Analysis of Processor Power Versus Network Bandwidth

Mark Claypool and John Riedl

{claypool,riedl}@cs.umn.edu
University of Minnesota
Computer Science Department

September 8, 1995

With the spread of networks of workstations and the wide-range of application communication and computation requirements, we may ask of each system: is the potential application bottleneck in the processor or the network? We propose an experimentally based model for visualizing and measuring the capacity of a system's processor and network. We apply our model to three typical networks of workstations: home, academic and research. Based on our analysis, we find that research clusters of high-performance workstations may still have computation bottlenecks when they are connected with the highest bandwidth networks. Academic clusters of Sun workstations, on the other hand, have the network as a potential application bottleneck. Home networks of personal computers have the network as a severe potential application bottleneck. Our model may direct work in application development and network of workstation design by exposing potential bottlenecks.

1 Introduction

Workstation performance has increased by a factor of 2 each year since 1985 [6], while costs have dropped dramatically. With the spread of networks and the decreasing price of powerful workstations, networks of workstations are becoming ubiquitous. Applications that run on such systems range from processor bound, such as ray-tracing, to network bound, such as remote file access. Given the range of application resource requirements, the following question arises for a given network of workstations: is the potential application bottleneck in the processor or the network?

We propose a model for visualizing and measuring the capacity of a system's processor and network. Our model can be depicted by a two-dimensional graph, with one axis representing computation and the other communication. A system is represented by an area bounded by its processor power and network bandwidth. We call this area *system capacity*. Areas outside of the system capacity indicate potential processor or network surplus, and indicate the magnitude and location of possible

[0]This work is sponsored in part by the Amry High Performance Computing Research Center under the auspices of the Department of the Army, Amry Research laboratory cooperative agreement number DAAH04-95-2-0003/contract number DAAH04-95-C-0008, the content of which does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred.

application bottlenecks. Our model may direct work in application development and network of workstation design by exposing potential bottlenecks.

Hypothesis 1: *Measurement of per-packet processor sending load can be used to accurately predict maximum processor sending bandwidth.*

Hypothesis 2: *Network bandwidth and processor power in research network of workstations may lead to an equal chance for network and processor application bottlenecks.* If true, application developers may need to focus on developing applications with balanced computation and communication loads. To keep future bottleneck potential equal, network of workstation designers may need to couple processor improvements with network improvements and vice versa.

Hypothesis 3: *Network bandwidth in academic network of workstations may lead to processor bottlenecks for some applications.* If true, application developers may need to focus on reducing processor load, perhaps by permitting additional network communication if it will reduce processor computation.

Hypothesis 4: *Processor power in home network of workstations may lead to network bottlenecks for some applications.* If true, application developers may need to focus on reducing network communication, perhaps by using the processor to compress network data. Network of workstation designers may also concentrate on increasing network bandwidths for links to the home, such as through modems or ISDN.

Section 2 describes related work. Section 3 details micro experiments to measure processor send loads. Section 4 experimentally tests hypothesis 1 above. Section 5 develops our model and tests hypotheses 2 through 4 above. Section 6 summarizes our conclusions and lists possible future work.

2 Related Work

2.1 Network Performance

Claypool, Riedl *et al* look at an application with high network bandwidth requirements [2]. We predict the effects of multiple servers, multiple users and compression on network bandwidth.

Riedl, Mashayeki, Schnepf, Claypool and Frankowski explore network loads of multi-person audioconferences under free mode and under two types of silence deletion [10]. We show that network loads under free mode quickly became prohibitive under increasingly larger audioconferences. Network loads under silence deletion are significantly lower.

Lin, Hsieh, Du, Thomas MacDonald measure the performance of an ATM network [9]. Although theoretical ATM performance is 155 Mbits/second, they find Sun Sparcstations 2 have significantly less throughput under existing Application Programming Interfaces.

Shoch and Hupp measure actual performance and error characteristics of an existing Ethernet installation with over 120 directly connected hosts [11]. They find latency, error and collision rates are extremely low under light loads and the system is stable and has a utilization of 98 percent under heavy loads.

Leu and Du propose a Cycle Compensation Protocol (CPP) as a better version of the Distributed Queue Dual Bus (DQDB) IEEE 802.6 standard [8]. Through detailed network simulation, they show the CPP ensures fairness regardless of the end-to-end propagation delay to the slot size and achieves almost the same throughput and delay as those of DQDB.

2.2 Processor Performance

Claypool, Riedl *et al* measure processor components of compression and decompression [2]. We look at the tradeoff between processor and network bandwidth with compression.

Riedl, Mashayeki, Schnepf, Claypool and Frankowski measure processor load under free mode using the counter process [10]. We show that processor loads under free mode quickly became prohibitive under increasingly larger audioconferences.

Claypool and Riedl study audioconference processor load under the effects of silence deletion [3]. We find silence deletion improves audioconference processor load more than most alternatives, including 10 times faster processors and multicasting.

Dongarra compares the performance of different computer systems in solving dense systems of linear equations [4]. He gives results for over a hundred computers, ranging from a CRAY Y-MP through workstations by SGI and Sun to IBM PCs.

Gustafson and Snell present HINT, a mathematically sound method for comparison of computational performance [5]. HINT scales to computing as slow as hand calculation to computing as fast as the largest supercomputers. It ports with little effort, permitting fair, low-cost comparison of any architecture capable of digital arithmetic.

Hennessy and Patterson analyze processor performance and growth [6]. In addition to a computer architecture overview, they present detailed examples of processor performance on three common applications: the Gnu C compiler; Common TeX; and the CAD Tool, Spice.

Lazowska parameterizes queuing network performance models to assess the alternatives for diskless workstations [7]. He uses a counter process to measure processor loads.

2.3 Our Contribution

Our contribution is the development of a model for communication and computation ability. We build upon previous performance analysis with our careful measurements of processor sending loads. We contribute a model for predicting maximum sending throughput and validate our model with experimentally-based macro experiments.

3 Micro Experiments

Experiments that measure components of an application we call *micro experiments.* We do micro experiments to allow us to predict the performance of applications built with those components. Examples of previous micro-experiments appear in [2, 3].

