Understanding performance of SMP clusters running MPI programs

Franck Cappello, Olivier Richard and Daniel Etiemble

LRI, Université Paris-Sud, 91405 Orsay, France

Abstract

CLUsters of MultiProcessors (CLUMPS) have an hybrid memory model, with message passing between nodes and shared memory inside nodes. We examine the performance of Myrinet clusters of SMP PCs when using a Single Memory Model (SMM) based on the MPICH-PM/CLUMP library of the RWCP, which can directly use the MPI programs written for a cluster of uniprocessors. The specificities of the communication patterns with the SMM approach are detailed. PC clusters with 2-way and 4-way nodes are considered and compared.

1 Introduction

With the continuously increasing performance of commodity computers and the existence of high speed networks, clusters of PCs or workstations look a promising solution to design low cost parallel machines. Multiprocessor motherboards are now widely available and clusters of multiprocessors (CLUMPs) can be designed and evaluated. Considering the cost of a high speed network such as Myrinet, 2-way PCs are only slightly more expensive than uniprocessor PCs. 4-way or 8-way PCs which are used for PC servers can also be used, but they are much more expensive.

CLUMPs mixes two memory models: each node is a multiprocessor using a shared memory and communication between nodes uses a ”message-passing” hardware. A Hybrid Memory Model (HMM) mixing the two approaches can be directly used. In [1], we have presented some results on a 8-node clusters for a HMM version which used MPI for communication between nodes and OpenMP for multithreading inside each node. Performance scaling with successive generation of IA32 processors was considered. In [2], results have been presented for a Myrinet cluster of 36 2-way PCs. HMM is supposed to deliver better performance as it matches the hardware features. However, the programming model is more complicated and MPI programs cannot be directly

Preprint submitted to Elsevier Preprint *25 February 2000*

executed on CLUMPS. A unified memory model (SMM) can be implemented either as a single "shared memory" model by using shared virtual memory environments or a single "message passing" model by using appropriate communication libraries. In [3], performance of HMM has been compared with a MPI SMM version using MPICH-PM/CLUMP [4] for the same cluster of 36 2-way nodes.

In this paper, we detail the performance of the SMM version by examining the communication times and the communication patterns of SMP clusters (MPI SMM approach) compared with uniprocessor clusters. Section 2 presents the methodology. Section 3 presents the overall performance. Section 4 focuses on the communication performance, considering communication times and the modification of communication patterns according to the number of nodes and the number of CPUs per node.

2 Methodology

2.1 SMM implementation

The library called MPICH-PM/CLUMP [4] provides a direct support for CLUMP architectures. Designed by the Parallel and Distributed System Software Laboratory (Real World Computing Partnership), it is based on MPICH 1.0. and implemented on top of a fast communication library called Score/PM. With MPICH-PM/CLUMP, a MPI application runs unchanged on a SMP cluster. Several MPI processes run on each CLUMP node. They share the network interface through a runtime that provide transparent message passing between the MPI processes.

MPICH-PM/CLUMP supports several protocols: *zerocopy* and *eager*. We used the *eager* protocol for two reasons: a) the *eager* protocol provides better performances for all benchmarks of the NAS NPB 2.3 except FT [4] when comparing dual processors PCs clusters with uniprocessor PCs clusters for the same number of nodes (twice the number of processors in the CLUMP). The reasons behind these results are presented in [4]. b) the *zerocopy* protocol was not stable during the experiments on the Parnass2 platform.

2.2 Hardware platforms

The Parnass2 cluster of PCs of the Institute for Applied Mathematics (University of Bonn, Germany) was used as the experimental platform with 2-way nodes. It consists of 36 (recently upgraded to 64) 2-way 400-MHz Pentium II connected by a Myrinet network. The network topology is a Fat-Tree. All processors can send and receive messages with a potential hardware bandwidth of 1 Gbit/sec full duplex. The software environment includes Linux 2.2.1, MPICH-PM/CLUMP version of the MPI library, F77 PGI 3.0 programming environment and Linux Pthread library. For the eager protocol, the performance of MPICH-PM/CLUMP, measured on one processor are: 1) a bandwidth of 96 MB/s between 2 SMP nodes and 128 MB/s inside a SMP node and b) a latency of 10 μs between 2 SMP nodes and 9 μs inside a SMP node (Dongarra's echo test). All benchmarks have been compiled with the o2, unroll and P6 options.

A Myrinet cluster of 4 IBM Netfinity PC servers was used as the second platform. The software environment was exactly the same as for Parnass2. Netfinity servers use 4-way nodes with 400-MHz Pentium II Xeon with 1 MB second level cache. Main memory consists of 1 GB EDO DRAM. The XCOM Myrinet network has the same features as for Parnass2.

