ORNL/TM-12830

Computer Science and Mathematics Division

Mathematical Sciences Section

BETA TESTING THE INTEL PARAGON MP

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: June 1995

Research was supported by the Office of Scientific Computing 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

1 Introduction 1
2 Paragon MP Architecture 1
3 Performance 3
4 Message passing 6
5 Summary 9
6 References 9
A Timeline 11
B Comparative Architectures. 11

BETA TESTING THE INTEL PARAGON MP

Thomas H. Dunigan

Abstract

This report summarizes the third phase of a Cooperative Research and Development Agreement between Oak Ridge National Laboratory and Intel in evaluating a 28-node Intel Paragon MP system. An MP node consists of three 50-MHz i860XP's sharing a common bus to memory and to the mesh communications interface. The performance of the shared-memory MP node is measured and compared with other shared-memory multiprocessors. Bus contention is measured between processors and with message passing. Recent improvements in message passing and I/O are also reported.

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. Intel has developed a family of distributed-memory multiprocessors, starting with the iPSC/1 hypercube in 1986. The Intel multiprocessors are members of a growing market of parallel processing systems that are being used by researchers and commercial organizations to tackle increasingly complex computational tasks. A Cooperative Research and Development Agreement (CRADA) between ORNL and 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 the results of the third and final phase of the CRADA, the test and evaluation of the Paragon MP.

In 1994, the first Paragon MP was delivered to ORNL. The Paragon MP extended the computational power of the Paragon by providing three i860XP processors with a shared memory on each node of the communication mesh. Appendix A provides a time-line of the events during the test and evaluation of the Paragon MP system. This report details the performance of the shared-memory node board and evaluates the performance of several parallel applications on the Paragon MP. Computational, shared-memory, and communication performance were measured with synthetic benchmarks, application kernels, and a few parallel applications. The Paragon's shared-memory performance is compared with the performance of other shared-memory parallel processors.

In the following section, the Paragon MP architecture is summarized. Section 3 describes the performance of the shared-memory MP node and compares its performance to other shared-memory architectures. In section 4, recent improvements in message passing and I/O are reported.

2. Paragon MP Architecture

The Intel Paragon system is a mesh-connected parallel processor. In the first member of the Paragon family, the GP system, each node on the mesh consists of two 50 MHz i860XP processors, memory, and communication hardware. One processor is used for computation, and the second processor is for communication.

A Paragon MP node consists of three 50 MHz i860XP processors, memory, and communication hardware (Figure 2.1). The nodes are interconnected by a 2-D mesh with 175 MB/second communication channels and a per-hop latency of only 40 ns. The nodes are logically subdivided into service nodes, compute nodes, and I/O nodes (Figure 2.1). The service nodes appear as a single host and support time-sharing through the OSF operating system. The compute nodes run OSF or SUNMOS. 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).

