Achieving a Balanced Low-Cost Architecture for Mass Storage Management through Multiple Fast Ethernet Channels on the Beowulf Parallel Workstation

Thomas Sterling Donald J. Becker
Center of Excellence in Space Data
and Information Sciences
Code 930.5 NASA Goddard Space Flight Center
Greenbelt, MD 20771
{tron, becker}@cesdis.gsfc.nasa.gov

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

Michael R. Berry
Department of Electrical Engineering
and Computer Science
The George Washington University
Washington, D.C. 20052
mrberry@seas.gwu.edu

Chance Reschke
Center of Excellence in Space Data
and Information Sciences
Code 930.5 NASA Goddard Space Flight Center
Greenbelt, MD 20771
creschke@cesdis.gsfc.nasa.gov

Abstract

Network-of-Workstations (NOW) seek to leverage commercial workstation technology to produce high performance computing systems at costs appreciably lower than parallel computers specifically designed for that purpose. The capabilities of technologies emerging from the PC commodity mass market are rapidly evolving to converge with those of workstations while at significantly lower cost. A new operating point in the price-performance design space of parallel system architecture may be derived through parallelism of PC subsystems. The Pile-of-PCs, PopC (pronounced "pop-see"), approach is being explored through the Beowulf Parallel Workstation developed to provide order-of-magnitude increases in disk capacity and bandwidth for a single user environment at costs commensurate with conventional high-end workstations. This paper explores a critical aspect of the architecture trade-off space for Beowulf associated with the balance of parallel disk throughput and internal network bandwidth. The findings presented demonstrate that parallel channels of commodity 100 Mbps Ethernet are both necessary and sufficient to support the data rates of multiple concurrent file transfers on a sixteen processor Beowulf parallel workstation.

1 Introduction

The Beowulf Parallel Workstation integrates off-the-shelf commodity subsystems to create a new operating point in price-performance for single-user scientific workstation environments. Beowulf's capabilities include a Gops peak performance, half a GByte of main memory, and disk storage capacity of 20 GBytes achieved at the cost of a conventional high-end scientific workstation (under $50K). These capabilities are accomplished through a parallel configuration of multiple processor subsystems, disks, and interconnection networks; all commodity components derived from the PC marketplace. The challenge is to define structures of such components that provide a balanced ensemble of resources in support of user needs. This paper looks at the specific problem of balancing parallel disk capacity and file transfer bandwidth with the message passing bandwidth of the system internal interconnect network.

To achieve high intra-system interconnect bandwidth, the Beowulf project has pioneered the use of multiple commodity networks within a single workstation. It has been shown [7] that network bandwidth can be scaled, at least across several parallel Ethernet networks (10 Mbps) to achieve useful sustained throughput gain. For example, dual Ethernets have delivered sustained bandwidth of almost 2 MBytes per second under favorable conditions of packet size. This was achieved in a user transparent manner through changes in the operating system. Beowulf employs the Linux operating system [4] which, among its other features, comes with source code and therefore is ideal for this class of research. Modifications to Linux were made to support channel bonding, allowing message packets to use any of the available networks.

Even with multiple networks and channel bonding, it was shown [6] that dual parallel 10 Mbps Ethernet can impose a bottleneck on disk file transfers under unfavorable conditions. One experiment revealed a discrepancy of about a factor of four with respect to disk throughput demand. Two approaches are being pursued by the Beowulf project. One approach alters the topology of interconnection to segment the two busses into four segments, each using moderate cost switches. The results of these experiments will be presented in a separate paper. The second approach is to exploit the very recent advances in 100 Mbps Ethernet (also referred to as *Fast Ethernet*). Only in the last few months has this technology reached the commodity market and at a price level commensurate with the objectives of the Beowulf workstation project. Early analysis showed that one such network might be marginal but that dual Fast Ethernets should be able to provide sufficient useful bandwidth to remove the internal interconnect as the limiting factor for transferring files among remote processor subsystems.

A new realm of system architecture has been created by the opportunities implied by the emerging low cost processor and network technology base. The PopC or Pile-of-PCs approach enables on-site configuration of essentially interchangeable components easily procured locally from multiple vendors and distributors. The flexibility in configuration permits the end user to match needs specific to the immediate workload requirements and to adapt resources as requirements evolve. However, PopC is still experimental and only made possible by the Linux operating system. Linux has provided a sophisticated and robust system software platform with source code availability and essentially no legal constraints. From this perspective, it is an almost unique tool for systems research and was the catalyst for PopC and the Beowulf experiment.

