A Design Study of Alternative Network Topologies for the Beowulf Parallel Workstation

Chance Reschke Thomas Sterling Daniel Ridge
Center of Excellence in Space Data and Information Sciences (CESDIS)
Code 930.5 NASA Goddard Space Flight Center
Greenbelt, MD 20771
{creschke, tron, newt}@cesdis.gsfc.nasa.gov

Daniel Savarese
Department of Computer Science
University of Maryland
College Park, MD 20742
dfs@cs.umd.edu

Donald Becker Phillip Merkey
CESDIS
Code 930.5 NASA Goddard Space Flight Center
Greenbelt, MD 20771
{becker, merk}@cesdis.gsfc.nasa.gov

Abstract

Coupling PC-based commodity technology with distributed computing methodologies provides an important advance in the development of single-user dedicated systems. Beowulf is a class of experimental parallel workstations developed to evaluate and characterize the design space of this new operating point in price-performance. A key factor determining the realizable performance under real-world workloads is the means devised for interprocessor communications. A study has been performed to characterize a family of interconnect topologies feasible with low cost mass market network technologies. Behavior sensitivities to packet size and traffic density are determined. Findings are presented which compare more complex segmented topologies to the earlier parallel "channel bonded" scheme. It is shown that in many circumstances the more complex topologies perform better, and in some circumstances software routing techniques compare favorably to more expensive hardware switch mechanisms.

1 Introduction

The potential for integrating mass-market commodity subsystems in desk-side ensembles to achieve high performance computing at low cost has been demonstrated by the Beowulf Parallel Workstation for single-user environments. Beowulf is a class of experimental scientific workstations incorporating up to 16 PC-technology derived processing nodes, integrated by means of multiple commodity networks. Once limited to undemanding consumer tasks, PC-based system technology is now performance competitive with workstation class computers while excelling in price-performance. Thus a Pile-of-PCs or PopC ("pop-see") approach is emerging to complement the Cluster of Workstations (COW) or Network of Workstations (NOW) [3] distributed computing approaches using more costly subsystems, such as the Princeton SHRIMP project [1], which incorporates custom communication hardware. Beowulf has been used to explore this new point in the design space for scientific workstations by exploiting a degree of parallelism rarely encountered within the constraints of a dedicated end-user terminal. It has been shown that a 16 node distributed system exclusively composed of PC-market derived subsystems can provide peak performance in excess of 1 GOPS and ten times the disk capacity and bandwidth routinely provided by high-end scientific workstations at comparable cost. This capability has been realized through a balanced system structure and enhancements to the Linux operating system, providing a user interactive interface in many respects similar to that typically found in conventional environments. Key factors determining the viability of this approach are the achievable interprocessor network bandwidth and the system support software for implementing effective parallel disk I/O. This paper addresses the first of these two issues. It presents new findings that quantify the interconnect design trade-off space and demonstrate the potential importance of alternate network topologies, even in the context of modest system configurations such as those implicit in the Beowulf approach.

Even in the context of a single-user workstation, employing multiple processors benefits both execution performance and disk access bandwidth. An additional consequence is the large disk capacity achieved at low cost through multiple commodity disk drives. This enables, for example, very large scientific data-sets to be temporarily buffered locally by the parallel workstation throughout a session of data browsing and visualization, avoiding repeated accesses to remote file servers. User response time is significantly reduced as is the burden on shared LAN and remote file server resources. But these advantages come at the expense of requiring an internal interconnection network of sufficient capacity to meet the demand of interprocessor data transfers, a problem not encountered on uniprocessor workstations. The experimental Beowulf workstation has been used to explore the feasibility of employing multiple cost-effective commodity networks in parallel to satisfy these internal data transfer rate requirements. User-transparent access to multiple parallel Ethernet networks across the processor nodes was achieved by "channel bonding" techniques developed through enhancements to the Linux operating system. In previous work it was shown that up to 3 networks could be ganged to achieve significant scaling of sustained throughput [8], validating the channel bonding method. Additional studies were performed with the new 100 Mbps Fast Ethernet demonstrating effective utilization of dual channels [10]. However, Fast Ethernet is only beginning to become cost effective with sustainable performance to cost of the new technology now approaching that of the older 10baseT interconnects. In either case, each network connected all nodes within Beowulf.

