Sun Fire Gigabit Performance Characterization

Jian Huang, Shih-Hao Hung, Gian-Paolo Musumeci, Miroslav Klivansky, and Keng-tai Ko

Performance and Availability Engineering

Sun Microsystems

901 San Antonio Road

Palo Alto, Ca 94303

jianh,hungsh,gdm,miroslav,ko@eng.sun.com

Abstract

The recent network-centric computing has been exercising tremendous pressure on servers' network performance. With the launch of Sun's new midframe Sun Fire server, and the increasing popularity of gigabit ethernet, the question of how Sun's new servers perform in the gigabit ethernet arena has become one of the most important issues that our engineering team is trying to address. This paper presents an overview of our Sun Fire Servers' performance using Sun's gigabit ethernet products in terms of TCP/IP networking.

In order to evaluate a server's TCP/IP networking capability, both a systematic evaluation methodology and a well-structured benchmark set are needed. In this paper, we use the NetFrame network evaluation framework developed in the Performance and Availability Engineering (PAE) group to examine the Sun Fire product line. In particular, we studied the scalability of network throughput and packet rate vs. the number of CPUs, the number of TCP connections, the number of gigabit ethernet cards, and the number of PCI buses. The maximum observed throughputs and packet rates for a single connection, a single CPU, a single network interface card, and a single system I/O controller are revealed. Some TCP/IP tuning tips for the Sun Fire servers are disclosed. In addition, comparisons with the existing Sun Enterprise servers are also presented.

In order to explain the system behavior, more in-depth analysis is done to characterize the resource consumption by TCP/IP network modules at the per packet basis. Studies are also done on how the system load changes when different throughput and packet rate requirements are presented to a system.

1   Introduction

Being the largest UNIX server vendor in the World, Sun has been providing the ultimate computing solutions to corporate customers in all areas. Sun servers have been used extensively in the net-economy and are powering popular sites such as Ebay and AOL, etc. Requests from HTTP clients, database clients, mail clients, directory-query clients, and other network service clients exert a great pressure on Sun servers through the attached network. The responses from the server also go out through network interfaces. This client-server model brings the networking capability of Sun servers to the spot. The performance and availability of network services is essential for Sun's success.

Current popular network interface cards (NIC's) include the HME©and QFE©cards. These interfaces are only capable of sending and receiving at the 100 MegaBit Per Second (Mbps) range, which is not sufficient to saturate any of the PCI buses in a system. However, the newer and faster GibaBit Ethernet (GBE) interface cards are gaining momentum, especially when the current category-5 copper cables can now replace the more expensive fiber cables to carry the gigabit traffic. While Sun servers are known to

be performing very well in the 100Mbps range, it remains a mystery how Sun servers perform in the new gigabit ethernet arena.

Recently, Sun started shipping the new Sun Fire midframe servers based on the UltraSparc III microprocessor. The UltraSparc III processors run at a higher frequency (750MHz and higher), with improved cache and translation look-aside buffer (TLB) structures. The Sun Fire product line also incorporates many features that were only available in the mainframe computers, e.g., dynamic reconfiguration and multi-domain computing. Hence, the customers would be interested to know how our new product line performs in the aspect of networking.

This paper tries to address the above two questions in one attempt. In order to study the network performance effectively, we developed a systematic evaluation mechanism with a comprehensive set of tools. This systematic study was conducted on the Sun Fire 6800 platform using Sun's gigabit ethernet products. Performance numbers at a single-connection, single-card, single I/O controller, and single I/O boat levels are presented. We also reveal how Sun servers' network performance scale with the number of CPUs and the number of NIC's.

The rest of the paper is organized as follows: Section 2 introduces the Sun Fire product line and Section 3 describes Sun's available gigabit ethernet products. Section 4 presents our systematic network evaluation methodology called NetFrame. Section 5 discusses the performance data while Section 6 concludes the study.

2   Sun Fire Product Line

The new Sun Fire product line was launched in March, 2001. This product line includes the Sun Fire 3800, 4800, 4810, and 6800 midframe servers. The architecture is based on the UltraSparc-III processors running at 750MHz and higher. Sun Fire servers offer greater performance, reliability, availability, and serviceability (RAS) compared to the existing Enterprise Server product line.