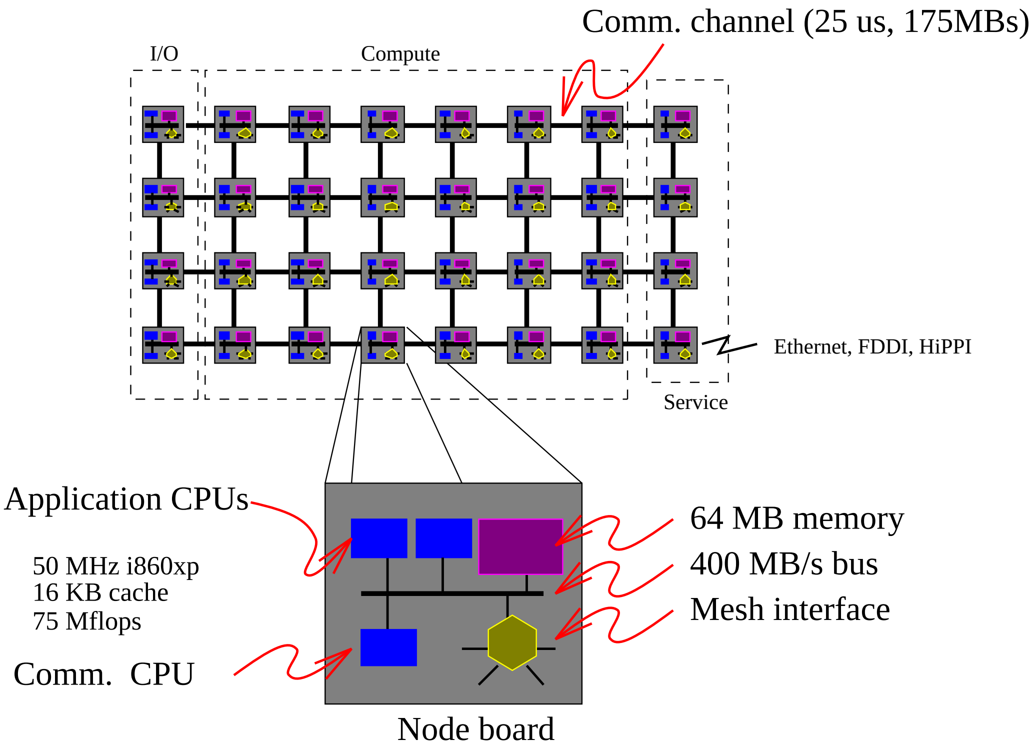

Figure 2.1: *MP node board.*

Each i860XP has its own 16 KB data and instruction cache, and each node has at least 64 MB of memory. The bus interconnecting the processors, mesh-interface, and memory operates at 400 MB/second. The 50 MHz i860XP is a super-scalar architecture capable of a peak 75 Mflops (double precision). Typical FORTRAN performance is only 11 Mflops ([2]). Early designs of the MP proposed five CPUs with L2 cache, but cost-performance analyses dictated the three-CPU configuration and no secondary cache. Intel's analyses showed the bus bandwidth to the local memory would not support five CPUs efficiently. Also the three-CPU configuration provided more board real-estate for memory than the five-

CPU design and the five-CPU design would have required using only every other backplane slot.

Message-passing libraries (NX, PVM, MPI, SUNMOS) are provided for inter-node communication. A node is the smallest addressable unit in the message-passing architecture. (Conceptually, each processor might be addressable in the message-passing software, but Intel's early design analyses favored node addressing.) Typically, one processor on each node is designated as the communication processor, leaving two processors for computational work. The compilers can provide automatic parallelization of the processors on a node, or a threaded library and compiler directives are provided for explicit parallelization.

3. Performance

In this section, we look at the performance of the shared-memory node. We measure CPU memory bandwidth and the effects of bus contention when multiple CPUs and message passing compete for the limited bus bandwidth. We compare the shared-memory performance and multi-threading primitives to other shared-memory multiprocessors, finally comparing performance of some parallel benchmark kernels.

An MP node has three CPUs, memory, and mesh-controller sharing a 400 MB/second bus (Figure 2.1). A 50 MHz i860XP is specified as being able to generate 400 MB/second of memory traffic. Clearly, the MP node architecture is likely to be bus limited. Large caches on each processor could mitigate the limited bandwidth, but the data cache is not large (16KB). With 90% to 95% cache hit rates and typical cache write-back rates, one can expect that only 15% to 25% of a CPU's memory requests actually generate a memory operation. In principle, the 400 MB/second bus could support four or five i860XPs. Programs can be contrived to demonstrate either extreme: where all data requests are satisfied from cache, and linear speed-up is possible; or where little or no cache hits occur, and the node runs at the speed of one (or less) processor.