These commodity networks are essentially multi-drop media and are well suited to system configurations in which each network channel connects all nodes. However, it was recognized that the potential for higher performance exists through more complex topologies, at possibly somewhat greater cost. The weakness of any multidrop network is that only one communication packet may be transferred at a time. Channel bonding permitted the number of simultaneous packets transferred to, at least in principal, be equal to the number of parallel channels employed. But these are limited by both cost and the number of network controllers that any processor node can practically support. An alternative approach for increasing the peak aggregate bandwidth with comparable resources is network segmentation. A given channel is divided into a number of non-overlapping sub-channels. Each subchannel can support independent packet transfers unless nodes on separate sub-channels must interact. A peak throughput gain of a factor of 4 is theoretically possible for Beowulf using segmentation at the cost of fully interconnected non-blocking switches. Through more complex topologies, still using the basic Ethernet technology, much of the possible sustained throughput gain might be realized without resorting to the extreme requirement of a fully connected switch for each channel.

This paper presents the experimental results of tests to characterize and quantify the design tradeoff space for these more complex topologies and compare the performance achieved to the original channel bonded approach. Both hardware switching among segments as well as software routing mechanisms were evaluated under synthetic packet generation workloads and parallel file copying applications. The following section specifies the basic architecture attributes of Beowulf parallel workstations. Section 3 discusses the Linux based software environment provided on Beowulf including the enhancements made to support interprocessor communication. The experimental methods employed to characterize the internal system communications and the results of the experiments are provided in Section 4. These findings are analyzed and their implications discussed in Section 5. A summary of the overall findings and their importance for future directions is concluded in Section 6.

2 Beowulf Architecture

The CESDIS Beowulf parallel workstation architecture, an emerging standard for PopC clusters, realizes high-performance distributed computing from strictly commodity hardware. The experiments detailed here were performed on the original Beowulf prototype, a 16 node cluster with each node running a copy of the Linux operating system and configured as follows:

- Intel 486 DX4 microprocessors at 100MHz
- SiS471 chipset
- 256K asynchronous cache
- 16MB 60ns RAM
- 540MB IDE disk
- dual 10Mbps NICs (Network Interface Cards)

The original Beowulf network topology consisted of a pair of Ethernet busses that spanned the entire cluster and operated in parallel as a single virtual bus.

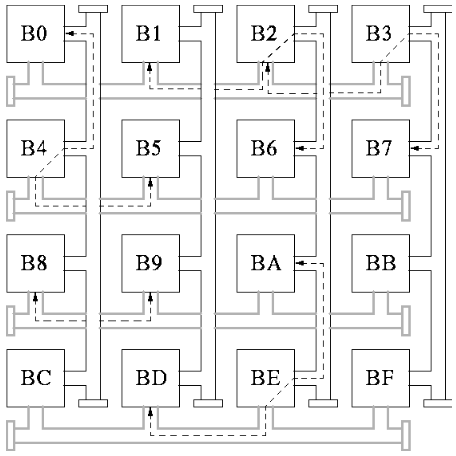

Figure 1. Software Routed Network Configuration

This configuration's aggregate network bandwidth of 20Mbps was a serious limiting factor on the cluster's performance. The alternative configurations explored here retain the 10Mbps technology and dual NICs per node of the original prototype but increase the aggregate network bandwidth by creating eight separate Ethernet busses. Each node attaches to two busses, each bus spans four nodes as shown in Figure 1. A variation on this interconnect scheme, depicted in Figure 2, adds two four-port Ethernet switches; the four vertical network segments are connected to each other this way as are the horizontal networks. While the theoretical maximum aggregate bandwidth has quadrupled, there are constraints on the effective aggregate bandwidth and performance.

The original Beowulf network topology (Bonded Dual Net) provided each node with a direct route to all nodes in the cluster. From Figure 1 we can see that the unswitched alternative topology (Routed Mesh) allows any node in the cluster to communicate directly with only six other nodes (local nodes), the three nodes with which it shares a vertical network segment and the three nodes with which it shares a horizontal network segment. For a node to communicate with the nine remaining nodes (remote nodes), an intermediate node which is local to both the sender and receiver must act as a router, forwarding the communication across network segments. Any two nodes may communicate using no more than one intermediate router node. The two consequences of the alternative topologies are:

- Routed packets consume bandwidth. A routed packet is replicated on two segments and consumes twice the aggregate bandwidth of non-routed packets.
- Routing is not free. There is an additional latency cost associated with routing, be it in hardware or software, although this cost is higher in software.

The experiments described in Section 4 do not include replicated packets in their measure of aggregate throughput. The bandwidth consumed by these extra packets actually reflects the degradation in performance for remote node transactions discussed in Section 5. Another factor reflected by the experiments is that the Beowulf architecture makes no distinction between I/O nodes and compute nodes. Packet routing consumes both CPU cycles and bus to memory bandwidth that causes contention with other operations such as file I/O.

