

PROCESSOR PARTITIONING: AN EXPERIMENTAL PERFORMANCE ANALYSIS OF PARALLEL APPLICATIONS ON SMP CLUSTER SYSTEMS

Xingfu Wu and Valerie Taylor

Department of Computer Science, Texas A & M University, College Station, TX 77843, USA

Email: {wuxf, taylor}@cs.tamu.edu

ABSTRACT

Currently, clusters of shared memory symmetric multiprocessors (SMPs) are one of the most common parallel computing systems, for which some existing environments have between 8 to 32 processors per node. Examples of such environments include some supercomputers: DataStar p655 (P655 and P655m) and P690 at the San Diego Supercomputing Center, and Seaborg and Bassi at the DOE National Energy Research Scientific Computing Center. In this paper, we quantify the performance gap resulting from using different number of processors per node for application execution (for which we use the term **processor partitioning**), and conduct detailed performance experiments to identify the major application characteristics that affect processor partitioning. We use the STREAM memory benchmarks and Intel's MPI benchmarks to explore the performance impact of different application characteristics. The results are then utilized to explain the performance results of processor partitioning using three NAS Parallel Application benchmarks. The experimental results indicate that processor partitioning can have a significant impact on performance of a parallel scientific application as determined by its communication and memory requirements.

Keywords

Performance analysis, processor partitioning, parallel applications, and MPI benchmarks

1. Introduction

Today, cluster systems with hierarchical SMPs, composed of multicore chips, are common parallel computing systems used by scientific applications. Examples of such systems include the DataStar p655 (P655 and P655m) and P690 at the San Diego Supercomputing Center (SDSC) [10], and Seaborg and Bassi at DOE National Energy Research Scientific Computing Center (NERSC) [7]. These systems have between 8 and 32 processors per node. When using these SMP clusters to execute a given application, one issue to be addressed is how many processors per node to use for efficient execution. It is expected that the best number is dependent upon the application characteristics and the system configuration. In this paper, we quantify the performance gap resulting from using different number of processors per node for application execution (for which we use the term **processor partitioning**), and conduct detailed

performance experiments to identify the major application characteristics that affect processor partitioning.

Other work in this area has focused on using all processors per node. Phillips et al. [9] presented the performance results for 4 processors per node and 3 processors per node on Lemieux Alpha cluster at Pittsburgh Supercomputer Center that has a maximum of 4 processors per node, and noted that leaving idle one processor per node reduces performance variability. Petrini et al. [8] found that application execution times may vary significantly between 3 processors per node and 4 processors per node on a large scale supercomputer, ASCI Q; they concluded that system noise within the nodes was the source of the performance variability, and used a discrete-event simulator to evaluate the contribution of each component of the noise to the overall application behavior. In contrast to these approaches, we conduct an experimental performance analysis to identify the application characteristics that affect processor partitioning and to quantify the performance difference among different processor partitioning schemes.

The experiments conducted to explore processor partitioning utilize systems with different number of processors per node. The P655, P655m and Bassi have 8 processors per node, Seaborg has 16 processors per node, and P690 has 32 processors per node. Further, each system has a different node memory hierarchy. In this paper, we use Intel's MPI benchmarks, IMB [3] and the STREAM memory benchmark [5] to provide initial analysis on the application characteristics that affect performance. These benchmarks indicated that the following characteristics affect processor partitioning:

- **Memory Accesses:** Applications with a large memory requirement can execute efficiently when large memory bandwidth per processor is available. The results using the STREAM indicate that memory access patterns at different memory hierarchy levels affect performance for different processor partitioning schemes. In some cases, it is best to use fewer processors per node because of large memory bandwidth requirements of applications.
- **Global Communication:** Global communication, such as that incurred with reduce, broadcast, barrier, allreduce, or alltoall operations, generally utilizes moderate to small message sizes. The main issue with such operations is the number of messages in the network and the amount of available internode (e.g. between nodes) bandwidth. The results using IMB indicate that it is best to use half of the maximum number of processors or fewer per node for efficient execution because the intra-

node (e.g. within a node) bandwidth (latency) is always larger (smaller) than that of the inter-node.

- **Message Size:** For the case of applications requiring large message sizes, the results using IMB depict that it is best to use half of the maximum number of processors or fewer per node. If a large number of processors per node is used, congestion can occur, resulting in a reduction in the effective bandwidth.

