For PDPTA'99 Conference Proceedings

Cluster Computing Technologies, Environments,
and Applications (CC-TEA)Technical Session

An Assessment of MPI Environments for Windows NT

K. Takeda[1], *ktakeda@soton.ac.uk*, Tel: +44 (0)1703 594467, Fax: +44 (0)1703 593058
N.K. Allsopp[2], *nka@pac.soton.ac.uk*, Tel: +44 (0)1703 760834, Fax: +44 (0)1703 760833
J.C. Hardwick[3], *jch@microsoft.com*, Tel: +44 (0)1223 744754, Fax: +44 (0)1223 744777
P.C. Macey[4], *patrick.macey@SERuk.com*, Tel: +44 (0)115 9357060, Fax: +44 (0)115 9357064
D.A. Nicole[1], *dan@ecs.soton.ac.uk*, Tel: +44 (0)1703 592703, Fax: +44 (0)1703 593903
S.J.Cox[1], *sc@ecs.soton.ac.uk*, Tel: +44 (0)1703 593116, Fax: +44 (0)1703 593903
D.J.Lancaster[1], *djl@ecs.soton.ac.uk*, Tel: +44 (0)1703 593943, Fax: +44 (0)1703 593903

[1] *Dept. of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK*
[2] *Parallel Applications Centre, 2 Venture Road, Chilworth, Southampton SO16 7NP, UK*
[3] *Microsoft Research Ltd, St George's House, 1 Guildhall Street, Cambridge CB2 3NH, UK*
[4] *SER Systems Ltd, 39 Nottingham Road, Stapleford, Nottingham NG9 8AD, UK*

An Assessment of MPI Environments for Windows NT

K. Takeda[1], N.K. Allsopp[2], J.C. Hardwick[3], P.C. Macey[4], D.A. Nicole[1], S.J.Cox[1] and D.J.Lancaster[1]
[1] *Dept. of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK*
[2] *Parallel Applications Centre, 2 Venture Road, Chilworth, Southampton SO16 7NP, UK*
[3] *Microsoft Research Ltd, St George's House, 1 Guildhall Street, Cambridge CB2 3NH, UK*
[4] *SER Systems Ltd, 39 Nottingham Road, Stapleford, Nottingham NG9 8AD, UK*

Abstract *In this paper we evaluate the MPI environments currently available for Windows NT on the Intel IA32 and Compaq/DEC Alpha architectures. We present benchmark results for low-level communication and for the NAS Parallel Benchmarks to allow comparison to other systems, but our primary interest is determining real application performance and robustness in production cluster environments. For this we use PAFEC-FE, a large FORTRAN code for finite-element analysis. We present results from three MPI implementations, two architectures, and three networking technologies (10 Mbit/s and 100 Mbit/s Ethernet and 1 Gbit/s Myrinet).**

Keywords: Clusters, MPI, Windows NT

1 Introduction

The rapid advances in computer technology over the last few years mean that it is now possible to build supercomputer-class systems at commodity prices. The NASA Beowulf project [1] has led the way in developing this technology using MPI [2] and open-source operating systems such as Linux. To complement this work, we have been introducing commodity supercomputing technology into industries that use Microsoft Windows NT [3, 4, 5, 6, 7].

Previous work by others in this field has concentrated on low-level communication and application performance over 10Mbit/s Ethernet on IA32-based systems [8]. The rapid development and deployment of MPI on NT means that it is now possible to compare the real-world performance of several different systems. In this paper we present benchmark results for low-level communication, the NAS Parallel Benchmarks, and a commercial industrial application, running on Intel Pentium II and Compaq/DEC Alpha systems, using three implementations of MPI, and running over 10 Mbit/s Ethernet, 100 Mbit/s Fast Ethernet and 1 Gbit/s Myrinet networks. We also discuss the robustness, security and cluster management issues related to each MPI implementation.

2 MPI on Windows NT

Most Unix MPI implementations are derived from the MPICH code base from Mississippi State University and Argonne National Labs [9], thanks to its portable device abstraction layer. MPICH has also been ported to Windows NT, resulting in the three different implementations evaluated in this paper.