3.1 Design

Our micro experiments are designed to measure the processor load for sending packets. We chose 3 different classes of Sun workstations: a 3/80, an SLC and a Sparc Classic to obtain results at several data points.

We used a process that increments a `double` variable in a tight loop to measure the processor load for a *send* process that does nothing put transmit fixed size packets. We did not use the Unix **`time`** command because the reporting of per-process processor consumption by most operating systems is unreliable. Often, the system gives an incorrect account of interrupt level processing and fails to capture processor degradation from DMA [3, 7].

To obtain a baseline for our counter, we ran the counter process on a machine with a minimum of system processes running. This gave us the processor potential for the machine. We then ran the send process with the counter process. The difference between the bare count and the count with the send process is the send-induced load. To verify that the counter process does indeed accurately report loads of processor-bound processes with which it runs concurrently, we ran the counter process with 1, 2, 3 and 4 other counter processes. Since we assume the counter processes will have same priorities, we expect the counter value to be `(count on bare machine) x 1/(N+1)`, where N is the number of other counters running. Figure 1 shows the results of this experiment. The predicted values were within the confidence intervals for all the measured values.

We ran the counter process for 200 seconds to amortize start-up costs. To determine the number 200, we took 5 counts for an increasing number of seconds. At just under 200 seconds the standard deviation of the 5 counts approached an asymptotic value.

In order to achieve the desired confidence intervals in measured counts, we performed 5 iterations for each machine at packet sizes of 1000 and 1400 bytes. One 200 second counter run was one iteration. The counter values were divided by the number of packets sent to get a per-packet count. Based on the count per second a processor can perform, the send counts were converted into processor time.

In running the send process, we did not want to send as fast as possible, as this might cause artificial load from overfull network buffers, unexpected network card behavior, etc. Thus, we used a Unix `signal` to send at fixed intervals. As the measurements contained the processor load for both the signal and the send, we performed an equal number of experiments with a process that did only the signal. We then subtracted the signal load from the send plus signal load to get load for just the send.

In order to determine if the processor load per send varied with the rate at which the packet was sent, we measured the processor load for sending for signal rates of 10 milliseconds between sends to 300 milliseconds between sends. Figure 2 depicts the results. The confidence intervals did not

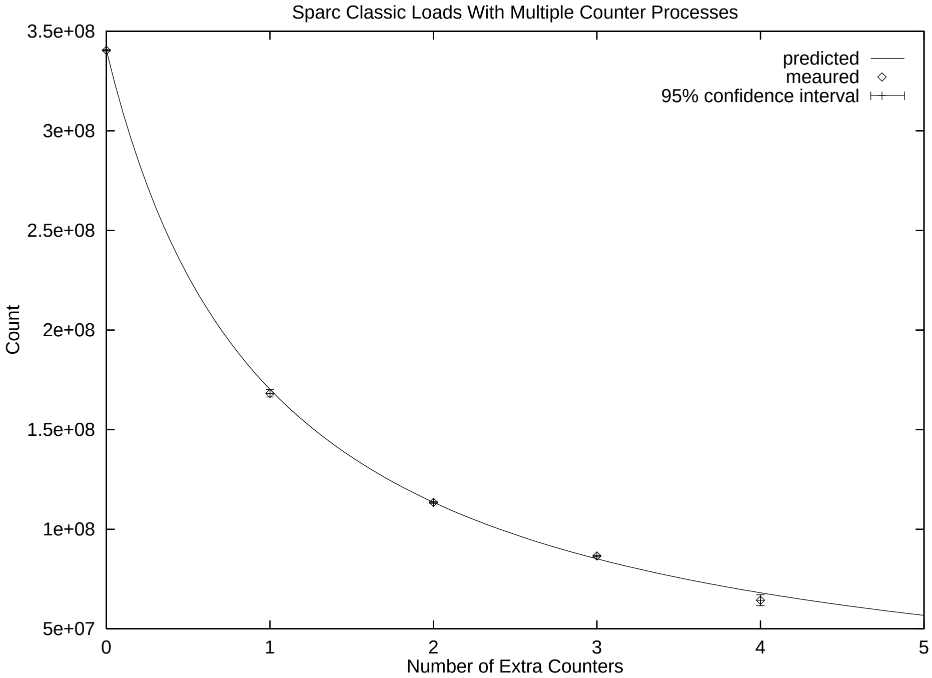

Figure 1: Count versus Number of Counters. The vertical axis is the count obtained by the counter process. The horizontal axis is the number of additional counters running. The curve represents the predicted value. All points are shown with 95% confidence intervals. The measurements were taken on a Sun Sparc Classic.

overlap in all cases. Thus we use linear regression to obtain the processor load for sending one packet. To avoid redundancy, we do not depict the similar graphs for the 3/80 and SLC.

At the receiver, we verified that the expected number of packets were actually sent by the sender. In all cases, the receiver did obtain the expected number of packets based on the signal frequency.

In determining the processor load for a packet, there is a per-packet overhead from kernel-user boundary crossing plus a per-byte overhead for copying the packet from the user's space to various buffers. Thus, we duplicated our experiments on the send process with packet sizes of 1000 and 1400 bytes. We fit a line through the two data points to obtain our send load equations.

3.2 Results

Table 1 lists the counter values of the three lightly loaded processors. We call these counts *barecounts*. Figure 3 depicts the three barecounts in a graph. Table 2 lists the per-packet times for sending 1000 and 1400 byte packets and gives the values for the sending equation for processor load (in milliseconds): `time = per-packet + per-byte * num-bytes`. Figure 4 depicts the linear equations.