2.1   Advantages of the Sun Fire Midframe Servers

In terms of performance, Sun Fire servers has the following improvements:

- Faster microprocessors. The UltraSparc III processor runs at a frequency of 750MHz or higher with larger cache (32KB level-1 instruction cache and 64KB level-1 data cache), and translation look-aside buffers (128-entry instruction TLB, and 512-entry data TLB).

- Wider bandwidth in system interconnect. The Sun Fireplane interconnect offers 9.6GB/second in sustained bandwidth and 24GB/second in aggregated bandwidth. Each Input/Output Assembly (IO Assembly) offers 1.2GB/second sustained bandwidth. Data can be moved in a larger batch with greater speed.

In addition to the performance enhancements, the Sun Fire servers offer the convenient RAS features as follows:

1. System interconnect segments. The Sun Fire servers have one interconnect segment by default, but can be configured into two segments. When the sytem is split into two segments, it logically behaves as if it were two separate systems with their own private system interconnect. Connections between the boards of one segment and the boards of the other segment are disabled. The benefit of system interconnect segment is that faults occurring in one segment do not directly impact applications running on the other segment.

2. Multiple system domains. A single Sun Fire server can have up to 4 logical domains. Different instances of Solaris operating system run on each domain independently, presenting an image of multiple independent machines.

3. Dynamic configuration. Dynamic Reconfiguration (DR) is a feature of the Sun Fire servers. It will be supported at general availability later.

DR is the ability to alter the configuration of a running system by bringing components online or taking them offline without disrupting system operation or requiring a system reboot.

4. Hot-pluggable I/O devices. The Sun Fire 3800 and 4810 servers support Compact PCI (cPCI) devices that can be serviced while the system is running, which reduces the system down time due to maintenance.

5. Capacity on demand (COD). The COD program enables customers to purchase a small Sun Fire configuration and increase the resources of their configuration as their needs grow.

2.2 A Closer Look at I/O Assembly

We use a Sun Fire 6800 (F6800) for the experiments. The F6800 can have up to 4 I/O assemblies. As shown in Figure 1, Each I/O assembly has 8 64-bit PCI slots, controlled by two PCI controllers (Schizos). Each Schizo controls 2 PCI buses – a 66MHz PCI bus and a 33Mhz PCI bus. There is one PCI slot that runs at 66MHz and 3 slots that share the 33MHz bus. Hence, the maximal number of cards an I/O assembly can support is 8, with 2 66MHz cards, and 6 33MHz cards. Theoretically, each 64-bit 66MHz PCI bus can support up to approximately 500MB/second, or 4Gigabit/second. A 64-bit 33MHz bus supports up to 2Gigabit/second. Hence, two schizos can bring about 1.5GB/second I/O traffic in theory. However, due to other factors like PCI bus overhead, the actual traffic the two schizos can generate is below the 1.2GB/second bandwidth supported by each I/O assembly, which excludes any possibility of bandwidth mismatch in the hardware.

3 Sun's Gigabit Ethernet Offerings

Sun offers two lines of gigabit ethernet products for Sbus and PCI bus. The GigabitEthernet/P 2.0 adapter, shown as "ge" by the *ifconfig* command, has both an Sbus and a PCI bus versions. The *ge* cards

Figure 1: An IO Assembly of Sun Fire 6800.

require fiber optic cable as the carrier. The newly released Gigabit Ethernet Adapter 3.0 (also called the GigaSwift Ethernet adapter), shown as "ce" by the *ifconfig* command, can use the Unshielded Twisted Pair (UTP) category-5 copper cables as media, which greatly reduces the cost of ownership.

Both adapters are fully compliant with the IEEE 802.3z standard and support full- and half-duplex mode. The PCI version of *ge* and *ce* cards comply to the PCI 2.1 specification and run at both 33MHz and 66MHz in 32-bit or 64-bit mode.

4 The NetFrame Methodology

Traditionally, networkers tend to use microbenchmarks, such as Netpipe [4], Iperf [5], and Netperf [3], to evaluate machines' network performance. These microbenchmarks, however, could only provide quick and simplified snapshots that often create confusions in discussions since the number of parameters that affect network performance is large. It will be nice if there exists a tool set that has the simplicity close to the microbenchmarks while still capable of providing the user a relatively comprehensive view of the network performance.

