Commodity Clusters: Performance Comparison Between PC's and Workstations

Russell Carter
1765 Shamrock Ave.
Santa Clara, CA 95051
rcarter@geli.com

John Laroco
Robert Armstrong
Sandia National Laboratories
Livermore, CA 94551
{jclaro,rob}@ca.sandia.gov

1 Introduction

Traditionally, the bulk of large scale scientific and engineering computations were performed on large specialized Supercomputers. In the last seven years, RISC based workstation technology has largely supplanted the Supercomputing market. Using techniques such as workstation clustering, wide classes of problems have been successfully attacked. Workstation clusters are networked RISC UNIX systems commonly provided by vendors such as IBM, Hewlett Packard, SUN, Silicon Graphics, and Digital Equipment Corporation. Each of these vendors is the predominant or proprietary supplier of both hardware and UNIX operating systems.

Applications run on proprietary workstation clusters use UNIX features that enable high performance floating point, fast disk drive transfer rates, and robust networking performance in a multi-user, multi-system environment. UNIX workstation clusters are capable of achieving supercomputer efficiencies on many computationally intensive tasks. Additionally, individual cluster systems can be used as desktop UNIX workstations.

The size of the PC market is about nine times larger than the proprietary UNIX workstation market. Intel Pentium based systems are the performance leader in this market. Until recently, proprietary workstation hardware and software greatly exceeded in sophistication, capability, and performance that available from the commodity PC distribution channel. Recent PC technology advances have dramatically increased processor, main memory and cache memory performance. Some high end models offered by proprietary workstation vendors still maintain a significant advantage in peak floating point performance. However, the widespread availability of numerous high performance PCI bus network, video, and disk controllers for PCs has erased the traditional hardware I/O performance advantage of proprietary UNIX workstations. A similar evolution has occurred in system software. The multi-user performance of several varieties of PC UNIX running on Intel Pentium CPUs is equivalent to proprietary workstation UNIX. The enormous size of the commodity PC market ensures lowest possible hardware costs. Other advantages exist: individual commodity workstations are well engineered PCs, and may be maintained by any competent PC maintenance vendor. Organizations have more options to manage computer resources efficiently, since every commodity workstation can run industry standard operating systems, such as Microsoft Windows 95, Windows NT, IBM OS/2, and SunSoft's Solaris.

With these considerations in mind, in 1994 the Distributed Computing Research group at Sandia National Laboratories, CA, constructed a testbed of sixteen Pentium workstations. Dubbed the DAISy (Distributed Array of Inexpensive Systems), this testbed was used to investigate the viability of commodity workstation clusters. Extensive functionality, performance and cost studies have been compared with performance and cost data from the proprietary workstation vendors. Full UNIX operating system functionality is provided by the BSD 4.4 based OS. The advanced networking applications required to manage clusters of workstations were found to be robust and full featured. Performance of I/O subsystems is equivalent or superior. Floating point performance is equivalent or superior to low-end offerings from the traditional workstation vendors. Initial and ongoing costs for installing and operating a commodity workstation cluster are about one half (50%) that of similar functional configurations from the traditional workstation vendors.

1.1 Related Work

A commodity workstation cluster project using similar technology is the Beowulf project [1]. The emphasis of this project is on lowest possible cost components combined with striped 10 Mb Ethernet.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Good results have been obtained for important parallel codes. Scalability has been limited because each PC in the cluster share the same 10Mbit/sec Ethernet.

2 DAISy Node Configuration

The DAISy Cluster is dual homed network of 16 Intel Pentium 90MHz workstations (see Figure 1) and very inexpensive UNIX compatible software. DAISy is a homogeneous research prototype used for scientific parallel distributed computing, and, a model for a minimum cost and fast distributed computational system. The motherboards support 3 PCI (Peripheral Component Interface)[2] bus cards and 4 ISA (Industrial Standard Architecture) bus cards. Each node has 256Kbytes of 2nd level cache and 64Mbytes of random access memory (RAM). The PCI bus is noteworthy for the reason that it is the first high performance, asynchronous I/O bus available for commodity PC architectures. Disk I/O functionality is handled by a bus-mastering PCI SCSI-II controller.

Each node in the model consists of :

Intel Pentium based workstation (3 PCI, 4 ISA slots), 90MHz

Motherboard:	*Intel Premier 90MHz w/Neptune chipset, P54C-PCI w/256k cache*
CPU:	*Intel Pentium P54C 90MHz*
RAM:	*64MB (2@8x36) 60ns 72 pin SIMMs, w/parity*
SCSI Controller:	*PCI fast SCSI II NCR53810 controller*
Ethernet:	*3COM 3C509 Etherlink III Combo, EISA* *SMC EtherPower 10/100, PCI*
Hard Drive:	*Quantum PD1080S, 1GB fast SCSI II 9.5ms internal*
Floppy:	*Teac, 1.44MB 3.5"*
Video:	*SVGA 512k, 1024 x 768*
Case & PS:	*medium tower case w/250W power supply*

D-00 also includes the following:

CDROM Drive:	*fast SCSI 3x speed NEC 3x1 CDR-510*
Tape Drive:	*8-16Gb Wangdat internal Dat*
Video:	*ATI Mach 64 Win Turbo, 2Mb VRAM*
Hard Drive:	*IBM fast SCSI II 4GB internal, <10ms*

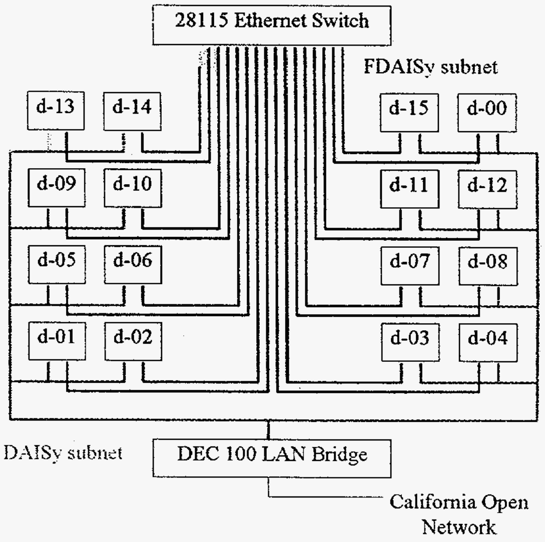

Figure 1. DAISy & FDAISy Subnet. Individual CPU's are labeled d-00 through d-15.

2.1 The P54C Pentium ™ 90 MHz Processor

In the early 90's, the installed large scale computing resources at Sandia National Lab, CA, evolved to a cluster of RISC UNIX systems. With free UNIX compatible software already available on architectures based on the i386 ™ and i486 ™ CPUs the Distributed Computing Research Group at SNL,CA decided to construct a network of commodity components based on the x86 CPUs, the P54C Pentium ™ Processor was the logical choice.

The P54C (90 MHz Pentium) was the state-of-the-art commodity PC CPU at the time of initial construction of DAISy. At the time of this writing, 166 MHz Pentiums and 200 MHz Pentium Pros are standard, leading to CPU performance increases of a factor of 2-4. Advances in cache memory and main memory technology have increased performance of these critical components by over 50%. The measured performance of the P54C-90 is used as the basis for the computation of price/performance.