This paper presents the first published results of experiments conducted to evaluate multiple Fast Ethernet channels as the interconnection medium for a parallel workstation. Through empirical means, this new form of communication is characterized. It is shown that its capacity is not fully utilized by a single processor because of software limitations. However, it is demonstrated that multiple processor subsystems using a single channel concurrently can exploit most of its available capacity which is precisely the mode of use required by the Beowulf approach. Even in single source mode, achievable bandwidth will be shown to exceed sustainable disk file transfer rates. Finally, quantitative data will be provided that demonstrates a balanced architecture for mass storage management based on Fast Ethernet technology and channel bonding techniques employed by Beowulf.

2 Beowulf Architecture Characteristics

The Beowulf Parallel Workstation architecture comprises 16 processor nodes interconnected by multiple parallel Ethernet channels and includes a keyboard and dual high resolution screens. Each processor node combines an Intel x86 processor with memory, and disk storage, and network interfaces. The initial Beowulf prototype previously described [6] used the Intel 80486 DX4 (100 MHz) processor connected by VESA-local bus to 16 MBytes of memory and single 520 MBytes IDE disk drives. Dual 10 Mbps Ethernet channels provided system connectivity. The entire system is housed in a single half-height rack as shown in Figure 1.

The Beowulf philosophy is to provide a general structure that may track the rapid evolution of commodity technology, providing capability growth while minimizing the need for changes to underlying software. This approach has been followed in the implementation of the recently completed Beowulf Demonstration system. This new system retains the general Beowulf architecture described above but incorporates new components that are incremental enhancements of those making up the prototype. The Beowulf Demonstration system processor is the new Pentium (100 MHz) connected by a PCI bus to 32 MBytes of main memory and 1.2 GBytes of disk. As will be shown, the most important difference between the prototype and demonstration systems is that the latter employs the new Fast Ethernet technology, only now available in the commodity market. This network technology has a peak performance of 100 Mbps, 10 times that of the regular Ethernet used by the prototype system. Although substantially more expensive than regular Ethernet, the improved bandwidth was required to achieve a balanced system architecture, as will be demonstrated.

Figure 1: Beowulf Parallel Workstation

A major objective of Beowulf is to provide rapid access to disk storage. The two elements of the Beowulf architecture that impact the movement of *spinning-bits-to-pixels* (one of Beowulf's primary uses is scientific data visualization) is the rate at which data moves between the disk and memory and the rate at which data moves between memories on separate processor subsystems. The Beowulf prototype was the target of empirical studies to characterize its principal attributes. Of primary importance was the sustainable performance achieved using multiple Ethernets in parallel. The ability to gang Ethernet channels was a key factor in enhancing interprocessor communication through low cost technology.

To determine the scaling properties of parallel Ethernets (10 Mbps), a set of experiments was conducted whose essential findings are captured in Figures 2 and 3. The results of the same experiments on the new Beowulf Demonstration system are presented in Figures 4 and 5. The experimental method and results of both sets of experiments are discussed in Sections 3 and 4, while the implications for distributed computing are presented in Section 5.

3 Experimental Results

Two key elements of the Beowulf Parallel Workstation design are critical to the performance characteristics of the system as a whole. These factors are the sustained usable bandwidth of the network and the raw throughput of the disks. Two experiments were devised to measure the efficiency of the network and disks on the original Beowulf prototype and these were repeated on the Beowulf Demonstration system. The results for the prototype system are presented in Figures 2 and 3 and those for the