In the experiments, the logical topology is static and constructed such that the routing load is distributed fairly for a uniform traffic case. The routed mesh scheme requires no hardware not already necessary for the original bonded dual net topology. The switched mesh topologies (Switched Mesh, Bonded Switched Mesh) retain the eight-segment basic physical interconnect of the routed scheme, but add high-speed Ethernet switches to perform the routing between segments. To avoid becoming a bottleneck, a switch backplane must support more bandwidth than the network segments. The experiments conducted on switched topologies would reveal any impact of the switch backplane on aggregate bandwidth.

A notable feature of the original logical interconnect was channel bonding (Section 3); this allows two or more network segments to be joined into a single logical segment in a fashion completely transparent to applications. While the original interconnect (Bonded Dual Net) bonded two identical networks into a single logical segment, channel bonding can be applied

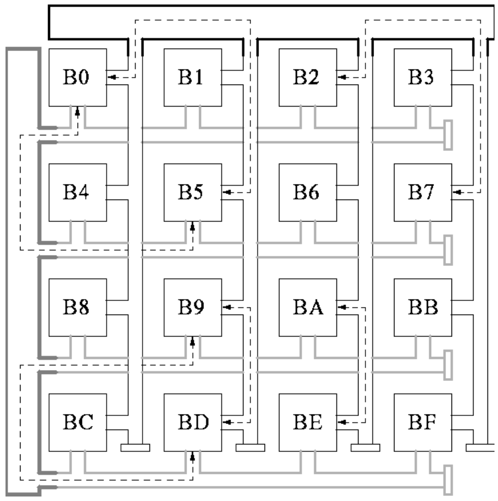

Figure 2. Switched Network Configuration

whenever there are multiple possible routes between two nodes. This technique has been applied to the switched interconnect scheme, and its effect on performance is presented and discussed in Sections 4 and 5.

3 System Software

The Beowulf Parallel Workstation is based on the Linux operating system. Linux [6] is a full-featured clone of the UNIX operating system originally designed for *x86* processors and recently extended to support other architectures. Its feature set includes POSIX compatibility, a TCP/IP protocol stack with a BSD-compatible sockets interface, very broad device support, dynamically linked shared libraries, interprocess communication and an efficient virtual memory subsystem with unified buffer cache.

A feature especially important to Beowulf is that Linux is a *free* implementation of UNIX. It is distributed under the terms of the Free Software Foundation's GNU Public License which insures that source code to the system is available, that we can easily share improvements, and that there are there no per-node royalties. This last feature is important, and not only financially because the administrative overhead associated with adding and removing nodes from a Beowulf cluster could be substantial.

Linux supports most of the available portable programming tools for distributed environments. The most popular system has been the public domain version of PVM [11], the *Parallel Virtual Machine* library. Other tools such as MPI [7] and an RPC library are also available.

Scalable communications is achieved by a technique called "channel bonding" [8]. With our implementation, the hardware address of a primary network adaptor is duplicated on the secondary interfaces, and all packets received on the bonded networks are marked as if they came from the primary interface. This scheme requires each Ethernet segment to span all nodes; switches provide a transparent connect between segments. With this constraint the Ethernet packet contents are independent of the actual interface used. The software routing overhead of handling more general interconnect topologies is avoided; the only additional computation over using single network interface is the computationally simple task of distributing the packets over the available device transmit queues. The current method used is initially alternating packets among the available network interfaces to the point that their private queues are full, and then distributing additional packets to the not-yet-full queues.

The Routed Mesh topologies take advantage of IP packet forwarding, a standard feature of the Linux kernel. This allows the kernel to function as an IP packet router, accepting a packet on one interface, determining that it must be forwarded, potentially fragmenting it, and transmitting it out a second interface. While the Ethernet adapters used on Beowulf are "bus masters" that reduce CPU load by copying packet data to and from main memory autonomously, the machine doing the routing still has the software load of handling a receive interrupt, allocating temporary space for the forwarded packet and copying it there, consulting its routing table, and placing the packet on the target interfaces' transmit queue.

4 Experimental Methods and Results

In this section, methods used to evaluate the relative communications capacity of five alternative network configurations are described in conjunction with the resulting experimental data. A synthetic program, the Network Throughput test, and a parallel file copy application, the Disk/Net Balance test, were run on each network configuration to make the comparison. Both experiments used TCP/IP as the networking protocol, and results were collected on a node not otherwise involved in the experiment.