2.2 Network

The networks hosted on each node of the cluster are standard 10Mb/s ISA bus (10BASE-2) Ethernet and switched 100Mb/s PCI bus (100BASE-TX) Fast Ethernet. The 10BASE-2 network is a bus broadcasting network topology. This interface is used for client node NFS mounts, and any client node interactive work users find necessary. The 100BASE-TX network facility uses a high speed frame switch (28115 Fast Ethernet Switch by Bay Networks) with PCI bus fast Ethernet (10/100BaseT) NICs connected in a point-to-point star topology. The designated use of the 100BASE-TX network is for user program message passing traffic. Hence, the architecture of the 100Base-TX network was designed to ensure contention free, high performance network communications. DAISy is a subnet on Sandia's Internet backbone via a DEC 100 bridge. The choice of networks was dictated by the architecture of the motherboards available at the time of design; the Intel motherboards used do not support more than one bus-mastering network interface card.

2.3 Frame Switched 100BASE-TX Fast Ethernet

The 100BASE-TX Fast Ethernet Network uses a Synoptics 28115 Frame Switch to reduce latency and increase aggregate bandwidth available to message passing functions in user programs on the DAISy cluster. Frame switching is used to enhance network performance by increasing the total amount of available aggregate bandwidth and decreasing overall communication latency. In the case of DAISy, an increased bandwidth of a factor of ten increases the number of parallel applications suitable for the DAISy cluster. Bandwidth is increased because contention is eliminated and multiple transmissions are allowed. For instance, ordinary shared media broadcast through a type of pipe communication. That is, all nodes connected to that pipe can see the broadcast, therefore; (1) all nodes look at the broadcast frame, (2) decide if the frame belongs to them, and. (3) act accordingly if the frame was addressed to them, otherwise (4) the nodes just continue to monitor the pipe. The advantage of a frame switch is frames are unicast only to the port attached to the destination, much like the crossbar interconnect network seen in multiprocessor machines. Because the frame is only transmitted on a single port, other ports are available for other simultaneous transmissions Frame Frame

3 DAISy System Software Configuration

3.1 Operating System

The operating system is the freely redistributable FreeBSD [3], a BSD 4.4Lite-Derived UNIX OS.. The choice of OS was made on the basis of support for high performance PCI devices, performance of device drivers on the high performance PCI network and disk drive devices, performance of NFS server and client services, availability of system source code, and overall cost. Interestingly, no commercial OS approaches the thoroughness by which FreeBSD satisfied the requirements. Linux [4] is an obvious possibility (and was the initial OS) but was replaced due to lack of adequate PCI device support, PCI device driver performance, and NFS server performance.

The directory hierarchy is a common for workstation clusters. Users home directories and the various local "/usr/local" applications are installed on the main drive of the master node, which functions as an NFS server to the remaining 15 client nodes.

3.2 Message Passing Software

The goal of the DAISy is to provide the highest possible price performance using commodity hardware and software resources. The parallel architecture of DAISy requires a user accessible means of parallel programming. The 100BASE-TX network is used solely for user application message-passing traffic. The 10BASE-2 network is used for NFS and system uses. Message passing libraries available to the DAISy users are PVM3[5] and MPI (MPICH)[6].

4 System Analysis

A goal of the DAISy project is to investigate the viability of commodity PC technology to the computation of scientific and engineering problems traditionally performed on "Supercomputers", and more recently high performance RISC workstations and clusters of RISC workstations. To this end a performance analysis of the various subsystems was carried out. Finally, performance of the cluster as a whole on a number of parallel applications was determined. The results are given in the following sections.

The compiler used for all performance tests is gcc-2.6.3. FORTRAN code is translated to C using a translator and then compiled using gcc-2.6.3. This particular version of gcc did not support any Pentium specific optimizations, which are becoming common in commercial compiler products.

System Description	Operating System	CPU	MHz	Year	List Price
FreeBSD/i586 (p5-90)	FreeBSD 2.1	Pentium 90	90	'94	5K
DEC Alpha	OSF1 3.2	Alpha	175	'92	24.4K
HP 9000/735	HP-UX A.09.05	PA-RISC	99	'92	33K
IBM RS6000	AIX 2.3.5	RS6000	40	'92	29K
SGI IRIX	IRIX 5.3	MIPs R4000	100	'92	33K
SUN SS10, SUN/TI	SUNOS 4.1.3	Super SPARC	51	'92	18.9K

Table 1. DAISy and HEAT Cluster system descriptions.

For comparison, there are various other system performance results shown throughout the paper. One such system configuration is another cluster at SNL,CA, HEAT (Heterogeneous Environment And Testbed). The HEAT cluster is a collection of five flavors of mainstream workstations. There are 50 workstations connected together to form HEAT, 10 each of the following: SUN SS10, SGI R4000 Indigo, DEC Alpha, HP 735, and IBM RS6000 350. The network media for HEAT is a bridging crossbar gigaswitch with 22 FDDI ports. Table 1 shows a description of the DAISy and HEAT clusters. A note as to where the other system performance results were obtained will be given in the respective subsections.

4.1 Subsystem Performance

The performance of a workstation cluster is strictly limited by the performance of applications on each of the individual nodes. The individual node performance is in turn a function of how efficient the important subsystems perform their tasks. For the class of scientific and engineering applications of interest the individual node performance measure most often quoted is the number of millions of floating point operations performed per second, or *MFLOPS*. The most important subsystems that determine individual node floating point performance are Main Memory and the CPU.

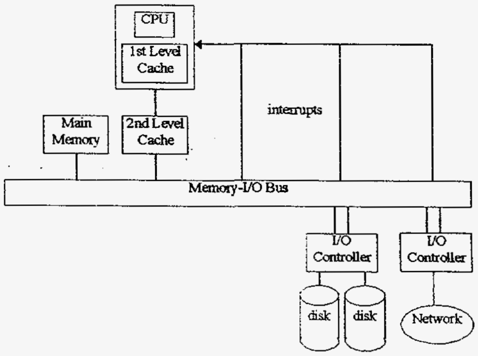

Figure 2. Typical collection of I/O devices on a computer.

The I/O subsystems of interest are disk and networking . The disk I/O performance is characterized by bandwidth through the file system, and is important for those applications that use local storage. Network File System (NFS) performance is crucial for efficient operation of the workstation cluster. Both client and server performance of NFS is important, the first in order to implement efficient distribution of executables at user program startup, and the latter to allow timely transferal of computed data to the user's home directory on the master node. Finally, networking performance is characterized by bandwidth and latency, and affects the performance of parallel message passing algorithms on the cluster. Figure 2 shows a typical collection of I/O devices which are of interest when determining system performance.

4.2 Floating Point performance

The performance of the Pentium CPU on floating point is discussed in detail in [7],[8]. The following discussion, and performance measurements, are for 64 bit floating point operations. The Pentium CPU has a single 8 stage pipeline for floating point operations, with a capability of producing one result per cycle, so the "not-to-be-exceeded' performance is 1 FLOP per cycle, or 90 MFLOPS for the 90 MHz P54C. However, the P54C architecture uses a stack of floating point registers rather than an independently addressable register set. A common operation is to swap operands within the stack. When paired with certain floating point operations this swap can happen simultaneously, so that no cycles are lost. This cannot be done in every case. The most highly tuned assembly coded kernels achieve a maximum floating point performance of roughly 2/3 FLOP per cycle, or 60 MFLOPS for the 90 MHz P54C.

