ORNL/TM-12194

Engineering Physics and Mathematics Division

Mathematical Sciences Section

EARLY EXPERIENCES AND PERFORMANCE OF THE INTEL PARAGON

Thomas H. Dunigan

Mathematical Sciences Section
Oak Ridge National Laboratory
P.O. Box 2008, Bldg. 6012
Oak Ridge, TN 37831-6367
thd@ornl.gov

Date Published: October 1994

Research was supported by the Applied Mathematical Sciences Research Program of the Office of Energy Research, U.S. Department of Energy.

Prepared by the
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831
managed by
Martin Marietta Energy Systems, Inc.
for the
U.S. DEPARTMENT OF ENERGY
under Contract No. DE-AC05-84OR21400

Contents

EARLY EXPERIENCES AND PERFORMANCE OF THE INTEL PARAGON

Thomas H. Dunigan

Abstract

Experiences and performance figures are reported from early tests of the 512-node Intel Paragon XPS35 at Oak Ridge National Laboratory. Computation performance of the 50 MHz i860XP processor as well as communication performance of the 200 megabyte/second mesh are reported and compared with other multiprocessors. Single and multiple hop communication bandwidths and latencies are measured. Concurrent communication speeds and speed under network load are also measured. File I/O performance of the mesh-attached Parallel File System is measured. Early experiences with OSF/Mach and SUNMOS operating systems are reported, as well results from porting various distributed-memory applications. This report also summarizes the second phase of a Cooperative Research and Development Agreement between Oak Ridge National Laboratory and Intel in evaluating a 66-node Intel Paragon XPS5.

1. Introduction

The Department of Energy selected Oak Ridge National Laboratory (ORNL) as one of its high performance computing centers as part of the government's High Performance Computing and Communications (HPCC) initiative. The initiative provided ORNL with funds to procure a massively parallel computer and to support various Grand Challenge applications. ORNL selected Intel to provide the massively parallel computer for the HPCC project. The agreement with Intel specified the staging of increasingly more powerful versions of its new Paragon multiprocessor. As part of the agreement, ORNL would receive pre-production models of the Paragon and assist in beta testing and product development.

This report summarizes our early experiences with the Intel Paragon. Our evaluation and testing of the Paragon involved testing end-user UNIX services (editors, compilers, file management, etc.), system administration services (account management, partition management, batch queuing support, network services, backup/restore, etc.), and porting various parallel applications onto the new software platform. The initial testing was done with test suites and small parallel applications that ran on Intel's iPSC/860 and Delta multiprocessors. As soon as the hardware and software had stablized, the Grand Challenge applications were ported to the Paragon. Bugs and problems were reported to on-site Intel staff, and the Intel design team consulted with ORNL in setting design directions and priorities for the evolving Paragon system.

This report also provides initial performance characteristics of the Paragon. Computational and communication performance were measured with synthetic benchmarks, application kernels, and a few parallel applications. The Paragon's performance is compared with the performance of other currently available parallel processors.

In the following section, the Paragon architecture is summarized, and the configuration of the ORNL Paragons is detailed. Section 3 describes the Paragon operating systems. In section 4, the performance of the i860XP is compared with the i860 processor. The message-passing performance of the Paragon mesh is reported in section 5, and preliminary performance of the Paragon's file system and local area network interfaces are analyzed in section 6. Parallel application performance of the Paragon is examined in section 7, and section 8 summarizes our initial experiences with the Paragon.

2. Paragon architecture and configuration

The Intel Paragon system is a mesh-connected parallel processor. Each Paragon node consists of two 50 MHz i860XP processors, memory, and communication hardware (Figure 2.1). One processor is used for computation, and the second processor is for communication. (The communication processor became operational in May, 1994.) The bus interconnecting the processors and memory operates at 400 MB/second. Each compute node is presently configured with 32 million bytes of memory. The initial configuration had only 16 million bytes of memory, but that proved inadequate.