2.3 Benchmarks

Floating point benchmarks of NAS NPB2.3 have been used. Measures have been done for all benchmarks on the two platforms, but we only present results for CG, FT, LU and MG in the rest of the paper. EP has very few communications and provides the same nearly perfect speedup with 2-way and 4-way nodes. SP and BT requires a square number of processes. Results have been presented in [3] for 2-way nodes. With 4-way nodes, only 1-node and 4-node configurations can be used, which is not enough to make significant comparisons. For all benchmarks, "constant size" scaling is used.

3 Overall performance

3.1 Performance with uniprocessor nodes

Figure 1 shows the overall performance for the NAS benchmarks running on Parnass2 according to the number of nodes when using only one processor per node. It allows a direct comparison of MFLOPS performance with the one of parallel supercomputers using the same number of nodes [3]. Parnass2 behaves like the proprietary supercomputers for the NAS benchmarks. Performance scales continously. For some benchmark like CG, efficiency decreases with the number of nodes.

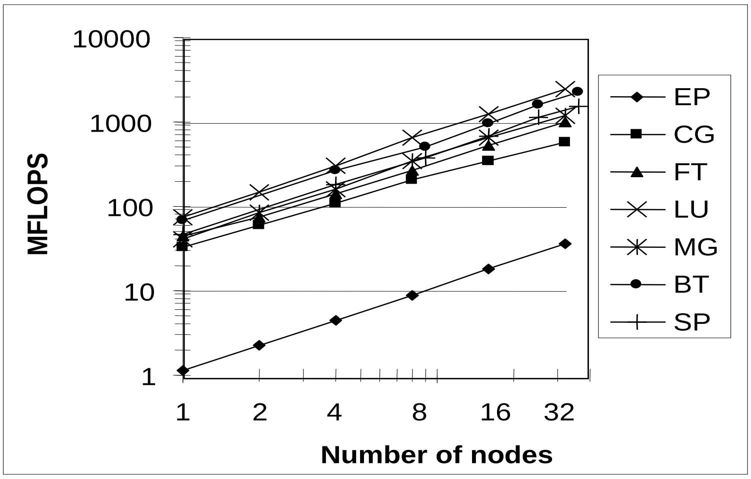

Fig. 1. Parallel efficiency of a network of 400-MHz Pentium II PCs for the NAS benchmark (class A) according to the number of uniprocessor nodes.

3.2 Parallel efficiency with Parnass2

To compare the performance of different configurations of clusters, the metrics that we use is the parallel efficiency, defined as 1-CPU execution time / (n-CPUs execution time * n). For clusters of multiprocessors, two parameters are significant: the number of nodes and the number of CPUs per node. Parallel efficiency can be expressed either according to the number of CPUs or according to the number of nodes. As we are mainly interested by the effect of using SMP nodes instead of uniprocessor nodes, we will present the evolution of parallel efficient according to the number of nodes. Results according to the number of CPUs can be easily deduced from our results. Figure 2 shows the parallel efficiency on Parnass2 according to the number of nodes.

1-way nodes Dark bars in Figure 2 shows the results for 1-way nodes. In that case, the parallel efficiency characterizes the computation to communication ratio for each benchmark when only using external message passing between nodes. It is the reference data, with which data for n-way nodes will be compared. Parallel efficiency is 1 for a perfect speedup. Parallel efficiency greater than 1 corresponds to superlinear behavior. LU exhibits superlinearity. As demonstrated in [5], this feature results from cache effects. The total L2 cache size increases with the number of processors up to the point where it can fit with the working size requirements. MG, SP and BT have a non monotonous behavior with the number of nodes, but parallel efficiency remains greater than 0.9 for most configurations. For FT and CG, parallel efficiency

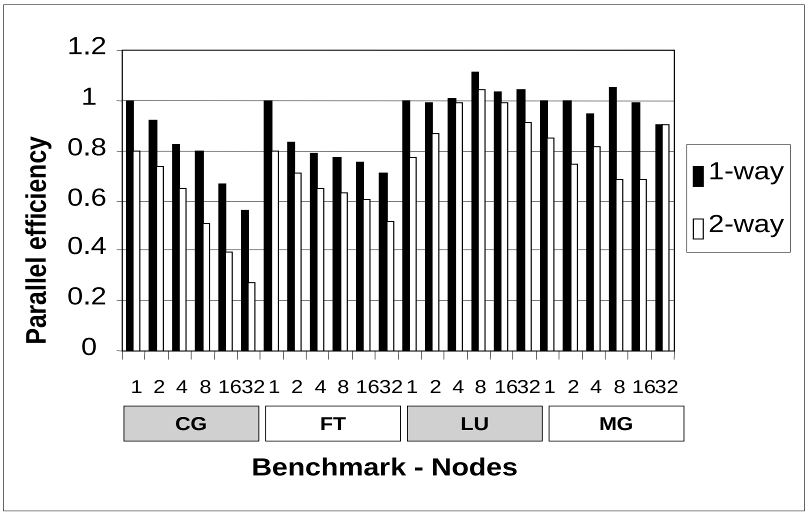

Fig. 2. Parallel efficiency according to the number of nodes with 1-way and 2-way nodes.

constinuously decreases down to 0.6 (FT) and 0.55 (CG).

