HPC ON DEC ALPHAS AND WINDOWS NT

Denis Nicole, Kenji Takeda and Ivan Wolton

Southampton HPCI Centre
Department of Electronics and Computer Science
University of Southampton
Southampton SO17 1BJ

ABSTRACT

We have obtained a dedicated computational cluster of eight DEC Alpha systems interconnected by 100 Hz switched Ethernet and running Digital Visual FORTRAN on Windows NT. This is an 8 Gflop/s (peak) system with 2 Gbytes of memory. The total cost was under £50,000. We have just finished porting MPI onto this environment and are now able to run mainstream UK HPC codes such as ANGUS. We believe that our system is a highly cost-effective environment for the development and medium-scale execution of science and engineering codes; it currently represents the biggest single computational resource at Southampton.

We present our early experiences with this leading edge medium-scale resource and some early performance results for sequential, MPI and PVM FORTRAN codes.

INTRODUCTION

The consolidation of the microprocessor market has resulted in there now being little distinction between the PC and workstation markets. By taking advantage of the inherent economies of scale in processors, memory, switching technology and software it is possible to utilise commodity components to build supercomputer-level machines at low cost. The Beowolf project [1] has concentrated on using Intel-based machines running Linux to provide very cost-effective production machines for a number of applications.

We have recently purchased a dedicated computational cluster of DEC Alpha workstations. These compete on a node for node basis with systems from IBM and SGI/Cray for many scientific and engineering applications, but by using commodity components the cost is lower by a factor of at least three. By also utilising the Windows NT operating system further savings can be made. Our long term goal is to provide a full, remote access service on this HPC system.

High Performance Computing
Edited by R. J. Allan *et al.*, Kluwer Academic / Plenum Publishers, New York, 1999

SYSTEM CONFIGURATION

The exact configuration of the installed DEC Alpha cluster is given in Table 1. As this is a compute cluster only two 21" and two 15" monitors were purchased. The other four main compute nodes share a single monitor through a manual switchbox. The total system cost was £50,000.

The server node uses Linux to allow good cross-compatibility with a variety of UNIX systems via Samba. We have used Debian Linux as it was the most up-to-date common distribution. The DLT drive represents a long-term investment for data backup.

ADVANTAGES

There are several advantages in opting to buy a DEC Alpha Windows NT system over other similar commodity supercomputing solutions. The 500 MHz DEC Alpha AXP21164 processor offers at least twice the performance of a Pentium system at only 50% higher cost (at Q3 1997 prices). By utilising fewer more powerful nodes, communication overhead can be significantly reduced which is the critical bottleneck in commodity supercomputer systems. These processors compete against Intel in the NT server market and are therefore priced aggressively. Additionally, the standard motherboard architecture means that true commodity components can be used, such as ordinary SIMMs, PCI bus cards and EIDE disks. In step with Intel's processor roadmap, Digital are continuing to keep pace and the onset of Samsung manufactured Alpha chips with enhanced clock speeds means more competition in the marketplace.

While the hardware is a considerable cost, software costs can also be a major issue. In order to run the best FORTRAN compilers the choice of operating system is limited to Digital UNIX and Windows NT.

Digital UNIX is expensive and this is exacerbated by the need to use SCSI rather than the cheaper EIDE disks. Windows NT Workstation is considerably cheaper than Digital UNIX, and the excellent Digital FORTRAN compiler is available complete with IMSL libraries at significantly lower cost than the UNIX version. Digital recently licensed Microsoft Developer Studio for use with its FORTRAN compiler and so programmers can benefit from this powerful, user-friendly environment. Additionally most Windows x86 packages run out of the box under the FX32! emulation package [2].

A fundamental problem of any 32-bit operating system, such as Windows NT 4.0, is that the maximum addressable memory space is limited to 2 Gbytes. Linux has a memory limit of 3-4 Gbytes. However, Digital's alliance with Microsoft in developing 64-bit Windows NT 5.0 means that Alpha systems will be able to overcome this limit as soon as the software becomes available.