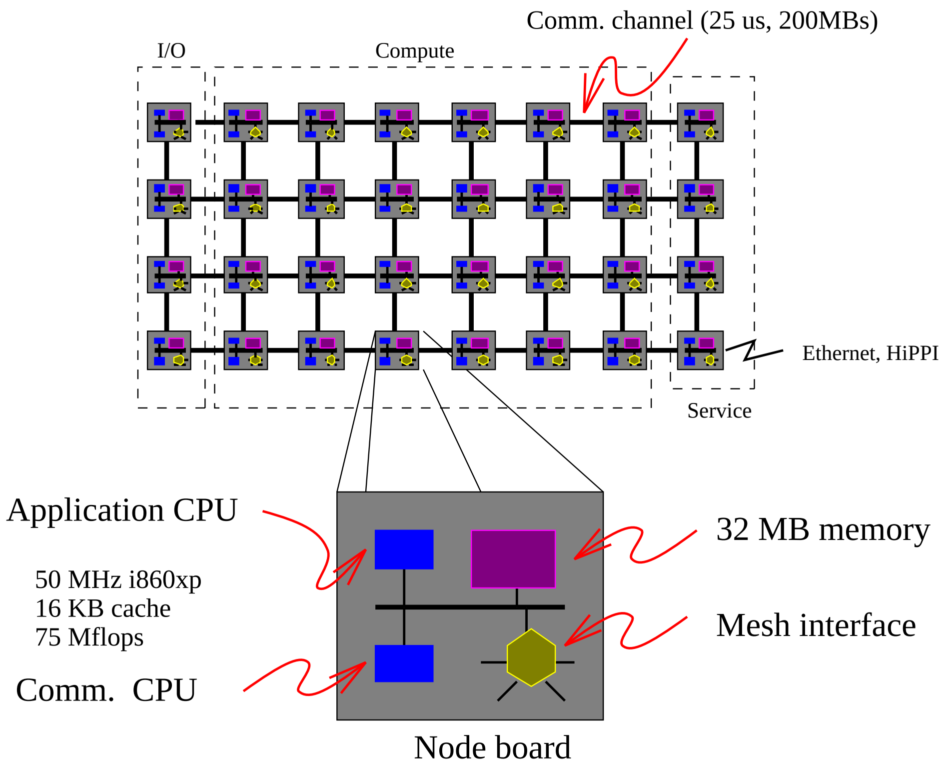

Figure 2.1: *Paragon mesh and nodes.*

The nodes are logically subdivided into service nodes, compute nodes, and I/O nodes. The service nodes appear as a single host and support time-sharing through the OSF operating system. The compute nodes also run OSF. The I/O nodes are connected to local networks and arrays of disks (RAID) and provide a UNIX file system, swap/paging space, and a Parallel File System (PFS). Since the service nodes are used for time-sharing and loading the compute nodes, a “host” is not required as in the earlier Intel architectures (Delta and the iPSC series).

The nodes are interconnected by a mesh. The speed of a single mesh channel was designed to be 200 MB/second, but the delivered Paragons provided a maximum of only 175 MB/second. Per hop delay through the mesh is only 40 nanoseconds. Based on analytical studies and simulations, Intel chose the mesh architecture because it provides the most efficient use of available wires. Given the same number of wires, a mesh will outperform any hypercube, toroidal, or tree-structured network for uniformly distributed communications traffic [12].

Two Paragons were delivered to ORNL in September, 1992. A 66-node system with 14 Gigabytes of disk was provided under a Cooperative Research and Development Agreement (CRADA) and was used primarily as the program development machine. A 512-node system with 150 Gigabytes of disk was the interim production machine, eventually to be replaced with a 2048-processor machine. The i860XP's were running at only 40 MHz in these initial machines. Each Paragon was connected to the local Ethernet, and later each was also attached to HiPPI.

For comparison, the following sections include performance data from the Intel Delta and Intel iPSC/860. The Delta is the one-of-a-kind predecessor to the Paragon. The Delta is a mesh-based multiprocessor based on the 40 MHz i860 processor and the NX node operating system. The peak bandwidth of a channel in the Delta mesh is 22 MB/second. The iPSC/860 is a hypercube multiprocessor based on the same processor and OS as the Delta. The peak bandwidth of one of the hypercube's channels is 2.8 MB/second. Both the Delta and iPSC/860 support a parallel file system (CFS) similar to the Paragon's PFS. The iPSC/860 and Delta configurations and details of the benchmarks are described in [5] and [7].

The Paragon is Intel's first production-oriented mesh multiprocessor. The Intel iPSC series were all based on a hypercube topology. The mesh has some potential advantages over a hypercube topology. Though both topologies are extensible, in practice, commercial hypercubes have a fixed maximum dimension. For example, the largest iPSC/860 is seven dimensions or 128 processors. Hypercubes must be expanded in powers of two, which is often prohibitively expensive. Meshes can be expanded at linear costs by adding an additional row or column. Of course, the hypercube topology has advantages as well. The maximum distance between two processors in an n processor system is only $log_2 n$ for a hypercube, compared with $\sqrt{n}$ for the mesh. The lower connectivity of the mesh may lead to communication "hot spots" in the mesh or to slower aggregate communication operations such as barriers. Our tests and analyses in the following sections will attempt to identify the strengths and weaknesses of the Paragon's mesh topology.

3. OSF and SUNMOS

The Paragon operating system support differs from both the Delta and iPSC series of hypercubes. For the older Intel parallel processors, compilers and editors were provided on a small host processor or by cross compilers on the user's workstation. For the Delta and the Intel hypercubes, a small kernel OS (NX) on the nodes provided message passing, memory management, and a UNIX I/O library. For the Paragon, time-sharing services are provided by OSF running on a set of mesh nodes (service nodes). In the later releases of the Paragon OSF, the service nodes provided some limited parallel processing of user services, so that different users would likely be running on different service nodes. Our beta testing included exercising the user services (editors, file system, accounting, compilers, linker, etc.) on the OSF service nodes. The OSF services worked, though initially performance was slow. Performance has improved with each software release, but overall UNIX performance as measured by a set of UNIX benchmarks is still slow in comparison with current workstations.

Instead of a tiny kernel like NX on the nodes, the Paragon provides the OSF micro-kernel on each node. OSF on each node provides a more comprehensive set of of services to the node programmer, but at a cost of memory and some additional overhead. The OSF kernel provides virtual memory, permitting larger node programs to be run on the nodes than might fit in physical memory. However, the paging of memory to the I/O nodes has proven a bottleneck to date, and the benefits of virtual memory have diminished. The OSF kernel on the compute nodes provides OS services through the OSF interprocess communications facility (NORMA IPC) which in turn sits on top of inter-node message passing services. The present implementation of the OSF IPC has limited the performance of file I/O and network I/O.

The software overhead of OSF and the inability to use the message co-processor initially prevented parallel applications on the Paragon from matching the performance of its predecessor the Delta. To provide an alternative node operating system, Sandia National Lab and the University of New Mexico developed a small (256K byte) compute node kernel called SUNMOS [14]. SUNMOS runs in the compute partition, supporting the same message-passing primitives as OSF and NX. OSF is still used on the service nodes. SUNMOS does not provide virtual memory, and its I/O support is not fully developed, but SUNMOS provides higher bandwidth for large messages than OSF.

4. Computational performance

The CPU for the Paragon is the 50 MHz i860XP, an enhanced version of the 40 MHz i860 CPU in the Delta and iPSC/860. The i860XP has the same instruction set as the i860 and so is software compatible. The i860XP has a 16KB instruction and data cache, twice that of the i860. In addition the speed of the memory bus has been increased from 160 MB/second to 400 MB/second. The super-scalar architecture is capable of 75 Mflops (double precision).