2-way nodes White bars in Figure 2 indicates the parallel efficiency for 2-way nodes. For all benchmarks, parallel efficiency is better when using 1-way nodes than 2-way nodes. For each benchmark, the evolution of the parallel efficiency with 2-way nodes "follows" the evolution of the 1-way node one, with a shift that depends on the benchmark. The evolution is monotonous for CG and FT, and more complex for LU and MG. These different behaviors depend on the performance of communications, as explained below. The minimum value of parallel efficiency with 2-way nodes is 0.5 for FT, 0.8 for LU and 0.68 for MG. Only CG has values less than 0.5. If 2-way node clusters have lower parallel efficiency, they are generally more cost effective. Assuming that for a low cost cluster as Parnass2, the network and single CPU cost per node are equivalent and that 2-way nodes are 30% more expensive than 1-way node, a cluster with 2-way nodes is more cost effective than a cluster with the same number of 1-way nodes if the parallel efficiency is greater than 0.65, which is the case for most benchmarks without any modification to the MPI program.

3.3 Parallel efficiency with 4-way nodes

Figure 3 shows the parallel efficiency for the cluster of Netfinity servers. Obviously, the global trends are the same as in Figure 2. Parallel efficiency still decreases when switching from 2 to 4-way nodes. With 4 4-way nodes, parallel efficiency goes down to 0.5 for MG, 0.4 for CG and less than 0.4 for FT. LU

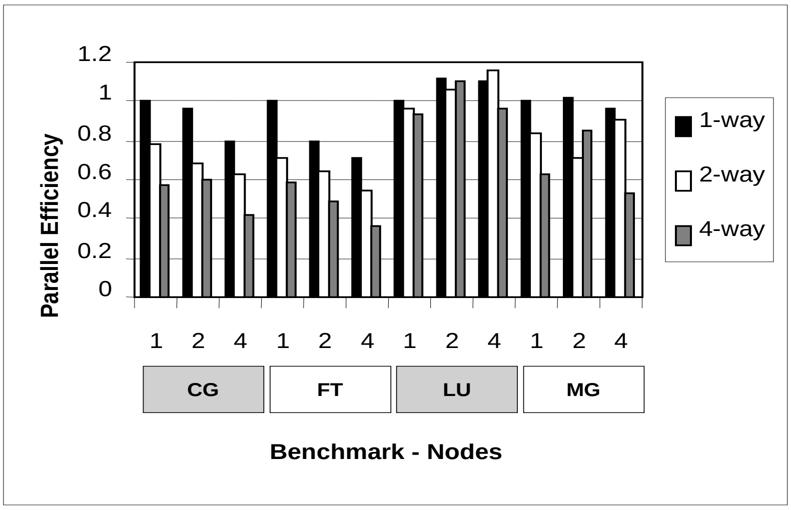

Fig. 3. Parallel efficiency according to the number of nodes with 1-way, 2-way and 4-way nodes .

efficiency remains close to 1 for all configurations. Cache size and speed of the Xeon processor doesn't compensate for the efficiency degradation, whatever number of CPUs is used per node.

4 Understanding the speedup efficiency

To explain the differences in parallel efficiency according to benchmarks, we decomposed the execution time into computation times and communication times. Figure 4 shows the results for Parnass2 and Figure 5 shows the results for Netfinity servers . Summing computation and communication time across all processors make easier the examination of speedup. Left scale and right scale are different for each benchmark. A perfect speedup on computation time corresponds to an horizontal line starting from the top of the left black bar (value comp1 for 1 node). Similarly, perfect speedup for communication times would give horizontal lines for comm1, comm2 and comm4.

4.1 Speedup on computation time

Figures 4 and 5 also give some insights on speedup on computation times. As these times are presented as cumulative sums on all CPUs, a perfect speedup would give the same computation time for any configuration as the comp1 value for 1 node. For all benchmarks except LU, the computation time for