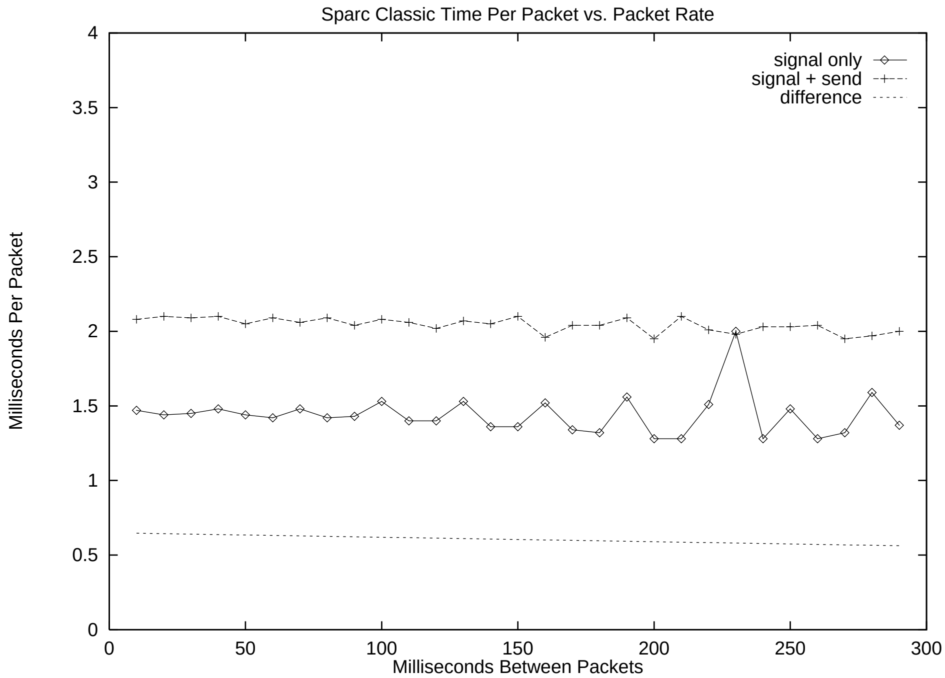

Figure 2: Per Packet Processor Load. The vertical axis is the time per packet in milliseconds. The horizontal axis is the milliseconds between packets, as sent to the `signal` call. The top line is the processor load per send plus signal. The middle line is the processor load for the signal only. The bottom line is a linear regression fit of a line through the difference between the top two. This line represents the processor load for sending one 1000 byte packet.

Processor Type	Count	Interval
Sun 3/80	3,470,034	[3,430,391 : 3,509,678]
Sun SLC	122,040,705	[121,959,971 : 122,121,439]
Sun Sparc Classic	375,601,242	[374,393,651 : 376,808,833]

Table 1: Workstation counter values on lightly loaded machines. The 95% confidence interval values are also given.

Processor Type	msec/1000 bytes	Interval	msec/1400 bytes	Interval	Per-packet	Per-byte
Sun 3/80	2.1	[1.8, 2.4]	2.8	[2.7, 2.9]	0.57	0.0016
Sun SLC	0.9	[0.8, 1.0]	1.2	[1.1, 1.3]	0.34	0.00061
Sun Sparc Classic	0.63	[0.60, 0.66]	0.80	[0.75, 0.85]	0.21	0.00042

Table 2: Processor loads for sending packets. Measured values are given for 1000 and 1400 byte packets, with 95% confidence intervals. Values for the linear equation in per-packet and per-byte times are also given. Times are in milliseconds.

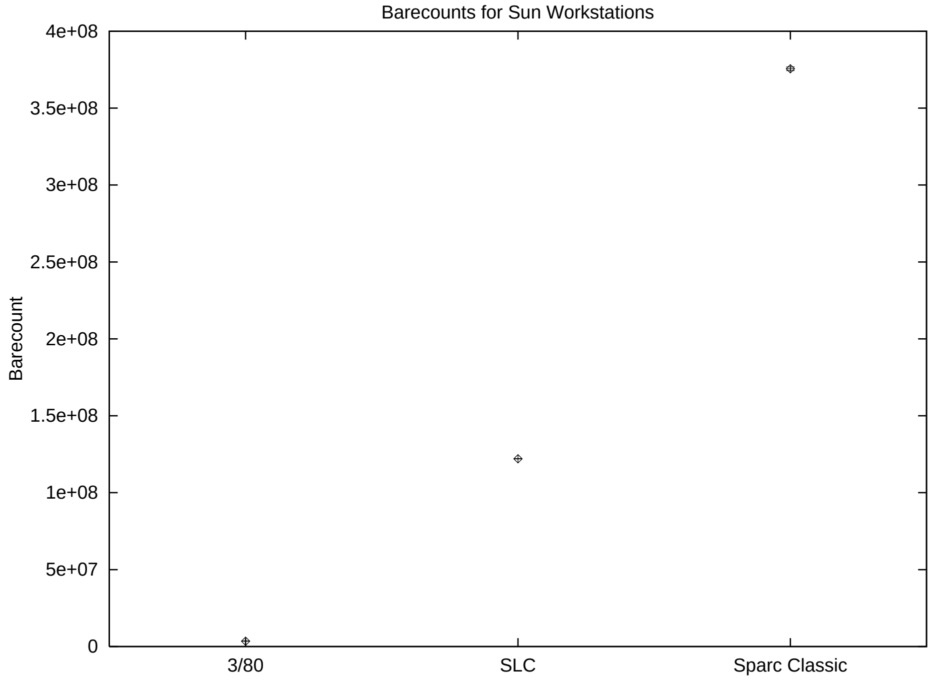

Figure 3: Counter values for three Sun workstations. Counters were run for 200 seconds on a lightly loaded machine. Each point is the mean of 5 iterations surrounded by a 95% confidence interval.

4 Macro Experiments

Experiments that measure performance of applications built with micro experiment components we call *macro experiments*. We do macro experiments to validate micro experiment-based predictions of application performance. Examples of macro experiments appear in [2, 3].

Our macro experiments were designed to measure the maximum sending rate (in Mb/s) of a workstation. We set the send process to transmit 1000 byte packets in a tight loop, without the `signal` present in the micro experiments. As in the micro experiments, we ran the send process for 200 seconds to amortize start-up costs. At the end of this time, we recorded the number of packets received by the receiver and converted this to Mb/s. We did 5 iterations for each machine to obtain confidence intervals.

We calculated the predicted network throughput from the per-packet loads obtained in the micro experiments (see Section 3). The predicted throughput is `1000 / milliseconds per-packet`. Since we have confidence intervals for the 1000 byte send loads, we can obtain confidence intervals for our predicted network throughput.