Our CRADA agreement with Intel resulted in our being able to evaluate early releases of the hardware and software. Our initial Paragon configurations had 40 MHz i860XPs until March, 1993. Single-node performance from these 40 Mhz chips and early software was disappointing. For example, single-node Linpack performance was actually slower than the 40 MHz i860. Of course, evaluation and development in these early months was concentrated on OSF reliability and stability issues and not on absolute performance.

With the 50 Mhz i860XP's installed, single node performance improved to roughly 20% faster than the i860 processor over the set of benchmarks described in [2]. For example, the 100×100 double-precision FORTRAN Linpack ([1]) ran at 10.9 Mflops on the 50 MHz i860XP versus 9.7 Mflops on the i860. A FORTRAN radiosity code ([10]) that includes some I/O ran at 4.8 Mflops on the i860XP versus 2.8 Mflops on the i860. A C Cholesky factorization and a C numeric integration ran 25% and 31% faster on the i860XP than on the i860.

Finally, application performance on a single-node is affected by the amount of memory available. The OSF kernel consumes about 6 megabytes. By contrast the SUNMOS kernel takes less than 1 megabyte. NX on the Delta consumes about 4 megabytes, and NX on the iPSC/860 consumes about 1 megabyte. Memory consumption varies based on message buffer allocations. Memory consumption was measured with a simple *malloc()* loop on NX and SUNMOS. For the virtual-memory OSF, a vector-touch loop was run over larger and larger vectors until performance drops indicating that paging has begun. The larger memory consumption of OSF has to be balanced against the additional features (e.g., virtual memory) it provides the application programmer. In general, message buffer requirements grow with the number of nodes, so the amount of memory available to an application will diminish as more nodes ares used. The need for larger-memory nodes in large (greater than 512 nodes) configurations is a system design issue that was identified in our early evaluation process.

5. Communication Performance

In this section, we analyze the communication performance of the Paragon mesh, first looking at adjacent node performance, then at communication to more distant nodes. The communication tests were performed under OSF 1.2 with the communication processor enabled. Various communication patterns are analyzed to determine how much concurrency the Paragon mesh can support and when contention degrades performance.

5.1. Node-to-node communication

In the first test, a simple echo test is used, where a message is sent and echoed back by the receiver. The sender measures the round-trip time for 1000 iterations. Figure 5.1 shows the data rate for two adjacent nodes echoing messages of various message lengths. The data rate increases with message sizes from 8 to 8,192 bytes. The Paragon using SUNMOS reaches a data rate of about 65 MB/s for a message size of 8,192 bytes. By contrast, the Paragon with OSF achieves 45 MB/s, though, as the figure illustrates, OSF's data rate exceeds SUNMOS for smaller messgaes. The cross-over point occurs roughly at where OSF segments messages into 1792-byte packets. SUNMOS does not segment messages.

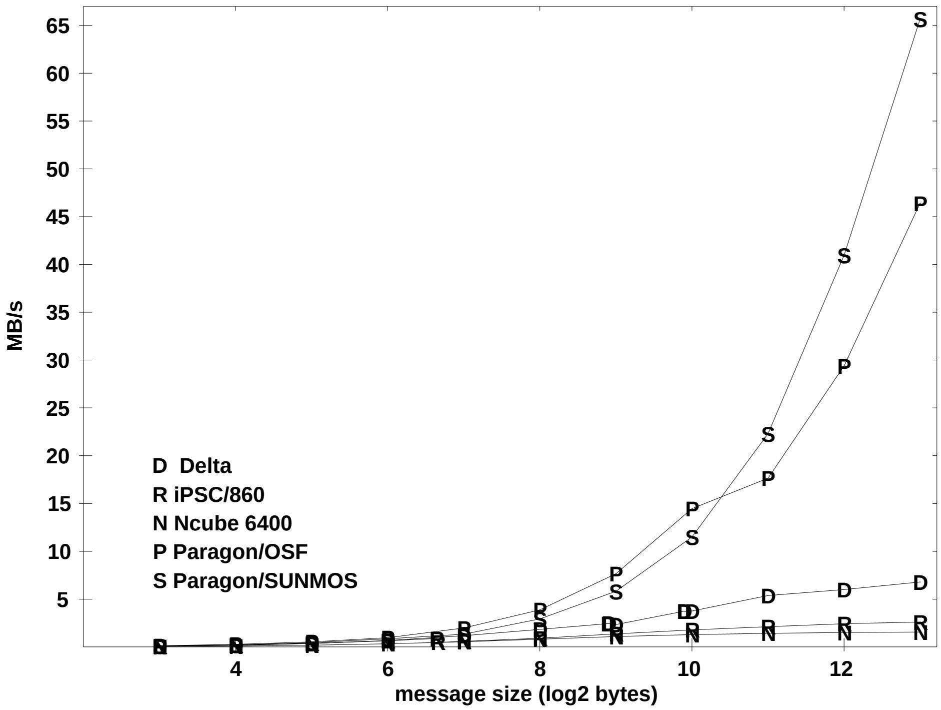

Figure 5.1: *Nearest neighbor echo data rates.*

Earlier generation message-passing machines (Intel iPSC/2 and iPSC/860) exhibited slower data rates if communication was not with the nearest neighbor ([7]). The Paragon, like its predecessor the Intel Delta (and to a lesser extent the Ncube 6400), communicates nearly as fast with the most distant node in the network as it does with its nearest neighbor. In particular, at the moment, differences in communication speed across the Paragon mesh are hidden in the measurement error of the experiments. The specifications for the Paragon mesh suggest that only 40ns are required for each hop [12]. Thus for a 16×32 mesh less than 2 μs are added to the communication times between most distant nodes. As noted below, the minimum nearest-neighbor communication times are currently about 45 μs, so multi-hop overhead is less than a few percent.