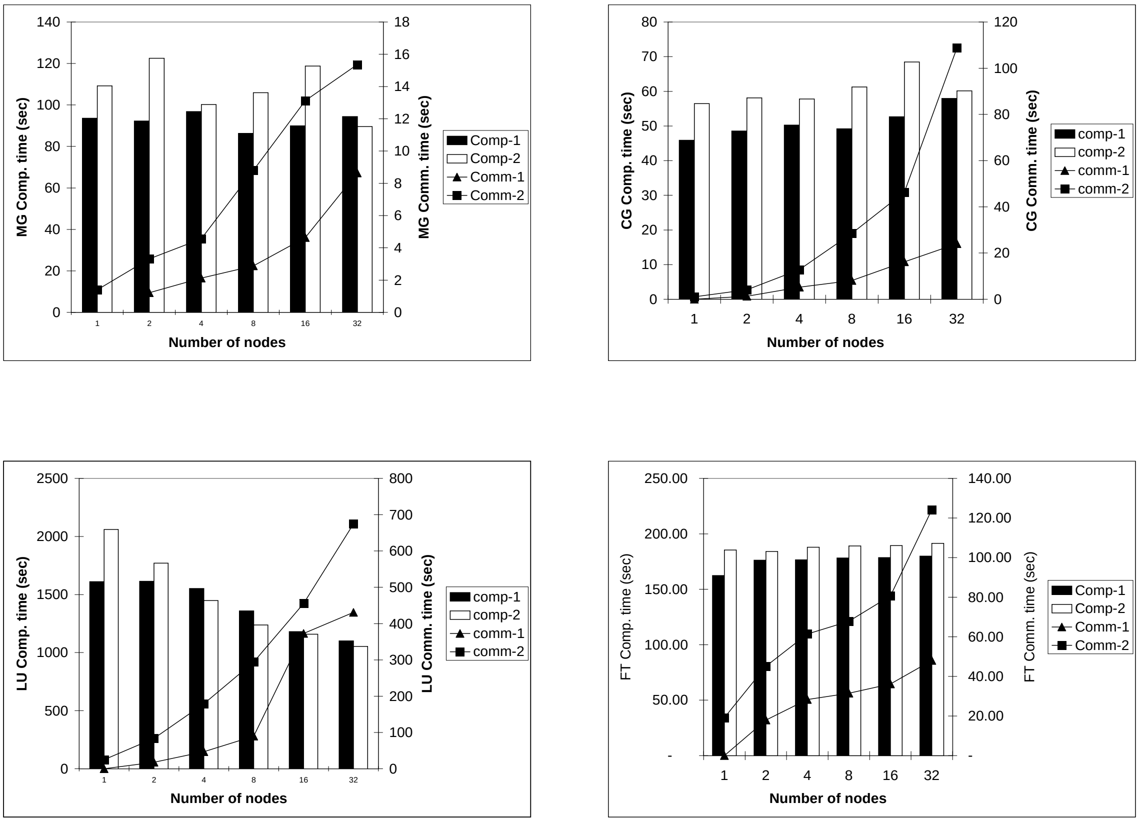

Fig. 4. Time Breakdown for MG, CG, LU and FT according to the number of 1-way and 2-way nodes. The figures breakdown the total execution time, summed across all the processors, into computation times (shown as bars with the left scale) and communication time (lines with the right scale). Comp (resp. Comm) is a shortcut for Computation (resp. Communication). 1 and 2 refers to 1-way and 2-way nodes.

2-way nodes is greater than for 1-way nodes, which means that the speed-up is less than 2. Computational speedup also degrades when switching from 2-way to 4-way nodes. We have demonstrated in [2] that the performance of the Pentium II system bus is the bottleneck.

4.2 Communication times

Comparing left scale and right scale for each benchmark indicates the relative impact of communication times on the overall execution time. CG and FT, for which parallel efficiency is lower, both have significant communication times. For all benchmarks and all configurations (2-way or 4-way nodes), the communication time increases with the number of CPUs per node. The communication times depends on the communication architecture of the CLUMP and the communication patterns of the benchmarks.

