to test the VisProxy's ability to recognize and classify the incoming clients (based on the IP address) according to their connection speed we have developed a tool, named Raw Packet Sender (RPS later in text). RPS allowed us to create TCP/IP and HTTP sessions to the VisProxy server from one physical machine which on the server side were regarded as coming from different computers with different IP addresses. The underlying technology bases on the solutions presented in the [4] and [15]. We could stress the VisProxy by simulating connections from thousands of network adapters with only one physical network card. RPS was designed to allow testing in two modes: constant rate and variable rate. Variable rate was implemented based on a Markov Modulated Poisson Process (MMPP), which is a Poisson process whose rate varies according to a Markov process. This is a natural model for point processes that combine irregular bursts of activity with predictable (e.g. daily and hourly) patterns, which can be found in web traffic [17]. MMPP model used in our system was developed basing on FIFA World Cup '98 server logs. Figure 4 presents RPS test performed at average rate of 200 sessions per second with MMPP option enabled. A measured average response time was 0.18s while median value was 0.13s for that test. RPS requested one HTML object with one image object embedded and total simulated IP count of 42000. The effect of burstiness and variability is clearly visible on that figure.

4.2 Redirection Tests

We have also conducted redirection tests which evaluated accuracy of classification under different network conditions. Our test environment consisted of a client computer connected to a VisProxy via Linux router (traffic shaper) with Dummynet installed. Dummynet

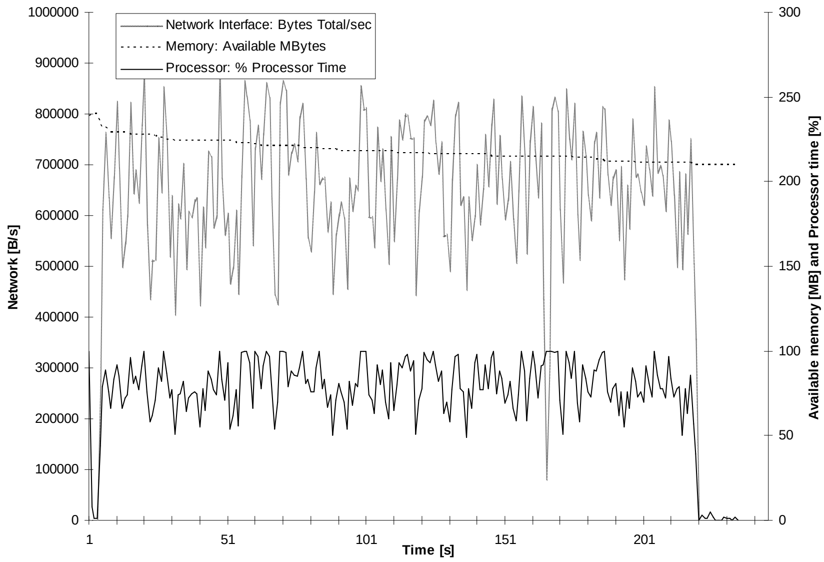

Fig. 4: VisProxy performance measurements

is a tool designed for testing networking protocols and for bandwidth management [16]. Dummynet works by intercepting communications of the protocol layer under test and simulating the effects of finite queues, bandwidth limitations and communication delays. It runs in a fully operational system, hence allowing the use of real traffic generators and protocol implementations. In order to verify classification accuracy, we have used *httperf* load generator requesting a complete web page (HTML plus embedded objects) at a constant rate. In the course of a test we were modifying network bandwidth, observing if the VisProxy reacts accurately to a changing network conditions[2] - see Figure 5. Redirection tests have shown that VisProxy responses properly to bandwidth change, redirecting the client to an appropriate resource. Due to the limitations of this publication we present only a small subset of the results.

5 Conclusion and Future Work

In this paper we have presented the design and an implementation of a working prototype of a proxy server that allows estimation of a client perceived performance by using two factors: the round trip time and bandwidth estimations passively obtained at a server side. Our system is server independent, making it possible to introduce the request handling modifications without altering the target platform and giving system administrator the

[2]*The disproportion in the values of bandwidth is caused by the way Dummynet shapes the traffic. It is important to note that the obtained measurements on the VisProxy side are proportional to changes at the traffic shaper, thus allowing for accurate classification.*

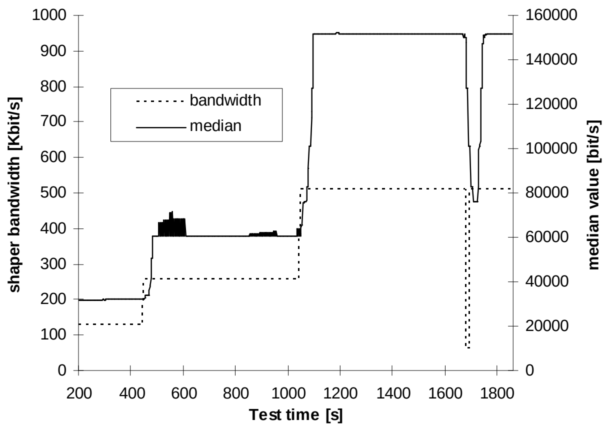

Fig. 5: Traffic shaper bandwidth adjustment and corresponding server side bandwidth median measurements

flexibility in choosing desired system architecture. Our approach assumes client classification being made "on thy fly" even if client IP does not exist in the client classification tables – in this case classification is made based on coarse RTT measurement before the redirection decision is made. Further redirection decisions for the previously classified clients are made based on more accurate bandwidth estimation being continuously measured during client's sessions. This approach allows fast, accurate and immediate client classification at minimal resources cost.