In addition to using the two benchmarks, we investigate processor partitioning using three NAS Parallel Application benchmarks, **BT**, **SP** and **LU** [1]. These application benchmarks are analyzed in terms of the aforementioned characteristics and results are given for processor partitioning schemes executed on the five supercomputer environments. The experimental results indicate that processor partitioning can have a significant impact on the execution time. For example, the execution time for BT with problem size of Class B on a total of 16 processors using 2 nodes with 8 processors per node is **two** times more than that using 16 nodes with 1 processor per node on P655 and P655m. Processor partitioning, however, has very little performance impact for the applications on systems with large page memory such as Bassi.

In the remainder of this paper, the processor partitioning scheme **NxM stands for N nodes with M processors per node (PPN)**. For instance, the processor partitioning scheme 8x4 indicates 8 nodes with 4 PPN for a total of 32 processors. When considering processor partitioning for an application on given number of processors, we assume that all system configurations use the same data partition pattern for the application execution. The job scheduler for each supercomputer always assigns one process to one processor. All experiments were executed multiple times to ensure consistency of the performance data.

The remainder of this paper is organized as follows. Section 2 describes the configurations of the five supercomputers. Section 3 uses the STREAM benchmark to investigate how processor partitioning is affected by the memory accesses of an application. Section 4 uses Intel’s IMB benchmarks to identify how communication overhead and message size impact processor partitioning. Section 5 discusses the lessons learned from the benchmarks STREAM and IMB. Section 6 uses the results from Section 5 to explain the performance results for the NAS Parallel Application Benchmarks. Section 7 concludes this paper.

2. Execution Environments

Five supercomputers used for our experiments are described as follows. These systems differ in the following main features: number of PPN, configurations of node memory hierarchy, CPU speed, and communication networks.

DataStar has 176 (8-way) compute nodes with 16 GB memory and eight 1.5GHz POWER4 CPUs per node, called P655, and 96 (8-way) compute nodes with 32 GB memory and eight 1.7GHz POWER4 CPUs per node, called P655m. DataStar P690 has 7 (32-way) compute nodes with 128GB memory and thirty-two 1.7GHz POWER4 CPUs per node.

Nodes are connected by IBM Federation switch. The use of 8-way nodes for P655 and P655m is exclusive, i.e., only one user is allowed to use a node at any given time, regardless of the number of CPUs one uses on that node. The use of 32-way nodes on P690 is shared among users. Further, access to the P690 is limited to five nodes.

Seaborg has 380 compute nodes with 16 POWER3 processors per node. Processors on each node have a shared memory pool of size of 16-64GB. Nodes are connected by IBM colony switch. The use of 16-way nodes is exclusive.

Bassi has 111 compute nodes with 32GB memory and 8 single-core POWER5 processors per node. Nodes are connected by IBM Federation switch. The use of 8-way nodes is also exclusive. Only Bassi is configured to use 20GB “large page” memory on each node [7]. The large page memory uses hardware prefetch mechanisms to eliminate costly TLB misses at the expense of an increase in process start-up time.

3. STREAM Benchmark: Memory Analysis

In this section, we use the MPI version of the STREAM memory benchmark [5] to investigate memory performance for different processor partitioning schemes. The STREAM benchmark is a synthetic benchmark program, written in standard Fortran 77 and MPI. It is specifically intended to eliminate the possibility of data re-use (either in registers or caches) by setting large enough array sizes. We use unit-stride TRIAD benchmark ($a(i)=b(i)+q*c(i)$) to measure the sustainable memory bandwidths. We find that only Bassi supports array sizes up to $128M (2^{27})$; the other systems, P655, P655m, P690 and Seaborg, support array sizes of at most $4M (2^{22})$ because of lack of sufficient memory to start the benchmark.