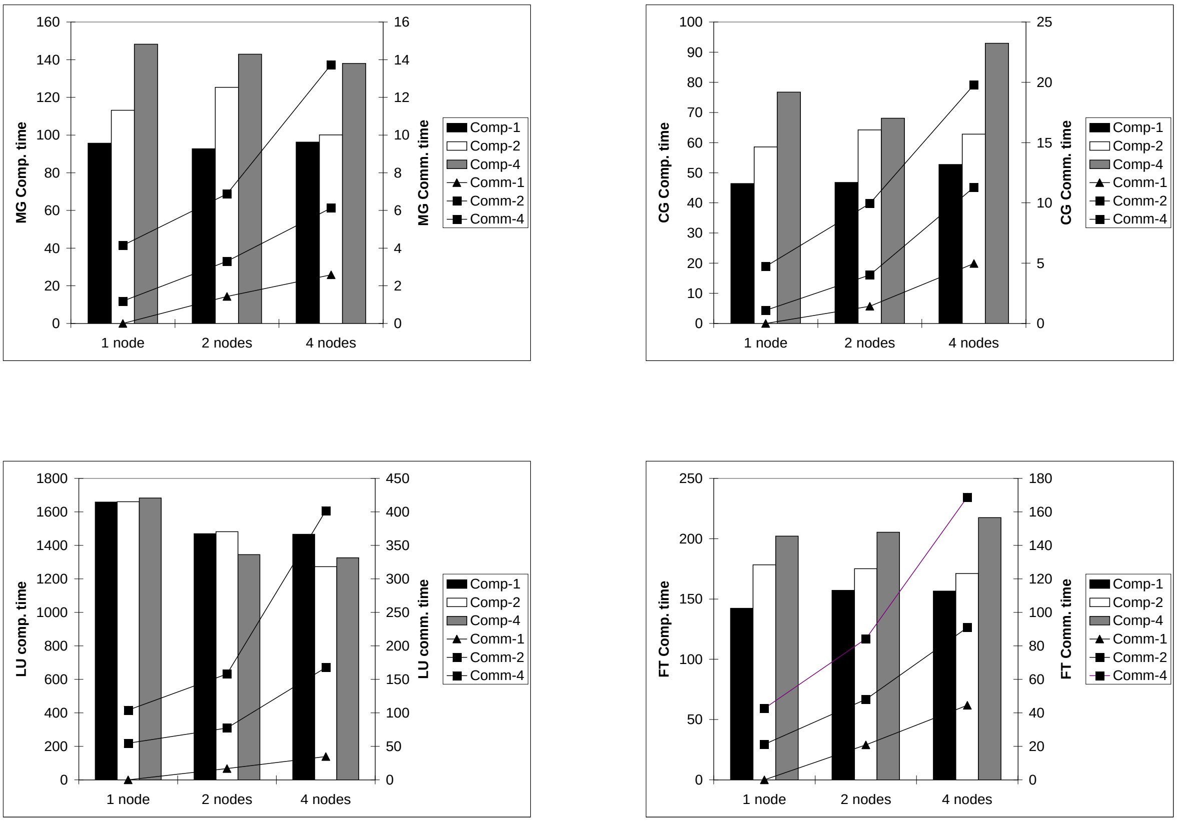

Fig. 5. Time Breakdown for MG, CG, LU and FT according to the number of 1-way, 2-way or 4-way nodes. The figures breakdown the total execution time, summed across all the processors, into computation times (shown as bars with the left scale) and communication time (lines with the right scale).

4.3 CLUMP communication architecture.

N-way nodes involve two communication supports: the network interface for communications between nodes and the memory hierarchy (cache, bus and main memory) for communications inside each node. Figure 6 presents the different communication schemes that occurs in clusters of SMP nodes. We outline that 2 simultaneous duplex messages between 2-way nodes share only one network interface on each node.

In LU, MG and CG, most communication patterns correspond to data exchange between MPI processes, for which each process sends and receives the same amount of data. Table 1 presents the bandwidth values (128 kB exchange) and the latency (8 Bytes exchange) for LU, MG and CG patterns using synchronous and asynchronous communications when the CPUs issue their communication at the same time. Values are given for 1-way nodes (Figure 6a), for processor to processor communication between 2 2-way nodes (

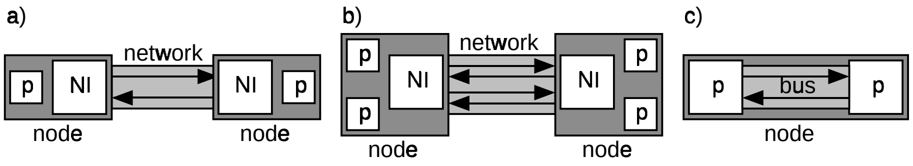

Fig. 6. Communications in clusters. a) external communication between 1-way node. b) external communication between 2-way nodes. c) local communication inside a node .

Figure 6b) and for processor to processor communication inside a 2-way node (Figure 6c). For 2-way nodes, values are the per processor performance, as both communicates simultaneously.

	Uniprocessor	Dual processor external com.	Dual processor local com.
Bandwidth (128 kB) Synchronous	68 MB/s	24 MB/s	120 MB/s
Latency Synchronous	$20\mu s$	$27\mu s$	$13\mu s$
Bandwith (128 kB) Asynchronous	80 MB/s	32 MB/s	132 MB/s
Latency Asynchronous	$12\mu s$	$21\mu s$	$11\mu s$

Table 1
Performance of 1-way and 2-way nodes for the patterns used in LU, CG and MG.

Two details are significant: 1) When two processors inside a node transfers a small message by using a simultaneous synchronous communication, the latency (27 μs) is lower than the latency of a 1-way node that transfers the same amount of messages ($2*20\mu s$). 2) The bandwidth of 1-way node (synchronous: 68 MB/sec and asynchronous: 80 MB/sec) is more than twice the per processor bandwidth of 2-way node (synchronous: 24 MB/sec and asynchronous: 32 MB/sec).

4.4 NAS NPB2.3 communication patterns.

For constant size problems, Won et al [5] showed that the size of a message generally decreases when the number of nodes increases. But the number of messages sent and received by each node is likely to increase, as more nodes lead to more complicated communication patterns.

SMM model places a MPI process on each processor. When using n-way nodes instead of 1-way nodes, the communications takes place between $n*p$ processors instead of p processors. The communication patterns are thus different. Figures 7, 8 and 9 present the main communication patterns for LU, CG and

MG when using 2, 4, 8 and 16 processors. Small white boxes are for the processors. Light grey rectangles correspond to the mapping of communications with 2-way nodes, and dark grey rectangles to the mapping on 4-way nodes. The decomposition into external and local communications is thus shown for 2-way and 4-way nodes. For 2-way nodes, external communications are the ones between light grey boxes. For 4-way nodes, external communications are between dark grey boxes. In each figure, communications are decomposed into small and large messages, which number of each type is given (number * 64-bit words).