Our NetFrame evaluation framework, which we will describe in detail in this section, is intended to serve this purpose. We use an extended version of Netperf [3], which we call MC-Netperf, as the baseline microbenchmark. Included in this tool set are a few scripts and programs that automate the evaluation process. NetFrame requires the users to provide

3

a few inputs before it fires up a series of runs and summarizes the experimental data in a spreadsheet-style table for further analysis. The NetFrame tool set runs a predetermined combination of experiments to measure throughput and packet rate (number of of packets per second). At the mean time, it also collects data that shows the amount of system resource that is consumed by the networking activities. The graphs based on the data collected by NetFrame should provide the user a comprehensive view of the network performance.

Different collections of the NetFrame data is comparable since it was obtained under a predetermined system setting. The data is comprehensive since a combination of interesting data points are included. In addition, this experiment is easy to perform since it is fully automated.

4.1 NetFrame Data Set

The NetFrame experiments include the following sets of measurement runs:

1. **Impact of Socket Buffer and Message Size.** The size of the socket buffer determines the amount of data the application can write to the socket in each attempt. This parameter, together with the Solaris TCP module parameter *tcp_xmit_hiwat*, determines the largest achievable TCP window size, and hence affects the maximal throughput. Message size is the amount of user data that is written to the socket buffer in each of the calls to the the *send* function using BSD socket interface. The number of calls to send also affects the amount of system resource required per unit amount of data and hence the overall throughput and packet rate. The sizes of socket buffers in the experiments vary from 24 Kilobyte (KB) to 1 Megabyte (MB) and the message sizes range from 4 bytes to 1MB.

2. **Impact of the number of TCP connections.** The number of TCP connections that go through a network interface simultaneously determines the pressure on the Solaris kernel modules and the queuing mechanism. Since the Solaris implementation uses *mutex* to guarantee mutual exclusion of access to some system resource, the number of TCP connections also affects how the system resource is competed for, and hense the amount of time required to process each packet. We chose 2, 4, 6, 10, and 20 simultaneous TCP connections in the experiments

3. **Impact of additional micro-processors (CPUs).** The Solaris kernel is multithreaded. This means when additional processors are added to a system, Solaris can distribute the tasks and get the requested job done faster. However, in order to get the job done in the correct way, some serialization effort is necessary. This means all of the CPUs in a system need to coordinate in a way that the semantics of the program is not violated. The serialization and communication work among the CPUs prohibit the CPUs from executing programs completely in parallel. The number of CPUs in a system directly affects the throughput and packet rate. We evaluate 1, 2, 4, 5, and 8 processors when the number of NICs in the system fewer than 2.

4. **Benefit of additional Network Interface Cards (NIC).** Each NIC functions independently but competes for the central system resource, for example, the Sun Fireplane in the Sun Fire midframe servers. The addition of a second NIC does not bring the overall throughput and packet rate to twice of that of a single NIC. The benefit of each additional NIC determines how well a server can scale to handle heavier load. The number of NICs evaluated in NetFrame goes from 1 to 4.

During the experiments, NetFrame collects the resources consumed by the system and the packet rate at a per second basis. The per packet cost in terms of system resources consumed can be calculated after the experiment. By comparing the per packet cost, we can know if the Solaris operating system and the NIC driver are efficient so that more CPU time can

4

be devoted to user applications, such as a web server.

4.2 TCP/IP Tuning

In order to obtain the maximal performance from the *ge* and *ce* interfaces, we changed some of the TCP parameters to optimize the system performance for bulk-transfer traffic. They include:

- **tcp_xmit_hiwat**: This parameter affects the default size of socket buffers and indirectly determines the largest possible window size during the TCP transfer operations. This value is changed to 65536 in our experiments. The default value for Solaris 8 is 24576.

- **tcp_deferred_acks_max**: This parameter sets the maximum number of packets that the receiving end can hold before sending an *ACK* packet to acknowledge receipt of the last batch of packets. The more packets the receiving end can hold on to before sending an *ACK* packet without overflowing the TCP transmission *WINDOW*, the lower the overhead the system incurs. We used 16 in our tests, which is the maximum allowed.

- **tcp_maxpsz_multiplier**: This parameter specifies the largest TCP segment that the applications can write to the TCP module. It saves the sending side overhead if the messages being sent were mostly moderate to small in size.