4.2.1 Matrix-Matrix multiply (using assembly coded DGEMM)

Matrix-Matrix multiply is a highly optimized kernel computation which, when well implemented, is for practical purposes an indicator of the upper bound of the floating point performance available to user applications on a single node. The performance figures provided here use the BLAS3 [9] DGEMM algorithm. The BLAS 3 routines are generally provided as tuned assembly coded routines by RISC UNIX workstation vendors. The P54C-90 Pentium results were obtained using a tuned version of the DGEMM algorithm, using assembly coded DAXPY routines implemented by one of the authors. The best performance obtained was 13.3 MFLOPS on 64x64x64 DGEMM. Properly tuned DGEMM implementations allow for the CPU to operate out of the highest level of the memory hierarchy, in this case, the first level cache.

4.3 Main Memory Performance

Commodity PC architecture is similar to low and mid-class RISC workstation architecture: The processor is fed operands from a series of memory hierarchies, which generally differ in speed, cost, and size. Competitive pressures require that the fast memory hierarchies (1st and 2nd level cache) be made as small as feasible, and thus many applications of interest have the characteristic that a significant amount of data is fetched relatively frequently from the large (and slow) main memory. If main memory is too slow, the performance of many applications will be limited by the speed with which memory can supply operands.

Prior to the availability of Fast Page Mode DRAM, commodity PC main memory was significantly slower than that available for most RISC UNIX workstations. Since implemented in the first quarter of 1994, Fast Page Mode DRAM has become the memory of choice for commodity PCs. Now commodity PCs exhibit main memory bandwidths that compare favorably to that of RISC workstations offered by the traditional workstation vendors.

4.3.1 Stream benchmark

The most widely used measure of main memory bandwidth is McCalpin's *Stream*[10] benchmark. This benchmark performs a carefully parameterized DAXPY, and returns information on several aspects of main memory performance. A wide range of results are available. The benchmark is notable because it emphasizes the measurement of the rate at which operands can be fetched to the CPU from the lowest level of the memory hierarchy, which for most workstation class systems is the large DRAM based main memory. Table 2 shows the results from both the DAISy and HEAT clusters.

Function Rate	MB/s p5-90	DEC Alpha	IBM RS6K	SGI IRIX
Assignment:	38.908	88.148	125.912	39.903
Scaling :	39.178	88.83	122.966	37.597
Summing :	46.03	93.155	130.42	38.856
SAXPYing :	46.032	91.699	130.285	36.285

Table 2. Stream results (MB/s).

4.3.2 lmbench benchmark (Memory Bandwidth)

The lmbench [11]suite measures the ability to read, and write data over a varying set of sizes. The benchmarks included in the *memory bandwidth* component include: bw_mem_rd and bw_mem_wr. The results shown in table 3 are from the DAISy and HEAT systems and from McVoy and Staelin's lmbench draft.

Memory reading bandwidth is measured by an unrolled loop that sums up a series of integers (typically a 4 byte integer). The benchmark bw_mem_rd allocates the specified amount of memory, zeros it, and then times the reading of the memory as a series of integer loads and adds. An 8MB area is specified in Table 3 to show memory bandwidth and not cache bandwidth.

Memory writing bandwidth is measured by an unrolled loop that stores a value into an integer (typically a 4 byte integer) and then increments the pointer. The benchmark bw_mem_wr allocates the specified amount of memory, zeros it, and then times the writing of that memory as a series of integer stores and increments. Again, an 8MB area is specified in Table 3 to show memory bandwidth and not cache bandwidth.

System	memory read (bw_mem_rd)	write (bw_mem_wr)
IBM Power2	205	364
Sun Ultra 1	129	152
DEC Alpha@300	120	123
HP K210	117	126
Unixware/i686	214	86
DEC Alpha @150	79	91
Linux/i686	208	56
Linux/Alpha	73	71
FreeBSD/i586	73	83
Linux/Alpha	73	71
Linux/i586	74	75
SGI Challenge	65	67
SGI Indigo	69	66
IBM PowerPC	63	26
Sun SC1000	38	31
DAISy systems		
FreeBSD/i586 (p5-90)	54.15	28.01
HEAT systems		
DEC Alpha	84.8	90.31
HP 9000/735	52.81	50.88
IBM RS6000	59.03	61.32
SUN SS10	42.6	28.18
SGI IRIX	44.47	48.94

Table 3. bw_mem_rd, bw_mem_wr results (MB/s).

SPECint92 and **SPECfp92** results showing the top 20 SPECx92 performance ratings as reported at the URL, *http://www.ideas.com.au/bench/spec/spec.html*, and the results of the P5-90 and P6-200 from URL *http://hpwww.epfl.ch/bench/SPEC.html*.

System Name	SPECint 92	SPECfp 92
DEC-AlphaServer 8200 5/300	341.4	512.9
DEC-AlphaServer 8400 5/300	341.4	512.9
Olivetti-LSX 7830	341.4	512.9
Olivetti-LSX 7860	341.4	512.9
DEC-AlphaStation 600 5/300	337.8	502.1
Sun-Ultra 2 Model 2200	332	505
DEC-AlphaStation 600 5/266	289	405
DEC-AlphaServer 2000 5/250	277.1	410.4
Olivetti-LSX 7560	277.1	410.4
DEC-AlphaServer 2100 5/250	277	410.4
Sun-Ultra 1 Model 170	252	351
Sun-Ultra 1 Model 170E	252	351
Sun-UltraServer 1 Model 170	252	351
Sun-UltraServer 1 Model 170E	252	351
Sun-Ultra 1 Model 140	215	303
Sun-UltraServer 1 Model 140	215	303
HAL-HALstation 350	212	271
HAL-HALstn 350 Application Svr	212	271
DEC-AlphaServer 2000 4/275	202.9	292.6
DEC-AlphaServer 2100 4/275	202.9	
Intel Xpress Pentium 60/90 512+8/8	**106.5**	**81.4**
Intel Alder PentiumPro 200 256+8/8	366	283.2

Table 4. SPECint92 and SPECfp92 performance ratings.

4.3.3 SPECx92: Standard Performance Evaluation Corporation

The Standard Performance Evaluation Corporation (SPEC) [12] was founded in, 1988, as a non-profit group of computer vendors, system integrators, universities, research organizations, publishers and consultants throughout the world. It was formed with the objective to establish, maintain and endorse a standardized set of relevant benchmarks that can be applied to the newest generation of high-performance computers. The SPEC92 benchmarks are the second generation of the SPEC benchmarks. A third major version is the SPEC95 suite.

Unfortunately, insufficient results were reported for systems that resemble DAISy systems for the SPEC95 suite. Shown in Table 4 are the top 20 SPECx92 performance ratings and, Intel P5 90MHz and P6 200MHz systems.

4.3.4 LINPACK (C and FORTRAN)

LINPACK[13] is a venerable floating point benchmark that primarily demonstrates performance on a very simple set of loops. Code is provided on the NETLIB[14] software repository for both C and FORTRAN versions.