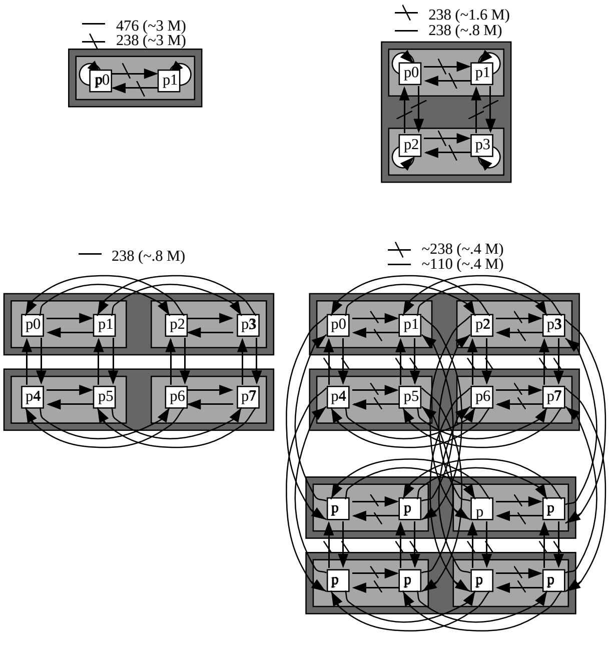

Fig. 7. Communication patterns for MG using 2, 4, 8 and 16 processors.

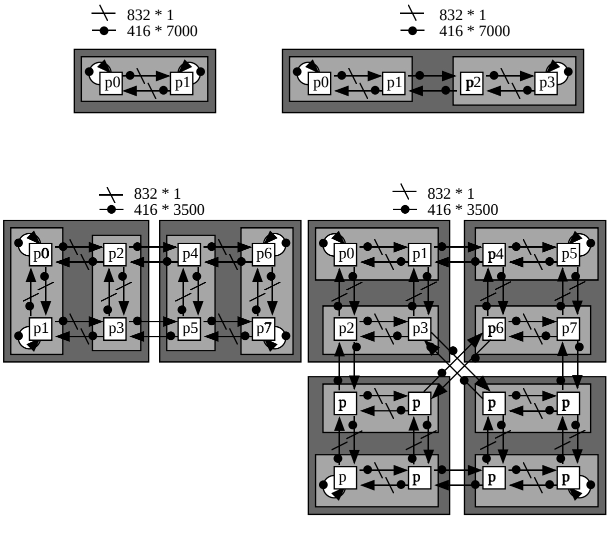

Fig. 8. Communication patterns for CG using 2, 4, 8 and 16 processors.

It is well known that latency is the main factor for small messages and bandwidth for large messages. The difference in message sizes of the different

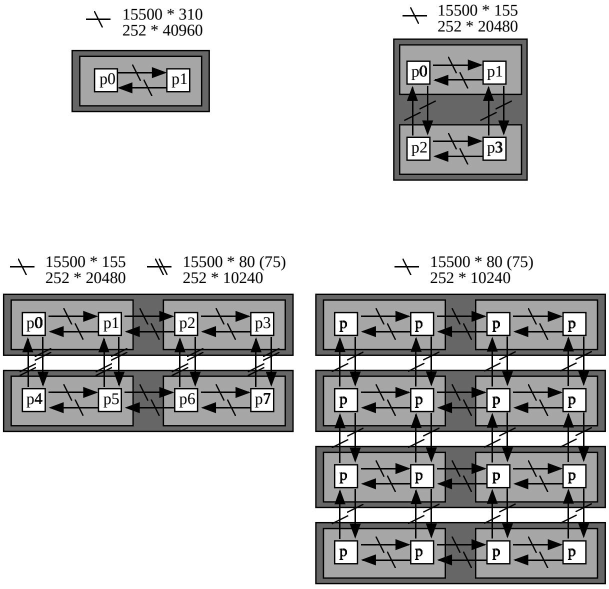

Fig. 9. Communication patterns for LU using 2, 4, 8 and 16 processors.

benchmarks is the main reason for the differences between SMM communication times (comm2 and comm4) and the uniprocessor communication time (comm1) in figures 4 and 5.

General trends MG communicates asynchronously messages of various sizes, but long messages dominates. For 8 processors, about 80% of the data are transferred in less than 16% of the messages. Communication bandwidth is the key factor and 2-way and 4-way node platforms spend more time in communication than 1-way node platforms. As shown in Figure 7, the CPUs inside SMP nodes actually share the network interface for communications with the other SMP nodes.