In our earlier analyses of message-passing systems ([7] [2] [3], we modeled the message-passing time, T, as a linear function of start-up time, α, a per-byte cost, β, and a per-hop delay, γ.

$$T = \alpha + \beta N + (h - 1)\gamma$$

We used a linear least-squares fit of our experimental results to calculate the startup and per-byte parameters. However, the experimental data from the Paragon and other new architectures are not as well supported by a linear fit, and the calculated parameters are very sensitive to the set of data points used in the fit. For purposes of comparison, Table 5.1 shows the communication coefficients for both OSF and SUNMOS on the Paragon. The sample data is from echo times for 8 byte to 8,192 byte messages. For the Paragon, the per-hop penalty is buried in experimental error.

Coefficients of Communication microseconds					
	OSF	SUNMOS	Delta	iPSC/860	N6400
Startup (α)	62	93	72	136	154
Byte transfer (β)	0.02	0.01	0.08	0.4	0.6
Hop penalty (γ)	0.04	0.04	0.05	33	2

Table 5.1: *Least-squares estimates of communication coefficients.*

To the extent that one can characterize communication with one or two numbers, we now prefer to use the time to send a zero-length message as one metric, and the data rate for a one million byte message as another metric. Table 5.2 shows our experimental measurements on the Paragon and other parallel systems for zero-length and megabyte messages for nearest neighbor communication.

Communication Parameters		
	0 bytes μs	1 megabyte MB/sec
Paragon/SUNMOS	65	166
Paragon/OSF	45	89
Delta	77	8
iPSC/860	65	3
CM5	95	9
IBM SP1	253	7
Cray T3D	1	120
KSR (tcgmsg)	73	8

Table 5.2: *Nearest neighbor time and data rates for 0 byte and one megabyte messages.*

The extra time required for a multi-hop message is more clearly seen if we look at the time for sending a zero-length message (Figure 5.2). Though the bandwidth between nodes has increased on the Paragon in comparison to the Delta and iPSC/860, the zero-length message time (latency) has improved only marginally, even though the 50 MHz i860XP is a faster processor. The latency is dominated by house-keeping chores (argument checking, context switch on interrupt, etc.) on both the sending and receiving nodes. In a separate study ([4]), the time to handle the time-slice interrupt on the iPSC/860 was about 50 microseconds, which suggests that interrupt context switch overhead could be the dominant factor in message latency. With the communication processor disabled, latency on the Paragon climbs to 85 μs and bandwidth is reduced by a factor of two. Intel hopes that the latency on the Paragon can be reduced to 25 μs.

Figure 5.3 further illustrates the difference in performance and variability of message passing with and without the message processor under OSF. The figure shows the distribution of round-trip times for 2,000 samples using an 8-byte message. The echo test was run both with a nearest neighbor and from corner-to-corner in the 512-node mesh using both OSF and SUNMOS. Notice that the variance is such that it is possible to observe round-trip times that are faster corner-to-corner than to nearest neighbor.

Even though the communication performance of the Paragon and Delta is generally better than the iPSC/860, the hypercube topology performs some communication primitives faster than the mesh. For example, using Intel's *gsync()*, barrier synchronization time grows with the number of nodes for the mesh, but only as the *log* of the number of nodes for the hypercube (Figure 5.4).