Performance on the C version is somewhat better, as is expected from the need to translate FORTRAN code first to C before compilation. However, FORTRAN performance is surprisingly good, and well within the competitive range of higher priced offerings from the traditional RISC Workstation vendors. Results in Table 5 and Table 6 are from DAISy and HEAT systems.

linpackc	p5-90	DEC Alpha	HP 735	IBM RS6K	SGI IRIX	SUN SS10
MFLOPS	7.4	19.3	18.7	12.6	8.1	10.1

Table 5. LINPACK C results.

linpackd	p5-90	DEC Alpha	HP 735	IBM RS6K	SGI IRIX
MFLOPS	6.151	9.909	5.607	4.862	4.86

Table 6. LINPACK FORTRAN results.

4.3.5 Discussion

As seen from the results, the main memory bandwidth of the 90MHz Pentium is well within that of many low-end workstations offered by the traditional workstation vendors. This result indicates that many applications that depend on the performance of main memory, can be expected to perform well, provided the speed of the CPU is sufficient. Also, note the significant gain in main memory bandwidth from the 90MHz Pentiums to the 133MHz Pentiums and to the now standard Pentium Pro's. This indicates that with little additional investment DAISy can be upgraded with a significant increase in performance.

4.4 Disk Performance

Many interesting applications require significant amounts of local DISK I/O capability.

4.4.1 Bonnie: Disk performance benchmark

The Bonnie[15] disk performance benchmark measures several aspects of disk performance. Table 7 shows the DAISy and HEAT disk performance benchmarks for sequential output, sequential input, and random seeks on a per character and a block size measure. The benchmark size was 100MB.

		Sequential Output						Sequential Input				Random	
		Per Char		Block		Rewrite		Per Char		Block		Seeks	
Machine	MB	K/sec	%cpu	K/sec	%cpu	K/sec	%cpu	K/sec	%cpu	K/sec	%cpu	/sec	%cpu
P5-90	100	2059	93.6	2429	57.6	971	18.7	1302	49.6	2280	29.9	156.5	13.9
DEC Alpha	100	3425	95.6	3491	14.5	1538	6.8	3497	96.9	3574	8.6	101.6	3.6
HP 735	100	1568	60.6	1492	31.7	610	4.5	1492	56.9	1530	7.5	141	6
IBM RS6K	100	1459	96.5	1558	13.3	530	6.4	1123	89.2	1939	13.2	44.8	6
SGI IRIX	100	1767	95.4	3307	26.5	1320	13.5	1336	81.6	2806	15.8	62.9	5
SUN SS10	100	1503	71.7	1552	15.4	523	8.4	1415	81.9	2275	20.5	71.9	6.6

Table 7. Bonnie results (100MB benchmark size).

		Sequential Output						Sequential Input				Random	
		Per Char		Block		Rewrite		Per Char		Block		Seeks	
Machine	MB	K/sec	%cpu	K/sec	%cpu	K/sec	%cpu	K/sec	%cpu	K/sec	%cpu	/sec	%cpu
P5-90	100	131	4.5	128	1.1	12	0.3	660	24.4	571	4.7	25.4	3.8
DEC Alpha	100	519	13.6	756	3.4	350	1.7	1065	27	1201	2.6	99.2	5.2
HP 735	100	645	25.4	679	11.9	330	2.3	835	31.6	1032	4.9	106.6	4.8
IBM RS6K	100	562	38.9	692	8.2	289	5	405	33.5	1440	11.6	35.4	2.9
SGI IRIX	100	258	16.6	287	3.4	188	6	174	14.4	425	7.8	26.6	5.8
SUN SS10	100	407	20.5	581	8.1	252	6	463	27.5	615	6.3	47	8.3

Table 8. Bonnie NFS Server Performance results (25MB benchmark size).

4.4.2 NFS Server Performance

NFS server performance (see Table 8) was assessed by running the bonnie benchmark on a client node to the user home directory file system on the master node, using the DAISy 10BASE-2 operational network.

4.4.3 NFS Client Performance

Client performance was measured by copying a large file from the master node to /dev/null on the client system. For an 11 MB file the best rate was measured at 602 KB/s

4.4.4 Discussion

The results from the bonnie benchmark show that the DAISy cluster systems have disk performance equivalent or superior to that of the measured HEAT systems. The NFS server performance shows the disadvantage of the current Intel motherboards inability to support more than one bus-mastering network interface card.

4.5 Networking Bandwidth and Latency

The DAISy project's first stage interconnect transport protocol for the message passing libraries is TCP (with RouteDirect PVM3 operations). The performance of the DAISy cluster as a whole on parallel message passing applications then is at minimum bounded by the underlying TCP performance of the 100BASE-TX interconnect. Two aspects of the switched 100BASE-TX network directly affect the performance of parallel message passing applications. In the simplest case two nodes communicate, in *point-to-point* fashion. The second, more complex case occurs when more than two nodes wish to communicate contemporaneously. The performance analysis of this important case is much more complex and is beyond the scope of this paper. In the point-to-point case, two common measures of TCP performance are bandwidth and latency. Bandwidth is defined as the asymptotic number of bytes transferred per unit time, as the size of messages size is increased. Latency is defined to be the extrapolated zero byte message transfer time obtained from a linear fit of a range of (small) message sizes. A useful and widely used tool to determine these quantities at the TCP socket level is Netperf.[16]. These network performance results were confirmed using the *lmbench* benchmark suite. Direct measures of the bandwidth and latency of point to point message passing latency and bandwidth PVM3 and MPI libraries were also performed.

4.5.1 Netperf suite: A Network Performance Benchmark

Netperf is a suite of benchmarks used to measure various aspects of networking performance and designed around the client/server model. The primary focus is on bulk data transfer and request/response performance using either TCP or UDP and the Berkeley Sockets interface. All benchmarks are run for an elapsed time of ~60 seconds. The various netperf performance benchmarks fall into two categories: (a) stream, and (b) request/response.

The most common use of the netperf suite is measuring bulk data transfer performance. This is referred to as "stream" or "unidirectional stream" performance. These tests measure how fast one system can send data to another and/or how fast that other system can receive it. The "tcp_stream_script" and "udp_stream_script" shell scripts supplied with the package were run to determine bandwidth..

Netperf request/response performance is quoted as "transactions/sec" for a given request and response size. A transaction is defined as the exchange of a single request and a single response. From a transaction rate, one can infer one way and round trip average latency. The "request/response" scripts that were run include: tcp_rr_script and udp_rr_script.

TCP Stream: tcp_stream_script is an implementation of the stream benchmark over TCP. The local send size ranges from 4096 to 32768 bytes with the local/remote send and receive socket buffer sizes of each ranging from 8102 to 57344 bytes. With the send and receive socket buffer sizes not remaining constant, the output shows throughput (Mb/s) as a function of range (bytes). For the results in Table 9 recv & send socket size = 57344, send message size = 32768.

System	Network	TCP stream	UDP stream send	receive
DAISy systems				
FreeBSD/i586 (p5-90)	10base2	5.19	5.73	5.73
FreeBSD/i586 (p5-90)	100baseT	50.64	68.5	35.28
HEAT systems				
DEC Alpha	fddi	85.87	89.23	37.14
HP 9000/735	fddi	79.31	87.38	52.18
SGI IRIX	fddi	71.08	69.66	11.45

Table 9. tcp_range_script, tcp_stream_script, udp_stream_script results (Mb/s).