WMPI is a TCP/IP port developed at the University of Coimbra, Portugal [10]. It has

---

* All trademarks and registered trademarks used in this paper are the property of their respective owners.

now been commercialized by GENIAS Software GmbH as PaTENT MPI [11]. For simplicity within this paper we will refer to this as WMPI. Similarly, initial work at Mississippi State University on an SMP, TCP/IP and Myrinet port of MPICH to NT [12] has resulted in MPI/Pro [13], a commercial implementation by MPI Software Technology Inc., supporting both TCP/IP and the VIA standard for system-area networks. Finally, the Fast Messages project provides HPVM [14], which supports MPI and other user-level protocols on top of TCP/IP and Myrinet.

3 Parallel Performance

In this section we discuss the performance of three different Windows NT 4.0 clusters, each of which represents a different computational environment that might be found in industry.

First, a cluster of eight DEC/Compaq Alpha 500MHz 21164 PCs with 256MB of RAM each, connected by switched Fast Ethernet, and using MPI/Pro and HPVM (which we have ported to Alpha socket networking). This represents a cluster optimized for floating-point arithmetic.

Second, a cluster of sixteen dual-processor 300MHz Pentium II PCs with 384MB of RAM each, connected by switched Ethernet and Myrinet, and using MPI/Pro and HPVM. This cluster is optimized for fine-grained problems, with a high-performance system-area interconnect.

Third, a cluster of four dual-processor 450MHz Pentium II PCs with 128MB of RAM each, connected by switched Ethernet and Fast Ethernet, and running MPI/Pro and WMPI. This represents a "found cluster" that in industry might be composed of existing office machines.

All measurements were performed with the clusters isolated from network traffic, and using only one processor per machine. Digital Visual FORTRAN v5.0 was used on all platforms.

3.1 Communication Performance

Interprocessor communication is often the limiting factor in determining the overall performance of parallel applications. Therefore, in Table 1 we give MPI latency and bandwidth measurements for the different cluster environments. For comparison, we also include MPI/Pro results for Giganet, a VIA-compatible SAN interconnect similar to Myrinet.

We expected all MPI implementations to be able to drive Ethernet and Fast Ethernet to near their bandwidth limits. However, the PII-450 cluster proved to be limited to about 70% of its theoretical Fast Ethernet bandwidth under both WMPI and MPI/Pro *[note to referees: we believe this to be due to a hardware interaction, and will include updated numbers and an explanation of the problem encountered in the final paper]*. In addition, Ethernet message latencies are an order of magnitude greater than those of Myrinet and comparable interconnects. These system-area interconnects provide about half of their advertised Gbit/s bandwidth to end-user applications.

From these results we would expect Ethernet to be suitable only for applications with very low communication requirements, Fast Ethernet to extend this range to include applications that have limited coarse-grained communication, and Myrinet or similar system-area networks to be required for applications with intensive or fine-grained communication.

Configuration	*Latency (μsecs)*	*Asymptotic Bandwidth (Mbytes/sec)*
Ethernet, PII-450, MPI/Pro v1.2.3	300	1.0
Ethernet, PII-450, WMPI 4.09	374	1.3
Fast Ethernet, PII-300, MPI/Pro v1.2.3	246	9.5
Fast Ethernet, PII-450, MPI/Pro v1.2.3	183	6.5
Fast Ethernet, PII-450, WMPI 4.09	259	7.1
Fast Ethernet, 21164-500, MPI/Pro v1.2.3	196	8.7
Myrinet, PII-300, HPVM 1.0	14	86
Giganet, PII-350, MPI/Pro (from [19])	24	77

Table 1. MPI communications performance, measured using MPI/Pro `lat` and `bw` benchmarks.

3.2 NAS Parallel Benchmarks

To enable more detailed comparison with other platforms, Table 2 gives results for version 2.3 of the NAS Parallel Benchmarks [15]. For reasons of space, only results for 8 processors (9 in the case of BT and SP) are given here, although full results may be found in [16].