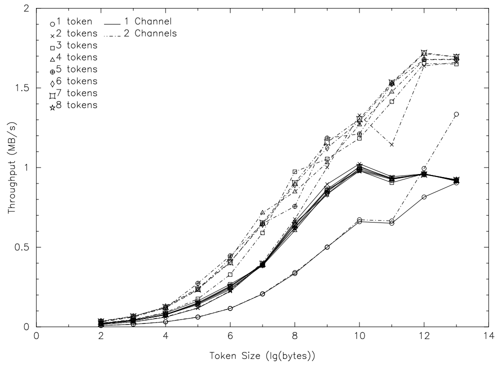

Figure 2: Beowulf Prototype Network Throughput

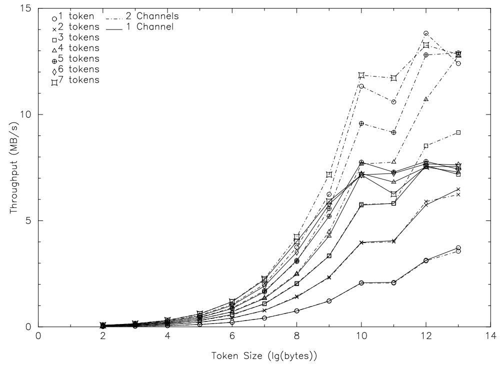

Figure 4: Beowulf Demonstration Network Throughput

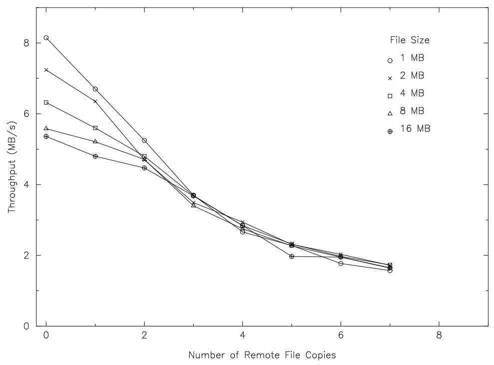

Figure 3: Beowulf Prototype File Transfers (2 channels, Total of 7 local and remote files)

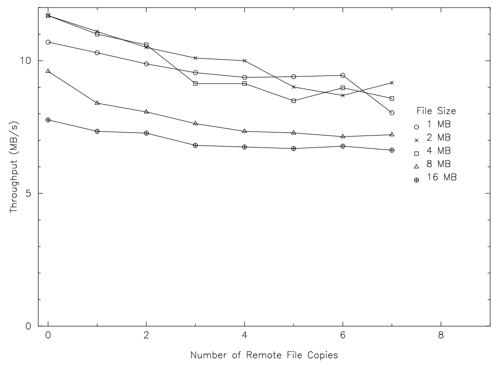

Figure 5: Beowulf Demonstration File Transfers (2 channels, Total of 7 local and remote files)

demonstration system in Figures 4 and 5.

3.1 Network Bandwidth

As has been done by others [2], artificial network traffic ultimately intended to maximally load the system was generated to approximate the sustained usable bandwidth of the network. The basic unit generating network load consisted of a pair of processes exchanging a fixed size token (message) an arbitrary number of times (we chose 1000 exchanges because it took sufficient time to minimize timing error). The token exchanges were performed by using the Linux implementation of BSD sockets and the `send()` and `receive()` system calls under TCP/IP. One process would send a token to its partner, the partner would receive and store it in a buffer, and immediately send the buffer back. To maximize throughput, the sockets used was for token exchange were made nonblocking through the Linux implementation of the POSIX `fcntl()` system call.

Each process in the pair was assigned to a different processor, and at most one traffic-generating process was assigned to any given processor. Network traffic was increased in two ways. The first was by increasing the size of the tokens being exchanged, and the second was by increasing the number of traffic-generating processes (i.e. increasing the number of tokens). At the time the experiment was performed, up to 14 processors and 7 concurrent tokens were employed.

The results of this experiment as performed on the Beowulf Demonstration system are shown in Figure 4. Data was taken on the demonstration system with one 100 Mbps network active (1 channel mode), represented by the solid lines, and also with two 100 Mbps networks active (2 channel mode), represented by the double-dashed lines. The number of tokens was varied from 1 to 7 and the token sizes used were 2^n where n was varied from 2 to 13.

As we had seen before with the prototype system utilizing 10 Mbps Ethernet (see Figure 2), maximal network utilization only occurred for larger token sizes. However, there now appears to be some variation with the number of tokens being exchanged. The maximum throughput achieved in 1 channel mode was 7.8 MB/s or about 62% of the 12.5 MB/s peak for Fast Ethernet. In 2 channel mode, the maximum throughput was 13.8 MB/s, only 55% of the theoretical 25 MB/s peak for dual channels, but higher than the peak for a single 100 Mbps net.