To measure actual memory bandwidth performance, we used a small unrolled assembler loop that did quad load's (*pfldq*) from memory. The memory locations were "pre-touched" to eliminate any virtual memory effects. A single CPU sustained a memory access rate of 251 MB/second, considerably less than the 400 MB/second specification. If we ran our test concurrently on two CPUs, the aggregate rate was still only 237 MB/second. If all three CPUs on the node concurrently accessed memory, the aggregate rate was 246 MB/second (Table 3.1). Table 3.1 also shows the data rates for a C double-precision inner-product on one, two, and three CPUs. The vector lengths are too long to be contained in the 16 KB cache, and speedups are sublinear due to bus contention. (The C inner-product for cacheable vectors gave linear speedups with a data rate of 88 MB/second per CPU.)

MP memory bandwidth (MBs)				
CPUs	1	2	3	Sum
pfldq-1	251			251
pfldq-2	118	119		237
pfldq-3	80	82	84	246
ddot-1	39			39
ddot-2	25	25		50
ddot-3	21	21	21	63

Table 3.1: *Memory bandwidth consumption.*

We compared the MP shared-memory node board with other shared-memory multiprocessors. We compared thread and fork creation, lock and unlock, barriers, and concurrent update of shared variable (no locks). The i860XP has no hardware "atomic" operations, so locks are implemented by software. Table 3.2 compares single processor performance of the 50 MHz i860XP with single processors on the KSR, BBN, and Sequent. The KSR is ring-based shared-memory multiprocessor using a 20 MHz custom processor. The Sequent Symmetry is bus-based shared-memory multiprocessor using 16 MHz 386 processors. The BBN TC2000 is a cascaded-switch based shared-memory multiprocessor using 20 MHz M88000 processors (see Appendix B). The performance of the i860XP and the Paragon

Time on one CPU (μs)				
	MP	KSR	BBN	Sequent
fork/wait	50,000	108,000	44,000	14,000
thread/join	1,191	130	79	26
lock/unlock	13	3	8	10
barrier	13	39	11	10
hotspot	0.12	0.41	1.3	1.4

Table 3.2: *Single processor performance of shared memory.*

thread library is comparable to the other multiprocessors. The thread/join times are slower, but the Intel programming model is such that threads are usually only created once at the start of the application.

Table 3.3 compares the performance of three processors for an MP node board with three processors for the KSR, BBN, and Sequent. The MP node compares reasonably to the three-processor performance of the other multiprocessors. There appear to be no architecture or implementation penalties in the MP thread primitives for one, two, or three processors.

Time on three CPUs (μs)				
	MP	KSR	BBN	Sequent
lock/unlock	104	17	20	38
barrier	110	119	38	24
hotspot	0.81	2.5	5.3	6.4

Table 3.3: *Three processor performance.*

Table 3.4 compares the MP node performance over a set of application kernels in C and FORTRAN. The same copy of the code was run on each multiprocessor, and the codes have not been tuned. The numeric integration kernels effectively operate from cache, so near linear speed-up is achieved. The Cholesky code is a little more memory intensive, and the slower MP performance results from lock and bus contention. The HiTC kernel is based on a double-precision complex ZAXPY. Using Intel's ZAXPY from the *kmath* library, the serial code runs at 36 Mflops. The vectors in the ZAXPY exceed the i860XP cache size, and bus contention prevents the parallel HiTC kernel from achieving any speedup on an MP node. Another version of the HiTC kernel, modeling only one atom per cell, has small enough vectors that near linear speedups can be attained.

Speedup on three CPUs			
	MP	KSR	Sequent
Integration (C)	2.9	2.9	3.0
Jacobi iteration (C)	2.7	2.8	3.0
Cholesky (1K $\times$ 1k) (C)	2.1	3.0	2.9
Integration (F)	2.9	2.9	3.0
HiTC kernel (F)	1.0	2.9	2.9

Table 3.4: *Speedup on three CPUs for various application kernels.*

To this point, we have considered only a single node board. In parallel applications, each Paragon node will communicate with other nodes in the mesh in solving a parallel application. The expected configuration is to use one CPU on each node board as a communication processor. To see the effect of communication and computation competing for the bus, we added a communication thread to our *pfldq* test. In the absence of computational activity, the communication thread ran at 119 MB/second, using an echo test to an adjacent node. In the absence of communication, the *pfldq* ran at 252 MB/second. With one *pfldq* thread and one communication thread running concurrently for identical durations, the aggregate data rate was 177 MB/second. The communication thread garnered 31 MB/second, and the *pfldq* thread achieved about 146 MB/second. So the limited bus speed can slow both computation and communication.