These results show that the latency and bandwidth measures of the previous section are too simplistic to accurately predict application behavior. For example, using Myrinet and HPVM instead of commodity Fast Ethernet and MPI/Pro gives a significant (i.e., greater than 10%) performance increase on only two of the seven benchmarks. These are CG and MG, the communication-intensive conjugate gradient and multigrid kernels. On the FT benchmark, HPVM is significantly slower, since the array transposes of the Fast Fourier transform stress MPI's all-to-all communication primitives. These use naïve algorithms in the original MPICH code base, but have been replaced by optimized version in MPI/Pro. Finally, HPVM hangs on the LU-factorization benchmark, indicating that it isn't robust in the face of the large numbers of very small messages being sent by this application.

The application performance of the Alpha cluster is also less than we might expect given that its floating-point performance is significantly greater than that of the Intel architecture. This advantage is only really demonstrated in the EP kernel, which generates pseudo-random floating-point numbers with almost no communication.

3.3 Real Application Performance

While benchmark programs can give an indication of the performance of new systems, their extrapolation to real-world application performance is not always straightforward. In this section we use the parallel PAFEC-FE code [7, 17] to test the application performance of our Windows NT cluster systems. PAFEC-FE is a large industrial finite-element analysis code, consisting of several hundred thousand lines of FORTRAN and over 18,000 subroutines. It is a legacy application whose performance-critical sections have been parallelized using a master/slave approach. The test case is a sonobuoy, consisting of a piston transducer embedded in a cylindrical baffle, shown in Figure 1. The structural model has 10923 degrees of freedom and a front size of 813, while the acoustic boundary element has 1664 acoustic degrees of freedom. For this test case, parallel PAFEC-FE uses four main stages.

Stage 1 merges the contributions from individual finite elements, and then reduces the resulting shared sparse stiffness matrix. To ensure stability on a client's network the very large global sums required were implemented using many small messages (see Section 4 below), and so performance is strongly affected by latency and bandwidth. On 10 Mbit/s Ethernet this stage takes longer as the number of processors is increased, as shown in Table 3. On 100 Mbit/s Ethernet it scales moderately well under MPI/Pro, with the Alphas fairing better than the PII clusters (see Table 5). WMPI does not scale beyond the second processor (see Table 4).

Stage 2 forms the boundary element matrices. This requires no communication after the

NAS Benchmark, Class A	*PII-300, Fast Ethernet, MPI/Pro v1.2.3*	*PII-300, Myrinet, HPVM 1.0*	*Alpha, Fast Ethernet, MPI/Pro v1.2.3*
BT	307.4	302.5	N/A
CG	87.6	154.1	68.7
EP	6.2	6.2	19.0
FT	145.2	116.7	168.5
LU	324.9	-	476.2
MG	221.3	269.0	278.8
SP	216.7	236.2	N/A

Table 2. NAS parallel benchmark performance in MFLOPS on 8 processors (9 in the case of BT and SP)

collocation points have been shared between processors, so scaling is perfect, and the results correspond to those of the NAS EP benchmark. Notably, the Alphas perform significantly better than the IA32 systems during this stage.

Stage 3 reduces the boundary element matrices. This is the most numerically intensive portion of the code, and although it still requires a significant amount of communication, speedups are obtained in all cluster environments. Note that WMPI is generally faster than MPI/Pro on this stage, whereas the reverse is true for Stage 1. This is due to Stage 3 having a stronger requirement for high bandwidth, whereas Stage 1 has a stronger requirement for low latency.

Stage 4 performs Gaussian elimination with partial pivoting of the columns between processors. This has a similar structure to the NAS LU benchmark, sending many short messages. It's performance is again limited by latency and bandwidth, with slowdowns occurring on the 10 Mbit/s Ethernet as more processors are used. On 100Mbit/s fast Ethernet the Alphas scale well up to four processors, while the IA32 systems achieve only low parallel efficiency.

Although we were able to run a smaller test case using HPVM 1.0 [16], PAFEC-FE causes the HPVM MPI implementation to leak memory, and we were not able to run the full sonobuoy test case. The more recent HPVM 1.2 release cannot even run the smaller test case.

