Session 2532

Undergraduate Research Participation: Designing and Building a New Generation Beowulf-Class PC Cluster

Nickolas S. Jovanovic, Zachary R. Kaufmann, Lance W. Laettner

University of Arkansas at Little Rock

Abstract

Massively parallel processors (MPP) are the laboratories for computational science and engineering. It is important for computational scientists and engineers to have a local platform for developing, testing, and debugging MPP codes, so that computer time on large national-resource MPPs such as those at the national laboratories and NSF supercomputing centers can be secured and used wisely. Undergraduate computer engineering technology students are well prepared to design and build Beowulf-class PC clusters that can serve this purpose.

1. Introduction

Due to the continuing decreases in the prices of commodity off-the-shelf (COTS) computer hardware (PC-class processors and Fast and Gigabit Ethernet switches), and the development of free parallel computer systems software (Linux operating system and MPI software that allows processors to share data with each other via message passing), it has become possible to build a personal MPP for a relatively modest cost. An example is the Beowulf-class PC cluster[1]. A Beowulf-class PC cluster consists of one or more front-end workstations, one or more node workstations, and a switch that serves as the electrical interconnect for data transfer between nodes. Since COTS hardware is undergoing continuous, rapid improvement, almost every Beowulf-class cluster is unique in some way. One objective of our research was to design, build, and test a Beowulf-class PC cluster at the University of Arkansas at Little Rock using the most appropriate COTS hardware that was available at the time of funding. Undergraduate computer engineering technology students were involved in the project from preliminary design through commissioning. Our cluster will be used to support computational science and engineering research in radiation transport and computational fluid dynamics, as well as for undergraduate and graduate education.

2. Design Constraints

2.1 Cost

A grant from the Arkansas Science and Technology Authority, along with matching funds from the University of Arkansas at Little Rock, provided us with a nominal budget of $52,500 for the cluster.

Cost effective choices needed to be made for the cluster design. Our cluster was designed to maximize performance within the constraints of our budget. We used a price-to-performance ratio to evaluate various system design options. We calculated the price-to-performance ratio by dividing the price of a single system by its SPECfp95 benchmark rating.

2.2 Number of Processors

We wanted to build a 32-node cluster so that we would be able to conduct scalability studies using 1, 2, 4, 8, 16, and 32 nodes. We knew that our budget would be adequate to reach this goal, based on the experiences of other Beowulf builders who had used Pentium II processors, but we also wanted to try to design a cluster with higher performance.

3. Design Choices

3.1. Processor Family

We chose the AMD Athlon processor for our cluster system. One of the potential strengths of the Athlon processor is its cache memory system. The on-chip level one cache size is 128 kB, four times that of the PIII. This memory runs at the runs at the central processing unit core clock speed and is most local to the processor. The level two cache size is 512 kB. In addition, the Athlon has a superscalar, fully pipelined, out-of-order, three-way floating-point engine. This feature was very attractive since floating-point performance can be critical in many scientific applications.

The price-to-performance ratio was much lower for the Athlon than it was for either Intel Pentium III (including Xeon) or Compaq Alpha 21264 processors. Alpha processors are now pseudo-commodity hardware, due to Compaq's licensing of the technology to AlphaProcessors, Inc. However, Alphas are still available from only a relatively small number of system providers compared to PC-class processors, and are still quite expensive. The absolute performance of the Athlon is equal to or better than that of the Pentium III, but poorer than that of the Alpha 21264, at the same clock speed. However, Athlon processors cost significantly less than Pentium III or Alpha processors, at the same clock speed. Figure 1 compares the price-to-performance ratios of Athlon and Pentium III processors, and shows the large advantage of the Athlon for this metric.

We used the SPECfp95 benchmark[2] for estimating performance, and the Computer Shopper web site[3] and various other sources for estimating prices. Prices from many sources were averaged.

One disadvantage of the Athlon processor is a lack of software that has been optimized specifically for the Athlon instruction set. In our case, we plan to develop our own software, so the only things we really lack are C and Fortran compilers that can optimize for Athlon. So far, the Athlon processor has performed well in the marketplace, however, and we hope that such compilers become available in the near future.