We are pursuing the long-term goal of delivering an effective remote and local parallel computing service directly under Windows NT. Windows NT is the wave of the future, whether we like it or not.... and it runs Microsoft Office.

Table 1. Configuration of DEC Alpha cluster

Eight nodes, each with
500 MHz Alpha 21164 processor
256 Mbyte RAM
2.5 Gbyte EIDE drive
Two additional 5Gbyte drives to support Windows NT 5.0 and Linux
Windows NT version 4 (service pack 3)
Server node
200 MHz Pentium
32 Mbyte RAM
4 x 5 Gbyte EIDE drive
30 Gbyte DLT backup
Debian Linux
Network connectivity
100 Mbit Ethernet
100 Mbit twelve port Ethernet switch
Compilers
Digital Visual FORTRAN
Visual C++ v4.1 (RISC, shortly to be upgraded to v5.0)

Uniprocessor Performance

For most small benchmarks the 500 MHz DEC 21164A Alpha is a 100 Mflop/s system. Linpack gives figures of 110 Mflop/s (201) and 97 Mflop/s (200). The Whetstone benchmark delivers 528926 kwhetstones per second. A performance breakdown running Livermore loops is given in Table 2. Enabling debugging which entails disabling compiler optimisations hurts performance badly, as shown in Table 3.

This level of performance is encouraging considering the price and is very respectable compared with what are traditionally regarded as "high-end" workstation systems such as those based on RS/6000, MIPS and UltraSPARC processors.

These benchmark figures translate into good real application performance. We have used the Alpha cluster to perform partitioning of a fifteen million element unstructured tetrahedral grid, which requires a two Gbyte memory region. Initially we performed this on one node of the Southampton IBM SP2 with 256 Mbytes of RAM and reconfigured to page off five SCSI disks simultaneously. This took nine hours to complete and required us to run in the overnight queue, and only on the one specifically configured node.

The same job took six hours to complete on one AlphaNT node. This used 256 Mbytes RAM and paged off a single EIDE drive. Setting up the necessary swapfile took just six mouse clicks and a reboot. We were therefore able to do eight partitioning jobs in parallel, overnight and without having to fight through any queues.

Table 2. Performance breakdown of 500 MHz DEC Alpha 21164 system running Livermore loops with full compiler optimisations (level 5).

Measurement	Mflop/s
Maximum rate	796.34
Quartile Q3	153.27
Average rate	141.33
Geometric mean	102.52
Median Q2	86.09
Harmonic mean	81.88
Quartile Q1	62.24
Minimum rate	29.76

Table 3. Performance breakdown of 500 MHz DEC Alpha 21164 system running Livermore loops with debugging enabled and no compiler optimisation.

Measurement	MFlop/s
Maximum rate	31.59
Quartile Q3	20.42
Average rate	15.11
Geometric mean	13.11
Median Q2	12.95
Harmonic mean	11.40
Quartile Q1	7.87
Minimum rate	4.63

DISADVANTAGES

There are many disadvantages in using Windows NT as it certainly is not as mature as UNIX. We can expect some improvements with version 5 which we are currently running under beta on a couple of machines. Other problems are being addressed at Southampton.

Filenames under Windows NT are not case-sensitive[1], ie: `prog.f = prog.F`, which can cause problems for some preprocessing makefiles. Resource leaks can occur, particularly in DLLs when processes are killed. NT is really only a one-at-a-time multi-user system as all users share, and can modify, the same drive map. It is only designed to support one interactive user at a time.

A significant area for improvement is in remote access. Remote logins using the Microsoft `telnet` daemon (in beta) are not very stable. Currently the only facility for remote windows graphical applications is WinVNC [3]. While initial tests demonstrate that this allows full graphical remote access from Win32 and X clients and performs well under Windows 95, its performance on the Alpha is unacceptably slow at present. This is being addressed; such remote access is also a feature touted in Windows NT version 5. We have also successfully tested a freely available `rlogin` daemon. `Edlin` is, however, the only editor we can run remotely through `telnet` presently. We are tackling all of these problems as they are crucial in being able to offer a remote, multi-user, high performance computing service.