- **tcp_recv_hiwat**: This parameter specifies the maximum socket buffer size at the receiving end. It works with the *tcp_xmit_hiwat* to decide the maximum transmission *WINDOW* size. We also used 65536 in our experiments. However, when the socket buffer is set to 1MB, these two parameters also need to be adjusted to 1MB.

- **tcp_wscale_always**: This parameter specifies whether the TCP transmission *WINDOW* is allowed to go beyond the 16-bit limit of 65536. We set the value to 1 to enable larger than 65536-byte *WINDOW*.

5 Performance Characterization

In this section, we present the experiment data we obtained using the NetFrame 0.8.0 tool set on two platforms with the *ge* interface card. We will also present a first look of the *ce* cards which use copper media.

5.1 Experiment Setup

Two systems were tested. The focus is the Sun Fire 6800 (F6800) midframe server, where most of the performance characterization work is conducted. Measurements on the Sun Enterprise 6500 (E6500) midrange server is done for the purpose of platform comparison. The machines are configured as follows:

- **F6800:** Two domains were used as the client and the server. Each with 12 750MHz Ultra-Sparc III processors. and 12GB of memory.

- **E6500:** 8 400MHz UltraSparc II processors and 12Gb memory. Clients include 2 Sun Enterprise 3500 with 6 400MHz UltraSparc II processors, and a Sun Enterprise 450 with 4 400MHz Ultra-Sparc II processors.

- Point-to-point Connections are used for *ge* to *ge* experiments. An Extreme Summit 5i switch is used for *ge* to *ce* experiments.

5.2 The Impact of Socket Buffer and Message Size

As described in Section 4, the size of the socket buffer and message determines the amount of system resource required to process each TCP segment and each "send" and "receive" attempt by the user. Hence, it will be beneficial to the end users if they know which combination of socket buffer and message sizes deliver the best overall performance.

Figure 2 summarized the throughputs achieved using socket buffers of 24KB, 64KB, and 1MB with message sizes range between 536 Bytes and 1MB. The system under test is an F6800 with 4 CPUs and 1 TCP connection in each direction is measured. We
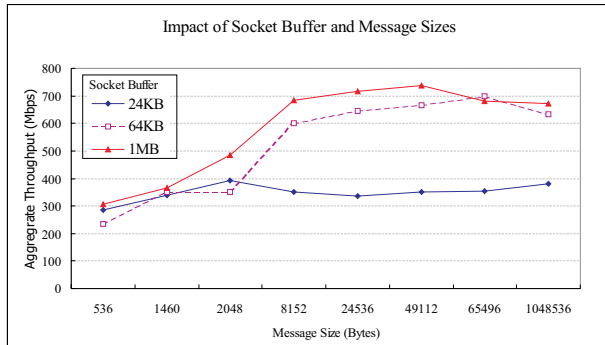
Figure 2: The Impact of Socket Buffer and Message Sizes on Network Performance.

Figure 3: The Benefit of Additional CPUs.

can see that the throughputs obtained using 24-KB socket buffer lags far behind the other two curves. However, the performance of 1-MB socket buffer outperforms the 64-KB socket buffer case by only up to 25%, while the size of the socket buffer increases 16 times. Hence, the 64-KB socket buffer seems to offer the best return if the total amount of memory used by network operations is a concern. When the size of the message is within 2KB, the system delivers rather poor performance (throughput below 500Mbps) even if we use 64KB and 1MB socket buffers. The overall throughput increases by up to 70% when the size of the message moves from 2KB to 8KB for 64-KB and 1-MB socket buffers. Hence, the applications should try to avoid sending messages smaller than 8KB to the network for maximal performance. In the later experiments, we will use 64KB as the size for socket buffers.

5.3 Scalability Analysis

Sun's products are designed for maximal scalability, which means that the additional hardware the customers invest in a single system always bring about significant performance gain. In this part of the paper, we will present how the throughput and packet rate changes when additional CPUs and gigabit NICs are installed in an F6800 system.