Table 1. Memory performance on Bassi (POWER5)

Processors Partition	1x8	2x4	4x2	8x1
Memory bandwidth (MB/s)	26843.55	80530.64	80530.64	80530.64
L1 hit rate	97.18%	97.16%	97.12%	97.11%
L2 bandwidth (MB/s)	201.07	241.74	237.60	233.21
% accesses from L2	2.816%	2.844%	2.885%	2.893%
TLB miss rate	0.0106%	0.0109%	0.0107%	0.0103%

Table 2. Memory performance on Seaborg(POWER3)

Processors Partition	1x16	2x8	4x4	8x2	16x1
Memory bandwidth (MB/s)	2147.48	2597.76	2928.39	3038.89	3097.33
L1 hit rate	99.41%	99.42%	99.44%	99.46%	99.51%
L2 bandwidth (MB/s)	4.32	5.11	5.61	5.69	5.90
%accesses from L2	0.254%	0.252%	0.245%	0.232%	0.210%
TLB miss rate	0.023%	0.022%	0.021%	0.021%	0.019%

Table 3. Memory performance on P655 (POWER4)

Processors Partition	1x8	2x4	4x2	8x1
Memory bandwidth (MB/s)	16106.13	16106.13	20132.66	40265.32
L1 hit rate	84.36%	84.05%	79.93%	78.94%
L2 bandwidth (MB/s)	7535.11	7211.68	9704.74	11418.79
% accesses from L2	4.905%	4.637%	6.295%	7.621%
TLB miss rate	0.020%	0.023%	0.028%	0.039%

Tables 1-3 present the following performance data for POWER3 (Seaborg), POWER4 (P655) and POWER5 (Bassi) systems using unit-stride TRIAD with the array size of $4M$; the rows represent sustainable memory bandwidth, L1 hit rate, L2 bandwidth per processor, percentage accesses from L2 per cycle, and TLB miss rate collected by hpmcount [2]. For all these systems, processor partitioning impacts the sustainable memory bandwidth. For example, in Table 1 for Bassi, the sustainable memory bandwidth increases from 26843.55MB/s using all 8 PPN to 80530.64MB/s using 4 or fewer PPN. For Seaborg, the sustainable memory bandwidth increases from 2147.48MB/s using all 16 PPN to 3097.33MB/s using 1 PPN; similarly, L2 bandwidth increases from 4.32MB/s to 5.90MB/s shown in Table 2. For P655, the sustainable memory bandwidth increases from 16106.13MB/s using all 8 PPN to 40265.32MB/s using 1 PPN; similarly, the L2 bandwidth increases from 7211.68MB/s to 11418.79MB/s as shown in Table 3. Similar results occurred with P655m and P690; the tables are not included due to space limitation. For P655m, the sustainable memory bandwidth increases from 13421.77MB/s using all 8 PPN to 40265.32MB/s using 1PPN; similarly, the L2 bandwidth increases from 6333.46MB/s to 12009.26MB/s. For P690, the sustainable memory bandwidth increases from 32212.25MB/s using all 32 PPN to 64424.51MB/s using 8PPN; similarly, the L2 bandwidth increases from 3399.62MB/s to 6993.12MB/s. Hence, using the fewer PPN results in better sustainable memory bandwidth.

Although processor partitioning significantly impacts the sustainable memory bandwidth, it has very little impact on L1 hit rate and TLB miss rate on Bassi and Seaborg as shown in Tables 1 and 2 because of equal workload per processor. In particular, the L1 hit rate remains above 97% on Bassi, and above 99.4% on Seaborg. For the supercomputers with POWER4 (such as P655, P655m, and P690), processor partitioning significantly affects the L1 hit rate and TLB miss rate as shown in Table 3. With decreasing the PPN, the L1 hit rate decreases from 84.36% to 78.94% on P655, from 87.84% to 79.91% on P655m, and from 89.01% to 84.70% on P690. It is also interesting that, with decreasing the PPN, the percentage accesses from L2 per cycle increases on P655 and Bassi, but it decreases on Seaborg. This is because of different cache hierarchies on the five supercomputers. Bassi has the most advanced cache hierarchy with 20GB large page memory.

4. IMB Benchmarks: MPI Performance Analysis

In this section, we use Intel’s IMB benchmarks to investigate the impact resulting from global communication and message size on different system configurations. The IMB benchmarks include the following classes of communication: single transfer, parallel transfer, and collective. Single transfer benchmarks such as *PingPong*, focus on a single message transferred between two processes. Parallel transfer benchmarks, such as *Sendrecv* and *Exchange*, measure message passing efficiency under global load. Collective benchmarks are collective in proper MPI convention, such as *MPI_Reduce*, *MPI_Bcast*, *MPI_Barrier*, and so on. In the following subsections, we focus on the performance of parallel transfer and collective benchmarks since these benchmarks are indicative of the communication mechanisms used in scientific applications. Each benchmark is executed with varying message lengths from 0 to 4MB, and timings are averaged over multiple samples (1000).