Table 3.5 summarizes speedups of the FORTRAN NAS parallel benchmarks on a Pargon MP (using two compute processors and one communication processor) as reported by Intel in the Spring of 1995 (OSF R1.3). Speedups are relative to a Paragon GP (one compute processor and one communication processor). The class B versions represent larger problems (larger arrays or more iterations). The NAS results are consistent with the early results of ORNL "grand challenge" applications on the Paragon MP. The material science parallel application realized a speed-up of 1.7 on the MP versus the GP. However, the shallow-water kernel showed little speedup on the MP, but that kernel is characterized by low data re-use.

Program	Speedup
EP class A	1.74-1.91
EP class B	1.94-2.00
FT class A	1.20-1.42
FT class B	1.24-1.42
MG class A	1.21-1.37
MG class B	1.32-1.39

Table 3.5: *Intel reported speedups of Pargon MP verus GP for FORTRAN NAS Parallel Benchmarks.*

4. Message passing

Our beta testing concentrated primarily on the shared-memory features of the MP, but we also re-evaluated message-passing performance and I/O. Most of our production research is conducted using Intel's OSF on the compute nodes, but we also continue to evaluate SUNMOS. Our communication tests uncovered several performance anomalies. Data rates were poor if message sizes were not a multiple of 32 bytes, and data rates of one-to-n communication degraded as n increased. Intel corrected the anomalies in subsequent software releases. Message-passing performance (latency and bandwidth) improved with each release of software. For nearest neighbor communication, we are currently measuring latencies of 25 to 30 μs for zero-length messages, and data rates of nearly 171 MB/second for one MB messages. These numbers were measured under OSF 1.0.4 R1_3 and

SUNMOS 1.6.2 and are much faster than those we reported just last year ([4]). Per-hop delay is nearly negligible. The additional delay for going corner to corner on the 1024-node MP Paragon (16×64) is less than 3 μs.

The compute nodes can be configured in a "turbo" mode, where all three processors are used as computation processors. Communication tasks are handled with context switches. Our early tests showed that communication performance was several orders of magnitude slower in turbo mode. However, recent software releases have greatly improved turbo mode communication. Latency slows to 75 μs, and bandwidth is reduced to 109 MB/second. Figure 4.1 compares the message-passing performance (transfer time) for OSF, SUNMOS, and OSF in turbo mode. Transfer times are half the round-trip time for 1,000 repetitions of an echo test to a neighboring node.

