2012 10th IEEE International Symposium on Parallel and Distributed Processing with Applications

Using low-Power Embedded Microcontrollers as Web Servers

Navid Mohaghegh, and Mokhtar Aboelaze
Department of Computer Science and Engineering,
York University,
Toronto, ON, Canada

Abstract – *Power consumption of server farms is becoming a huge issue in the server operation community. The cost of powering the farm (both to operate servers and to cool them) is surpassing the amortized server cost. In this research, we investigate the use of low-power embedded controllers as servers. We built a small cluster of 4 embedded controller boards and compared its performance to a more traditional (and more powerful) server. We report on our findings with respect to the system configuration and performance.*

1 Introduction

Lrge server farms are powered by powerful servers that receive users' requests, process them and send back the response. One of the major problems facing providers is how to condition the servers in order to produce the agreed-upon quality of service (QoS) and at the same time minimize their cost.

Reducing energy consumption of processors or complete systems has been of a long time interest to mobile and wireless system designers to extend battery life of mobile devices. However, recently energy has become a major problem for large data centers and server farms. The cost of energy consumption does not only include the price of the electricity to drive the system, but also includes the price of cooling the system as well. With today's very powerful servers, small form factor, and tight packing the heat density is a major problem. Powerful and expensive cooling systems are required in order to avoid reliability problems.

It is estimated that the total cost of ownership of a rack in a data center is approximately $120K over the data center lifetime. In 2010, TelTub Inc. paid monthly fees of $275 CAD for each 15A circuit breaker and $850 for rental space and cooling of a standard rack mount (standard racks are 6.5 feet high with capacity of 10 4U servers per rack). Each 2MB/s bandwidth costs $50 CAD monthly. This will lead to a yearly cost of $20K for a standard rack ($120K if the average lifetime of a server is estimated to be 6 years) [1].

To address the energy crisis Google built one of their data centers in Oregon next to a power generation station (The Dalles, OR) [2]. According to Google, it has the right combination of energy infrastructure. Clearly there is a need to lower the energy demands of servers especially in large server farms and data centers

Traditional energy-saving techniques like DVFS (Dynamic Voltage and Frequency Scaling) and server shut off do not work as expected for large servers, due to the following reasons: First, for complete server shutoff, it takes a long time to bring the server back on-line. Without almost a perfect prediction of the workload; it is difficult to get a noticeable reduction in energy without performance degradation (web traffic is extremely volatile). Second, voltage scaling has a minimum effect since modern processors work near their minimum voltage anyway. Third even if we can reduce the energy consumption at light load, the infrastructure required for cooling is conditioned at the maximum load anyway. Fourth, DVFS can reduce the energy consumption in CPUs but it can not address the energy consumption in other parts of the system (disks, RAM, …). We need a new approach to energy saving that includes the entire system not only the CPUs .

In this research we investigate the possibility of using a cluster of low power small embedded microcontrollers as a web server. We build a small cluster (4 nodes) of low power embedded microcontrollers and configure them as a web server. We compare the performance (both power consumption and response time) to a more powerful web server. Our preliminary results show that we can reduce energy by a factor of 80% while increasing the response time by 60%.

2 Motivation

In this research, we propose a green, cheap and reusable embedded hardware server accelerator, clustered together with free and open-source software to reduce the power consumption of internet server farms. Our idea is based on the fact that throughput is much more important for web servers than response time (within limit of course). The user might not notice the difference between serving his/her request (at the server level) in 100 msec, or 900 msec. However, there is a lot of difference between the server being able to serve 100 or 900 request per second (The same factor of 9).

Using small, low-power embedded microcontrollers will result in almost a factor of 7-9 slowdown in both response time and throughput. However the power reduction is much more than that (a factor of 10-20 in our implementation). By using a cluster of low-power embedded microcontrollers, we can make up the difference on the throughput side, with a considerable energy saving.

978-0-7695-4701-5/12 $26.00 © 2012 IEEE
DOI 10.1109/ISPA.2012.137

Our proposed system can be used in two different scenarios. First, it could be sued to replace a server in a server farm. Thus maintaining the same throughput (albeit a slower response time) at a reduced power consumption level. It also could be used a standby server to added to the farm if the load exceeds a specific throughput. Our small controllers can load Linux in less than a seconds, a time that is much less than the start up time of big powerful servers.