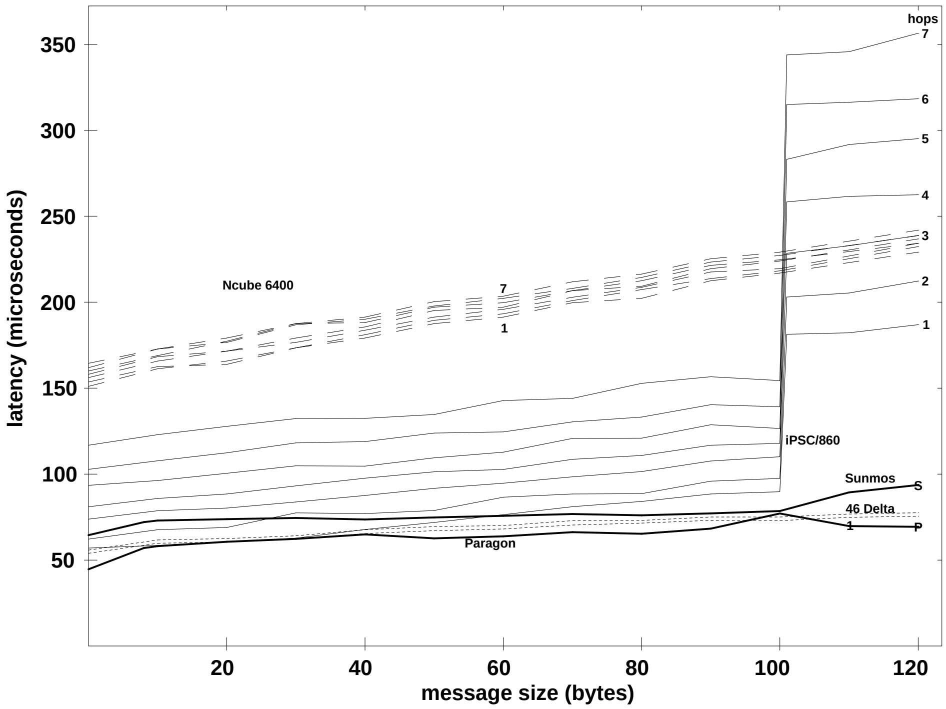

Figure 5.2: *Echo test message latency.*

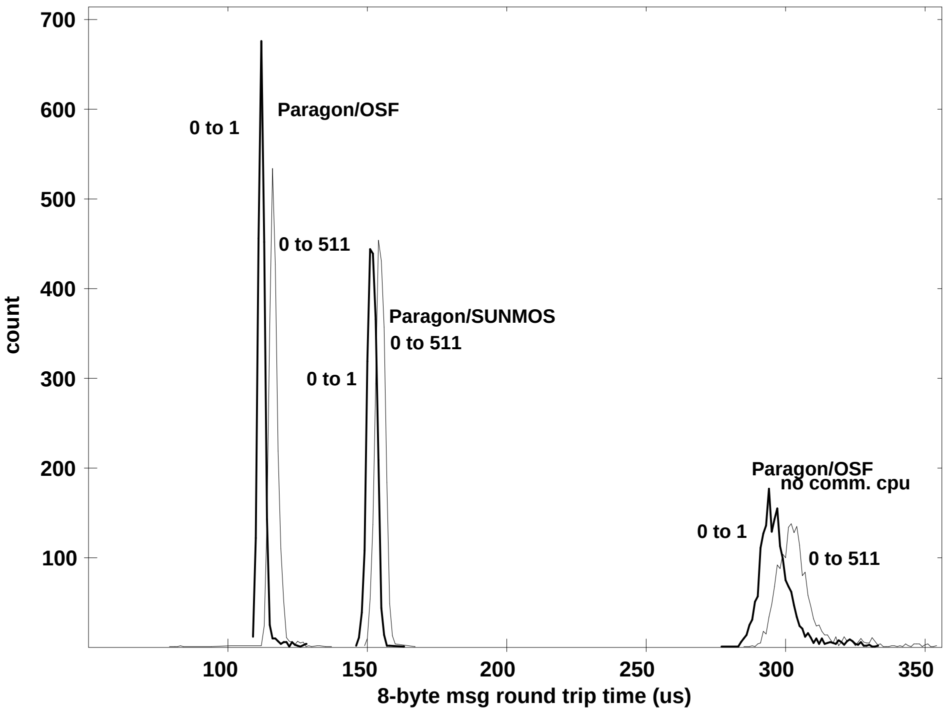

Figure 5.3: *Paragon roundtrip times for 8-byte message.*

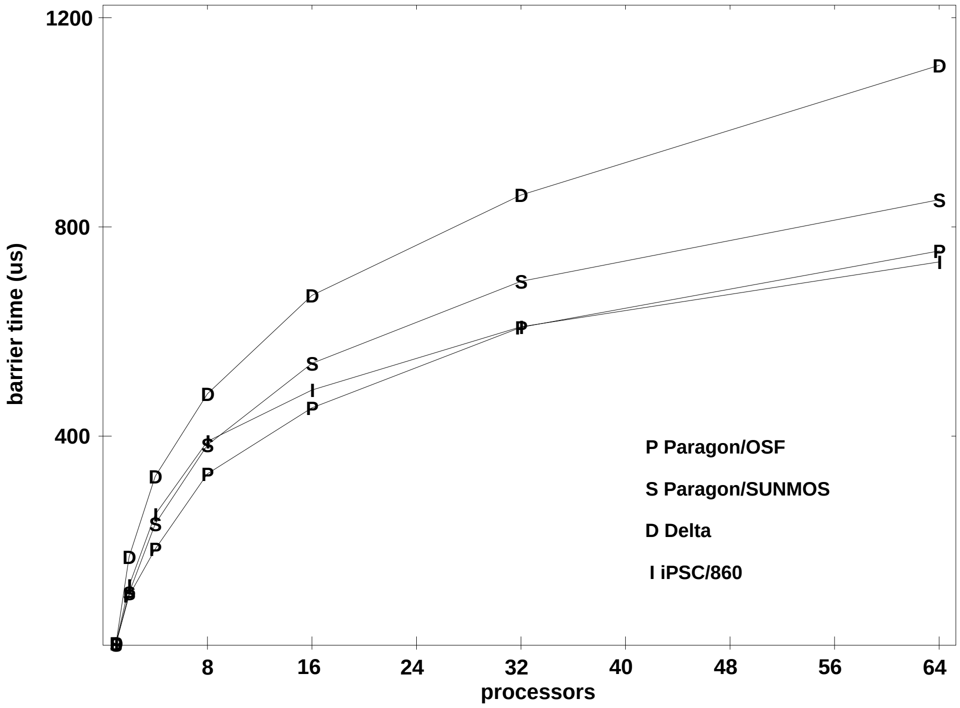

Figure 5.4: *Barrier times.*

5.2. Contention

All of the communication data rates that we have reported have been measured on idle systems. In actual applications, other message traffic may compete for the communication channels, either from the application itself or from applications in other partitions. One partition may need to use another partition's communication channels to reach the I/O processors or other service nodes. The Paragon, iPSC/860, Delta, and Ncube 6400 use circuit-switching to manage the communication channels. When a message is to be sent, a header packet is sent to reserve the channels required. When this "circuit" is established, the message is transmitted, and an end-of-message indicator releases the channels. SUNMOS reserves the channel for the entire message. Paragon/OSF breaks a message up into packets (usually 1792 bytes), and the circuit is only reserved for the packet. This packetizing can add to the overhead of a message, but permits multiplexing the links of the circuit with other nodes.

A program was developed to measure the effect of contention on the data rate of a communication channel and to measure the capacity of a given physical link. The link-contention program developed for the hypercube [7] proved inadequate for the higher speed meshes of the Delta and Paragon. Link contention was measured on a row of the mesh with varying numbers of pairs doing synchronous sends of one megabyte messages in one direction. It was observed that the interior pair completed first, followed by the next innermost pair, and so on. The outermost pair finished last. (Note the inner pairs continued to send data after the timed portion of their transmission completed.) For both OSF and SUNMOS, contention occurs when the aggregate data rate exceeds about 160 MB/second (Figure 5.5). (Recall, the peak channel bandwidth is 175 MB/second.) The slower data rate of the OSF nodes, means that more OSF nodes can be sending before contention occurs. For the Delta, aggregate channel throughput under contention is about 11 MB/second. The effect of contention can vary from run to run and can slow down an application. Since a mesh has fewer channels between nodes than the hypercube architecture, one would expect increased contention for the mesh channels. But contention will occur on both mesh and hypercube channels when the aggregate sending rate of nodes on the channel exceeds the channel bandwidth. A more detailed analysis of channel contention on the Paragon is reported in [13].