With all of the mesh configurations evaluated here, the relationship between a pair of nodes falls into one of two categories. When the members of a pair of nodes share one of the two network segments to which they are connected, communication between them is referred to as a *local-wire* transaction. Alternatively, when the members of a pair of nodes share no common network segments, communication between them is referred to as a *remote-wire* transaction. For those network configurations where the local-wire/remote-wire distinction was meaningful (all segmented topologies), a complete set of measurements was collected for local-wire only and remote-wire only transactions. In all cases, the experiments were designed to avoid contention between transactions to the extent possible.

Three principal features describe the experiments that were conducted in evaluating the different network configurations:

Routing system - Software or switch based.

Number of routes for a packet - Channel bonding allows two paths, no channel bonding allows only one.

Location of send/receive processes - Some of the experiments placed all send/receive pairs on nodes sharing the same physical link (local-wire), while the rest of the experiments placed send/receive pairs on nodes on different physical links (remote-wire).

These situations are encapsulated by Figures 1 and 2. The figures themselves are differentiated by routing system, Figure 1 representing software routing and Figure 2 switch based routing. The two packet paths from B0 to B5 in Figure 2 show channel-bonding as well as illustrating remote-wire transactions. In Figure 2, the BA to BE packet path illustrates a local-wire transaction.

4.1 Internode Throughput

In the Network Throughput test, one node (a producer) generates a message (a token) and then sends it to a second node (a consumer), which then returns the token to the producer. A node is either a producer or a consumer, never both, and is a member of, at most, one producer-consumer pair. The network load in this experiment was increased by increasing the token size over a range from four bytes to eight kbytes, and by increasing the number of producer-consumer pairs over a range from one to seven. Several runs consisting of several thousand exchanges were performed for each combination of token size and number of producer-consumer pairs. For each combination, the best observed throughput for the system was recorded. Throughput was measured as the number of tokens written multiplied by the token size and divided by the elapsed time. The results of this experiment are presented in Figure 3.

Not surprisingly, the local-wire experiments achieved the highest throughput, peaking at 6.3 MB/s for an 8 KB token size. This is almost a factor of four improvement over the original bonded dual net Beowulf configuration. The remote-wire experiments succeed in bettering the original Beowulf performance, showing almost a factor of 2 improvement, but are unable to match the local-wire experiments. This is not unexpected because the remote-wire experiments require that packets be duplicated through either software or switch routing mechanisms,introducing a latency penalty caused by contention for network resources and packet routing overhead. Unexpectedly, the switched channel bonded configurations did not exceed the performance of the equivalent switched non-channel bonded schemes, actually demonstrating marked degradation in throughput. These results are further discussed in Section 5.

The Switched Mesh Local-Net and the Routed Mesh Local-Net experiments are operationally identical and exhibit the same behavior. Because in both cases all transactions are between independent, local wire pairs, no messages must be passed through either a switch or a routing node. In other words, all transactions are of the type between nodes B8 and B9 in Figure 1 and nodes BA and BE in Figure 2. Therefore all transactions avoid the additional latency incurred by packet duplication through a switch or routing node in reaching a remote node.