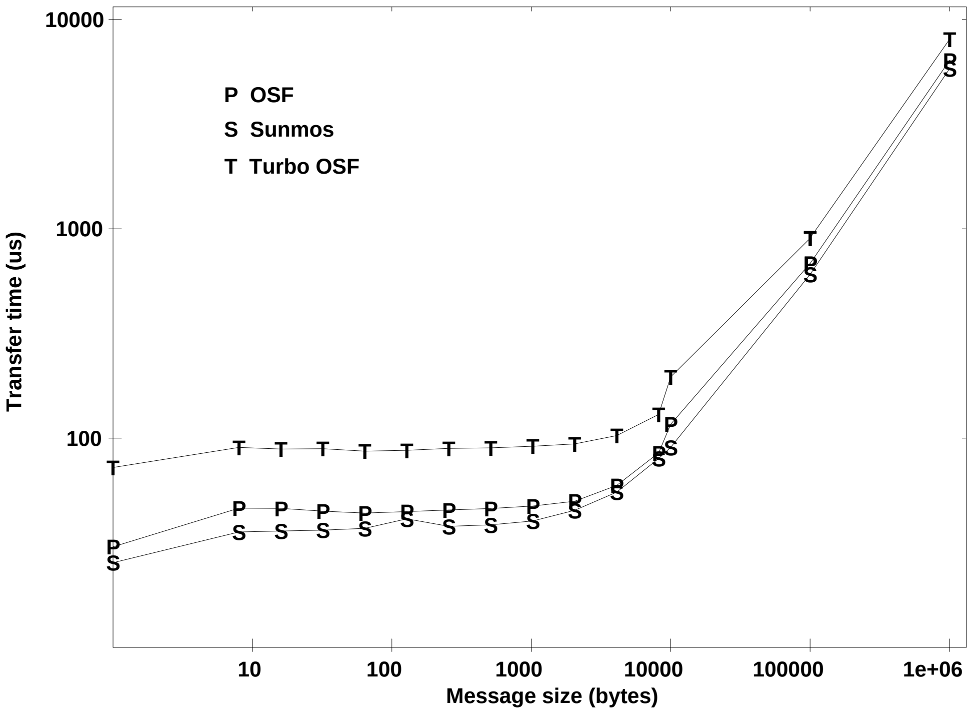

Figure 4.1: *Message transfer time for the Paragon MP.*

Our CRADA analysis also re-evaluated I/O performance. The Paragon OSF provides both a standard UNIX file system and a larger, high performance parallel file system (PFS). PFS is typically configured across a set of I/O nodes and disks. The PFS is striped across one or more I/O nodes and their disk RAID arrays and appears to the UNIX system as a separate mountable file system (e.g., */pfs*). The striping factor is 64 KB. PFS performance improves with additional I/O nodes and larger block sizes and is limited on each node by the throughput of the OSF

NORMA IPC communication. IPC is used by OSF to communicate between the compute nodes and I/O nodes. The IPC communication is in turn transported by the Paragon message passing hardware and software. Using 2 MB records to 64 I/O nodes, a single MP node achieves an 18 MB/second read rate and a 36 MB/second write rate (Figure 4.2). Figure 4.2 also shows that the NORMA IPC data rate between adjacent MP nodes peaks at about 45 MB/second, and the read and write I/O data rates follow the same basic curve as the IPC performance. The IPC performance probably limits I/O performance, since the IPC data rate is well below the 171 MB/second data rate available from the underlying mesh.