Figure 5 shows the predicted versus actual Mb/s throughput on each machine. The vertical axis is in Mb/s for sending. The horizontal axis has the three Sun Microsystem machines. The horizontal line indicates the maximum Ethernet capacity. All points are shown with 95% confidence intervals. Table 3 lists the numeric values for the predicted versus actual throughputs.

For the 3/80 and the SLC, the predicted throughput confidence intervals overlap the measured

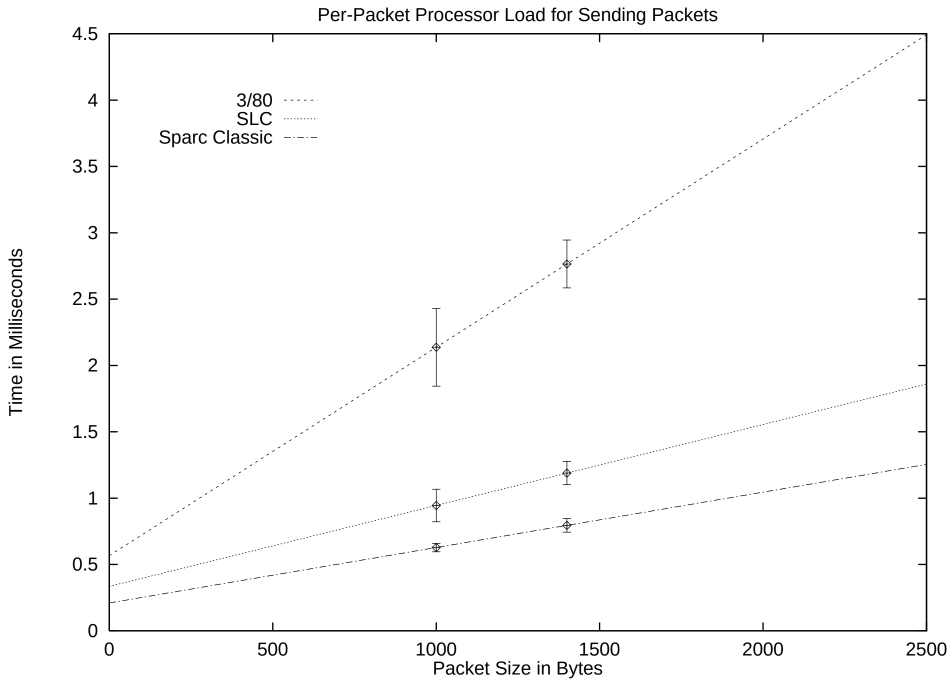

Figure 4: Processor loads for sending packets. The vertical axis is the processor load per-packet in milliseconds. The horizontal axis is the packet size in bytes. The points are measured values surrounded by 95% confidence intervals. The lines are linear fits through the points. The vertical axis intercepts represent the per-packet times and the slopes represent the per-byte times.

Machine	Predicted Mb/s	Confidence Interval	Measured Mb/s	Confidence Interval
3/80	3.7	[3.2, 4.2]	4.3	[4.2, 4.4]
SLC	8.6	[7.5, 9.7]	8.8	[8.3, 9.2]
Sparc Classic	12.3	[11.6, 13.0]	8.9	[8.7, 9.1]

Table 3: Predicted versus. Measured Processor Throughput. Units are in Mb/s. Confidence intervals are at 95%.

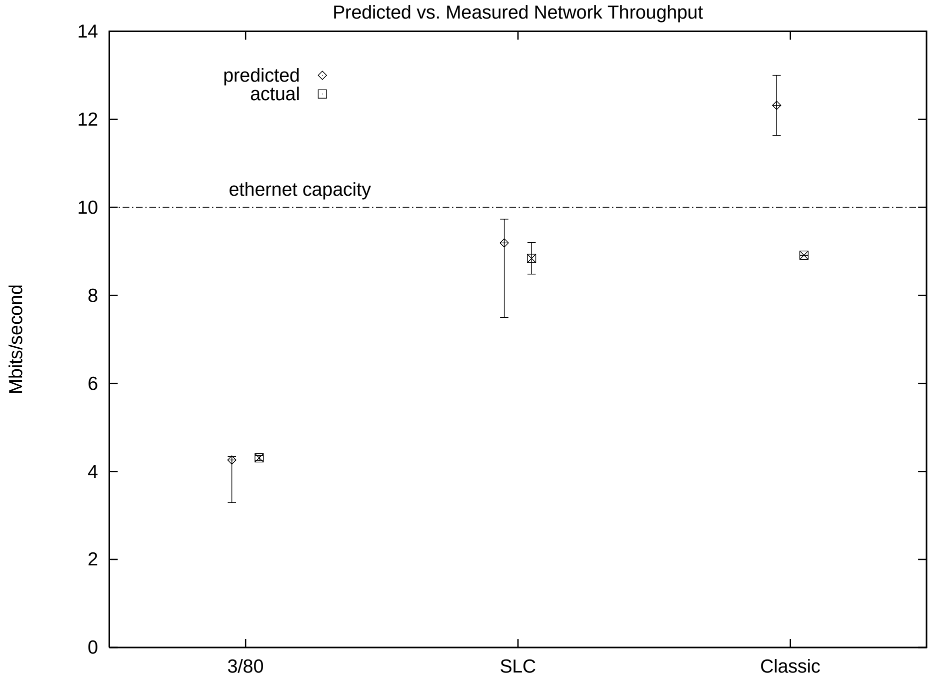

Figure 5: Predicted versus Measured Processor Throughput. The vertical axis is in Mbits/second for sending. The horizontal axis has the three Sun Microsystem machines. The horizontal line indicates the maximum Ethernet capacity. All points are shown with 95% confidence intervals.

throughput confidence intervals. This indicates that there is no statistically significant difference between our measure of predicted throughput and the actual measured throughput. For the Sparc Classic, the predicted throughput is above the maximum Ethernet throughput. This indicates our predicted measure of throughput is different than the actual measured throughput. We assume the actual measured throughput is limited by the maximum ethernet capacity of 10 Mbits/second.