3.2 Parallel Disk I/O

We designed another artificial test to determine the approximate disk bandwidth of the Beowulf prototype and the limiting factors on remote interprocessor file accesses. The results of this experiment as originally run on the Beowulf prototype are shown in Figure 3 and the demonstration system results are shown in Figure 5. The experiment measured the throughput of simultaneous file transfers across a mix of intra-processor and interprocessor copies for a range of file sizes. Seven simultaneous file transfers were performed. Each file transfer could be either remote or local. A local file transfer involved only one processor, which would copy a file from its local disk to another file on its local disk. A remote file transfer involved two processors. One processor would run a process reading a file from its local disk and writing it across the network to another processor running a process reading the data from the network and writing it to its local disk. No processor was ever involved in more than one file transfer, avoiding local disk contention. A problem that arises when conducting an experiment of this type is that the Linux operating system automatically caches files as they are accessed in an attempt to reduce the cost of future accesses. To ensure that all file transfers involved only uncached files, we copied a dummy 32 MB file prior to each run. Local file transfers were performed using the Linux implementation of the POSIX `read()` and `write()` system calls while remote file transfers additionally used TCP/IP for transferring the files across the network.

Figure 5 shows the results of running this experiment in 2 channel mode for file sizes ranging from 1 to 16 MB and varying remote file copies from 0 (all local copies) to 7 (all remote copies). As one would expect, a file transfer rate of 11.7 MB/s, the largest achieved, occurred when all file transfers were local. The smallest transfer rate achieved was 6.6 MB/s. In the prototype system (see Figure 3) the network clearly constrained the disk throughput. This is no longer the case. As the number of remote file transfers increase, the curves no longer converge on the maximum sustained network performance. They now remain rather flat, only degrading by about 3.5% for each additional remote file transfer, unlike the 15% seen in the Beowulf prototype with 10 Mbps Ethernet.

4 Discussion

The preceding experiments demonstrate two qualitatively distinct operational behaviors between the Beowulf system based on 10 Mbps Ethernet and that based on the new Fast Ethernet technology. The first system is characterized with a parallel disk bandwidth that significantly exceeds the interprocessor communications bandwidth. Therefore, the network imposes a bottleneck to the system. The second system provides sufficient interprocessor bandwidth to match the demands of the disk throughput tested. Therefore the key qualitative distinction is that the Fast Ethernet based Beowulf is a balanced architecture while its predecessor is unbalanced.

It had been previously shown that 10 Mbps Ethernet channels could be used in parallel to provide increased bandwidth. While no one packet would go faster, the aggregate communications bandwidth supporting multiple concurrent traffic sources could be effectively increased. Figure 2 showed that under favorable conditions, the bandwidth gain of two channels with respect to one could reach 70%.

It is a new finding of this paper that Fast Ethernet channels can also scale comparably. Figure 4 demonstrates that Fast Ethernet can achieve a gain in bandwidth from one to two channels of about 74%, essentially the same as the 10 Mbps case. In terms of utilization, the slower channels are superior, achieving 80% of peak compared to Fast Ethernet's 65% of peak for a single channel. While the ratio of the peak performance of the Fast Ethernet versus regular Ethernet is 10, the ratio of their respective sustained throughputs is less at a factor of 8; still a respectable enhancement. But to be fair, the current per port cost of Fast Ethernet compared to regular Ethernet is also a factor of 8 so there is no direct price performance advantage when considering just the networks alone. The performance to cost gain occurs when this network capability is related to the file transfer traffic created by the parallel disk array.

As previously discussed, Figure 3 clearly demonstrates the bandwidth bottleneck caused by the limitations of the 10 Mbps Ethernet, even when more than one is used together. As all the disk traffic is forced to go between processors, the total sustained file transfer bandwidth is throttled down to that capable of being supported by the network, about 1.7 MBytes per second. This is a degradation of greater than a factor of 4 over the best concurrent local disk file transfer rate. It should be noted that a higher aggregate local file transfer bandwidth could be achieved than was done in the actual experiment. To hold the traffic demand constant and at the same time avoid processing node contention, each processor was used only in the role of either a producer or consumer of a file and not both.

The operational behavior of the Fast Ethernet based system is not only better than the prototype system, it performs in a new regime, one that achieves a balance between the demands of the parallel disk array throughput and the capacity of the interprocessor communications to support concurrent remote file transfers. Figure 5 shows the sustained throughput of the system as the seven concurrent file transfers range from entirely local to entirely remote. While the corresponding curves for the Beowulf prototype drop dramatically as file copies become exclusively remote, constrained by the 10 Mbps Ethernets, the Fast Ethernet based system experiences only a small degradation in system file transfer throughput. Instead of the loss of a factor of 4 to 5, the throughput degrades only about 20% across the range of tests.