CG also uses asynchronous communications, with two message sizes: 1 word and x words, where x depends on the number of processors. About half of the messages are x words long. x equals 1750 with 32 processors. Such long messages still require 1 ms compared to 21 μs for 1 word messages. As for MG, communication bandwidth dominates and 2-way and 4-way node platforms are disadvantaged. Again, the sharing of the network interface between nodes and of the memory hierarchy inside the SMP nodes explains this behavior.

LU mainly uses synchronous communications. Up to 4 nodes, large message dominates and 2-way and 4-way node platforms are disadvantaged (bandwidth disadvantage). For higher number of nodes, LU spends more time for small messages than for large ones and 2-way node platforms are more efficient (latency gain).

FT exhibits a specific behavior. It mainly uses global exchange communication patterns (MPI_alltoall). As shown in [3], for this communication pattern, the communication time is nearly the same for the n MPI processes (1-way nodes) and the $2 * n$ MPI processes (2-way nodes), except for $n = 4$ for which communication time is 25% longer with 2-way nodes.

4-way nodes versus 2-way nodes Table 2 gives the precise communication times for MG, CG, LU and FT. It also presents the communication time ratios: (2-way nodes/1-way nodes) and (4-way nodes/ 2 way nodes).

	MG		CG		LU		FT	
	2 nodes	4 nodes	2 nodes	4 nodes	2 nodes	4 nodes	2 nodes	4 nodes
1 way (s)	1.4	2.6	1.4	5	16.9	34.7	20.9	44.5
2 ways (s)	3.3	6.1	4	11.3	77.3	167.6	48	91
4 ways (s)	6.9	13.7	9.9	19.8	158.1	401.9	84.2	168.8
2 ways / 1 way	2.3	2.6	2,8	2	4.6	4.8	2.3	2
4 ways / 2 ways	1.8	1.8	2.5	1.7	2	2.4	1.7	1.8

Table 2
Communication time and ratio for MG, CG, LU and FT according to 1, 2 and 4 ways nodes (for 2 and 4 node configurations).

The performance degradation of the communication times for the 4-way nodes is due to the increase of the communication complexity. 4-way configurations use two times more processes than 2-way configurations (for the same number of nodes). Despite the communication times of 4-way nodes are higher than the ones with 2-way nodes (as shown in Figure 5), the performance degradation of the communication times is lower from 2-way nodes to 4-way nodes than from 1-way nodes to 2-way nodes.

The communication topologies of the NAS benchmarks and the way they are mapped on the Clump architecture establish the communication performance. Figures 7, 8 and 9 show that NAS benchmarks use regular mesh topologies. Most of the communications occur with direct neighboring processors on the topology. Since processes are mapped on multiprocessor node in a block way, some communications stay local within the multiprocessor. When the number of processors per node increases, there are more local communications inside the multiprocessor node. Table 3 presents the number of external and internal communications for MG, CG and LU and for all CLUMP configurations .

	MG			CG			LU		
	4	8	16	4	8	16	4	8	16
External:Internal (2 ways)	2:3	4:1	6:1	1:2	2:3	3:2	2:1	3:1	3:1
External:Internal (4 ways)	0:8	4:4	8:4	0:5	2:6	3:5	0:4	4:3	4:3

Table 3
Number of external and internal communications (per processor) for the MG, CG and LU according to 2 and 4 way nodes and for 4, 8 and 16 processor configurations

For 4-way nodes, there is a lower ratio (external/internal communication) than for 2-way nodes. As internal communications are performed faster than external ones (as shown in table 1), performance degradation of the communication from 2-way nodes to 4-way nodes is lower than from 1-way nodes to 2-way nodes (cf. Table 2.

5 Related works

The performance issues of the NAS benchmarks have been presented in [5]. Especially, Wong et al have examined the architectural requirements and scalability. This work is restricted to uniprocessor nodes, comparing performance of a cluster of workstations and the Origin 2000 machine. Performance of Message Passing SMM has been detailed in [6] for a cluster of SUN SMPs. The authors claimed that a cluster of uniprocessors can be faster than a cluster of multiprocessors. They also note that the memory hierarchy and especially the population of the memory banks have a significant impact on performance.

Other programming models have been designed for the CLUMPs. For example, Shared virtual memory environments (DSVM) provide another alternative to unify the memory models, as presented in [7] [8] [9][10]. Recently, OpenMP [11] has been implemented on a cluster of SMPs on top of the Treadmark DSM system.

In[12], a hybrid shared memory/distributed memory model is proposed for CLUMPs. Communications between nodes use message-passing and remote memory operations. The model is based on the SPMD paradigm in which the programmer first distributes the data set between the different nodes. Within each node, every portion of the distributed data set is partitioned among the threads.

6 Conclusion

In this paper, we have examined the performance of Myrinet clusters of SMP PCs when using a unified message passing programming model based on MPICH-PM/CLUMP library.