4.1 Single Transfer: PingPong

Table 4 presents the minimum latency and the maximum bandwidth with the message size from 1 byte to 4MB using PingPong. The intra-node bandwidths for P655, P655m and P690 reach the maximum with the message size of 128KB, Seaborg gets the maximum bandwidth with the message size of 2MB, and Bassi gets the maximum bandwidth with the message size of 4MB. For inter-node bandwidths, when the message size is 2MB or 4MB, these five supercomputers get the maximum bandwidths. It is noted that the Federation switch used in P655, P655m, P690 and Bassi has lower latency and higher bandwidth than the Colony switch used in Seaborg. Compared to the sustainable memory bandwidths shown in Tables 1-3, the intra-node MPI bandwidths are much smaller.

Table 4. Uni-directional Latency and bandwidth

Platform	Communication Mode	MPI Latency (μ s)	MPI Bandwidth (MB/s)
P655	Intra-node (1x2)	1.68	2862.22
	Inter-node (2x1)	5.66	1390.60
P655m	Intra-node (1x2)	1.51	3380.19
	Inter-node (2x1)	4.76	1543.07
P690	Intra-node (1x2)	3.52	2052.89
	Inter-node (2x1)	6.74	1370.28
Seaborg	Intra-node (1x2)	9.93	485.39
	Inter-node (2x1)	26.22	224.24
Bassi	Intra-node (1x2)	1.28	6096.93
	Inter-node (2x1)	4.16	1701.34

4.2 Parallel Transfer: Sendrecv and Exchange

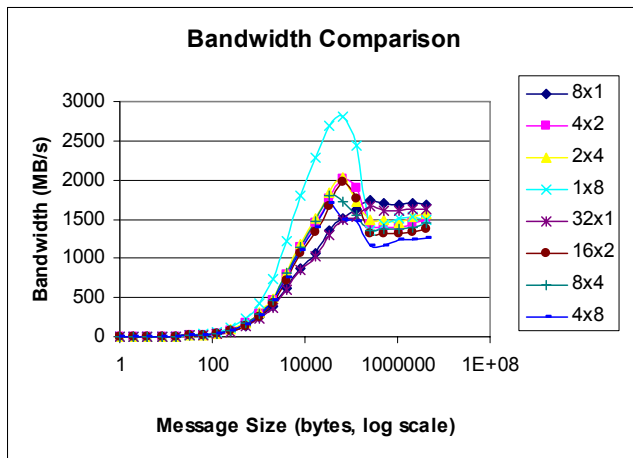
In this section, we use the performance results for the benchmark *Sendrecv* to analyze bi-directional latency and bandwidth, and use the performance results for the benchmark *Exchange* to discuss how processor partitioning affects the bandwidth available to applications.

Table 5. Bi-directional latency and bandwidth

Platform	Communication Mode	MPI Latency (μ s)	MPI Bandwidth (MB/s)
P655	Intra-node (1x2)	2.72	4118.52
	Inter-node (2x1)	6.71	1600.55
P655m	Intra-node (1x2)	2.7	4249.44
	Inter-node (2x1)	5.68	1796.73
P690	Intra-node (1x2)	4.91	2606.86
	Inter-node (2x1)	8.01	1504.12
Seaborg	Intra-node (1x2)	14.45	932.84
	Inter-node (2x1)	29.89	295.61
Bassi	Intra-node (1x2)	2.09	11885.67
	Inter-node (2x1)	5.55	2284.93

Table 5 reports the bi-directional latency and bandwidth on each supercomputer. For intra-node bandwidth, the bandwidths on P655, P655m and P690 reach the maximum with the message size of 128KB, and Seaborg gets the maximum bandwidth with the message size of 2MB. For inter-node bandwidths, P655 and P655m reach the maximum intra/inter-node bandwidths with the message size of 256KB, P690 has the maximum bandwidth with the message size of 128KB, and Seaborg has the maximum bandwidth with the message size of 1MB. Bassi has the maximum bandwidths with the message size of 4MB, and has the lowest latency and the highest bandwidth. The performance data collected on P690 may be affected by other users' applications because the nodes on P690 are shared among all users.

