A Measurement Based Memory Performance Evaluation of High Throughput Servers—Extended Abstract

Garba Isa Ya'u and Abdul Waheed

E-mail: {dygarba, awaheed}@ccse.kfupm.edu.sa

Phone: 966 3 860 1423

Fax: 966 3 860 2440

Computer Engineering Department

King Fahd University of Petroleum and Minerals

Dhahran 31261, Saudi Arabia

ABSTRACT

Performance of high throughput servers largely depends on memory issues, like cache, main and virtual memory, and disk. According to Moore's law, processor speed have been roughly doubling every eighteen months while memory and disk only get faster at about 10% per year[3]. Bottleneck in server performance has shifted from processors to memory and disk. Memory hierarchy performance is a limiting factor in the performance of high throughput servers. In this work, we are conducting measurement-based performance study of key high throughput servers namely: streaming media servers, web servers and web proxy servers, and routing server based on Linux kernel. Our aim is to identify where on-chip and memory becomes bottleneck in the performance of this high throughput servers. We conducted some initial experiments to substantiate on our claim that memory hierarchy is a limiting factor for these high throughput servers.

1. INTRODUCTION

Growing demands of the Internet requires high performance servers for such applications like World Wide Web and real-time multimedia applications. Web servers, streaming servers and proxy servers are essentially high throughput servers that normally serve a large number of clients. The continuous explosion of the Internet further makes demand on these servers very high and stringent; hence the performance of these servers must meet up with the demands in today's Internet applications and large number of clients.

Streaming servers play a key role in providing streaming services. To offer quality-streaming services, streaming servers are required to process multimedia data under timing constraints and support interactive control operations such as pause/resume, fast forward, and fast backward. Furthermore, streaming servers need to retrieve media components in a synchronous fashion. These servers deliver live or on-demand audio or video content to potentially thousands of clients distributed across the Internet. Because of the stringent timing and quality-of-service requirements, high-bandwidth demands, and the CPU and memory intensive characteristics of these applications, the performance of the server hardware is very critical for efficient performance and delivery of high quality multimedia contents.

Proxy servers are now highly in use and caching proxies have gained widespread popularity on the Internet. Both Web proxies and streaming media proxies are now widely deployed both on the global Internet and organizations' intranet. The proxies store frequently requested objects close to the clients in the hope of satisfying future client requests without contacting the origin server. Highly localized request patterns, which exhibit hot-spots, i.e., frequent requests for a small number of popular objects, have made caching highly successful in reducing server load, network

congestion, and client perceived latency [1]. While most of the caching research to date has focused on caching of textual and image objects, streaming proxies becomes increasingly popular. Caching streaming media objects with proxy servers poses many new challenges [2].

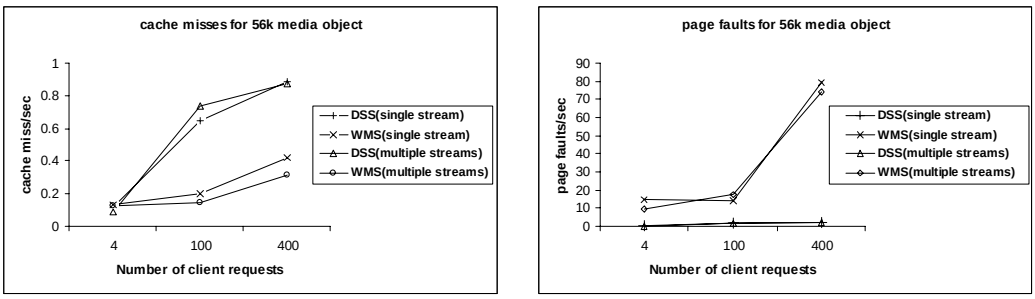
There has been tremendous progress in microprocessor technology that leads to high speed CPUs. Also, advances in memory and magnetic disk technology have significantly leads to improvement in memory density and magnetic disk density much more than access and cycle times. Density of semiconductor DRAM increases by 60% per year, quadrupling in three years, but cycle time has improved very slowly, decreasing by about one-third in 10 years. In a similar fashion, magnetic disk density has been improving by about 50% per year, almost quadrupling in three years. Access time has improved by only one-third in 10 years [3]. It is obvious that memory and disk performance can limit the performance of any busy high throughput server like streaming media server, Web server and proxy servers. In this work we are interested in the memory performance evaluation of high throughput servers to determine the specific conditions under which on-chip cache or memory becomes a major bottleneck on performance of the server. We are also interested in comparing the performance of different high throughput servers under identical workload conditions.

2. MEASUREMENT-BASED CASE STUDIES

In this section we report some initial results obtained from our measurement-based performance study of high throughput servers; streaming media servers, web and proxy servers and routing server. The results presented are preliminary as the work is in progress.

2.1 Streaming Media Servers

We setup measurement test bed with a dual boot server machine running windows 2000 and Linux Red Hat 7.2. We also have multiple client machines running windows 2000 and Linux Red Hat 7.2. All the machines are connected through a LAN with 100 Mbps 3Com ethernet switch. We used software tools; vmstat, sar, netstat, Windows 2000 performance and Intel Vtune performance analyzer. Our metrics comprises of on-chip cache misses, page faults, throughput and CPU utilization. To study effects of number of concurrent streams, encoding rate and stream distribution, we used three factors; number of streams (number of clients), encoding rate (56kbps and 300kbps) and stream distribution (single and multiple). We setup streaming media clients requesting media objects from the streaming media server to mimic real world clients. Experiments under same conditions were run for both Darwin streaming media server running on Linux Red Hat 7.2 and Windows media server running on Windows 2000 server. Figure 1 shows cache miss and page fault rates.