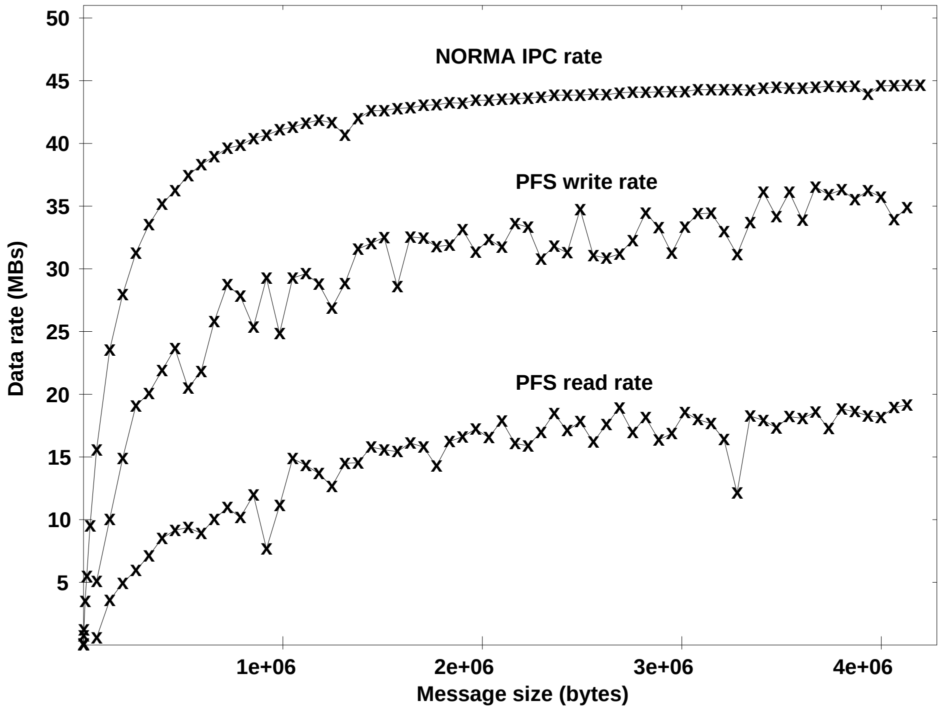

Figure 4.2: *Paragon MP NORMA IPC rate and PFS read/write data rates.*

Figure 4.3 shows aggregate read data rate using 16, 32, and 64 I/O nodes, with from 1 to 128 compute nodes doing concurrent I/O. Aggregate read data rates of 95 MB/second are achieved with 64 I/O nodes and 128 compute nodes, an improvement over earlier results ([4]). The PFS tests use 64 KB blocks, and each compute processor reads a 32 MB section of a file. Our PFS tests use the M_RECORD mode of *gopen()*, and open and close times are included in the timings.

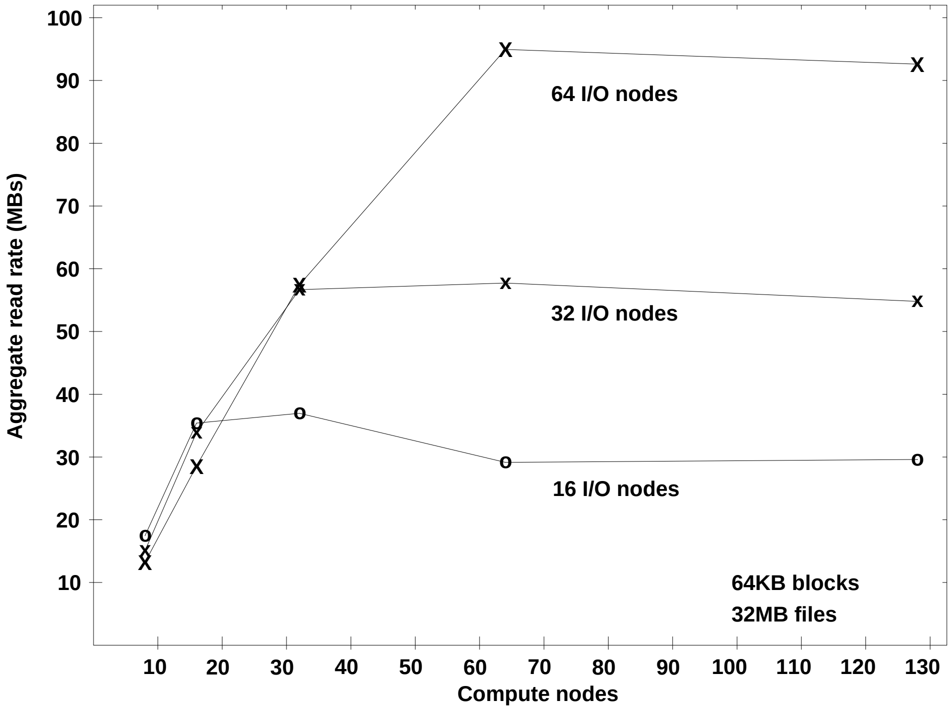

Figure 4.3: *Aggregate read data rate for varying compute and I/O nodes.*

5. Summary

The final phase of the ORNL/Intel CRADA provided value to both parties, Intel getting feedback from early users and performance analyses, and ORNL getting an opportunity to do leading-edge computer science and computational science. The shared-memory performance of the Paragon MP is competitive with other shared-memory multiprocessors. The limited bus bandwidth of each MP node requires that the application programmer exploit data locality to garner noticeable speedups. The automatic parallelization compilers help in utilizing the multiple processor nodes, but in complex programs, the application programmer usually needs to assist in the parallelization process. Message-passing performance and I/O continue to improve. The 96-node MP Paragon CRADA machine continues to be a valuable computational resource for ORNL, even after delivery of the production 1024-node MP Paragon in January, 1995.