UDP Stream: udp_stream_script is an implementation of the stream benchmark over UDP. The difference between udp_stream and tcp_stream is that the send size cannot be larger than the smaller of the local and remote socket buffer sizes. The local send size ranges from 64 to 1472 bytes with the local/remote send and receive socket buffer sizes of each remain constant at 32768 bytes. With the send and receive socket buffer sizes remaining constant, the output shows throughput (Mb/s), as a function of range (bytes) for both send and receive. For the results inTable 9: socket size = 32768, message size = 1472.

TCP Request/Response: tcp_rr_script is an implementation of the request/response benchmark over TCP. The request/response sizes are varied with the local/remote send and receive socket buffer sizes of each being the default of that particular system. With the local/remote send and receive socket buffer sizes remaining constant (the default), the output shows performance (transactions/s) as a function of request/response sizes (bytes). For the results in Table 10: send & recv socket = default bytes, request/resp. size = 1/1.

UDP Request/Response: udp_rr_script is an implementation of the request/response benchmark over UDP. The request/response sizes are varied with the local/remote send and receive socket buffer sizes of each being the default of that particular system. With the local/remote send and receive socket buffer sizes remaining constant (the default), the output shows performance (transactions/s) as a functions of request/response sizes (bytes). For the results inTable 10: send & recv socket = default bytes, request/resp. size = 1/1.

System	Network	TCP request/ response	UDP request/ response
DAISy systems			
FreeBSD/i586 (p5-90)	10baseT	1331	1659
FreeBSD/i586 (p5-90)	100baseT	1638	2096
HEAT systems			
DEC Alpha	fddi	1772	1937
HP 9000/735	fddi	2423	2473
SGI IRIX	fddi	993	34

Table 10. tcp_rr_script, udp_rr_script results (transactions/s).

4.5.2 *lmbench suite (IPC Bandwidth)*

lmbench addresses the performance issues of interprocess communication bandwidth with the TCP bandwidth micro-benchmark. The results shown in Table 11 are from the DAISy and HEAT systems and from McVoy and Staelin's lmbench draft [11]. bw_tcp (Table 11), the TCP micro-benchmark, is a client/server program that moves 3M bytes of data over a TCP/IP socket. The sockets are configured to use the largest receive/send buffers that the OS will allow.

System	Network	TCP (bw_tcp) remote host
SGI PowerChallenge	hppi	79.3
Sun Ultra 1	100baseT	9.5
HP 9000/735	fddi	8.8
FreeBSD/i586	100baseT	7.9
SGI Indigo 2	10baseT	0.9
HP 9000/735	10baseT	0.9
Linux/i586@90Mhz	10baseT	0.7
DAISy systems		
FreeBSD/i586 (p5-90)	10baseT	0.76
FreeBSD/i586 (p5-90)	100baseT	6.26
HEAT systems		
DEC Alpha	fddi	9.76
HP 9000/735	fddi	9.02
IBM RS6000	fddi	4.54
SUN SS10	fddi	0.76
SGI IRIX	fddi	4.84

Table 11. TCP bandwidth results (MB/s).

4.5.3 *lmbench suite (IPC Latency)*

The cost of communicating between processes or IPC overhead consists of the time required to execute a system call and the time to move the data between processes. The *lmbench* suite implements both TCP and UDP latency IPC micro-benchmarks. The results shown in Table 12 are from the DAISy and HEAT systems and are compared to figures reported in [11].

		TCP		UDP	
System	Network	local host	remote host	local host	remote host
Sun Ultra1	100baseT	162	280	197	308
FreeBSD/i586	100baseT	256	365	212	304
SGI Indigo2	10baseT	278	543	313	602
DAISy systems					
FreeBSD/i586 (p5-90)	10base2	407	731	340	615
FreeBSD/i586 (p5-90)	100baseT	442	572	378	470
HEAT systems					
DEC Alpha	fddi	386	567	412	1089
HP 9000/735	fddi	222	419	225	403
IBM RS6000	fddi	1178	2033	936	1893
SUN SS10	fddi	495	1243	515	1293

Table 12. IPC latency results (microsec).

TCP: TCP connections are typically used in low bandwidth latency sensitive applications. TCP latency is measured by having a server process which waits for connections and a client process that connects to the server. The benchmark passes a token back and forth between the two processes through a TCP socket and measures the round trip time

UDP: UDP sockets are an alternative to TCP sockets. UDP messages are commonly used in client server applications. UDP latency is measured by having a server process which waits for connections and a client process that connects to the server. The benchmark passes a token back and forth between two processes through a UDP socket and measures the round trip time.

4.5.4 Switch performance

The following measurement uses a modified version of the *lmbench* TCP latency micro-benchmark. Specifically, all function calls were replaced with macros in order to minimize overhead. The TCP latency of DAISy's 28115 LattisSwitch was measured using this code as follows. First , two nodes were connected to the frame switch and a request response latency was measured. The test was repeated, this time connecting both nodes together directly using a crossover cable. The difference between the two latencies is attributed to the overhead incurred by sending packets through the switch. The results show a switch latency of 13.74 micro seconds.

LAT_TCP	
571.95 us	*w/switch*
558.21 us	*point-to-point*
13.74 us	*latency through switch*

4.5.5 Discussion

For the *Netperf* suite, a measure of the one-way latency for communication between two workstations is obtained by dividing the request/response time by two. In this case, the performance of DAISy systems was roughly comparable to that of the FDDI connected HEAT systems.

For the *lmbench* suite IPC bandwidth tests, in both the 10Mb and 100Mb network tests the DAISy cluster achieved slightly better than 50% of available bandwidth through the interface. The 100Mb DAISy network lagged significantly behind the performance achieved from the FDDI network connected to each HEAT cluster node. The cause of this performance lag is due to the memory bandwidth of motherboards used in the DAISy cluster. Using Triton PCI chipset based motherboards that support Pipelined Burst SRAM, current Pentium 100 Mhz systems exceed 72 Mbit/s TCP throughput over Fast Ethernet as measured by *Netperf*. The DAISy nodes configured with P5-90 CPUs are not powerful enough to drive the networking hardware to the theoretical maximum.

4.6 Message Passing Library Performance

The application user on the DAISy cluster may use either of two different message passing libraries, PVM3 and MPI. Parallel application performance using message passing libraries can be strongly affected by the performance of the message passing libraries. This performance may be characterized as having a message passing bandwidth and latency. The PVM3 and MPI libraries as used on the DAISy cluster use TCP as the underlying network protocol for node to node communication. In an efficient message passing library implementation the performance of the library communication routines should be close to that of the underlying protocol.

4.6.1 PVM3

PVM3is a popular message passing library available from NETLIB[14]. The design of this library is intended to facilitate heterogeneous network computing, and thus is designed to ensure compatibility of messages passing operations across diverse platforms. However, a goal of DAISy was to construct the highest performing workstation cluster for the least cost. To this end, the design goals for DAISy consider only the homogeneous cluster of Pentium workstations. Thus certain optimizations were made. In particular, the RouteDirect option, which specifies point to point connections between message passing nodes, (and TCP transport) was used for the following measurements.