Although Sun midframe servers rarely ship with 1 or 2 CPUs, it would still be interesting to see how much performance can be delivered when only 1 and
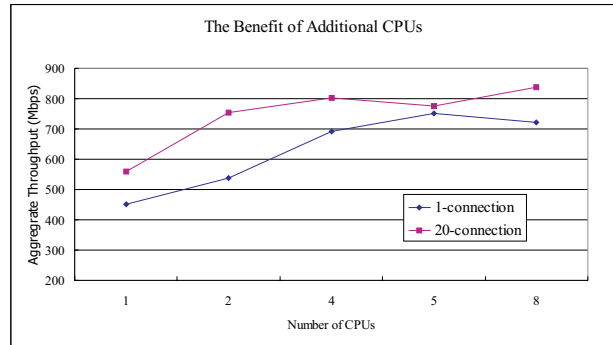
2 CPUs are present. In the CPU scalability analysis, we start with 1 CPU in an F6800 domain, and move to 2, 4, 5, and 8 CPUs for measuring the throughput and packet rate delivered by a *ge* NIC. We measured the cases with either 1 connection or 20 connections in each direction. The results are summarized in Figure 3.

As show in the figure, we see that The benefit of adding CPUs becomes rather trivial after we have 4 CPUs in the F6800. The 1-connection curve peaked with 5 CPUs, but the return of adding the fifth CPU is only 8.7%. The 20-connection curve peaked with 8 CPUs, indicating that multiple TCP connections can take advantage of more CPUs. However, the benefit of adding 4 CPUs is a merely 4.4%. On the other hand, it appears that a system of 2 CPUs can not take full advantage of a *ge* interface.

Since a single NIC does not require more than 4 CPUs to deliver close-to-optimal performance, a system with 8, 12, or 24 CPUs should be able to support multiple NIC's at their maximal performance. It is essential that the customers know how much performance gain can be archived when additional NIC's are added to a system with sufficient number of CPUs. However, since the PCI slots in an I/O assembly are not uniform (2 66MHz slots, and 6 33MHz slots), the additional cards added to the system can have many configuration possibilities. We choose to measure performance gain of each additional NIC in the following way:
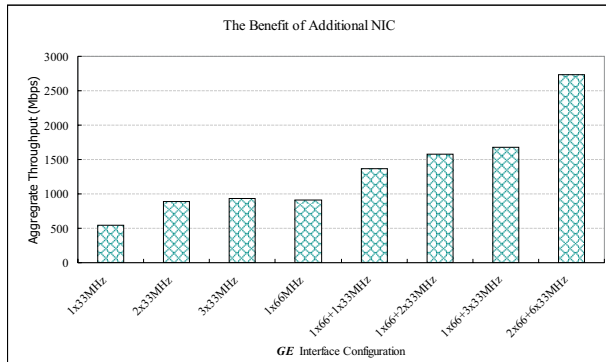
- Start from 1 *ge* NIC in the 33MHz slots. Then

Figure 4: The Benefit of Additional NIC's.

Figure 5: Comparing GE on F6800 and E6500 on the Impact of Message Size (4 CPUs and 1 connection each way).

move to 2 NIC's in 33MHz slots and 3 NIC's in 3 33MHz slots (controlled by 1 Schizo). This will show how the 33MHz PCI bus is saturated by the NIC's.

- Add a NIC in the 66MHz slot controlled by the same Schizo as that of the 33MHz slots in the last step.

- Configure 2-card and 3-card cases differently using 1 NIC in the 66MHz slot and 1 or 2 NIC's in the 33MHz slots.

- Skip the cases with 5 to 7 cards, Add 4 NICs to populate the whole I/O assembly for the case of 8 cards.

As shown in Figure 4, when only 1-NIC is present, installing it in a 66MHz slot offers 72% higher throughput than in a 33MHz slot. When two NIC's are present, installing them in the same 33MHz PCI bus under the same Schizo saturates the PCI bus by offering 884Mbps. If the same two NIC's spread to two busses under one Schizo, the performance is 55% better. When three NICs are needed, both the 66MHz and the 33MHz slots need to be used in order to show benefit. The performance of 8 NIC's controlled by two Schizos is 63% better than 4 NICs' under 1 Schizo.