3 System Architecture

. In our system, we used TS-7800 [3] a Single Board Computer, which fully utilizes Marvell® 88F5182 ARM architecture with maximum power consumption of 6 W @ 5 v. TS-7800 SBC is a small fan-less embedded device providing 128 MB DDR-RAM and 12k LUT programmable FPGA for further cryptographic engine implementation. This system boots Linux in 700 ms from internal flash allowing fast switching from sleep mode which uses only 200 mA.. A TS7800 board is used for modeling and proof of concept of accelerators.

We have already built a system of 4 TS-7800 boards [4]. Our system is composed of the 4 boards, a gigabit switch, a load balancer (to distribute incoming requests to the 4 boards), and a set of 3 laptops to generate traffic for testing. In order to compare it with a single powerful server, we used a powerful 6-core AMD Phenom clocked at 3.7 GHz with 16 GB of RAM (clocked at 2.0 GHz). Table 1 show a sample of our results, full results can be found in [4].

The first two rows in Table 1 show the response time of a single microcontroller and compares it with the AMD server. The test was performed using SpecWeb2009 [5]. The Table show a slowdown of 4-7 compared with the AMD server (4 for a static page and 7 for dynamic page).

Then we tested our 4-node cluster using Apache AB tool [6] since we could not configure SpecWeb2009 to test a system behind a fire wall. In our experiment, we maintained an average CPU utilization of 40-60% (typical for web servers). It is implied that the throughput is the same since we use the same input traffic for both systems. Under these conditions, the average response time for a static page is 97 ms compared with 14 ms for the AMD server, while for dynamic page the average response time was 700 ms for the AMD vs. 90 ms for the 4-node cluster.

Table 1: A comparison between a 4-node embedded cluster and a powerful server.

	AMD-6	TS7800
No-load Static	1.1 ms	4.3 ms
No-Load Dynamic	1.01 ms	7.6 ms
Static	14.5 ms	97.0 ms
Dynamic	91.4 ms	706.2 ms
Power	500W	21 W

From Table 1, we can see that the slowdown in the response time is 4-7 under no load, and 7-8 for a loaded system, while the reduction in energy consumption is 25.

The next step is to compare the throughput and the effect of the admission control proposed in [7]. Our goal here is to increase the traffic until either the response time is unacceptable or the percentage of the packets suffering unacceptable delay.

Table 2 shows the result of our experiment using the admission protocol proposed in [7]. The first column is the number of requests per second sent to the system. The second column is the CPU utilization. The third column is the number of microcontrollers used in the system. The last three columns represent the average response time, the maximum response time, and the percentage of the packets suffered delay more than 200msec. Note that when the number of controllers is one, we did not use any admission control. The results show almost a linear increase in the throughput with the number of microcontrollers. However, we used only 4 microcontrollers and a simple testing strategy.

Table 2: The effect of increasing traffic on average response time

Requests/sec	CPU	#	$T_{average}$	T_{max}	T>200ms
5000	45%	1	20ms	317ms	<1%
10000	90%	1	34ms	3542ms	>5%
10000	45%	2	20ms	317ms	1%
20000	44%	4	21ms	320ms	1%

Currently, we are in the process of expanding our system to 10-12 boards (that will make it comparable to the AMD server), also we will configure our system as a 3-tier server and tested under more realistic workload.

4 References

[1] TelTub Inc. "TelTub Weblog at blog.teltub.com," Last Checked on June 2011; http://www.teltub.com.

[2] Tippit. Inc. "The Google Datacenter in Oregon," Last Checked on June 2011; http://www.itmanagement.com/features/googles-oregon-datacenter_110107.

[3] Technologic Systems. "EmbeddedArm TS-7800 SBC," Last Checked on June 2011; http://www.embeddedarm.com/products/board-detail.php?product=TS-7800.

[4] N. Mohaghegh "A green cluster of low-power embedded hardware server accelerator". M.Sc. thesis, Dept, of Computer Science and Engineering. York University Sept. 2011.

[5] Standard Performance Evaluation Corp. "SpecWeb2009 – Benchmark for Evaluating server performance" www.spec.org checked March 2012.

[6] Apache Software Foundation "Apache HTTP server benchmarking tool" located at http://httpd.apache.org/docs/2.0/programs/ab.html Checked March 2012.

[7] M. Ghazy, N. Mohaghegh, M. Aboelaze "Controlling the response time of a web server". Proceedings of the International Conference on Internet Computing ICOPM2011. Las Vegas, NV. July 2011.