It is clear that the overall performance of parallel PAFEC-FE is not dominated by any single feature of the cluster architecture. However, we have been able to characterise the individual parallel code sections in terms of their sensitivity to processor and network performance, and to relate them to NAS benchmark and low-level communication results.

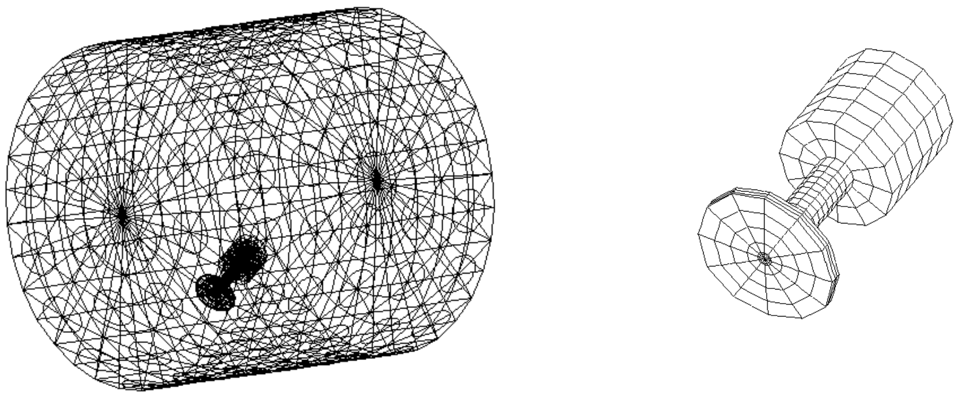

Figure 1. The PAFEC-FE test case is a finite/boundary element model of a sonobuoy, consisting of a cylindrical baffle (left) containing a piston transducer (right).

	1 proc		*2 procs*		*3 procs*		*4 procs*	
	MPI/Pro	*WMPI*	*MPI/Pro*	*WMPI*	*MPI/Pro*	*WMPI*	*MPI/Pro*	*WMPI*
Stage 1	3974	3974	3069	3135	2961	3683	2980	3624
Stage 2	240	240	142	168	84	85	69	66
Stage 3	4496	4496	2731	2525	2004	2016	1925	1631
Stage 4	212	212	255	252	358	362	425	354

Table 3. PAFEC-FE performance on PII-450 cluster with 10 Mbit/s Ethernet, comparing MPI/Pro 1.2.3 (left-hand columns) and PaTENT WMPI 4.09 (right-hand columns). Figures are timings in seconds.

	1 proc		*2 procs*		*3 procs*		*4 procs*	
	MPI/Pro	*WMPI*	*MPI/Pro*	*WMPI*	*MPI/Pro*	*WMPI*	*MPI/Pro*	*WMPI*
Stage 1	3974	3974	2539	2401	1696	2524	1833	2451
Stage 2	240	240	148	142	85	90	64	68
Stage 3	4496	4496	2696	2329	1620	1622	1621	1258
Stage 4	212	212	162	142	128	130	162	113

Table 4. PAFEC-FE performance on PII-450 cluster with 100 Mbit/s fast Ethernet, comparing MPI/Pro 1.2..3 (left-hand columns) and PaTENT WMPI 4.09 (right-hand columns). Figures are timings in seconds.

	1 proc		*2 procs*		*3 procs*		*4 procs*	
	PII-300	*Alpha*	*PII-300*	*Alpha*	*PII-300*	*Alpha*	*PII-300*	*Alpha*
Stage 1	5920	7047	3082	3823	2396	2845	1946	2259
Stage 2	450	186	191	106	127	66	96	47
Stage 3	7469	6347	3544	3195	2528	2143	1925	1624
Stage 4	374	202	219	108	184	80	153	63

Table 5. PAFEC-FE performance on PII-300 (left-hand columns) and Alpha 21164-500 (right-hand columns) clusters, both using 100 Mbit/s Fast Ethernet with MPI/Pro 1.2.3. Figures are timings in seconds.