5.4 Platform Comparison

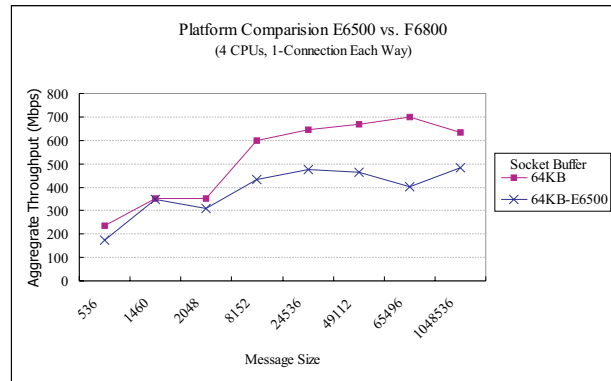A lot of Sun's current customers own the Enterprise x500 series. With the launch of the new Sun Fire midframe servers, it is essential that we show our customers how the new products perform comparing to the existing product line. Hence, we conducted similar experiments as in Section 5.3 on an E6500 server. Since the PCI slots and the I/O assemblies are configured differently from the those of an F6800, we will only include data when 1 to 8 CPUs are used with one *ge* NIC.

As shown in Figure 5, single-connection throughput for E6500 trailing that of F6800 by about 30% when the message size is above 8KB. CPU Scalability comparison is pending ....

5.5 Characterization of System Resource Consumption

Network traffic comes from the need of the applications to send and receive data. Although it is important that the NIC's deliver the highest possible throughput and packet rate when the system is idle, it is even more important for the NIC's to deliver high performance when the system is 100% busy running commercial applications. Hence, the amount of system resource required to achieve a certain level of network throughput and packet rate is key to the performance of the overall system.

Table 1 summarized some important resources consumed by a 66MHz *ge* interface when it is delivering the maximal throughput with 10 TCP connections in each direction using 64-KB socket buffer and

messages of different sizes. Note that the system-time is obtained using a 4-CPU F6800 system. We can see that the system time remains below 31% and that the instructions/packet stays around 5000 no matter which message size is chosen. The *ge* interface coalesces from 7.49 to 10.99 packets for each interrupt it sends to the CPU, which greatly reduces the CPU burden of processing interrupts. The system-mode data cache miss rate per instruction moves gradually from 9.45% to 15.92% when the message size increases 100 times from 1460 bytes to 1MB.

5.6 A First Look at Sun Gigabit Ethernet 3.0

As described in 3, the *ce* NIC is Sun's newest addition to the gigabit ethernet products. The *ce* card uses the cheaper category-5 UTP copper cables to reduce cost of ownership. The *ce* hardware provides some advanced features such as the packet reassembly hardware assist function, and multiple descriptor rings for reception. The device driver of *ce* also provides multiple task queues that handle delayed protocol processing for better scalability and for relieving workload on the CPU that processes interrupt.

We do not intend to perform a full evaluation of *ce* cards in this paper, but feel the need to present some preliminary observations since the product is in the market. Figure 6 compares *ge* and *ce* on the impact of message sizes using a 64KB socket buffer. Both *ge* and *ce* offer similar throughput when the message size is below 48KB. However, *ce* seems to be underperforming when the message size is larger than 64KB. Figure 7 compares the benefit of additional CPUs. I appears that *ce* scales better than *ge* when there is only 1 connection in each direction, although the overall throughput offered by *ce* is mostly lower than that by *ge*. For 20-connections in each direction, *ce* outperforms *ge* when the system has no more than 2 CPUs, but underperforms *ge* when there are more than 2 CPUs present.
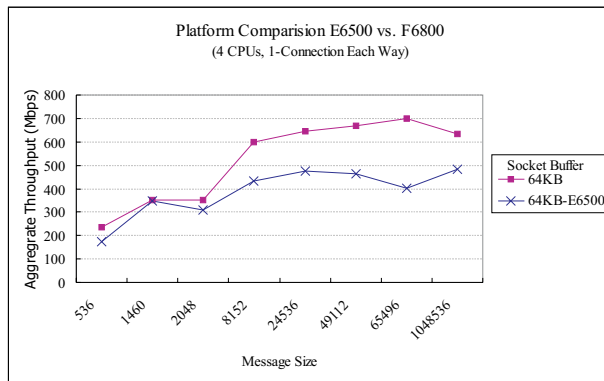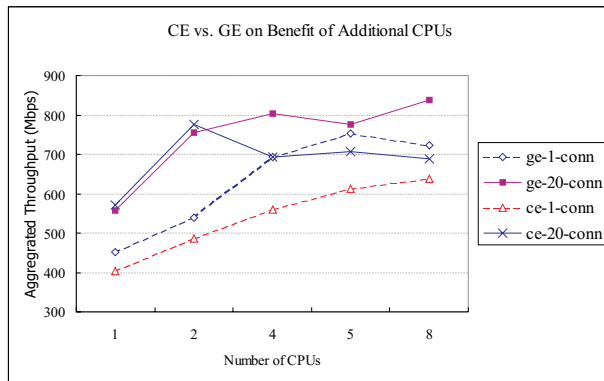

Figure 6: CE vs. GE on Impact of Message Size.