5 Analysis

The macro experiments in the previous section suggest that per-packet processor loads and maximum network bandwidth may determine a processor's maximum communication throughput. If a processor does nothing but sends and receives packets, it achieves its maximum communication throughput. If a processor sends and receives no packets, it can spend all its time on computation. In this case, we say it achieves its maximum computation throughput. A processor may divide its time between communication and computation depending upon the applications it is running. The range of communication and computation combinations is determined by the network and processor. We develop a system model that depicts this range via a 2-D graph. The vertical axis represents the communication throughput. The horizontal axis represents the computation throughput. The units for both axes are similar to MIPS that have been converted to Mb/s. Horizontal lines represent networks. Downward sloping lines represent processors. Areas represent networks of workstations. Areas are bounded by the processor and network bandwidths. Figure 6

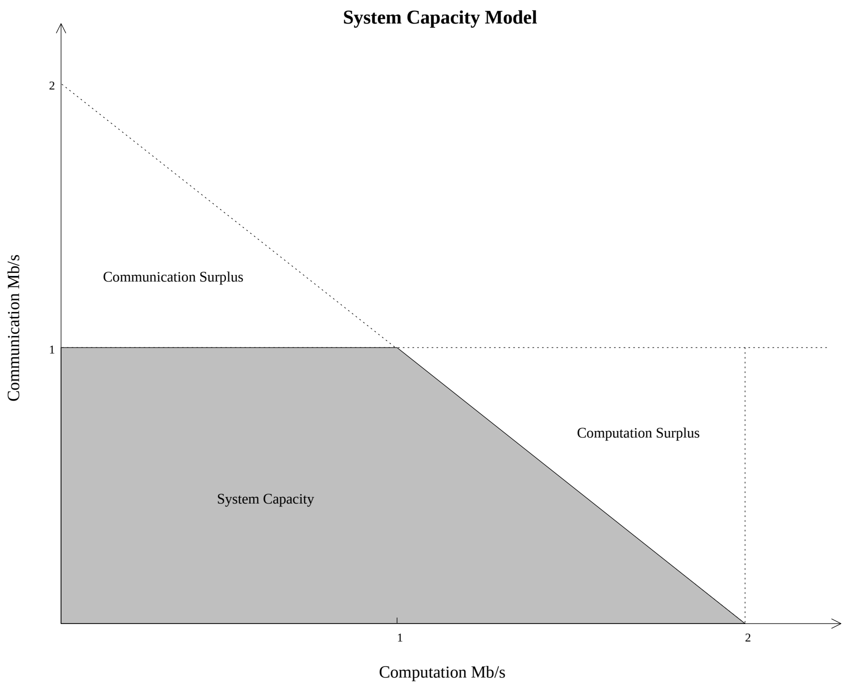

Figure 6: System Capacity Model. The vertical axis is communication Mb/s. The horizontal axis is computation Mb/s. The shaded area represents the network of workstations capacity. The upper triangle represents communication surplus. The lower triangle represents computation surplus.

depicts our model on a theoretical system.

Our model provides two network of workstation attributes:

1. *Capacity.* Capacity is the area bounded by the processor and the network bandwidth. Capacity determines what applications a system can run. The larger the capacity, the wider the range of applications a system can run. Capacity can range from 0.01 $(Mb/s)^2$ for a small PC and a modem to 150,000 $(Mb/s)^2$ or more for a high-performance workstation with a high-speed network. In Figure 6, the system has a communication bandwidth of 1 Mb/s and a computation bandwidth of 2 Mb/s. The total capacity of this system is depicted by the shaded area. The value is 1.5 $(Mb/s)^2$. In general, given a network bandwidth N and a processor communication/computation throughput P, the area of a system built with those components would be:

$$capacity = N * (P - 0.5 * N)$$

   Using the above formula, we can compute capacities for combinations of network bandwidths and processor communication/computation throughputs. Figure 7 depicts a 3-D graph of system capacity versus communication and computation throughputs. The surface represents the capacity of systems built with various networks processors. The two points represent a Sun Sparc 5 and an Ethernet and a SGI Challenge and a Fibre Channel. (See below for details.)

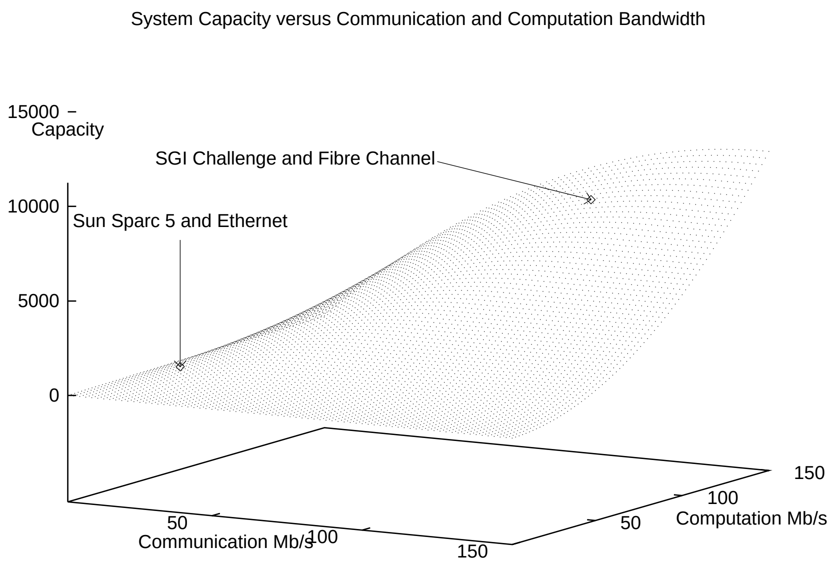

Figure 7: Capacity versus Communication and Computation Bandwidth. The X axis is communication Mb/s. The Y axis computation Mb/s. The Z axis is the system capacity. The surface represents the capacities of systems built with various networks and processors.

Network of Workstations Type	Processors	Networks
Home	Intel Personal Computers	ISDN, Modems, Ethernet
Academic	Sun Workstations	Ethernet
Research	SGI Challenge XLs	Fibre Channel, HIPPI

Table 4: Three networks of workstations with some typical processors and networks.