4 Deployment Issues

Key issues to the successful deployment of Windows NT clusters in production MPI environments include robustness in the face of application crashes and user interrupts, integration into the NT security model, and manageability features (which can range from simple remote application startup to a full batch scheduling system.)

Of the three MPI implementations tested, MPI/Pro v1.2.3 goes the furthest towards achieving these goals. Its `mpirun` startup mechanism proved very reliable, and shut down applications cleanly after crashes and user interrupts. MPI/Pro is also integrated into the NT domain security model, allowing password protection of applications and data files. It has no specific manageability features, although it can be used with batch scheduling systems such as Platform Computing's LSF [18], and MPI Software Technology have announced plans for a competing product [19].

PaTENT WMPI also includes NT domain security features. However, application cleanup is unreliable, and failed jobs often require logging out of and back into the affected machine.

HPVM offers just the communication mechanism, with no specific startup, manageability or security features. However, it does include a Java-based front-end that fully integrates it into the LSF batch scheduling system, if present. We also used HPVM successfully with MPI/Pro's `mpirun` mechanism.

As an example of the issues and opportunities faced, WMPI and the parallel PAFEC-FE code have been installed at Celestion International. Celestion is a small loudspeaker manufacturer employing fewer than 100 people, and already made heavy use of the serial PAFEC-FE codes. Office PCs are their only computational resource. During installation, a problem was encountered with long MPI messages at the end of Stage 1, which resulted in intermittent memory errors. This was traced to insufficient buffering on the Ethernet cards in the presence of intensive network traffic. Rather than replacing Celestion's existing hardware, Stage 1 was modified to send a sequence of smaller messages, with a synchronisation step between each message. This compromised the network performance but ensured the stability of the code.

Celestion originally envisaged using parallel PAFEC-FE on their office PCs in an overnight batch mode. However, the increases in performance and the ability to run larger test cases has caused them to re-evaluate this practice. They now utilise all available machines during the day, enabling them to increase their design throughput substantially.

5 Conclusions

In this paper we have assessed the performance and manageability of current MPI environments (specifically, MPI/Pro 1.2.3, PaTENT WMPI 4.09 and HPVM 1.0) under Windows NT on the IA32 and Compaq/DEC Alpha platforms, using a range of benchmarks.

Of these, MPI/Pro and WMPI successfully ran all the benchmarks, and had similar performance over the Ethernet and Fast Ethernet networks likely to be encountered on "found clusters" in existing NT installations. MPI/Pro is currently the more robust of the two, and also supports the VIA standard for gigabit-class SANs if a dedicated cluster is required.

Our experiences with HPVM on the Myrinet interconnect lead us to two important conclusions. First, although testing low-level communication performance is an important sanity check for a new cluster (as evidenced by the bandwidth problems it found in our PII-450 cluster), it is a worse predictor of application performance than we were expecting, as shown in the NAS numbers for Fast Ethernet versus Myrinet. Second, HPVM's failure to run the PAFEC-FE application raises the issue of coding to a particular MPI implementation (as opposed to just optimizing for one, for example by re-writing Stages 1 and 3 to better suit WMPI and MPI/Pro respectively).

Since MPI is a well-defined standard, applications should not have to be rewritten to use a different MPI implementation, and indeed we rely on this for portability of applications between Unix and NT. However, it is possible to write legal MPI codes that work on some implementations but cause problems (such as resource leaks) on others. We intend to explore this issue in the context of PAFEC-FE and HPVM, and perhaps draw some conclusions about "coding standards" for MPI portability.

In general, we believe that MPI on NT is now a viable platform for high-performance parallel applications, as we have shown in the deployment of the commercial PAFEC-FE code within Celestion. The one remaining area where MPI on NT lags behind Unix is in job management software for small-scale clusters, although there are a number of active research projects in this area, including Symera [20] and Condor [21].

Acknowledgements

This work has been funded as part of an ESPRIT project no. 24871, part of the HPCN TTN Network supported by the EC, and by EPSRC. The authors would like to thank Microsoft Research Ltd, MPI Software Technology Inc. and Genias GmbH for supporting this work. We would also like to acknowledge the HPVM team for their collaboration with us.