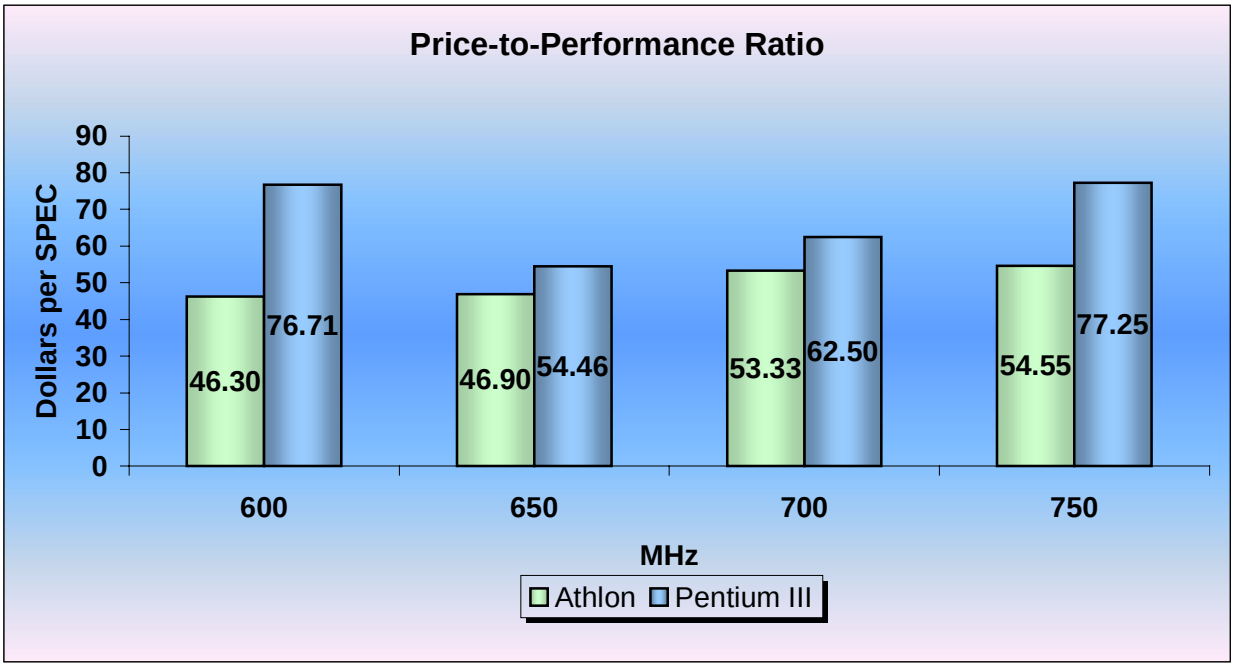

Figure 1

3.2 Processor Clock Speed

We chose the 600 MHz Athlon because it had the lowest price-to-performance ratio at the time that we purchased components (February 2000). Although 650-1000 MHz Athlons were also available by then, their costs were too great for our budget and their price-to-performance ratios were higher.

3.3. Motherboard

When the Athlon processor was introduced in September 1999, it was very difficult to obtain a Slot A motherboard for it. AMD had previously been unable to supply adequate quantities of new products, and the motherboard producers seemed to be taking a wait-and-see approach to the Athlon. Over the next few months, several motherboards did become available, however.

We chose the Gigabyte GA-7ixe motherboard for our cluster. The Athlon requires a Slot A motherboard. This motherboard allows only for a single processor, limits the total memory to 768 MB of PC100 SDRAM, and meets all of the latest standards such as AGP 2X, PCI 2.2, and UDMA/66. We wanted a reliable motherboard and didn't want to have any compatibility issues or problems arise that a common occurrence with bargain motherboards. A few other Slot A

motherboards were available (from Asus, Biostar, FIC, and Microstar) but a quick poll of computer technicians showed a definite preference for Asus and Gigabyte products. We experimented with both the Gigabyte GA-7ix and the ASUS K7M motherboards before finalizing the cluster design, but chose the Gigabyte board because it was on the AMD-approved list, whereas the ASUS board was not. Since then, ASUS has released a newer Slot A motherboard which is approved by AMD.

In late 1999, when we ordered parts, no SMP motherboards were yet available for Athlon. However, since some researchers report[4] that memory and network adapter contention by SMPs can degrade performance relative to single-CPU systems, this was not of particular concern. The Athlon motherboard features a bus architecture, called EV6, that is borrowed from the Alpha. This system bus, operating at 200 MHz, can deliver a theoretical bandwidth of 1.6 gigabytes per second and can scale to 3.2 gigabytes per second at 400 MHz[8].

3.4. Memory

We wanted to have as much RAM per node as we could afford, but memory prices spiked in the last few months of 1999, just when we were designing our system. Many applications need to have a lot of RAM per node so that they can be parallelized with a fairly course granularity and achieve high parallel efficiency. Higher processor clock speeds also require more RAM, to maintain high performance. Although we would have preferred 384 MB/node, based on a thumbrule[1] of 1 MB for every flop/s, we determined that we could afford only 256 MB/node within the constraints of our budget. This choice provided aggregate memory of approximately 8 GB for the cluster as a whole. As stated above, our selected motherboard will support 768 MB of SDRAM, and future upgrades can be made without problems.