The PVM timing example is a simple program used to measure PVM message passing bandwidth and latency under PVM. It is a part of the example programs that are included in the PVM distribution. Table 13 shows the results from both the DAISy and HEAT clusters.

System	Network	PVM timing ex.
DAISy systems		
FreeBSD/i586 (p5-90)	10baseT	0.695
FreeBSD/i586 (p5-90)	100baseT	5.284
HEAT systems		
DEC Alpha	fddi	8.242
HP 9000/735	fddi	9.5
SGI IRIX	fddi	6.66

Table 13. PVM timing example results (avg bytes/usec).

4.7 Application Parallel Performance

To verify the viability of commodity workstation clusters, DAISy has been used to perform various parallel computations. The NAS Parallel Benchmarks (PVM [17] and MPI [18] versions) along with a Parallel Seismic Inverse Problem have demonstrated DAISy's cost effectiveness.

4.7.1 NAS Parallel Benchmarks 1.0, PVM version

The NAS Parallel Benchmarks 1.0 (NPB 1.0) consisted of eight benchmark problems. Five of these were kernel benchmarks and three were simulated computational fluid dynamics (CFD) applications. We obtained the PVM versions of the NPB1.0. Unfortunately, only one of the eight benchmarks were able to run in class A mode on the DAISy cluster. This was the "embarrassingly parallel" benchmark EP. As a note, a few of the recently published NPB 2.0 benchmarks running under MPI have run successfully on DAISy and the results are described in the next section.

Kernel EP: Briefly, Kernel EP executes 2^28 iterations of a loop in which a pair of random numbers are generated and tested for whether Gaussian random deviates can be made from them according to a specific scheme. The number of pairs of the Gaussians in 10 successive square annuli are tabulated. The pseudorandom number generator used in this , and in all NAS benchmarks which call for random numbers, is of the linear congruential recursion type. This kernel is viewed and named as an "embarrassingly parallel" application. In other words, improved throughput rather than turn around time. Based on the partitionability of the problem, no data or functional dependencies are incurred, and there is little or no communication between processors.

size = 2^28	Benchmark time (sec)
# of processors	p5-90, 100Mb/s sw
15	287.97
14	307.83
13	331.45
12	358.17
11	391.82
10	430.70
9	481.27
8	540.89
7	616.46
6	718.62
5	865.28
4	1076.66
3	1440.13
2	2152.87
1	4304.07

Table 14. Kernel EP results (sec).

Table 14 shows the scalability of the EP Kernel benchmark on the DAISy cluster. Note that the time it takes to execute the benchmark on one processor is almost exactly fifteen times slower than it would be to execute the benchmark on 15 processors. Hence, “embarrassingly parallel”.

Table 15 shows the results from the DAISy and HEAT clusters using 8 nodes each.

System	Network	Kernel EP
DAISy systems		
FreeBSD/i586 (p5-90)	10baseT	537
FreeBSD/i586 (p5-90)	100baseT	541
HEAT systems		
DEC Alpha	fddi	408
IBM RS6K	fddi	775
SGI IRIX	fddi	1193

Table 15. Kernel EP results (sec).

4.7.2 *NAS Parallel Benchmarks 2.0, MPI versions*

NAS Parallel Benchmarks (NPB) 2.0 [19] currently includes five of the original eight benchmark problems, two of which are kernel benchmarks (FT and MG) and three which are computational fluid dynamics (CFD) application benchmarks (LU, SP, and BT). Results were obtained for the CFD application benchmarks. The benchmarks are based on FORTRAN 77 and the MPI message passing standard. Table 16 shows the various problem sizes for the NAS parallel benchmarks. DAISy runs the Class A problem size. Table 17 shows the standard operation count for the individual benchmarks with a Class A problem size and the MFLOPS results for the DAISy cluster, with the CRAY Y-MP/1 being the standard.

Benchmark Code	Class A	Class B	Class C
Embarrassingly Parallel (EP)	2^28	2^30	2^32
Multigrid (MG)	256^3	256^3	512^3
Conjugate Gradient (CG)	14000	75000	150000
3-D FFT PDE (FT)	256^2x128	512x256^2	512^3
Integer Sort (IS)	2^23	2^25	2^27
LU Solver (LU)	64^3	102^3	162^3
Pentadiagonal Solver (SP)	64^3	102^3	162^3
Block Tridiagonal Solver (BT)	64^3	102^3	162^3

Table 16. NAS Parallel Benchmarks Problem Sizes. From D. Bailey, T. Harris, W. Saphir, R. van der Wijngaart, A. Woo, and M. Yarrow’s “The NAS Parallel Benchmarks 2.0” [1995].

		CRAY Y-MP/1		p5-90/16 100baseT	
Name	Nominal Size, Class A	Operation Count (x10^9)	MFLOPS	MFLOPS total	MFLOPS per process
(EP)	2^28	26.68	211	NA	NA
(MG)	256^3	3.905	176	NA	NA
(CG)	14000	1.508	127	NA	NA
(FT)	256^2x128	5.631	196	NA	NA
(IS)	2^23	0.7812	68	NA	NA
(LU)	64^3	64.57	194	56.01	3.5
(SP)	64^3	102	216	20.5	1.28
(BT)	64^3	181.3	229	73.77	4.61

Table 17. NAS Parallel Benchmarks Standard Operation Counts. From S. Saini, and D. H. Baile’s “NAS Parallel Benchmark Results” [1995].

LU is a simulated CFD application which uses symmetric successive over-relaxation (SSOR) to solve a block lower triangular-block upper triangular system of equations resulting from an un-factored implicit finite-difference discretization of the Navier-Stokes equations in three dimensions. SP and BT are simulated

CFD applications that solve systems of equations resulting from an approximately factored implicit finite-difference discretization of the Navier-Stokes equations. BT solves block-tridiagonal systems of 5x5 blocks; SP solves scalar pentadiagonal systems resulting from full diagonalization of the approximately factored scheme.

Application Benchmark (LU): The LU benchmark code requires a power-of-two number of processors. A 2-D partitioning of the grid onto processors occurs by halving the grid repeatedly in the first dimensions, alternately *x* and then *y*, until all power-of-two processors are assigned, resulting in vertical pencil-like grid partitions on the individual processors. The ordering of point based operations constituting the SSOR procedure proceeds on diagonals which progressively sweep from one corner on a given *z* plane to the opposite corner of the same *z* plane, thereupon proceeding to the next *z* plane. Communication of partition boundary data occurs after completion of computational on all diagonals that contact an adjacent partition. This constitutes a diagonal pipelining method and is called a "wavefront" method. It results in a relatively large number of small communications of 5 words each. Table 18 shows the approximate sustained performance per dollar of DAISy and various systems for the Class A LU benchmark. Results for systems other than DAISy taken from [19].