References

[1] D. Ridge, D. Becker, P. Merkey and T. Sterling. "Beowulf: Harnessing the Power of Parallelism in a Pile-Of-PC's". *Proc. 1997 IEEE Aerospace Conference*

[2] Message Passing Interface Forum. MPI: A Message-Passing Interface Standard, 1994, `http://www.mpi-forum.org/docs`

[3] D.A. Nicole, K. Takeda and I.C. Wolton. "Running HPC Codes on DEC Alphas and NT". *Proc. HPCI Conference '98*, Manchester, January 1998.

[4] S.J. Cox, G.J. Daniell and D.A. Nicole. "Maximum Entropy, Parallel Computation and Lotteries". *Proc. 1998 International Conference on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, 1998.

[5] D. Emerson, K. Maguire, K. Takeda and D.A. Nicole. "An Evaluation of Commodity Supercomputers for CFD Applications". *Proc. Parallel CFD '98*, Taiwan, May 1998.

[6] K. Takeda, O.R. Tutty and D.A. Nicole. "Parallel Vortex Methods on Commodity Supercomputers; an Investigation into Bluff Body Far Wake Behaviour". *Proc. 3rd International Workshop on Vortex Flows and Related Numerical Methods*, Toulouse, August 1998.

[7] P.C. Macey, N.K. Allsopp, K. Takeda and D.A. Nicole. "Using Office PCs for the Vibroacoustic Analysis of Loudspeakers". S*ubmitted to Europar '99*, Toulouse, August 1999.

[8] M.A. Baker. "MPI on NT: The Current Status and Performance of the Available Environments". *Proc. EuroPVM/MPI 98*, September 1998

[9] W. Gropp, E. Lusk, N. Doss, and A. Skjellum. "A High-Performance Portable Implementation of the Message Passing Interface". *Parallel Computing*, vol. 22, pp 457-468, 1996

[10] M. Marinho and J.G. Silva. "WMPI Message Passing Interface for Win32 Clusters". Instituto de Engenharia de Coimbra, Portugal and Departamento de Engenharia Informatica, Universidade de Coimbra, Portugal. http://dsg.dei.uc.pt/wmpi/intro.html

[11] GENIAS Software GmbH. PaTENT MPI. http://www.genias.de/products/patent

[12] B. Protopopov. MPI for Windows NT. ftp://aurora.cs.msstate.edu/pub/mpi/Ntfiles/winMPICHpresent.ps , 1996.

[13] L.S. Hebert, W. Seefeld, A. Skjellum, C.D. Taylor and R. Dimitrov. "MPI for NT: Two Generations of Implementations and Experience with the Message Passing Interface for Clusters and SMP Environments". *Proc. 1998 International Conference on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, July 1998.

[14] M. Lauria and A. Chien. "MPI-FM: High-Performance MPI on Workstation Clusters". *Journal of Parallel and Distributed Computing*, vol. 40 no. 1, January 1997

[15] D. Bailey, T. Harris, W. Saphir, R. Wijngaart, A. Woo and M. Yarrow. "The NAS Parallel Benchmarks 2.0". *NAS Report number NAS-95-020*, December 1995.

[16] K. Takeda, N.K. Allsopp, J.C. Hardwick, P.C. Macey, D.A. Nicole, S.J. Cox and D.J. Lancaster, "An Assessment of MPI Environments for Windows NT", Dept. of Electronics and Computer Science Technical Report, University of Southampton, April 1999.

[17] P.C. Macey. "Advanced design tools for loudspeakers using vibroacoustic finite and boundary element models". *Proc. IOA.*, **17**, pt. 7, pp 3356-3367, 1995.

[18] Platform Computing. Load Sharing Facility. http://www.platform.com

[19] MPI Software Technology Inc. MPI/Pro. http://www.mpi-softtech.com

[20] The Symera Project. http://symera.ncsa.uiuc.edu/

[21] The CONDOR project. http://www.cs.wisc.edu/condor