The comparative behavior is best represented in Figure 6. Here, the curves in Figures 3 and 5 for the single case of 2 MByte file transfers is repeated to expose the differences between the 10 Mbps and 100 Mbps Ethernet channels. Even where all the file copies are to their respective local disks, a significant file transfer throughput gain is achieved. This has nothing to do with the networks and is entirely a consequence of the processing node. The improved processor architecture of the Pentium versus the 80486 and the superior properties of the PCI bus [5] with respect to the VESA-local bus combine to provide a performance gain of about 62%. The relative gain increases to a 550% when all file copies are

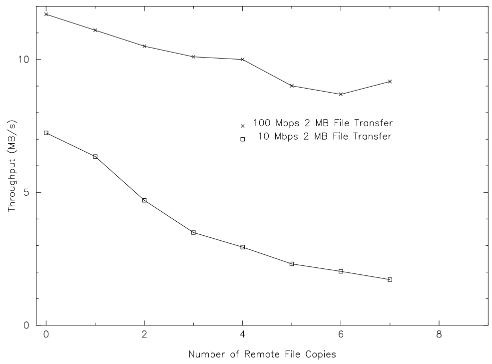

Figure 6: Beowulf Disk Bandwidth Comparison (2 channels, Total of 7 local and remote files)

remote at a system cost increase of about 14% for the Fast Ethernet channels.

5 Conclusions

The fundamental contribution of this paper is that a new balanced operating point for distributed computing systems has been identified and evaluated with direct implications for real-world computing. It has been shown that within the cost constraints of scientific workstations, such systems can be implemented that incorporate close to order-of-magnitude greater mass storage capacity and data access bandwidth than conventional workstations. This achievement is enabled through the exploitation of PC mass market commodity computing subsystems in a parallel structure and demonstrated by a series of experiments conducted with the Beowulf Parallel Workstation.

The seminal finding implied by the results presented in this paper is that the sustained data transfer rates possible with a 16-way parallel disk array can be supported by the parallel combination of dual 100 Mbps Ethernet channels. Using Beowulf as a testbed, it was shown that this configuration is both necessary and sufficient to achieve interprocessor communications rates comparable to those of the disk array. The application of this rapidly emerging Fast Ethernet technology makes scientific workstations of this type feasible for the first time.

The implications for real world computing are significant in that for specific environments substantial benefits can be achieved at little additional cost. In the realm of scientific computing, large data sets must often be manipulated, explored, and visualized. The working set size may easily exceed the capacity of conventional scientific workstations and require repeated access to shared file servers over common local area networks. The reason for repeated access is because ordinarily the entire data set can not be maintained in a conventional workstation and because the data usually requires many repeated examinations. The new capability provided by Beowulf is to stage such large working sets entirely in the workstation with only a single access required to the remote file server upon which the data set resides. The impact on the user is much faster response time, approaching an order of magnitude in some cases, permitting innovative ways of working with research data. For the more global system, it means significantly reduced burden on shared resources, reducing contention and improving response time there as well.

There remains the open question of software support for the use of PopC systems like Beowulf. Distributed computing systems such as PVM and MPI for message passing applications programming and Condor for job stream scheduling have been brought up on Beowulf and are being used. Management of parallel disk arrays, especially from the applications programmer perspective, continues to be a challenge and is the topic of active research by a number of groups around the country. The Beowulf project is evaluating some experimental packages. While the findings reported here do accurately characterize the capabilities of the Beowulf architecture, it does not represent a programming interface that is transparent to the user. Currently, only low level techniques have made this opportunity for superior mass storage in a single user context available. Without improved software tools, the potential for distributed disk arrays in a workstation environment will be lost.

REFERENCES

[1] S. Baylor, C. Wu, "Parallel I/O Workload Characteristics Using Vesta," *Proceedings of the 3rd Annual Workshop on Input/Output in Parallel and Distributed Systems*, April 1995, pp. 16-29.

[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] D. Kotz, N. Nieuwejaar, "Dynamic File-Access Characteristics of a Production Parallel Scientific Workload," *Supercomputing '94*, November 1994, pp. 640-649.

[4] Linux Documentation Project, Accessible on the Internet at World Wide Web URL *http://sunsite.unc.edu/mdw/linux.html*.

[5] T. Shanely and D. Anderson, "PCI System Architecture," MindShare, Inc., Richardson, TX, 1992.

[6] 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.

[7] 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.