Computer System	# of Proc.	Memory	Time in seconds	Ratio to CRAY Y-MP/1	List Price Million Dollars	Performance per Million Dollars	Date
CRAY Y-MP	1	NA	333.5	1	NA	NA	Aug-92
Convex SPP1000	32	4 GB	126	2.65	2.5	1.06	Mar-95
CRAY J916	16	2 GB	47.59	7.01	1.05	6.67	Jul-95
CRAY T3D	1024	64 MB/PE	7.09	47.04	3.6	13.07	Mar-95
DEC Alpha Server 8400 5/300	12	2 GB	79.13	4.21	0.718	5.87	Oct-95
IBM RS/6000 SP Wide-node1 (67MHz)	128	128 MB/PE	15.2	21.94	5.08	4.32	Mar-95
IBM RS/6000 SP Wide-node2 (77MHz)	64	128 MB/PE	19.2	17.37	5.74	3.03	Oct-95
IBM RS/6000 SP Thin-node2 (67MHz)	128	64MB/PE	15.9	20.97	3.48	6.03	Mar-95
SGI PC XL (75MHz)	16	2 GB	65.3	5.11	0.895	5.71	Jun-94
SGI PC XL (90MHz)	16	2 GB	65.9	5.06	1.02	4.96	May-95
DAISy	16	64 MB/node	2897.49	0.12	0.06	1.92	Nov-95

Table 18. Approximate sustained performance per dollar for Class A LU benchmark. From S. Saini, and D. H. Baile's "NAS Parallel Benchmark Results" [1995].

Application Benchmark (SP and BT): The SP and BT algorithms have a structure similar to the LU algorithm: Each solves three sets of uncoupled systems of equations, first in the *x*, then in the *y*, and finally in the *z* direction. These systems are scalar pentadiagonal in the SP code, and block tridiagonal with 5x5 blocks in the BT code.

The implementations of the SP and BT solve these systems using a multi-partition scheme. In the multi-partition algorithm each processor is responsible for several disjoint sub-blocks of points ("cells") of the grid. The cells are arranged such that for each direction of the line solve phase the cells belonging to a certain processor will be evenly distributed along the direction of solution. This allows each processor to perform useful work throughout a line solve, instead of being forced to wait for the partial solution to a line from another processor before beginning work. Additionally, the information from a cell is not sent to the next processor until all sections of linear equation systems handled in this cell have been solved. Therefore, the granularity of communications is kept large and fewer messages are sent.

Both the SP and BT codes require a square number of processors. Table 19 and Table 20 show the approximate sustained performance per dollar of DAISy and various systems for Class A SP and BT benchmarks respectively.

Computer System	# of Proc.	Memory	Time in seconds	Ratio to CRAY Y-MP/1	List Price Million Dollars	Performance per Million Dollars	Date
CRAY Y-MP	1	NA	471.5	1	NA	NA	Aug-92
Convex SPP1000	64	4 GB	102	4.62	2.5	1.84	Mar-95
CRAY J916	16	2 GB	77.54	6.08	1.05	5.79	Jul-95
CRAY T3D	1024	64 MB/PE	5.41	87.15	3.6	24.21	Mar-95
DEC Alpha Server 8400 5/300	12	2 GB	102.75	4.59	0.718	6.39	Oct-95
IBM RS/6000 SP Wide-node1 (67MHz)	128	128 MB/PE	18.7	25.21	5.08	4.96	Mar-95
IBM RS/6000 SP Wide-node2 (77MHz)	64	128 MB/PE	26.46	17.82	5.74	3.10	Oct-95
IBM RS/6000 SP Thin-node2 (67MHz)	128	64MB/PE	20.6	22.89	3.48	6.58	Mar-95
SGI PC XL (75MHz)	16	2 GB	67.2	7.02	0.895	7.84	Jun-94
SGI PC XL (90MHz)	16	2 GB	63.18	7.46	1.02	7.32	May-95
DAISy	16	64 MB/node	3883.83	0.12	0.06	2.02	Nov-95

Table 19. Approximate sustained performance per dollar for Class A SP benchmark. From S. Saini, and D. H. Bailey's "NAS Parallel Benchmark Results" [1995].

Computer System	# of Proc.	Memory	Time in seconds	Ratio to CRAY Y-MP/1	List Price Million Dollars	Performance per Million Dollars	Date
CRAY Y-MP	1	NA	792.4	1	NA	NA	Aug-92
Convex SPP1000	64	4 GB	78	10.16	1.25	8.13	Mar-95
CRAY J916	16	2 GB	98.8	8.02	1.05	7.64	Jul-95
CRAY T3D	1024	64 MB/PE	4.56	173.77	3.6	48.27	Mar-95
DEC Alpha Server 8400 5/300	12	2 GB	103.47	7.66	0.718	10.67	Oct-95
IBM RS/6000 SP Wide-node1 (67MHz)	128	128 MB/PE	20.1	39.42	5.08	7.76	Mar-95
IBM RS/6000 SP Wide-node2 (77MHz)	64	128 MB/PE	29.01	27.31	5.74	4.76	Oct-95
IBM RS/6000 SP Thin-node2 (67MHz)	128	64MB/PE	20.8	38.10	3.48	10.95	Mar-95
SGI PC XL (75MHz)	16	2 GB	91.8	8.63	0.895	9.64	Jun-94
SGI PC XL (90MHz)	16	2 GB	80.2	9.88	1.02	9.69	May-95
DAISy	16	64 MB/node	2641.61	0.30	0.06	5.00	Nov-95

Table 20. Approximate sustained performance per dollar for Class A BT benchmark. From S. Saini, and D. H. Bailey's "NAS Parallel Benchmark Results" [1995].

Again, data for systems other than DAISy is taken from [19].

4.7.3 *Discussion of NPB Results*

The evaluation of the performance of the DAISy cluster on the three NPB application benchmarks LU, BT, and SP is complicated by the source and intentions of the data which is being examined. First, the results reported in [19] were obtained under NPB 1.0 rules. However, the data reported for DAISy was obtained under NPB 2.0 rules. The difference in the two sets of rules is essentially that NPB 1.0 rules allow intensive optimization of codes in order to assess the absolute maximum performance obtainable on the algorithm from a particular architecture, while NPB 2.0 rules are intended to determine the performance of a parallel architecture on a portable parallel code using MPI as the message passing standard. The codes run on DAISy were modified only to the extent needed to run; i.e., no algorithmic optimizations were made In view of these differences in the source of data, it is impressive that NPB SP, BT, and LU implementations run on DAISy have price/performance with effectively unmodified, portable MPI message passing codes that exceed that of highly optimized codes on several architectures, and is competitive with many, including shared memory architectures. This comparison between NPB 1.0 results for other systems and NPB 2.0 results from DAISy is necessitated by the lack of NPB 2.0 data at the time this analysis was performed. It is expected that DAISy will exhibit much better comparative performance against most other systems when NPB 2.0 data becomes available for them.

4.7.4 Parallel Seismic Inverse Problem

The DAISy cluster has been used to calculate an inverse problem in seismic tomography. The project goal [19] is to demonstrate a parallel seismic inverse code that runs scalably on inexpensive IBM compatible platforms, incorporating a modular design that separates the parallel algorithm from the specific model used for seismic imaging. The seismic data generated by means of impacts on the earth's surface, consists of timings between generation and reception. The data can be inverted through a tomographic scheme to give a three-dimensional picture of the local rock velocity.

The algorithm is a hybrid of bisection ray tracing and a P-wave Huygens' principle approach and parallelizes in an embarrassingly parallel manner. The algorithm has an adjustable parameter that controls the resolution of the resulting 3D velocity distribution. High resolutions will require ~1 sec between communications, while lower resolutions require ~.01 sec. Though, this code is embarrassingly parallel, it is ideal to test the sensitivity of the cluster to network latency.