(a) (b)

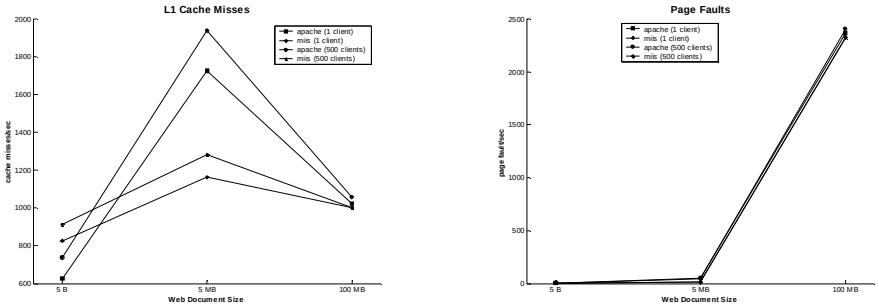
Figure 1: (a) cache miss rate for 56 kbps media (b) page faults rate for 56 kbps media

Figure 1 (a) shows that for both Darwin streaming server (DSS) and Windows media server (WMS), cache miss increases with the number of streams served to clients by

the server. Figure 1 (b) exhibits similar characteristics where the page fault rate increases with number of clients. Frequent cache misses and page faults like this can significantly affect the performance of the server.

2.2 Web Servers

Our performance test on Web servers utilizes the same test bed as the one outlined for streaming media servers as described in the previous sub-section. We used the industry standard web server performance benchmarking tool; Webstone. Webstone creates a load on a Web server by simulating the activity of multiple clients, which are called Web clients. For this study our metrics include cache miss, page faults, total server throughput (in terms of Mbit/sec delivered), total connection rate (connections/sec), average latency (delay) and CPU utilization. We also used the following factors: number of web clients and size of web objects. Like in the case of streaming media servers, we setup web clients requesting for web objects from the web server. Similar experiments were conducted for Apache server and Microsoft Internet Information server (MIIS). Figure 2 show cache misses and page faults behavior for the two servers.

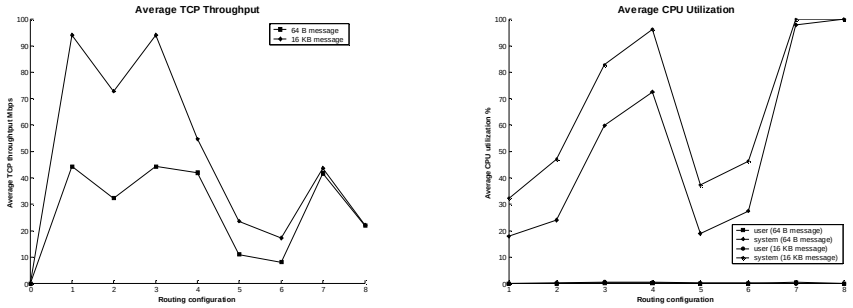


(a) (b)

Figure 2: (a) L1 cache misses (b) page fault rate

2.3 Routing Servers

We also carried measurement-based performance study of Linux routing server: using Linux operating system as a router. We setup a Pentium IV system with five PCI 100 Mbps ethernet network card to serve as router port. We connected five PCs serving as routing clients sending packets to be routed to various destinations. We used routed (Linux routing daemon) to dynamically create and maintain the routing table. At the client end, we used netperf [4] to send packets to the routing server for routing to its destination. We used various configurations to study the performance of Linux router. Our metrics include TCP connection throughput, CPU utilization and number of context switching. Figure 3 shows some of our initial results on throughput and CPU utilization.



(a) (b)

Figure 3: (a) Average TCP stream throughput (b) CPU utilization

2.4 Proxy Servers

We are presently setting up our measurement-based experiment to evaluate the memory performance of Microsoft proxy server.

3. CONCLUSIONS

From our initial experimental results, it is obvious that cache, memory and disk bottleneck could be key factor affecting the performance of high throughput servers. We are setting up

more experiments to have an in-depth study on the memory and disk performance of streaming media servers, web servers, proxy servers and Linux routing server. Currently, we observe that as this servers serve more clients, memory performance degrades due to frequent cache misses and page faults, the penalty of which is several CPU cycles of significant latency.

REFERENCES

[1] M. Reisslein, F. Hartanto and K. Ross, "Interactive Video Streaming with Proxy Servers," INFOCOM 2000

[2] S. Sahu, P. Shenoy, and D. Towsley, "Design Considerations for Integrated Proxy Servers," in Proc. of International Workshop on Network and Operating Systems Support for Digital Audio and Video, Basking Ridge, NJ, June 1999.

[3] J. L. Hennessy and D. A Patterson, "Computer Architecture: A Quantitative Approach," Morgan Kaufmann Publishers, Inc., 1996.

[4] "Netperf: A Network Performance Benchmark", Information Networks Division Hewlett-Packard Company, February 1996.