Recall that P655, P655m and Bassi have 8 PPN, Seaborg has 16 PPN, and P690 has 32 PPN. We investigate how processor partitioning impacts the effective bandwidth on 32 processors using the benchmark *Exchange*.

**Figure 1. Bandwidth comparison on P655m**

The performance results on P655m shown in Figure 1 indicate that, on 8 processors, the scheme 1x8 has the highest bandwidth for messages of 128 KB or less. However, the scheme 8x1 has the highest bandwidth for messages of 256 KB or more on 8 processors. On 32 processors, the scheme 8x4 has the highest bandwidth with messages of 16 KB or less, but the scheme 32x1 has the highest bandwidth for messages of 256 KB or more. It is noted that P655 and P690 have the similar performance

trends to that of P655m. We also find the similar performance trends on Seaborg and Bassi.

4.3 Collective Benchmarks

The collective benchmarks include *Reduce*, *Reduce_scatter*, *Allreduce*, *Allgather*, *Allgatherv*, *Alltoall*, *Bcast*, and *Barrier*. Our performance results show, in most cases, when the number of requested processors is equal to the maximum number of PPN, using all of the PPN has the lowest latency. For the case of the number of processors is bigger than the maximum number of PPN, it is the best to use half or all of the number of PPN. For the collective benchmarks, it was never the case that one processor per node resulted in the lowest latency.

5. Lessons Learned from STREAM and IMB

We used IMB and STREAM to provide initial analysis on the application characteristics that affect performance. The results using STREAM indicate that memory access patterns at different memory hierarchy levels affect performance for different numbers of PPN. The fewer PPN are used, the higher the sustainable memory bandwidth can be achieved. Using the maximum number of PPN never results in the highest memory bandwidth. Applications with a large memory requirement can execute efficiently when large memory bandwidth per processor is available.

The results using the IMB benchmarks indicate that processor partitioning significantly affects the communication bandwidth. In most cases, when the number of requested processors is equal to the maximum number of PPN, using all of the PPN had the lowest latency and highest bandwidth, with the exception of large messages; for the case of large messages using one or two PPN resulted in the best performance. When the number of requested processors is larger than the maximum number of PPN, half of the maximum number of PPN or fewer resulted in the lowest latency and highest bandwidth. Global communication (such as that incurred with reduce, broadcast, barrier, allreduce, or alltoall, etc.) generally utilizes moderate to small message sizes. The main issue with such operations is the number of messages in the network and the amount of available internode bandwidth. The results indicate that it is best to use half of the maximum number of processors or fewer per node for efficient execution because the intra-node bandwidth (latency) is much larger (smaller) than that of the inter-node.

6. Performance Analysis of NAS Parallel Application Benchmarks

In this section, we use three NAS Parallel Application Benchmarks (version 2.3) [1]: LU, BT, and SP with problem sizes of Class B and C running on 16 processors to investigate the impact of processor partitioning. LU is a simulated CFD application which uses symmetric successive over-relaxation to solve a block lower triangular

and upper triangular system of equations resulting from an un-factored implicit finite-difference discretization of the Navier-Stokes equations in three dimensions. BT and SP are simulated CFD applications that solve systems of equations resulting from an approximately factored implicit finite-difference discretization of the Navier-Stokes equations. The BT solves block-tridiagonal system of 5x5 blocks; the SP solves scalar pentadiagonal systems resulting from full diagonalization of the approximately factored

scheme. The LU benchmark requires a power-of-two number of processors. BT and SP require a square number of processors. The problem sizes all are 102x102x102 for Class B and 162x162x162 for Class C.

Based on the initial analysis using the benchmarks STREAM and IMB, we can hypothesize that processor partitioning will significantly affect the performance of BT and SP, and it will have very little impact on the performance of LU because of very small message sizes.