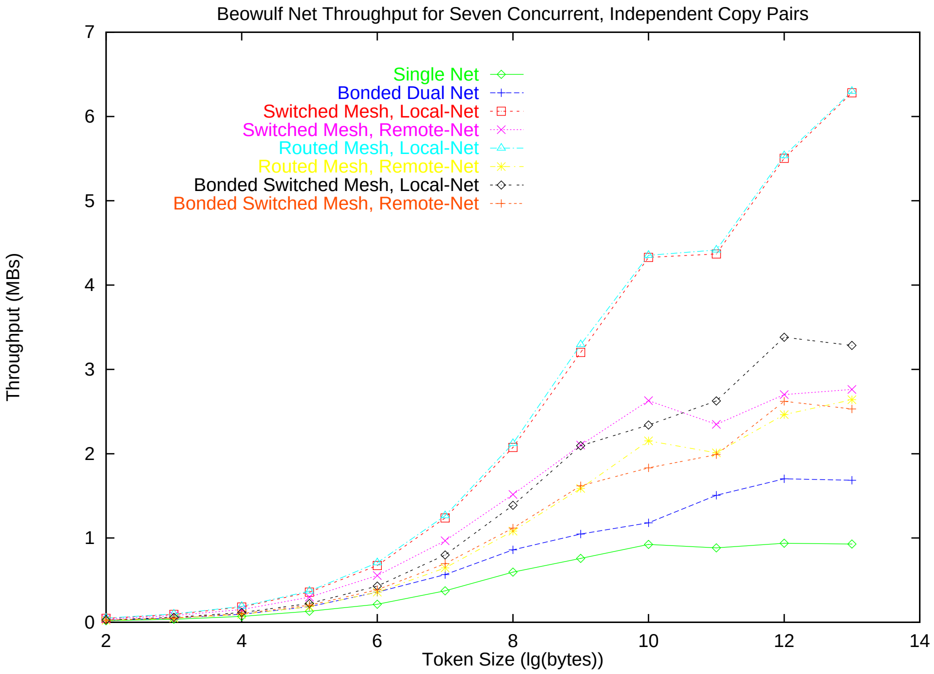

Figure 3. Network Throughput Test

4.2 Parallel Disk I/O

In the Disk/Net Balance test, a producer node copies a file from its local disk (the Linux buffer cache is flushed before each copy) to a consumer node. The consumer can be either the producer node itself (an intranode copy) or a separate node in the system (an internode copy). Each node is a member of a maximum of one producer-consumer pair. To evaluate the degree to which the system disk and network subsystems achieve balanced performance over a range of conditions, measurements were made with seven producer consumer pairs for all possible combinations of intranode and internode file copies. This series of measurements was performed with files sizes ranging from one to 16 MBytes. Again, system throughput was measured as the number of files copied times the files size, divided by the time needed for all copies to complete. Because of space considerations, we only present the results for the 2 MB file copies in Figure 4 to elucidate the performance differences of the studied network configurations.

The disk I/O experiments mirror the results of the network throughput tests, demonstrating the overall poorer performance of the channel bonded network configurations and the superiority of local-wire transactions over remote-wire ones. With respect to interprocessor copies, a peak sustained throughput of about 6 MB/s was achieved by both the software routed and switched local-wire tests, while the channel bonded configurations were able to peak at 7.2 MB/s for one remote file copy, before degrading. The one remote file copy data shows more clearly than the network throughput tests channel bonding's ability to better handle small workloads. These data are discussed in more detail in the following section.

5 Discussion

The experiments described in the previous section exhibit a varied set of behaviors, demonstrating the performance characteristics of different network topologies on the Beowulf Parallel Workstation. These results are discussed in relation to the original Beowulf topology, designated as Bonded Dual Net in Figures 3 and 4. In this configuration, each of the Ethernets independently connects all 16 nodes and each node uses channel bonding to balance its communications between the two networks. Unlike previous studies [8, 9, 10], where it was clear that a channel bonded dual net configuration was always desirable, there are now situations where one topology may be preferable to another depending on application requirements. It is also seen that with the alternative topologies, significant performance gains over the original Beowulf configuration can be achieved.