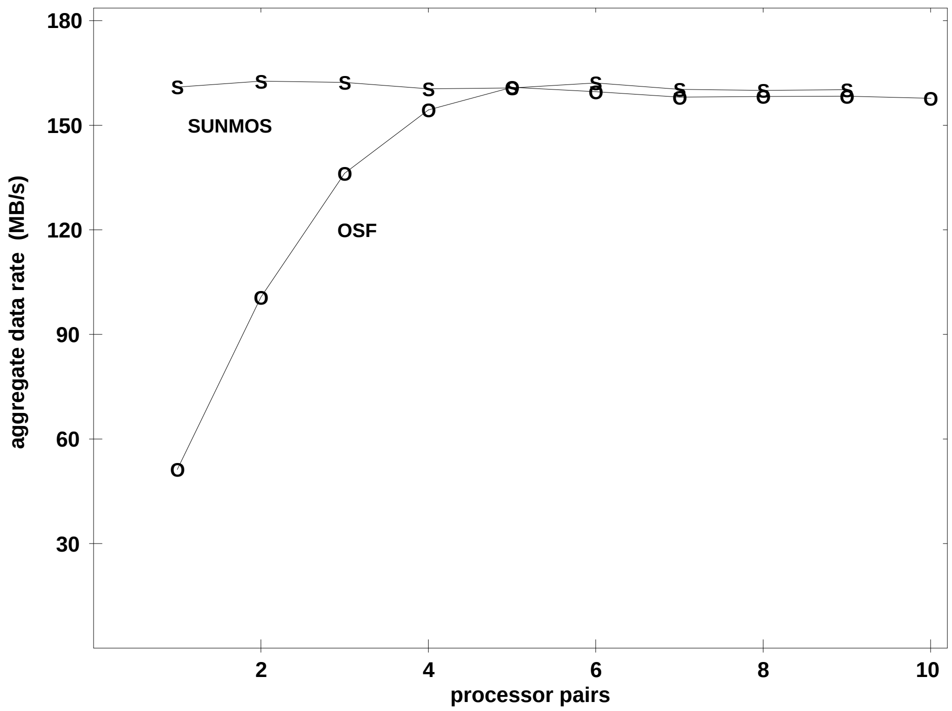

Figure 5.5: *Aggregate data rate for pairs sending on the same channel.*

5.3. Concurrent Communication

The message-passing performance of a node may be improved by utilizing more than one of its communication channels at the same time. A fan-in test was used in our earlier tests on hypercubes [6], but only the Ncube was able to show a higher aggregate receive date rate. The Intel machines (including the Paragon) have a single receive FIFO and a single transmit FIFO, so it is only possible to receive from one channel at a time. However, for the iPSC/860, it is possible to nearly double the aggregate data rate of a node by doing an exchange using FORCE_TYPE, that is, a node concurrently sends and receives with another node. However, so far, we have not been able to achieve the same result on the Paragon.

6. File and Network Performance

Paragon file I/O and access to local area networks are provided through one or more I/O or service nodes. These nodes usually reside on the outer columns of the mesh. Communication to the I/O or network nodes uses OSF interprocess communication (NORMA IPC) layered on top of underlying mesh communication primitives. The OSF IPC is presently limiting performance.

6.1. Parallel File System

The Paragon OSF provides both a standard UNIX file system and a larger, high performance parallel file system (PFS). The system manager can configure the I/O nodes and disks into combinations of UNIX and PFS file systems. A typical configuration would be to allocate the disks of an I/O node as a mountable partition in the UNIX file system. PFS is typically configured across a set of I/O nodes and disks. The PFS is striped across one or more I/O nodes using the disk RAID arrays and appears to the UNIX system as a separate mountable file system (e.g., */pfs*). Normal C and FORTRAN I/O operations can be used on PFS, but optimum performance is achieved using special open calls.

Several I/O benchmarks were used to characterize the performance of PFS. The benchmarks measured I/O throughput from a single compute node and from many compute nodes doing I/O concurrently. The tests were run with a varying number of I/O nodes in the PFS configuration. Figure 6.1 shows the aggregate read throughput of PFS when varying number of compute nodes are concurrently reading an independent portion of a file. The test (written in C) uses *gopen()* with the M_RECORD option. The aggregate data rate increases with the number of compute nodes doing I/O, though the individual node I/O rate decreases as parallelism is increased. More I/O nodes yields higher I/O throughput, though eight I/O nodes actually performs better than ten I/O nodes. (We do not yet have a satisfactory explanation for that anomaly.)

For comparison, Figure 6.1 also illustrates I/O performance for the Intel Delta and iPSC/860 [7]. The I/O nodes on both of these systems are based on 80386 processors with only 4 megabytes of memory. The Delta system provides 32 I/O nodes supporting the Concurrent File System (CFS). The iPSC/860 configuration had only 10 I/O nodes. Like PFS, CFS files are striped across the drives. On the