Figure 7: CE vs. GE on Benefit of Additional CPUs.

Message (Bytes)	System-Time	Instructions/Packet	Packets/Interrupt	D-Cache Miss Rate
1460	30%	4633	10.99	9.45%
24536	28%	4922	8.23	13.92%
65496	28%	5158	7.52	14.17%
1048536	31%	5621	7.49	15.92%

Table 1: Some Important Resources Consumed by *ge* on a 4-CPU F6800 system (64-KB Socket Buffer, 10 connections each way)

5.7 Sun Fire 6800 Gigabit Ethernet Performance Summary

In the next step, we try to determine the maximal network performance that can be obtained on the F6800 in the following cases:

- Single-Connection: There is one TCP connection from the server to the client and one TCP connection from the client to the server through a single NIC. The number of CPUs is up to 8.

- Single-CPU: Only one CPU is used. The number of TCP connections is up to 20 per direction through one NIC. There will be two numbers reported using a 33MHz NIC and a 66MHz NIC.

- Single-NIC: One NIC is evaluated. The number of TCP connections through the NIC is from 1 to 20, and the number of processors is up to 8.

- Single-Schizo: Using two PCI buses (1 33MHz bus and 1 66MHz bus) controlled by one Schizo chip. This means 3 NICs in the 33MHz slots and 1 NIC in the 66MHz slot will be used. The number of TCP connections in each direction through each card varies from 1 to 20 and the number of CPUs used is up to 8.

- Single-IO-Assembly: Using all four PCI buses (2 33MHz buses and 2 66MHz buses) available in one I/O Assembly. A total of 8 NICs will be used, which includes 6 cards in the 33MHz slots and 2 cards in the 66MHz slots. The number of TCP connections through each NIC is up to 20 and the number of CPUs used is up to 12.

Metrics	Throughput	Packets/Second
Single-connection	936Mbps	103,000
Single-CPU	590Mbps	76,000
Single-NIC	936Mbps	124,000
Single-Schizo	1687Mbps	220,000
Single-IO-Assembly	2733Mbps	334,000

Table 2: Maximal Observed Throughput and Packet Rate for an F6800 System with 8 750MHz Ultra-Sparc III CPUs.

This set of numbers is intended to show the networking capability of an F6800 using the *ge* interface cards when the amount of NIC hardware investment changes. The customers of F6800 servers can decide on the number of NICs to purchase depending on their different workloads. Table 2 summarizes the total throughput and packet rate.

What do we say about the table here?

6 Summary

With the launch of Sun's new midframe Sun Fire server and the increasing popularity of gigabit ethernet, we are facing the question of how Sun's new servers perform in the gigabit ethernet arena. In this paper, we evaluated the performance of an F6800 using Sun Gigabit Ethernet 2.0 with the NetFrame methodology. Extensive experiments are conducted to study the impact of the size of socket buffer and message, the benefit of additional CPUs, and the benefit of additional NIC's. Results show that the new

Sun Midframe server scales well with the addition of extra CPUs and NIC's and hence offers rather impressive network performance, with a single gigabit NIC reaching 936Mbps, a single PCI controller supporting 1640Mbps, and a single I/O assembly observing 2733Mbps.

References

[1] Adrian Cockcroft and Richard Pettit. "Sun Performance And Tuning, Section Edition". ISBN 0-13-095249-4. Sun Microsystems Press. Also a Prentice Hall Title. Prentice Hall, 1998.

[2] Gary R. Wright, and W. Richard Stevens. "TCP/IP Illustrated, Volumn 1". ISBN 0-201-63354-X. Addison Wesley, December, 1999.

[3] Rick Jones. "The Public Netperf Home Page". http://www.netperf.org.

[4] Quinn O. Snell, Armin R. Mikler and John L. Gustafson. "NetPIPE: A Network Protocol Independent Performance Evaluator". http://www.scl.ameslab.gov/netpipe/.

[5] Iperf Reference. Pending.