An immediate observation to be gained from the

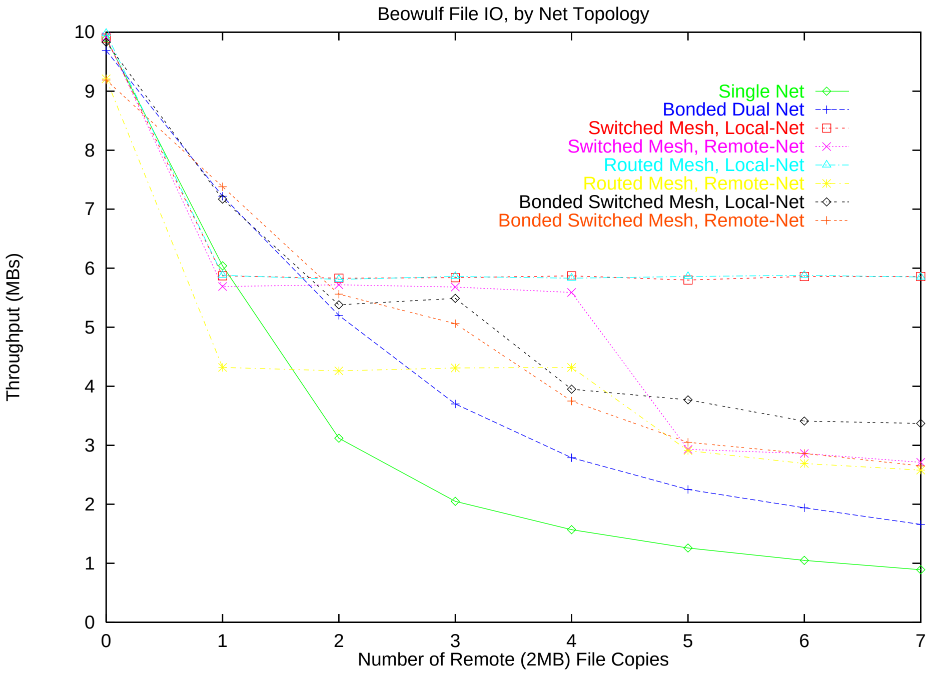

Figure 4. Disk/Net Balance Test

network throughput experiment (Figure 3) is that the alternate network topologies consistently outperform the two original interconnect schemes (Single Net and Bonded Dual Net) presented in [8, 9]. As one would expect, the local-wire experiments achieve the highest throughput, peaking at 6.3 MB/s for an 8 KB token. This bests the original Beowulf Bonded Dual Net configuration by a factor of 3.7, which is close to the theoretical limit of 4. The roughly equivalent performance of the 1 and 2 KB tokens reflects IP packet fragmentation caused by the Ethernet 1536 byte maximum packet size [2]. While the same number of tokens are being exchanged for both cases, the 2 KB case actually generates twice as many Ethernet packets; each 2 KB token is broken up into two Ethernet packets, while each 1 KB token fits in one Ethernet packet.

As expected, there is no meaningful difference between the Switched Mesh Local-Net and the Routed Mesh Local-Net data because as indicated in the previous section, both are operationally identical. In the case of remote-wire transactions, the software routed scheme demonstrates lesser performance than the switched scheme. This can be attributed to the overhead introduced as part of the critical path time for performing the packet routing in software and also the contention for processor resources between routing software and packet generating code on a routing node. For the switched network configurations, the switch backplane does not appear to be a bottleneck even though the switching units used were of relatively low cost. A benefit of the software routed network configurations is that they involve no additional equipment cost. Switch based network approaches increase the cost of Beowulf by about 5%.

The sensitivity of interconnect throughput with respect to message demand is shown in Figure 5 where the curves presented differ in the number of messages being generated simultaneously. This is for the case of software routed remote packet transfers; every packet must travel through an intermediate node for routing. These results show that throughput increases with message demand as it does with packet size. As the traffic density increases, the throughput grows but less than linearly. The first four curves from the bottom represent an operation in which no node performs more than one of three roles: token supplier, token router, or token consumer. But beyond that, some contention occurs as overlapping of some paths result in one or more nodes doing double duty. Beyond 1 KB packet size, tokens are fragmented into multiple packets. The first point at which this occurs yields no throughput gain because the overhead increases with the amount of information transferred and the packets are not fully filled producing reduced efficiency. But beyond this point, the majority of multiple packets are fully filled and significant performance increase is observed with token size. For the top three curves, the contention for system resources results in poorer throughput increase