3.5 Storage

We chose Fujitsu 6.4 GB 5400 RPM EIDE hard disk drives for the nodes, for an aggregate storage capacity of about 200 GB for the cluster as a whole. A thumbrule[1] is that local storage capacity should be at least ten times the local memory, and our design exceeds this by about a factor of two. Higher drive speeds, both EIDE and SCSI, were available, but were significantly more expensive. It is also possible to design a cluster with diskless nodes, but we wanted the flexibility of having physical storage on each node.

3.6 Case and Power Supply

We chose Antec KS-282 ATX midtower cases with Antec PP303X 300 Watt power supplies. These components were chosen because they were on the AMD-approved lists. We also installed two additional cooling fans in each case: one which draws air in through the front of the case, and one which pushes air out the back of the case. These fans are in addition to the power supply cooling fan and the processor heat sink cooling fan.

3.7 Local Area Network

In our cluster, the front-end and node workstations are interconnected via a private network by using restricted IP addresses. The front-end workstation is the only PC that is connected to the Internet (via a separate network adapter).

We chose a Fast Ethernet network for our cluster: Intel EtherExpress Pro100+ network adapters and a 36-port 3Com fast ethernet switch. Other switches, such as the Hewlett Packard ProCurve 4000, were considered as candidates in the selection process, but had lower throughput and backplane speed specifications. Higher performance networks, such as Gigabit Ethernet and Myrinet are still quite expensive compared to Fast Ethernet. The network is expected to be the weak link of our cluster for some scientific applications. However, several points can be made: (1) a high performance network can be added later, as an upgrade, (2) the cost of a high performance network can double the cost of the cluster, and (3) some parallel applications (termed *embarrassingly parallel)* do not require a high performance network to perform well.

We wanted a network adapter that could support the Fast Ethernet 100-BASE-TX standard and which was reliable. The 3Com 3c905C Fast Etherlink TX and Intel EtherExpress Pro 100+ both stood out as reliable NIC's and both support the 100-BASE-TX standard. Either would have sufficed. We chose the Intel EtherExpress Pro 100+ cards because we had previous experience with them.

3.8 Video Adapter

The node workstations in our cluster are not generally accessible to users, e.g., they do not have monitors, keyboards, and mice. There are not even generally available for remote login because they are not connected to the Internet. Therefore, the nodes do not need video adapters, except for assembly, testing, installation of the operating system, and troubleshooting. We did, however, opt to include relatively inexpensive video adapters in our node workstations so that they would be easier to troubleshoot when a problem arises.

3.9 Operating System

Linux is the primary operating system of choice for Beowulf-class clusters. Linux has the advantages of being low cost, very stable, and open source. In addition, the software development and execution environment is compatible with the Unix operating systems used in commercial MPPs.

We installed Red Hat Linux version 6.1 on each of our nodes. This version of Red Hat is based on the Linux Kernel 2.2.12. Many other Linux distributions are available, but we chose Red Hat because our hardware vendor was most familiar with that particular distribution. There are also alternatives to Linux, such as FreeBSD; however, they are not supported as well as Linux.

3.10 Front-End Workstation

The front-end workstation is almost identical to the node workstations. The exceptions are that we used a different case and power supply, added additional memory (up to the maximum of 768 MB), added additional storage (3 hard disk drives total, each identical to the hard drives in the node workstations), installed a more capable video adapter, and added a second network adapter. The only reason we used a different case and power supply was that we converted our prototype node workstation into the front-end workstation. Of course, the front-end workstation has a monitor, keyboard, and mouse also.

We hope to upgrade the front-end workstation to have high speed, high capacity SCSI hard disk drives. We also hope to upgrade the front-end motherboard and processor when new technology becomes available later this year. For example, Tyan recently announced a Slot A motherboard that uses PC133 memory, and several companies are working on Slot A motherboards that will support SMP. As this paper goes to print, AMD has already released a 1000 MHz (1 GHz) Athlon processor, and predictions go as high as 2 GHz by the end of this year. The extent to which we upgrade the front-end workstation depends mainly upon how much money we have remaining after building the cluster.

3.11 Message Passing Interface (MPI) Software

There are at least three choices for MPI implementations on Linux: MPICH[5] from Argonne National Laboratory, LAM[6] from Notre Dame University, and MPI/Pro[7] from MPI Software Technology, Inc. All three are free, and we plan to evaluate the performance of each implementation on our cluster.

3.12 Cluster Software

We purchased Cluster Development Kit (CDK) software from Portland Group, Inc. This is the only commercial software that we have purchased for our cluster. The primary reason we decided to purchase the CDK was for the parallel C, C++, and Fortran compilers that are included in it.