With low cost 2-way PC nodes, the performance that are obtained without any modification of the MPI program shows that the clusters of 2-way PCs are cost effective for most of the NAS benchmarks. A detailed examination of the communication times and the communication patterns for each benchmark shows that the programming model is more efficient with synchronous communications using small messages. With applications using a lot of asynchronous communications with long messages, the parallel efficiency can decrease significantly, up to the point where cost effectiveness of 2-way nodes becomes debatable. In that case, using a hybrid memory model is more efficient [3].

With 4-way nodes, which are presently used in more expensive PC servers, the performance trends that are observed with 2-way nodes become more pronounced. The impact of system bus limitations is also exacerbated. 4-way nodes increase the performance degradation of the communication times, but the degradation is lower from 2-way nodes to 4-way nodes than from 1-way nodes to 2-way nodes. This is mainly due to 1) the regularity of benchmark topologies and processor mapping on the CLUMP, 2) the better performance of local communications compared to external communications. To-day, using 4-way nodes for numerical applications is not a cost effective solution for PC clusters.

Present results show that using directly MPI programs with clusters of SMP PCs is a worthwhile solution. Future works include investigating the performance of the MPI unified programming model with SMP motherboards using a more efficient local communication systems with Alpha, Power3 or Itanium CPUs.

7 Acknowledgments

Gerhard Zumbusch and Prof. Michael Griebel from the Institute for applied mathematics of Bonn University (Germany)allowed us to use the Parnass 2 Cluster. (http://Parnass.net)

IBMFrance provided 4 Netfinity servers during three months for the experiments on 4-way nodes.

We thank Prof. Ishikawa and the other researchers of the Parallel and Distributed System Software Laboratory for providing us MPICH-PM/CLUMP (http://www.rwcp.or.jp/lab/pdslab/dist/).

References

[1] F. Cappello, O. Richard O and D. Etiemble. Performance of the NAS Benchmarks on a cluster of SMP PCs using a parallelization of the MPI programs with OpenMP. In *Proceedings of Parallel Computing Technologies (PaCT-99), LNCS 1662*, 1999.

[2] F. Cappello and O. Richard. Performance characteristics of a network of commodity multiprocessors for the nas benchmarks using a hybrid memory model. In *Proceedings of PACT'99.* Also available at: http://www.lri.fr/ fci/goinfreWWW/PACT99.ps, 1999.

[3] F. Cappello, O. Richard O and D. Etiemble. Investigating the performance of two programming models for clusters of SMP PCs In *Proceedings of HPCA6.* Also available at: http://www.lri.fr/ fci/goinfreWWW/HPCA2K.ps, 2000.

[4] T. Takahashi, F. O'Carrol, H. Tezuka, A. Hori, S. Sumimoto, H. Harada, Y. Ishikawa, and P.H. Beckman. Implementation and evaluation of MPI on an SMP Cluster. In *Proceedings of Workshop on Personal Computers based Networks Of Workstations.* IPPS/SPDP, 1999.

[5] F. C. Wong, R. P. Martin, R. H. Arpaci-Dusseau, D. T. Wu, and D. E. Culler. Architectural requirements and scalability of the nas parallel benchmarks. In *Proceedings of Supercomputing'99*, 1999.

[6] Steven S. Lumetta, Alan Mainwaring, and David E. Culler. Multi-protocol active messages on a cluster of SMPs. In *SC'97: High Performance Networking and Computing: Proceedings of the 1997 ACM/IEEE SC97 Conference: November 15–21, 1997, San Jose, California, USA.*, 1997.

[7] D. J. Scales, K. Gharachorloo, and A. Aggarwal. Fine-grain software distributed shared memory on SMP clusters. In *Proc. of the 4th IEEE Symp. on High-Performance Computer Architecture (HPCA-4)*, February 1998.

[8] R. Stets, S. Dwarkadas, N. Hardavellas, G. Hunt, L. Kontothanassis, S. Parthasarathy, and Michael Scott. Cashmere-2L: Software coherent shared memory on a clustered remote-write network. In *Proc. of the 16th ACM Symp. on Operating Systems Principles (SOSP-16)*, October 1997.

[9] R. Samanta, A. Bilas, L. Iftode, and J. P. Singh. Home-based SVM protocols for SMP clusters: Design and performance. In *Proc. of the 4th IEEE Symp. on High-Performance Computer Architecture (HPCA-4)*, February 1998.

[10] Andrew Erlichson, Neal Nuckolls, Greg Chesson, and John Hennessy. SoftFLASH: Analyzing the performance of clustered distributed virtual shared memory. In *Proceedings of the Seventh International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 210–220, 1996.

[11] Charlie Hu Honghui Lu Alan Cox and Willy Zwaenepoel. OpenMP for Networks of SMPs. In *Proc. of the Second Merged Symp. IPPS/SPDP 99*, 1999.

[12] Y. Tanaka, M. Matsuda, M. Ando, K. Kazuto, and M. Sato. Compas: A pentium pro PC-based SMP Cluster and its experience. In *IPPS Workshop on Personal Computer Based Networks of Workstations*, pages 486–497. LNCS, 1998.