Figure 3 shows a three-dimensional rendered image of the subterranean galleries of the "Lucky Friday" silver mine located in Northern Idaho. For acceptable tomographic feature prediction 1 sec to .1 sec is required per task (on the DAISy 90 MHz Pentium). This is useful as a check on the inverse model because the topography of the mine tunnels are measured.

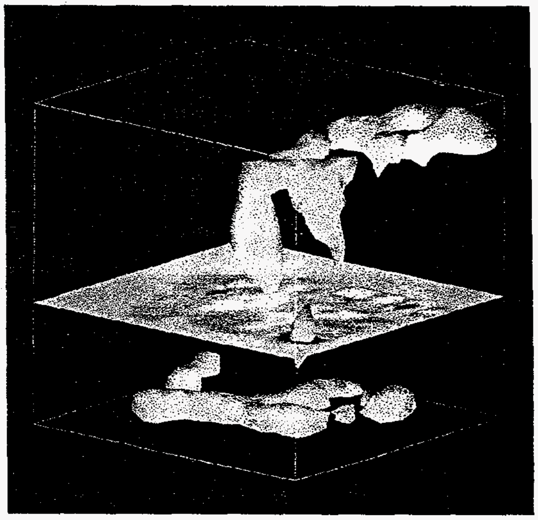

Figure 3. Parallel Seismic Inverse Model. This is the tomographic rendering from seismic data for the "Lucky Friday" silver mine in Northern Idaho. The gold features accurately predict the known locations of the mine galleries. The blue plane is an orthogonal slice through the observation volume. Colors on this plane indicate the effective "sound" velocity of the rock: red is faster; blue is slower.

Figure 4 shows the execution time for the parallel seismic inverse model on various platforms. All runs used the same source code and the GNU C++ compiler (G++) for the native OS without optimization. The PC cluster performs admirably against the considerably more costly workstations.

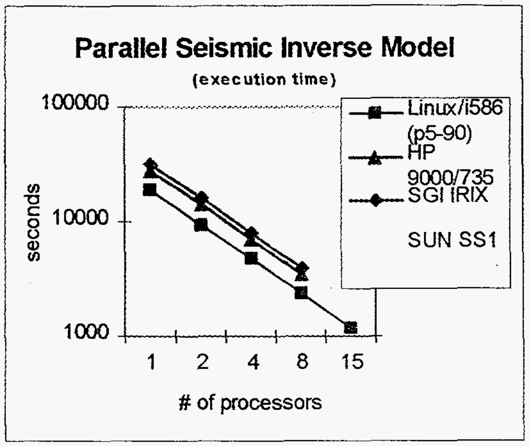

Figure 4. Execution time for Parallel Seismic Inverse Model.

5 Summary

Workstation clusters were originally developed as a way to leverage the better cost basis of UNIX workstations to perform computations previously handled only by relatively more expensive supercomputers. Commodity workstation clusters take this evolutionary process one step further by replacing equivalent proprietary workstation functionality with less expensive PC technology. As PC technology encroaches on proprietary UNIX workstation vendor markets, these vendors will see a declining share of the overall market.

As technology advances continue, the ability to upgrade a workstations performance plays a large role in cost analysis. For example, a major upgrade to a typical UNIX workstation means replacing the whole machine. As major revisions to the UNIX vendor's product line come out, brand new systems are introduced. IBM compatibles, however, are modular by design, and nothing need be replaced except the components that are truly improved. The DAISy cluster, for example, is about to undergo a major upgrade from 90MHz Pentiums to 200MHz Pentium Pros. All of the memory - the system's largest expense - and disks, power supply, *etc.*, can be reused. As a result, commodity workstation clusters ought to gain an increasingly large share of the distributed computing market.

[1] Donald J. Becker, Thomas Sterling, Daniel Savarese, Bruce Fryxell, Kevin Olsen, "Communication Overhead for Space Science Applications on the Beowulf Parallel Workstation", High Performance Distributed Computing Conference, August 1-4, 1995, http://cesdis.gstc.nasa.gov/linux/beowulf/hpdc95.html.

[2] Edward Solari and George Willse, "PCI Hardware and Software", Annabooks, San Diego, CA, 1994.

[3] http://www.freebsd.org.

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

[5] Al Geist, Adam Beguelin, Jack Dongarra, Weicheng Jiang, Robert Mancheck, Vaidy Sunderam, "PVM 3 User's Guide and Reference Manual", Oak Ridge National Laboratory Technical Memorandum 12187, May, 1994.

[6] "MPI: A Message-Passing Interface Standard", Message Passing Interface Forum, May 5, 1994.

[7] Stephen S. Fried, "Pentium Optimization and Numeric Performance", Dr.Dobb's Journal, pp.18-29, January, 1995.

[8] Michael L. Schmit, "Pentium Processor Optimization Tools", Academic Press, Cambridge MA, 1995.

[9] J. J. Dongarra, J. Du Croz, I. S. Duff, and S. Hammarling, A Set of 3 Basic Linear Algebra Subprograms, ACM Trans. Math. Soft., 16 (1990), pp. 1-17.

[10] McCalpin, John D, "Memory Bandwidth and Machine Balance in Current High Performance Computers." Invited for submission to IEEE Technical Committee on Computer Architecture newsletter. To appear December 1995.

[11] L. McVoy, C. Staelin. "lmbench: Portable Tools for Performance Analysis", Proceedings of the USENIX Annual Technical Conference, San Diego, CA, 1996.

[12] R. Weicker, J. Reilly [1995]. "SPEC Frequently Asked Questions (FAQ) / SPEC Primer". WWW Tech. Doc., Web Page (December 15), Siemens Nixdorf, Paderborn/Germany, weicker.pad@sni.de, and (Intel) and R. Weicker, jwreilly@mipos2.intel.com. URL: *http://hpwww.epfl.ch/bench/SPEC.FAQ.html.*

[13] Jack J. Dongarra, "Performance of Various Computers Using Standard Linear Equations Software", University of Tennessee Computer Science Technical Report CS-90-85, February 24, 1995.

[14] http://www.netlib.org

[15] Tim Bray, Bonnie source code, 1990.

[16] Information Networks Division, Hewlett-Packard Company [1995], "Netperf: A Network Performance Benchmark, Revision 2.0", Tech. Rep. (February 15), Hewlett-Packard Company, URL http://www.cup.hp.com/netperf/NetperfPage.html.

[17] S. White, A. Alund, and V.S. Sunderam, [199x]. "Performance of the NAS Parallel Benchmarks on PVM Based Networks", Tech. Rep., Department of Mathematics and Computer Science, Emory University, Atlanta, Georgia.

[18] D. Bailey, T. Harris, W. Saphir, R. van der Wijngaart, A. Woo, and M. Yarrow, [1995]. "The NAS Parallel Benchmarks 2.0", Tech. Rep. (December), NASA Ames Research Center, Moffett Field, California.

[19] Subhash Saini and David H. Bailey, "NAS Parallel Benchmark Results, 12-95", Report NAS-95-021, Dec. 1995, NASA Ames Research Center, Moffett Field, CA.

[20] Rob Armstrong, [1994]. "Parallel Seismic Inverse Model". Tech. Demo. Handout (November), Sandia National Laboratories, Livermore, California.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.