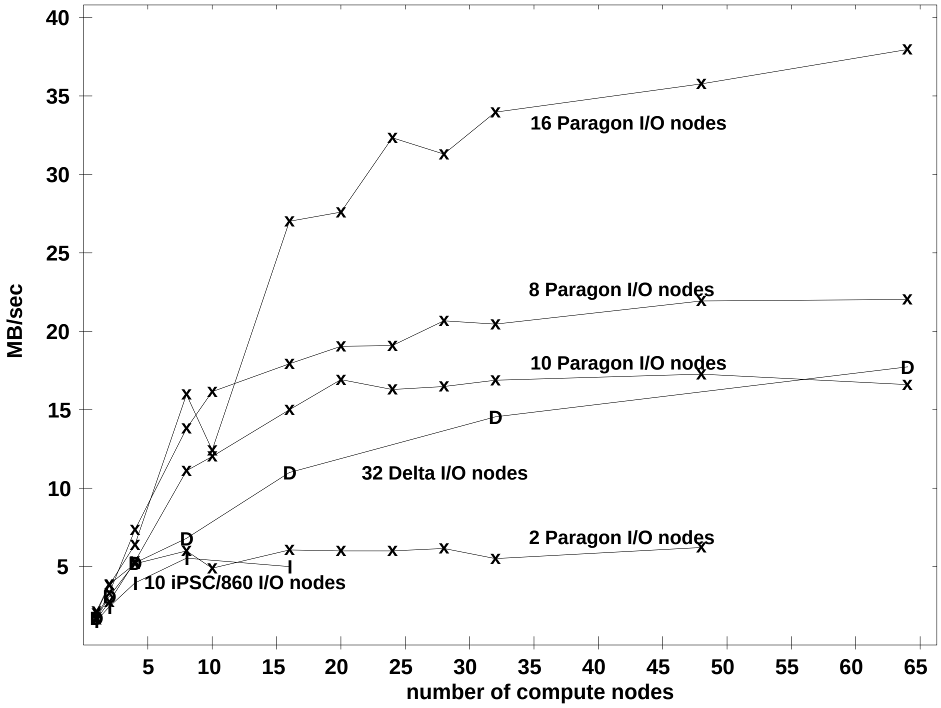

Figure 6.1: *PFS read throughput.*

Paragon, a single compute node can read disk data at about 2 MB/second. (We suspect that this data rate is limited by the overhead of the OSF NORMA IPC.) The aggregate read data rate flattens as the number of compute nodes doing I/O exceeds the number of I/O nodes. The steady-state aggregate data rate appears to be about 2.4 MB/second per I/O node for the Paragon, about five times faster than the Delta or iPSC/860. With 16 I/O nodes, PFS delivers an aggregate read data rate of 38 MB/second.

6.2. Network Performance

The ORNL Paragons have both Ethernet and HiPPI interfaces. The maximum speed of Ethernet is 1.2 MB/s, and HiPPI is 100 MB/s. The Paragon's Ethernet performance is limited by the OSF NORMA IPC overhead in communicating between the Ethernet service node and a computation node. Paragon

Ethernet/TCP performance reaches only 435 KB/s for 16 KB messages. That throughput modestly exceeds what we measured on the Delta and iPSC/860 Ethernets ([7]), but still is slower than most commercial workstations. Our tests with the HiPPI yielded 8 MB/s using a native API (not TCP/IP). We achieved 3 MB/s for 64 KB messages using TCP between the HiPPI attached Paragons. FDDI support for the Paragons is presently provided through a HiPPI-FDDI router. The Paragon FDDI/TCP performance was 2.5 MB/s for 64 KB messages.

7. Performance summary

The Intel Paragon mesh provides improved communication performance over the Intel Delta mesh and iPSC/860 hypercube. The Paragon mesh provides wider and faster communication channels between nodes, plus faster routing hardware, but the reduced connectivity of the mesh slows some communication primitives such as barriers. The message startup times are nearly identical for the Paragon and earlier Delta and iPSC/860. Table 7.1 summarizes the communication and computational performance of the Paragon. The data rates represent the 8192-byte transfer speeds, and the megaflops rate is calculated from a five operation expression [6]. The 8-byte transfer time is based on the 8-byte, one-hop, echo times. The structure of a parallel algorithm will be dictated by the amount of memory available on a node, the host-to-node communication speed, and the ratio of communication speed to computation speed. As can be seen from the table, the three Intel machines have roughly equivalent communication-to-computation ratios. (The ratio was calculated using the 8-byte transfer and multiply times.) For larger messages, the Paragon and Delta would show a more balanced ratio than the iPSC/860.

Figures of Merit				
	Paragon	Delta	iPSC/860	N6400
Data rate (MB/s)	65.7	11.9	2.6	1.6
Megaflops	22	18	18	2.5
8-byte transfer time (μs)	59	91	80	161
8-byte multiply time (μs)	0.07	0.08	0.08	1.5
Comm./Comp.	843	1138	1000	107

Table 7.1: *Summary performance metrics.*

To compare the performance of the Paragon to the earlier machines in an application involving both communication and computation, we solved a 1024×1024 linear system of equations (C double precision) using Cholesky factorization on 16 nodes. The Paragon with OSF ran at 46.1 Megaflops and at 51.2 Megaflops with SUNMOS. The Delta ran at 30.7 Megaflops compared with 22.3 Megaflops from the iPSC/860 (the Ncube 6400 was 5.3 Megaflops). These results are consistent with the LINPACK results reported in [1]. The LINPACK peak performance (measured by solving the largest linear system the memory can support) for 128 nodes was 4.1 Gigaflops for the Paragon/OSF and 3.6 Gigaflops for the Delta versus 1.9 Gigaflops for the iPSC/860 (8 Megabytes) and 0.24 Gigaflops for the Ncube 6400 (4 Megabytes) [1]. Using all available nodes, the peak LINPACK was 18.4 Gigaflops for the 512-node Paragon/OSF and 13.9 Gigaflops for the 512-node Delta versus 1.9 Gigaflops for the 1024-node Ncube. (The maximum number of nodes for an iPSC/860 is 128.)

To measure the performance of all of the Paragon subsystems (computation, communication, and I/O), we ran the FORTRAN SLALOM benchmark (version 1) [11] on a 64-node mesh. On a 64-node Paragon OSF mesh, SLALOM ran at 639 Megaflops. SLALOM on the Delta ran at 258 Megaflops on a 64-node mesh, as compared with 172 Megaflops for a 64-node iPSC/860.

8. Experiences

We have evaluated several serial-number-one parallel processors at ORNL, beginning in 1985 with Intel iPSC/1 hypercube. These early machines were used for algorithm development, performance analysis, and, to the degree possible, porting existing applications or developing new applications. Our initial testing of the Paragon was through a Cooperative Research And Development Agreement (CRADA) between ORNL and Intel. This first phase of the CRADA provided us with an iPSC/860 running OSF. Our testing involved evaluating the new OSF as well as porting some of our iPSC/860/NX hypercube applications to the new OSF environment. We were pleasantly surprised at the quality of the OSF implementation.