In terms of networking, NT does not seamlessly integrate into a UNIX environment. We use a Samba server to bridge this gap but individual file security is not propagated through Samba, NISGina provides a hack to support NIS logins but this has not been tested on Alphas at present, and domain logins currently require NT Server software.

MPI Performance

At present the only implementation of MPI available for Windows NT running on DEC Alphas is that developed by Mississippi State University based on MPICH and known as WinMPICH [4]. We have ported this fully to Digital Visual FORTRAN and we believe we are the only group running FORTRAN MPI on Alpha NT. WinMPICH is still in Beta (as of January 1998). While the performance between MPI processes on a single machine is reasonable for a layered OS-based implementation, its performance is disappointing between machines as can be seen in Table 4. We are working to enhance the performance of MPI between computers as a matter of urgency both by layering over Windows Named Pipes and by implementing our own protocols at the NDIS level.

[1] Except in the POSIX box which is otherwise almost useless.

Table 4. COMMS1 MPI benchmark performance.

Measurement	Between two processes	Between two machines
rinf (kbyte/s)	10800	59.2
nhalf (byte)	9070	130
Startup (μs)	842	2200

MPI under Windows NT is seriously immature, not only in terms of its inter-processor performance. As it was designed primarily for shared memory, when operating across machines it runs under Administrator accounts (equivalent to UNIX `superuser`) with full system privileges. It may also leave dead processes hanging when not exiting properly; this also occurs with some UNIX-based MPI implementations.

PVM Performance

We have also acquired a public domain implementation of PVM [5] and rlogin/rlogind for the DEC Alphas. We have not yet ported PVM for Digital Visual FORTRAN. As can be seen from Figures 1 and 2, the performance of PVM is much better than WinMPICH between machines.

This port of PVM for Windows NT will be included in the general distribution of PVM version 3.4. However it still is far from perfect. It fails to redirect i/o and requires manual starting of pvmd3 daemons on remote processors.

HPC Performance

Initial tests of the real application performance of this system have been carried out. The benchmark version of the combustion code ANGUS was compiled to run under Windows NT with only a single modification to the main source code; replacing the UNIX "/dev/null" with "NUL" for dummy file output. ANGUS is a finite-difference code which uses a regular grid and straightforward domain decomposition. The most intensive part of the program is solving the Poisson equation for the pressure. A number of solution algorithms have been implemented and the multigrid solver was used in this test. Performance for a 40x40x40 grid with 2x2x1 processor decomposition is given in Table 5 for running on a single DEC Alpha and on two machines. For comparison T3D performance using MPI is quoted. Note that in the production code Cray-specific SHMEM libraries replace the MPI calls to optimise performance on Cray T3D and T3E machines.

Table 5. Performance of ANGUS code on Alpha NT cluster, with T3D performance for comparison.

Configuration	time per iteration (s/iteration)	comms time (s)
One CPU, one process	20	2.4
One CPU, four processes	38	21.3
Two CPUs, four processes	285	263.0
T3D using MPI (four nodes)	61.6 (nodesecs/iteration)	14.0 (nodesecs)