Table 6. Runtime (seconds) for Problem sizes Class B and C on 16 processors

System Name	Processors Partition	LU		BT		SP	
		Class B	Class C	Class B	Class C	Class B	Class C
P655	16x1	32.15	149.05	53.64	310.83	41.42	213.88
	8x2	32.24	152.75	64.41	365.59	42.39	232.30
	4x4	39.67	186.08	90.72	417.23	56.61	279.46
	2x8	41.65	230.18	121.15	507.25	70.22	345.35
P655m	16x1	32.02	131.67	53.6	286.01	41.41	189.30
	8x2	32.26	134.90	62.04	338.91	42.51	216.98
	4x4	33.97	168.85	84.32	394.63	50.44	251.87
	2x8	37.24	211.65	115.78	489.16	64.63	321.29
Bassi	16x1	23.44	93.62	32.67	167.94	30.26	140.83
	8x2	23.46	93.67	32.53	167.67	30.07	140.81
	4x4	23.36	93.76	32.55	167.81	30.05	140.75
	2x8	23.37	93.51	32.67	168.40	29.9	149.33
Seaborg	16x1	110.53	496.45	306.48	1237.64	164.53	836.41
	8x2	110.59	494.78	309.7	1254.63	166.46	843.14
	4x4	110.33	499.80	314.99	1281.76	168.01	869.59
	2x8	111.8	512.92	341.13	1367.72	174.24	900.38
	1x16	113.95	540.34	415.48	1684.40	214.74	1065.29
P690	4x4	34.06	143.87	71.91	399.83	40.69	266.47
	2x8	34.93	201.78	86.39	469.27	42.15	265.74
	1x16	34.53	208.91	101.49	537.97	49.08	325.04

Table 7. Communication rate, L1 hit rate, L2 bandwidth, and TLB miss rate for BT with Class B

System	Metrics	16x1	8x2	4x4	2x8	1x16
P655	Communication Rate	5.95%	5.32%	6.95%	5.11%	NA
	L1 hit rate	85.79%	85.69%	85.93%	85.87%	
	L2 bandwidth (MB/s)	11824.88	10212.10	7789.92	5820.77	
	% accesses from L2	6.26%	5.63%	4.29%	2.84%	
	TLB miss rate	0.035%	0.034%	0.029%	0.020%	
Bassi	Communication Rate	5.26%	5.04%	5.39%	5.39%	NA
	L1 hit rate	96.84%	96.94%	97.00%	97.03%	
	L2 bandwidth (MB/s)	44.56	39.81	37.56	36.46	
	% accesses from L2	3.161%	3.061%	3.017%	2.973%	
	TLB miss rate	0.012%	0.012%	0.012%	0.010%	
Seaborg	Communication Rate	6.08%	5.76%	5.65%	6.20%	5.47%
	L1 hit rate	99.49%	99.45%	99.42%	99.40%	99.39%
	L2 bandwidth (MB/s)	0.58	0.56	0.55	0.51	0.42
	% accesses from L2	0.213%	0.236%	0.249%	0.255%	0.248%
	TLB miss rate	0.020%	0.021%	0.022%	0.031%	0.025%

Table 6 provides runtimes for LU, BT and SP with different number of PPN on a total of 16 processors. For LU, there is very little difference in execution times between using half of the maximum number of PPN down

to using one processor per node. LU has very small communication overhead and small message sizes (e.g., 40 bytes). For BT and SP benchmarks, however, there are significant differences in execution times for the different

PPN. For example, for BT with Class B executed on P655 and P655m, there is over 116% time difference between the execution times on the 16x1 scheme and the 2x8 scheme. For BT using fewer numbers of PPN has the best performance for P655, P655m, Seaborg, and P690. It is noted that on Bassi, there was very little difference in the execution time across all the partitioning schemes with BT, which we attribute to the large intra-node bandwidth available on Bassi. For the SP benchmark with Class B, there is 70% difference between the execution time for the scheme 16x1 versus 2x8 on P655; there is a 56% difference between the two schemes on P655m. Similarly, for SP executed on Bassi, there is very little difference in the execution times across all the partitioning schemes. Hence, for BT and SP, the memory requirements determine the best partitioning scheme. Note that P690 only allows up to five nodes to be used.