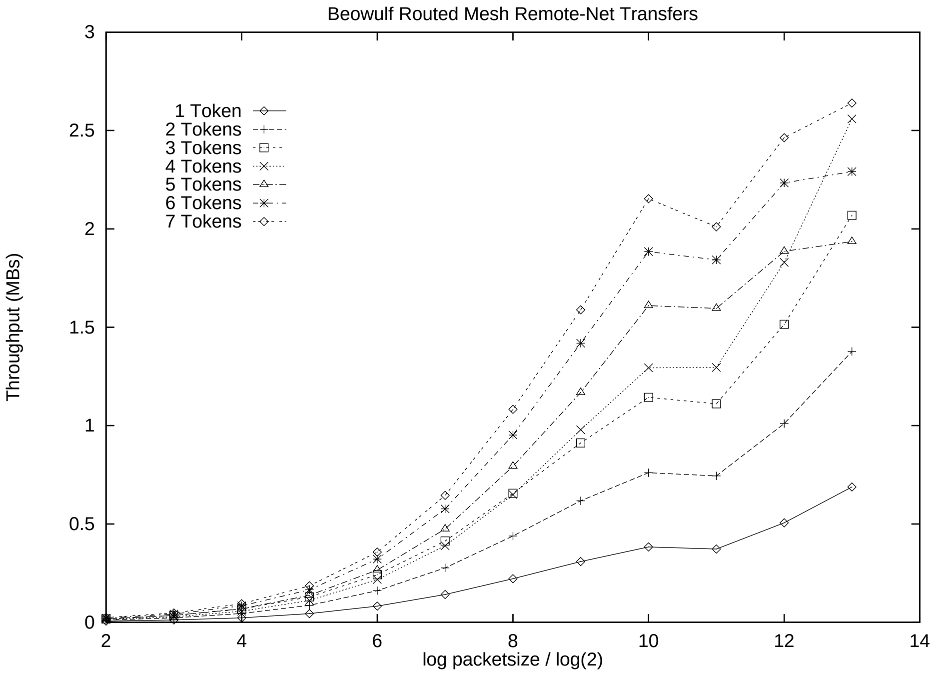

Figure 5. Sensitivity of Throughput to Message Demand

and at some points negative throughput increase. This is most dramatically demonstrated with larger token sizes where the lower message demand case (fewer tokens) results in greater throughput than the high message demand case (more tokens).

Bonded Switched Mesh Local-Net throughputs are about half that of the non-channel bonded local transaction cases due to the packet duplication caused by switch routing. The two data packet paths from nodes B9 to BD in Figure 2 illustrate this behavior. One path traverses only a local wire while the other path must traverse a switch in reaching the local node, effectively causing the local transaction to be a remote one. In the switched scheme, channel bonding causes successive packets to alternate paths to their destination, resulting in half of the traffic taking a longer route and incurring the latency of packet duplication. In light of the data, the channel bonded configuration is not desirable when most message traffic in an application will be between nodes sharing a local wire. While this indicates that special conditions must exist to attain the best network performance, this is a reasonable condition. Parallel programmers often tailor their codes to specific topologies on machines like the Intel Paragon [5] and Cray T3D [4]. This is not to say that it is desirable to force the programmer to contend with architectural details in writing his code, but rather to point out that while Beowulf does not relieve this problem, it does not impose any new impediments.

The Disk/Net Balance experiments were conducted to determine the operating balance between the disk and network I/O systems. A balanced machine should match the disk system's ability to generate data with the network's capacity to convey the traffic. As in the network throughput experiments, the local-wire disk transfers attain the highest sustained throughput, maintaining a rate of about 6 MB/s. However, the channel bonded network configurations exceed that performance for 1 remote file copy, achieving 7.2 MB/s, before degrading as more remote file copies are added. Channel bonding is able to improve throughput for light workloads by distributing traffic between two paths, but for heavier workloads, the increased latency inherent to remote-wire transactions eliminates the usefulness of this load-balancing method.

Again looking at the one remote file copy situation, the software routed network with remote-wire transactions exhibits the lowest throughput at just over 4 MB/s. This particular situation highlights the overhead associated with software routing. The equivalent switched network experiment achieves a throughput 1.7 MB/s higher, enabled by the faster routing of hardware switching. The performance benefit of switching can be seen by comparing the switched remote-net experiment to the software routed remote-net experiment. Up until 4 internode file copies, the switched configuration consistently outperforms the software routed configuration by about 1.5 MB/s. After 4 internode file copies, the attained throughput converges. This can be explained by the way the experiments were conducted. Up until 4 copies, no file transfer suffers from contention for access to any given wire. Starting with 5 file copies, contention is introduced. The situation is illustrated in Figure 1. The transfer between nodes B1 and B6 contends for the same wire as the transfer between B2 and B7, and B2 acts as both an end-node and a router.

From the results of the Network Throughput and Disk/Net Balance experiments, it can be said that throughput is sensitive to 3 factors: message size, available aggregate bandwidth, traffic density/demand. Message size determines the degree of utilization of network resources. Small messages do not take full advantage of the Ethernet packet size, while larger ones cause message fragmentation and increased traffic. Available aggregate bandwidth sets the upper bound for attainable throughput. Traffic density and demand determine to what degree the available bandwidth is utilized, in addition to establishing the amount of network contention. Although the performance measurements do not measure one-to-many or many-to-one transactions, we expect that the essential performance attributes of the system would be preserved under such conditions. Specifically, the channel bonded configurations would excel under a light workload, while the single-channel configurations would perform better under greater traffic.

6 Conclusions

Small ensembles of commodity PC-derived processing subsystems have been shown to be performance competitive with scientific workstations at lower cost while providing dramatically improved secondary storage capacity and bandwidth. Sustained performance for distributed computation is sensitive to the interprocessor communications network employed. This paper has presented findings of experiments conducted with the Beowulf parallel workstation to characterize the design space of interconnect topologies that may be implemented using strictly mass market network technology.

The original networking strategy for Beowulf was to use multiple Ethernet networks in parallel, each connecting all the nodes within the system. Both 10 Mbps and the new 100 Mbps Fast Ethernet were employed in separate Beowulf systems. The parallel networks were managed through a technique called channel bonding that uniformly distributed packets among the interconnects in a manner transparent to the user code.