3.13 Miscellaneous

We also purchased five heavy duty steel and particle board shelving units (from Home Depot), 16 Back UPS 400 uninterruptible power supplies from APC, and 32 pre-made stranded CAT 5+ patch cables of various lengths to build our cluster.

4. Prototype Node Workstation

When we first considered using the Athlon processor, it was not completely clear whether Linux would run on it. Although AMD listed RedHat Linux as compatible software for the Athlon, RedHat did not list the Athlon as a supported processor (RedHat did eventually list the Athlon processor as Tier 2 compatible hardware many months later). So we decided to build a single node workstation in order to do some compatibility testing. This prototype node was purchased

for less than $1000, and its success gave us the confidence to proceed with our final design strategy. The original prototype contained a 500 MHz Athlon processor, an ASUS K7M motherboard, a CasEdge ATX case, a PC Power and Cooling 300 Watt power supply, 64 MB of RAM, the Intel Pro/100+ network adapter, an inexpensive video adapter, a hard disk drive, a floppy disk drive, and a CDROM drive. When we decided to use the Gigabyte GA-7ix motherboard, we swapped out the ASUS motherboard and upgraded to 256 MB of RAM. Once again, the success of these modifications gave us the confidence to proceed. We have now converted the prototype workstation into the front-end workstation.

5. Acquisition

By November 1999, we had settled on a design for our cluster and tested our prototype node. We submitted a purchasing requisition to the university purchasing department in early December. Since the anticipated cost was more than $10,000, a sealed bid process was required. The request for bids was published in the newspaper in early January 2000, and the sealed bids were opened and read on January 20. The two lowest bids did not meet the specifications and were disqualified. The remaining qualified bids were too expensive for our budget. We modified the specifications slightly and began the sealed bid process again. The second set of sealed bids was opened on February 22. This time, the lowest bid did meet the specifications and the total cost was about $15,000 less than the lowest qualified original bid.

6. Baby-Beowulf

While we were involved with the preliminary design of our Beowulf cluster, we built a tiny Beowulf from spare parts and computers that had failed their Y2K tests. These computers and parts were of a surplus or obsolete nature, and so were completely free. But the students were able to build a few boxes, install Linux and MPI on them, and network them for practice, while waiting for our new parts to arrive, and without fear of breaking anything. If you don't have a grant like we did, you can still build a powerful cluster from such hardware.

7. Conclusions

Building a Beowulf is an excellent undergraduate student design project. There are many areas of active, cutting-edge research in parallel clusters, especially in the area of system software, that are perhaps more appropriate for graduate research. But there are also many aspects of engineering design that need to be addressed every time a Beowulf cluster is built. The hardware capabilities and prices change so fast that every cluster is essentially unique, so the design work must be done every time. This is entirely appropriate for undergraduate engineering and technology students, and very satisfying as well.

Bibliography

1. Sterling, T. L., Salmon, J., Becker, D. J., and Savarese, D. F., *How to Build a Beowulf: A Guide to the Implementation and Application of PC Clusters*, The MIT Press, Cambridge, 1999.
2. Standard Performance Evaluation Corporation (SPEC) Web Site, http://www.spec.org/
3. Computer Shopper Web Site, http://www.computershopper.com/

4. Loncaric, J. Web Site, *Commodity Supercomputers: High Performance, Low Cost,* http://www.icase.edu/~josip/HPCS/
5. MPICH Web Site, http://www-unix.mcs.anl.gov/mpi/mpich/
6. LAM Web Site, http://www.mpi.nd.edu/lam/
7. MPI/Pro Web Site, http://www.mpi-softtech.com/
8. AMD Web Site, http://www.amd.com/

NICKOLAS S. JOVANOVIC
Dr. Jovanovic received the B.S.M.E. degree from Northwestern University, the M.S.M.E. degree from Rensselaer Polytechnic Institute, and M.S., M.Phil., and Ph.D. degrees in Engineering and Applied Science from Yale University. He is an Assistant Professor of Computer Systems Engineering in a newly created Department of Systems Engineering at the University of Arkansas at Little Rock.

ZACHARY R. KAUFMANN
Zachary Kaufmann is a senior Computer Engineering Technology undergraduate student at University of Arkansas at Little Rock and is also completing his Associate of Science degree in Computer Science. He has worked as an Assistant Network Administrator at Conway Regional Health Systems in Conway, Arkansas and he has served as Vice President for the UALR student chapter of IEEE. He has also worked on infrared galaxy image processing.

LANCE W. LAETTNER
Lance Laettner is a senior Computer Engineering Technology undergraduate student at the University of Arkansas at Little Rock. He has been a Senior LAN Technician at the Department of Information Systems, State of Arkansas, from 1998 to present. He is a member of IEEE, and his primary related interests include computer hardware design, artificial intelligence, and programming.