2. *Surplus.* Surplus is a comparison of processor and network bandwidth. In Figure 6, the triangle labeled “Communication Surplus” represents system capacity that cannot be used because of the processor limitations. The triangle labeled “Computation Surplus” represents system capacity that cannot be used because of the network limitations. The total surplus is the sum of the communication surplus and the computation surplus. We report surplus as the fraction of the computation or communication surplus to the total surplus. Surpluses range from 0 to 1. In Figure 6, the system has a processor surplus of 0.5 and a network surplus of 0.5.

   Systems that have a high communication surplus may have the processor as a potential bottleneck. Systems that have a high communication surplus may have the processor as a potential bottleneck. The exact location of the bottleneck is determined by the system applications. In our model, an instance of an application exists at a point determined by the application's computation and communication requirements. A system can run any application that lies inside its capacity space.

   Analyzing the computation and communication requirements of an application can be extremely difficult. Instead, we assert that most networks of workstations run applications with a variety of different computation and communication requirements. Surpluses then indicate location and magnitude of potential application bottlenecks.

   Figure 8 depicts a 3-D graph of computation surplus versus communication and computation throughputs. Notice that computation surplus is highest when the network bandwidth is much larger than the processor bandwidth. The graph of communication surplus would have a similar shape, but would be flipped over.

We apply our model to analyze the capacity and surplus of three typical networks of workstations: home, academic and research. Table 4 gives typical processors and networks for these networks of workstations.

To predict the processor bandwidth for systems other than the ones we measured, we assume that sending processor bandwidth is inversely proportional to SPEC integer ratings[1]. SPEC is a non-profit corporation formed by leading computer vendors to develop a standardized set of benchmarks. Founders, including Apollo/Hewlett-Packard, DEC, MIPS and Sun, have agreed on a set of real programs and inputs that all will run. The benchmarks are meant for comparing CPU speeds. The SPEC numbers are the ratio of the time to run the benchmarks on a reference system and the system being tested. Table 5 lists our the SPEC integer values and the processor bandwidths Mb/s in our typical networks of workstations. Table 6 lists the networks and their maximum bandwidths.

Our model may be applied to more than one workstation on a network. We call more two or more workstations a *cluster*. We assume the home cluster has an Intel 386-20 and an Intel 486-33. The

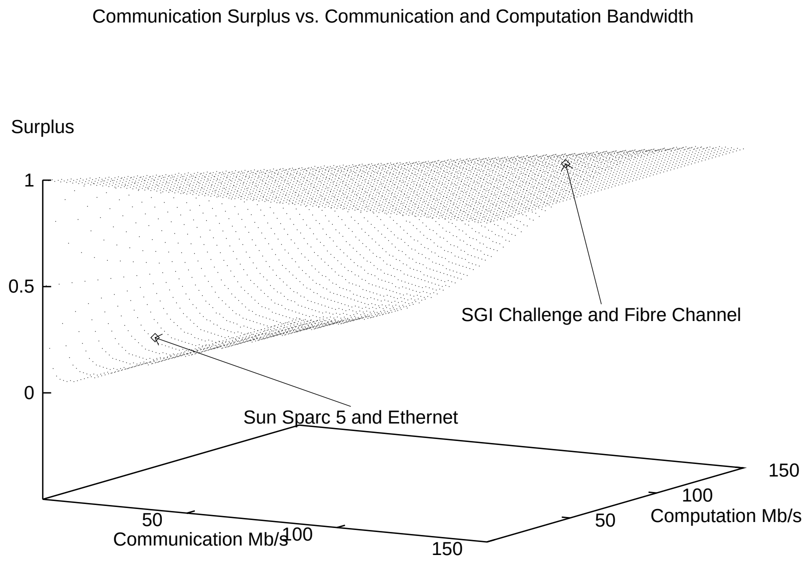

Figure 8: Computation Surplus versus Network and Processor Bandwidth. The X axis is communication Mb/s. The Y axis is computation Mb/s. The Z axis is network surplus. The surface represents the computation surplus' of systems built with varying network and processor bandwidths.

Processor Type	SPEC int	Processor Mb/s
Intel 386-20	2.2	1.2
Intel 486-33	17.5	9.9
Sun Classic	21.8	12.3
Sun Sparc 5	60.0	33.9
SGI Challenge XL	235.0	133.6

Table 5: Network of workstations processors, their SPEC int values and estimated maximum processor bandwidths.

Network Type	Bandwidth Mb/s
Modem	0.014
ISDN	0.064
Ethernet	10
Fibre Channel	100 to 800
HIPPI	800

Table 6: Network of workstations networks and their maximum bandwidths.

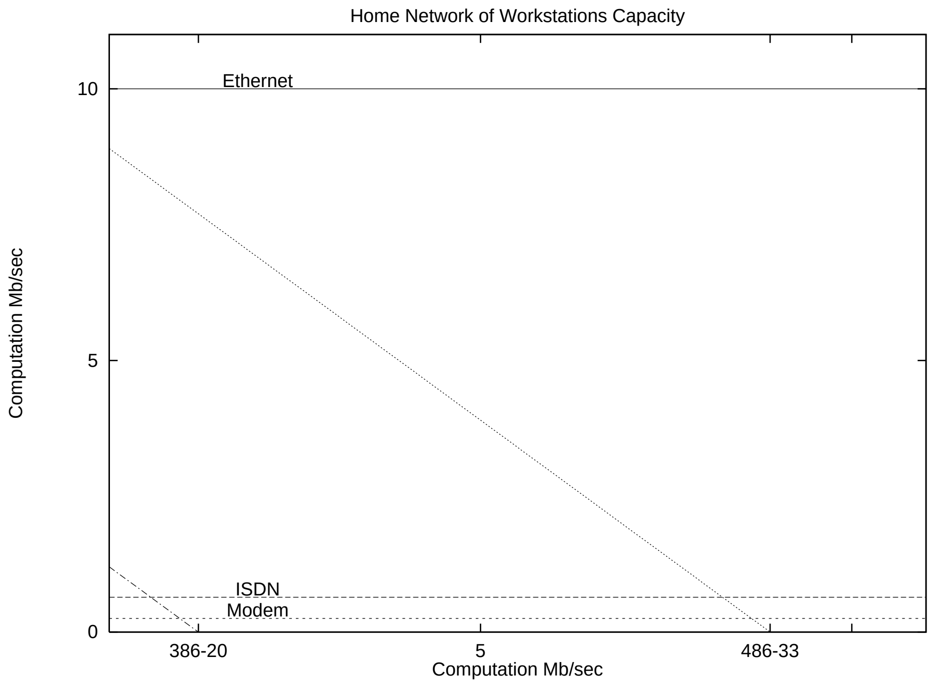

Figure 9: System Capacity Model for Home Networks of Workstations.

academic cluster has a Sun 3/80, a Sun SLC, a Sun Sparc Classic and a Sun Sparc 5. The research cluster has 4 SGI Challenge XL workstations. The super research cluster has 4 four-processor SGI Challenge XL workstations. We assume that processor sending throughput scales linearly with number of processors.