For the sake of space, we provide the results of BT with Class B executed on 16 processors on POWER4 (P655), POWER5 (Bassi) and POWER3 (Seaborg) systems. Table 7 indicates the MPI communication rate, L1 hit rate, L2 bandwidth, the percentage accesses from L2, and TLB miss rate, where the communication rate is average using IPM [4] and MPI Trace Library [6]. When increasing the PPN, the L1 hit rate changes a little bit across all systems because of equal workload per processor; it is very interesting that the MPI communication rate does not increase much although the global communication time increases with decreasing the PPN. We find that the subroutines MPI_Wait (which waits for a MPI send or receive to complete) and MPI_Waitall (which waits for all given communications to complete) take more than 90% of the total communication time across all platforms. For P655, the L2 bandwidth, the percentage accesses from L2, and TLB miss rate decrease significantly with decreasing the PPN. For example, the L2 bandwidth for the 16x1 scheme is over twice as large as that for the 2x8 scheme. Although the 16x1 scheme has the largest percentage accesses from L2, 6.26%, each processor uses the “dedicated” L2 with the bandwidth of 11824.88 MB/s. This is one main reason that the execution time for the 16x1 scheme has less half of that for the 2x8 scheme. For Bassi, there is very small difference in TLB miss rate and percentage accesses from L2 due to the 20GB large page memory, which uses hardware prefetch mechanisms to eliminate costly TLB misses. Hence, the execution time is flat across the different PPN. For Seaborg, with increasing the number of PPN, the L2 bandwidth decreases from 0.58MB/s to 0.42MB/s. But TLB miss rate and percentage accesses from L2 increase a little bit because of the decrease of L1 hit rate. P655 has the lowest L1 hit rate and highest percentage accesses from L2.

In summary, processor partitioning significantly affects the performance of BT and SP on P655, P655m, P690 and Seaborg; the better performance results from fewer PPN. Processor partitioning, however, has very little performance impact for applications with small message sizes such as LU or on systems with large page memory such as Bassi. For these cases, it is better to use as many PPN as possible.

Conclusions

This paper investigated experimentally the performance impact of processor partitioning using the five supercomputers: P655, P655m, P690, Seaborg and Bassi. First, we used STREAM and IMB to explore the impact of memory accesses, global communication requirements, and message sizes on the performance for different PPN. We then applied this knowledge to understand the results from NAS Parallel Application benchmarks. The results indicated that there can be a significant difference in execution time for different PPN. For example, the execution time for BT with problem size of Class B on 16 processors with the scheme 2x8 is over 116% more than that with the scheme 16x1 on P655 and P655m. However, processor partitioning has very little performance impact for applications on systems with large page memory such as Bassi.

Currently, most researchers in high performance computing present their performance results using all PPN as default. Our work, however, indicates that using all PPN resulted in the worst performance. Hence, based on the processor partitioning analysis, our further work will focus on exploring and optimizing the applications for efficiently using all PPN, especially BT and SP by increasing the utilization of the memory hierarchy on P655 where the L1 hit rate is less than 86% for BT as shown in Table 7.

References

- [1] D. Bailey, T. Harris, W. Saphir, et al., *The NAS Parallel Benchmarks*, Tech. Report NAS-95-020, Dec. 1995.
- [2] hpmcount, <http://www.nersc.gov/nusers/resources/software/ibm/hpmcount/>.
- [3] Intel MPI Benchmarks (Version 2.3), <http://www.intel.com/cd/software/products/asmona/eng/cluster/mipi/219848.htm>.
- [4] ipm, <http://www.nersc.gov/nusers/resources/software/tools/ipm.php>.
- [5] John McCalpin, STREAM: Sustainable Memory Bandwidth in HPC, <http://www.cs.virginia.edu/stream>.
- [6] MPI Trace Library, http://www.sdsc.edu/user_services/datastar/docs/trace.html.
- [7] NERSC Seaborg and Bassi, <http://www.nersc.gov/nusers/resources/>.
- [8] Fabrizio Petrini, Darren J. Kerbyson, and Scott Pakin, The Case of the Missing Supercomputer Performance: Achieving Optimal Performance on the 8,192 Processors of ASCI Q, *SC03*.
- [9] James C. Phillips, et al., NAMD: Biomolecular Simulation on Thousands of Processors, *SC02*, 2002.
- [10] SDSC DataStar, http://www.sdsc.edu/user_services/datastar/.
- [11] Valerie Taylor, Xingfu Wu, and Rick Stevens, Prophecy: An Infrastructure for Performance Analysis and Modeling System of Parallel and Grid Applications, *ACM SIGMETRICS Performance Evaluation Review*, Volume 30, Issue 4, 2003.
- [12] Xingfu Wu, Valerie Taylor and Rick Stevens, Design and Implementation of Prophecy Automatic Instrumentation and Data Entry System, *the 13th IASTED International Conference on Parallel and Distributed Computing and Systems*, 2001.