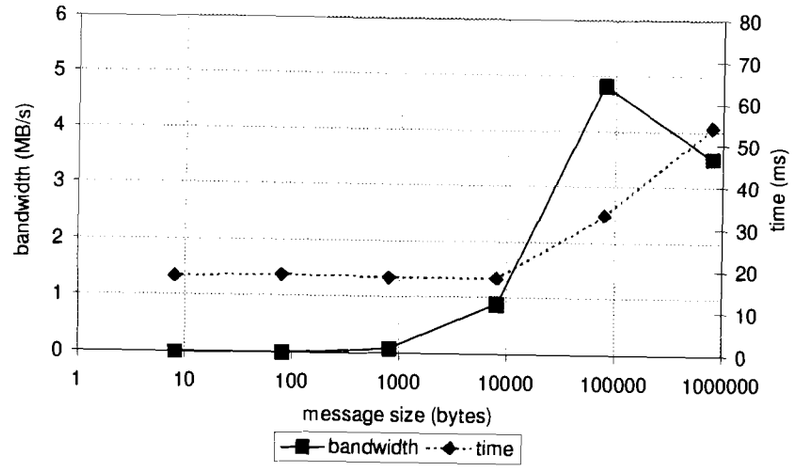

Figure 1. bwtest PVM benchmark performance on a single machine between two processes.

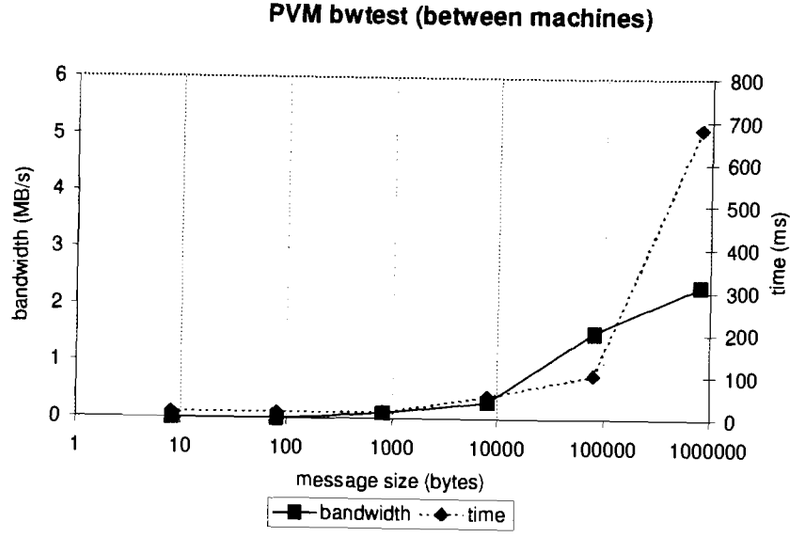

Figure 2. bwtest PVM benchmark performance between two machines (note timescale differs from Figure 1).

CONCLUSIONS

In this paper we present our early experiences of using DEC Alpha workstations running Windows NT for high performance scientific computing. Using Digital Visual FORTRAN, this is a good development environment and offers reasonable shared memory MPI and PVM performance for testing purposes. However, parallel programming between machines is at the bleeding edge; we still have a lot of work to do before this becomes a suitable platform for real users. For production work a combination of Digital UNIX compilation nodes and Linux compute nodes can provide cost effective, medium-scale commodity supercomputing today.

ACKNOWLEDGMENTS

The authors would like to thank Stewart Cant (Cambridge) and David Emerson (Daresbury) for the ANGUS code, and Ken Morgan (Swansea) for the unstructured grid and related software.

REFERENCES

1. The Beowolf Project, `http://cesdis.gsfc.nasa.gov/beowolf`
2. R.J. Hooway and M.A. Herdeg, DIGITAL FX!32: Combining emulation and binary translation, *Digital Technical Journal*, 9(1) (1997).
3. *WinVNC* from Olivetti & Oracle Research Lab, `http://www.orl.co.uk/vnc`
4. A. Skjellum, B. Protopopov, S. Hebert, P.J. Brennan and W. Seefeld, *MPI on Windows NT. 0.92 Beta release* (1997). Currently available at: `http://www.erc.msstate.edu/mpi/mpiNT.html`
5. *PVM for Windows NT*, `http://www.epm.ornl.gov/pvm/NTport.html`
6. D. Emerson, D.A. Nicole and K. Takeda, An Evaluation of Cost Effective Parallel Computers for CFD, to be presented at the *10th International Conference on Parallel CFD*, Taiwan (May 1998).