We notice that the workstation buses will not be the bottleneck for any system. The SGI workstations have a bus capable of 1 Gb/s throughput. The Sun workstations have bus capable of 100 Mb/s throughput. The Intel workstations have a bus that can achieve at least 10 Mb/s.

With the processor and network bandwidths above, we can apply our model to the three networks of workstations. Figure 9 depicts our model applied to the home network of workstations. Figure 10 depicts our model applied to the academic network of workstations. Figure 11 depicts our model applied to the research network of workstations. Table 7 lists the numeric values for all workstation-network pairs within our networks of workstations.

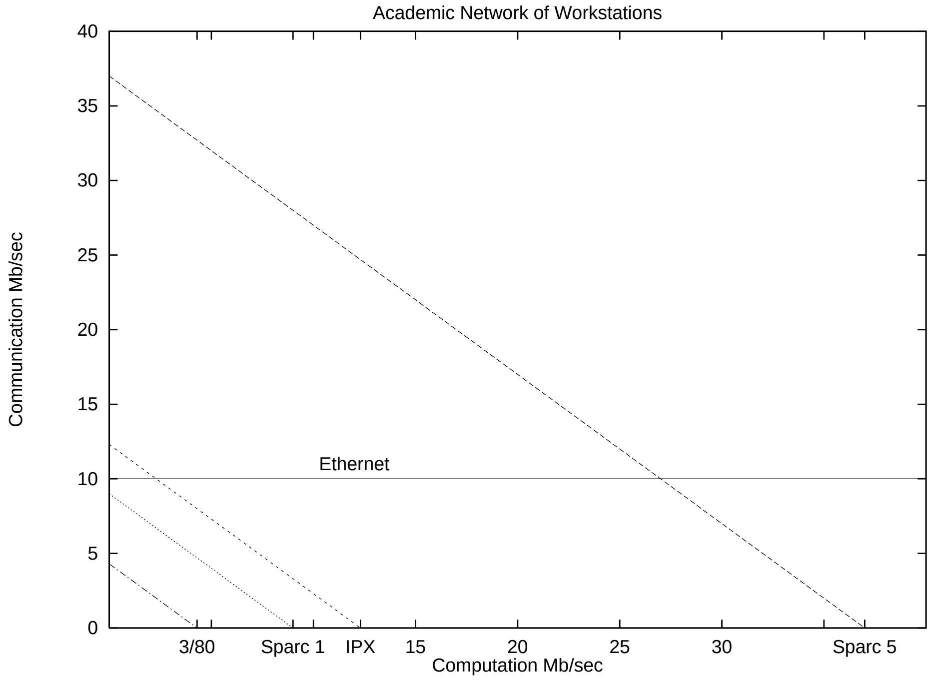

Figure 10: System Capacity Model for Academic Networks of Workstations.

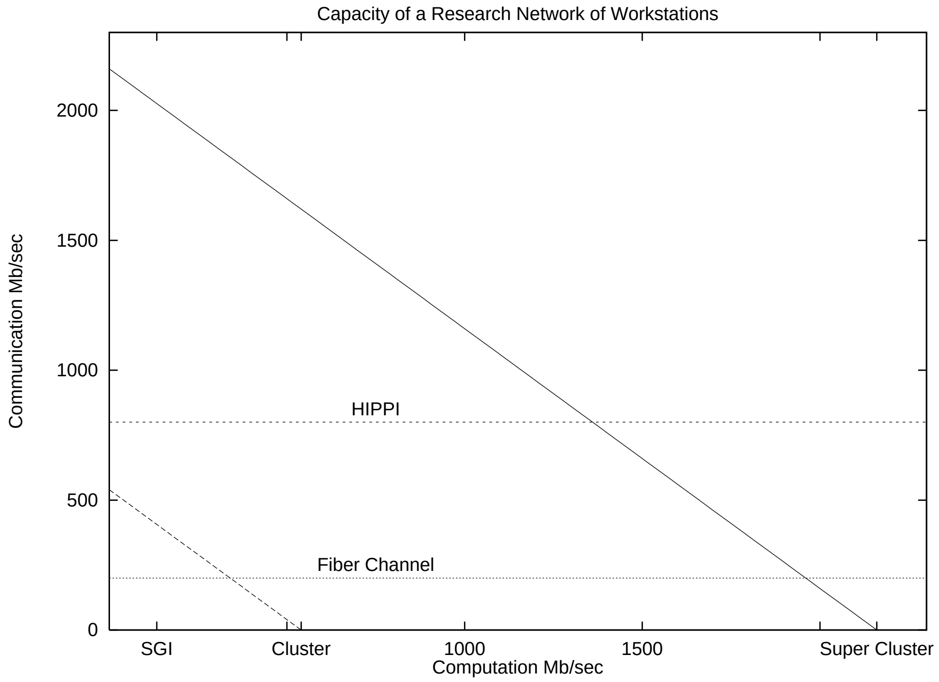

Figure 11: System Capacity Model for Research Networks of Workstations. The SGI machines are Challenge XLs. The cluster has 4 SGI Challenge XL workstations. The super cluster has 4 four-processor SGI Challenge XL workstations.