Synthetic programs that generated a controlled network demand were used to compare the original Beowulf topologies to the two new segmented topologies. Both segmented topologies used eight separate Ethernet segments arranged as if the nodes formed a 4×4 two dimensional grid with 4 segments connecting rows and 4 segments connecting the columns. In the Software Routed topology, traffic between nodes that do not share a segment uses an intermediate node as a router. In the Switched topology, the four horizontal and four vertical segments are connected by two 4 port switches. Routing in each of these topologies was statically set.

As anticipated, sustained throughput was highly sensitive to packet size and traffic demand. However, the relationship was not always simple. As traffic increased, sustained throughput increased to the point where contention became an important source of performance degradation. As packet size increased, in general there was a tendency towards higher throughput until saturation for the given channel configuration was achieved.

In almost every situation evaluated, the new segmented topologies demonstrated superior performance to the dual network channel bonded technique using the same number of communicating producer/consumer process pairs. Two interesting situations encountered in the study are: software versus switched routing under light traffic and channel bonded switched versus single-net switched routing under heavy traffic. Software routing yielded performance near that of the hardware switch based systems except in light traffic conditions where the software latency became a dominant factor. In the case of high traffic demand on the switched topology, the additional load caused by packet replication inherent to routing for the channel bonded configurations actually reduced the effective aggregate throughput compared to the non-bonded configuration.

The importance of this study is its immediate application to real-world systems. The data in this paper can be used as a direct guide to configuring the hardware and software of interprocessor communications in a Beowulf class system. While the experiments were performed for 10 Mbps technology, it directly relates to the higher bandwidth Fast Ethernet, although the switch costs for this emerging technology are still high. For example, an important intermediate step will be to use Fast Ethernet without switches but rather with the lower cost repeaters and software routing. It can be anticipated that many of the performance attributes of 10 Mbps Ethernet will be retained.

Current work is being conducted to explore the implications of network topology in the Beowulf context driven by end-user applications. A range of problems taken from the Earth and space sciences community are providing well understood benchmarks with many different characteristics. It is anticipated that while some problems may exhibit sensitivities to the network choices, many others will be less dramatically impacted than the rather severe tests shown in this paper.

References

[1] M. Blumrich, K. Li, R. Alpert, C. Dubnicki, E. Felten, and J. Sandberg, "Virtual Memory Mapped Network Interface for the SHRIMP Multicomputer," *Proceedings of the Twenty-First International Symposium on Computer Architecture (ISCA),* Chicago, April 1994, pp. 142-153.

[2] D. Boggs, J. Mogul, and C. Kent, "Measured Capacity of an Ethernet: Myths and Reality," *WRL Research Report 88/4,* Western Research Laboratory, September 1988.

[3] K. Castagnera, D. Cheng, R. Fatoohi, et al. "Clustered Workstations and their Potential Role as High Speed Compute Processors," *NAS Computational Services Technical Report RNS-94-003,* NAS Systems Division, NASA Ames Research Center, April 1994.

[4] Cray Research, Inc., "CRAY T3D System Architecture Overview," Eagan, Minnesota.

[5] Intel Corp., "Paragon User's Guide," Beaverton, Oregon 1993.

[6] Linux Documentation Project, `http://sunsite.unc.edu/mdw/linux.html`.

[7] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, "MPI: The Complete Reference,", The MIT Press, Cambridge, Massachusetts, 1996.

[8] T. Sterling, D. Becker, D. Savarese, et al. "BEOWULF: A Parallel Workstation for Scientific Computation," *Proceedings of the 1995 International Conference on Parallel Processing (ICPP),* August 1995, Vol. 1, pp. 11-14.

[9] T. Sterling, D. Savarese, D. Becker, B. Fryxell, K. Olson, "Communication Overhead for Space Science Applications on the Beowulf Parallel Workstation," *Proceedings of the Fourth IEEE Symposium on High Performance Distributed Computing (HPDC),* August 1995, pp. 23-30.

[10] T. Sterling, D. Becker, D. Savarese, M. Berry, C. Reschke "Achieving a Balanced Low-Cost Architecture for Mass Storage Management through Multiple Fast Ethernet Channels on the Beowulf Parallel Workstation," *to appear in Proceedings of IPPS 96*

[11] V. Sunderam, "PVM: A Framework for Parallel Distributed Computing," *Concurrency: Practice and Experience,* December 1990, pp. 315-339.