The second phase of the CRADA included the delivery of 66-node, i860XP-based, Paragon mesh. Initially, this unit was to have preceded the 512-node machine by several months, but schedules slipped, and the 66-node machine and 512-node machine arrived within a week of each other. As expected in a beta test, the hardware and software had bugs, and our initial efforts were directed at identifying the critical problems and working out solutions with the Intel staff. As part of the contract, Intel provided hardware and software personnel on site, so feedback was fast and effective. The developers at Intel Corporate would often have new software releases the day following a critical bug report.

Our testing on the Paragon consisted of program development of benchmark codes, porting working iPSC/860 codes, and performing system administration functions. A number of UNIX applications that run on single-processor UNIX systems were compiled and run under the OSF beta system. These applications included UNIX commands, benchmarks, various network servers, simulators, PVM [9], PICL [8], and component tests. These applications exercised system services such as file I/O, shared memory, semaphores, process creation, pipes, signals, network sockets, and shell scripts. In addition, POSIX and UNIX test suites were run. Though performance was not an issue during this early development and testing, the results from the various benchmarks did reveal various component inefficiencies that were relayed to the Intel team. The time-sharing services of OSF remained reasonably stable, though compile times were slow initially. File and network I/O were unusually slow, due to inefficiencies in the interprocess communication facilities of the OSF implementation. File and network I/O performance have improved, but still are not competitive with typical workstations.

The porting of parallel applications (working iPSC/860) codes was successful for smaller applications. Those codes that depended on a host (SRM) had to be recoded to be hostless. Some applications did not port because of the limited application memory space on the compute nodes. Initially, the OSF kernel was taking nearly 8 megabytes of memory on each of the compute nodes. Virtual memory was supported, but if an application started swapping, performance was very very poor and often was the cause of crashes. Though problems were fixed quickly, patches and new releases required re-running all of our tests. Occasionally, features that had been working would fail in a new release.

The 9-cabinet, 512-node system had early problems with grounding and noise on the communication channels. Many scaling problems with OSF were uncovered. Operating system tables were not properly sized for hundreds of processors. For several months, the maximum number of nodes in a single application was limited to 256. Multiprocessing of the service nodes was not initially supported, and memory bottlenecks hurt service node performance. Eventually, both the service nodes and compute nodes were upgraded to 32 megabytes of memory.

Although the evaluation of early systems and the CRADAs were partial justification for procuring the Paragons, the primary purpose of the ORNL Paragon was to provide a tool for computational science. Much of the testing and evaluation centered around porting the three Grand Challenge applications to the Paragon. The material science application was already running an early version on the 128-node iPSC/860. Porting that code to the Paragon was successful. The application uses PFS, dynamic memory allocation, and is achieving near linear speedups. On a per node basis, the Paragon version performs 1.7 times faster than the iPSC/860 version, and delivers 17 Gigaflops on the 512-node Paragon. Porting the global climate modeling application to the Paragon from the Delta has been more difficult than anticipated, primarily because of file I/O bugs and inefficiecies. However, the Paragon version is running about 1.5 times faster than the Delta version. The contaminant transport application did not have a fully developed parallel implementation, so progress on the Paragon has been difficult to measure.

The 66-node and 512-node Paragon systems are providing parallel computing cycles to a nationwide community primarily working on the three Grand Challenge projects. Performance and reliability continue to improve with each release, but performance still remains below expectations. The usefulness of OSF on the compute nodes is still a matter of debate in view of its performance and memory liabilities.

9. References

[1] J. Dongarra. Performance of various computers using standard linear equations software. Technical report, University of Tennessee, January 1993. CS-89-85.

[2] J.J. Dongarra and W. Gentzsch. *Computer Benchmarks*. North-Holland, Amsterdam, 1993.

[3] T. H. Dunigan. Performance of the Intel iPSC/860 hypercube. Technical report, Oak Ridge National Laboratory, Oak Ridge, TN, 1990. ORNL/TM-11491.

[4] T. H. Dunigan. Hypercube clock synchronization. Technical report, Oak Ridge National Laboratory, 1991. ORNL/TM-11744.

[5] T. H. Dunigan. Performance of the Intel iPSC/860 and Ncube 6400 hypercubes. *Parallel Computing*, 17:1285 – 1302, 1991.

[6] T. H. Dunigan. Performance of the Intel iPSC/860 and Ncube 6400 hypercubes. Technical report, Oak Ridge National Laboratory, 1991. ORNL/TM-11491.

[7] T. H. Dunigan. Communication performance of the Intel Touchstone DELTA mesh. Technical report, Oak Ridge National Laboratory, 1992. ORNL/TM-11983.

[8] G. A. Geist, M. T. Heath, B. W. Peyton, and P. H. Worley. A Users' Guide to PICL A Portable Instrumented Communication Library. Technical report, Oak Ridge National Laboratory, October 1990. ORNL/TM-11616.

[9] G. A. Geist and V. S. Sunderam. Network Based Concurrent Computing on the PVM System. Technical report, Oak Ridge National Laboratory, June 1991. ORNL/TM-11760.

[10] J. Gustafson, 1991. personal communication.

[11] John Gustafson, Diane Rover, Stephen Elbert, and Michael Carter. The design of a scalable, fixed-time computer benchmark. Technical report, Ames Laboratory, 1990.

[12] Intel. *Paragon XP/S Product Overview.* Intel, Beaverton, Oregon, 1991.

[13] E. Smirni, C. A. Childers, E. Rosti, and L. Dowdy. Thread placement on the intel paragon: Modeling and experimentation, 1994. submitted to ASPLOS-VI.

[14] S. Wheat, 1993. electronic communication.