Processor	Network	Capacity	Communication Surplus	Computation Surplus
Intel 386-20	Modem	0.02	0.00	1.00
Intel 486-33	Modem	0.14	0.00	1.00
Intel 386-20	ISDN	0.07	0.00	1.00
Intel 486-33	ISDN	0.63	0.00	1.00
Intel 386-20	Ethernet	0.72	1.00	0.00
Intel 486-33	Ethernet	49	1.00	0.00
Home Cluster	Ethernet	61	0.99	0.01
Sun 3/80	Ethernet	9	1.00	0.00
Sun SLC	Ethernet	42	1.00	0.00
Sun Sparc Classic	Ethernet	73	0.95	0.05
Sun Sparc 5	Ethernet	290	0.15	0.85
Academic Cluster	Ethernet	540	0.04	0.96
SGI Challenge	Ethernet	140	0.01	0.99
SGI Challenge	Fibre Channel	8400	0.90	0.10
Research Cluster	Fibre Channel	48000.00	0.05	0.95
SGI Challenge	HIPPI	8900	1.00	0.00
Research Cluster	HIPPI	140000	1.00	0.00
Super Research Cluster	HIPPI	1400000	0.26	0.74

Table 7: System Capacity Model for Various Processors and Networks.

As expected, home networks of workstations have the least capacity. Academic networks of workstations have an almost 10 times greater capacity, even when the home network of workstations has an Ethernet. Research networks of workstations have by far the most capacity, with the research cluster having over two orders of magnitude more capacity than the academic cluster.

Home networks of workstations have severe potential network bottlenecks, as evidenced by their large computation surplus'.

While a single workstation on an academic network of workstations does have a large communication surplus, the academic cluster may have the network as a bottleneck.

Research networks of workstations have large potential network bottlenecks with an Ethernet. A Fibre channel puts the potential bottleneck in the processor. A research cluster once again puts the potential bottleneck in the network. However, a HIPPI network puts the potential bottleneck in the processor, even for a research cluster. A super research cluster and a HIPPI has a fairly equal computation and network surplus.

6 Conclusion

In our micro experiments, we did careful measurements of processor loads for sending packets. In our macro experiments, we verified Hypothesis 1 from Section 1 that processor sending load can be used to accurately predict maximum processor sending bandwidth. In our analysis, we developed a model for visualizing and measuring the capacity of a processor and network. Our model allows us to examine current networks of workstations for possible application bottlenecks. We applied our model to three networks of workstations to test our Hypotheses 2 through 4 from Section 1. Results:

- Based on our analysis, high performance workstations, such as those found in a typical research network of workstations, need a higher bandwidth than can be supplied by an Ethernet. But the fastest networks have too much bandwidth for even clusters of high-performance workstations. However, clusters of the most powerful, multi-processor workstations have an equal chance for network and processor application bottlenecks.
- Single workstations on an academic network of workstations may have processor bottlenecks. While academic clusters may have network bottlenecks.
- Home networks of workstations very likely have network bottlenecks when connected via ISDN or modem.

Future work includes:

- *Applications.* In our model, applications may be represented as points based on their computation and communication requirements. An application that existed within a system capacity area could then be run by that system. Multiple applications may be represented by adding their communication and computation requirements.

- *Efficiency.* Their may be a notion of the 'best' system, based on dollar cost, application requirements, or application quality. Efficiency would then be a comparison of an existing system capacity to the best system capacity.
- *Verification.* We assumed that the prediction for maximum processor bandwidth holds when we consider multiple processors and workstation clusters. We would like to validate this hypothesis or find out the true relationship.
- *Capacity Rate.* For a given system, we can determine the capacity increase for a given network or processor increase. Coupled with a simple cost analysis, the capacity rate may provide a dollar per capacity increase measure, directing system spending during updates or in system purchases.

References

[1] Spec primer. July 1994. The SPEC primer is frequently posted to comp.benchmarks. SPEC questions can also be sent to spec-ncga@cup.portal.com.

[2] M. Claypool, J. Riedl, J. Carlis, G. Wilcox, R. Elde, E. Retzel, A. Georgeopolous, J. Pardo, K. Ugurbil, B. Miller, and C. Honda. Network requirements for 3d flying in a zoomable brain database. *IEEE JSAC Special Issue on Gigabit Networking*, 13(5), June 1995. To appear.

[3] Mark Claypool and John Riedl. Silence is golden? The effects of silence deletion on the CPU load of an audio conference. In *Proceedings of IEEE Multimedia*, Boston, May 1994.

[4] Jack J. Dongarra. Performance of various computers using standard linear equations software. Technical Report CS-89-85, University of Tennessee, February 1994. To obtain a postscript copy of the report send mail to netlib@ornl.gov and in the message type: send performance from benchmark.

[5] J.L. Gustafson and Q.O. Snell. Hint: A new way to measure computer performance. In *Proceedings of the 28th Annual Hawaii International Conference on System Sciences*, volume 2, pages 392 – 401, IEEE Computer Society Press, January 1995. Technical report available via ftp from ftp.scl.ameslab.gov.

[6] John L. Hennessy and David A. Patterson. *Computer Architecture: A Quantitative Approach.* Morgan Kaufmann Publishers, Inc., 1990.

[7] Edward D. Lazowska, John Zahorjan, David R. Cheriton, and Willy Zwaenepoel. File access performance of diskless workstations. *ACM Transactions on Computer Systems*, 4(3):238–268, August 1986.

[8] Yean-Shiang Leu and David H.C. Du. Cycle compensation protocol: A fair protocol for the unidirectional twin-bus architecture. *IEEE Transactions on Computers*, January 1994.

[9] Mengjou Lin, Jenwei Hsieh, David H.C. Du, Joseph P. Thomas, and James A. MacDonald. Distributed network computing over local atm networks. *ATM LANs: Implementation and Experience with An Emerging Technology*, 1995.

[10] John Riedl, Vahid Mashayekhi, Jim Schnepf, Mark Claypool, and Dan Frankowski. Suitesound: A system for distributed collaborative multimedia. *IEEE Transactions on Knowledge and Data Engineering*, August 1993.

[11] John F. Shoch and Jon A. Hupp. Measured performance of an Ethernet local network. *Communications of the ACM*, 23(12):711–720, December 1980.