References

[1] Allman, M. Measuring end-to-end bulk transfer capacity. In *ACM SIGCOMM Internet Measurement Workshop 2001* (San Francisco, CA, USA, November 2001).

[2] Borzemski, L., and Nowak, Z. Estimation of http throughput and tcp round-trip times. In *Proceedings of the 10th Polish Teletraffic Symposium* (Cracow, Poland, September 2003), pp. 335–352.

[3] Borzemski, L., and Nowak, Z. Wing: A web probing, visualization, and performance analysis service. *Lecture Notes in Computer Science 3140/2004* (July 2004).

[4] Degioanni, L., Baldi, M., Risso, F., and Varenni, G. Profiling and optimization of software-based network-analysis applications. In *Proceedings of the 15th IEEE Symposium on Computer Architecture and High Performance Computing (SBAC-PAD 2003)* (November 2003).

[5] Downey, A. B. Using pathchar to estimate internet link characteristics. In *ACM SIGCOMM Measurement and Modeling of Computer Systems 1999* (Cambridge, MA, USA, September 1999), pp. 241–250.

[6] Fox, A., Gribble, S. D., Brewer, E. A., and Amir, E. Adapting to network and client variability via on-demand dynamic distillation. In *Proc. Seventh Intl. Conf. on Arch.*

Support for Prog. Lang. and Oper. Sys. (ASPLOS-VII) (Cambridge, MA, USA, 1996).

[7] Gerla, M., Ng, B. K. F., Sanadidi, M., Valla, M., and Wang, R. Tcp westwood with adaptive bandwidth estimation to improve efficiency/friendliness tradeoffs. *To appear in Computer Communication Journal* (2003).

[8] Gribble, S. D., Welsh, M., von Behren, R., Brewer, E. A., Culler, D., Borisov, N., Czerwinski, S., Gummadi, R., Hill, J., Joseph, A., Katz, R. H., Mao, Z. M., Ross, S., Zhao, B., and Holte, R. C. The ninja architecture for robust internet-scale systems and services. *Comput. Networks 35*, 4 (2001), 473–497.

[9] Hryń, G., and Jerzak, Z. Możliwości poprawy postrzeganej jakości połączenia klientów www z wykorzystaniem pośredniczącego serwera proxy. In *Współczesne problemy sieci komputerowych - zastosowanie i bezpieczeństwo* (2004), WNT, pp. 465–474.

[10] Krishnamurthy, B., and Wills, C. E. Improving web experience by client characterization driven server adaptation. In *Proceedings of the eleventh international conference on World Wide Web* (2002), ACM Press, pp. 305–316.

[11] Krishnamurthy, B., Zhang, Y., Wills, C. E., and Vishwanath, K. Design, implementation, and evaluation of a client characterization driven web server. In *Proceedings of the twelfth international conference on World Wide Web* (2003), ACM Press, pp. 138–147.

[12] Midgley, J. T. J. Autobench.

[13] Mosberger, D., and Jin, T. httperf—a tool for measuring web server performance. In *Proceedings of the First Workshop on Internet Server Performance* (June 1998).

[14] Olshefski, D. P., Nieh, J., and Agrawal, D. Inferring client response time at the web server. In *Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems* (2002), ACM Press, pp. 160–171.

[15] Risso, F., and Degioanni, L. An architecture for high performance network analysis. In *Proceedings of the 6th IEEE Symposium on Computers and Communications (ISCC 2001)* (July 2001).

[16] Rizzo, L. Dummynet: a simple approach to the evaluation of network protocols. *SIGCOMM Comput. Commun. Rev. 27*, 1 (1997), 31–41.

[17] Scott, S. L., and Smyth, P. The markov modulated poisson process and markov poisson cascade with applications to web traffic data. In *Bayesian Statistics 7* (2003), Oxford University Press, pp. 671–680.

[18] Singh, A., Trivedi, A., and Ramamritham, K. Ptc : Proxies that transcode and cache in heterogeneous web client environments. *World Wide Web Journal, Special Issue on WISE 2002 Papers 7* (2003).

[19] Wald, L. A., and Schwarz, S. The 1999 southern california seismic network bulletin. *Seismological Research Letters 71(4)* (July/August 2000).

GRZEGORZ HRYŃ
grzegorz.hryn@polsl.pl
Politechnika Śląska w Gliwicach
Instytut Informatyki
44-100 Gliwice, Akademicka 16

ZBIGNIEW JERZAK
zbigniew.jerzak@inf.tu-dresden.de
Dresden University of Technology
Systems Engineering Group
D-01062 Dresden, Dürerstr. 26
tel:[+49] (0)351 463 39711