6. References

[1] J. Dongarra. Performance of various computers using standard linear equations software. Technical report, University of Tennessee, January 1991. CS-

89-85.

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

[3] T. H. Dunigan. Kendall Square multiprocessor: Early experiences and peformance. Technical report, Oak Ridge National Laboratory, 1992. ORNL/TM-12065.

[4] T. H. Dunigan. Early experiences and performance of the intel paragon. Technical report, Oak Ridge National Laboratory, 1993. ORNL/TM-12194.

[5] R. P. LaRowe and C. S. Ellis. Experimental comparison of memory management policies for numa multiprocessors. Technical report, Duke University, April 1990. CS-1990-10.

[6] R. D. Rettberg, W. R. Crowther, P. P. Carvey, and R. S. Tomlinson. The monarch parallel processor hardware design. *Computer*, 23:18–30, April 1990.

Appendix

A. Timeline

May, 1993. Inititial "planned" delivery of beta MP system to ORNL.

July, 1993. Intel changes from MP5 to MP3.

October, 1993. Fat-node programming model selected.

April, 1994. Training on MP at Beaverton.

May, 1994. 10-node MP delivered to ORNL.

June, 1994. MP training at ORNL and parallel FORTRAN delivered.

July, 1994. MP expanded to 28 nodes and additional training.

October, 1994. Applications testing on 1024-node MP system (XPS150) at Beaverton.

January, 1995. MP expanded to 96 nodes and XPS150 delivered.

B. Comparative Architectures.

A node on the Paragon MP supports a shared-memory architecture with three i860XP processors sharing a 400 MB/second bus and memory. The report compares shared memory performance of an MP node with several shared-memory multiprocessors. A summary of the shared-memory multiprocessors compared with the MP node follows.

BBN TC2000

The BBN TC2000 at Argonne National Laboratory (ANL) is a 45 processor shared-memory parallel processor. Each processor is a Motorola 88000 running at 20MHz with 16 MB of memory fronted by a 16KB data cache and a 16KB instruction cache. All of the memories are interconnected by a 2-stage 8-way switch. The system can be expanded up to 512 processors. The Uniform programming environment (under nX 2.0.6) provides the program with both local and explicitly allocated shared memory. The shared memory may be allocated in another processor's memory, and thus a non-uniform memory access (NUMA) model is supported. In the absence of contention, a remote reference typically takes less than two microseconds, and a single channel of the switch has a bandwidth of 40 MBs [6]. The architecture could be used with other memory management policies [5]. Compiles on the BBN were done with -O -lus. LINPACK 100×100 double-precision on a single processor was 1.0 Mflops using -OLM -autoinline. Dhrystone (v1.0) was 19.4 Mips.

Kendall Square

The Kendall Square uses custom-designed 20 MHz processors that share memory on a one gigabyte per second ring. Each processor has a 256KB cache, and the global memory is managed as a cache. A single processor generates a maximum of 40 MBs against the ring. LINPACK 100×100 double-precision on a single processor was 15 Mflops [3].

Sequent Symmetry

The 26 processor Sequent Symmetry located at ANL is based on 80386/387 processors (16 MHz) with a Weitek 3167 floating point co-processor. Each processor has a 64KB cache, and 32 MB of memory is shared by all processors on a 54 MBs bus. The maximum configuration is 30 processors. The processors run Dynix 3.1.2, and compiles were done using -O. LINPACK 100×100 double-precision on a single processor was 0.37 Mflops [1]. Dhrystone (v1.0) was 3.6 Mips. Processor 4.8 MBs versus a 26 MBs bus.