

# An Application Performance Comparison of HPCx to EPCC's Blue Gene/L Service

A. Gray, L. Smith, J. Hein, J.M. Bull, F. Reid,  
O. Kenway, B. Dobrzeleck, A. Trew  
*EPCC, The University of Edinburgh, James Clerk Maxwell Building,  
Mayfield Road, Edinburgh, EH9 3JZ, UK*

November 28, 2005

## **Abstract**

The results of benchmarking several popular applications on EPCC's Blue Gene/L service, BlueSky, are presented and compared with results from running on HPCx, a 1600 processor p690+ system (which is currently the main UK academic super-computing resource). The HPCx to BlueSky performance ratio is generally smaller than that expected from the clock frequencies and floating point unit architecture of the processors, and this is likely due to better memory bandwidth on BlueSky. The codes generally scale well on HPCx and on BlueSky. When considering other cost factors, such as hardware cost, power consumption and floor space, BlueSky has a large advantage over HPCx.

**This is a Technical Report from the HPCx Consortium**

**© HPCx UoE Ltd 2005**

Neither HPCx UoE Ltd nor its members separately accept any responsibility for loss or damage from the use of information contained in any of their reports or in any communication about their tests or investigations.

# 1 Introduction

BlueSky, an IBM Blue Gene/L machine, has recently been added to EPCC's super-computing facilities. This paper reports on the performance of a selection of popular scientific applications on BlueSky, and compares to performance results from HPCx (an IBM p690+ cluster which is currently the main UK academic facility).

The applications benchmarked in this report are CASTEP [1], DL\_POLY [2], H2MOL [3], PCHAN [4], MDCASK [5], LAMMPS [6], NAMD [7], and LUDWIG [8].

## 2 Architectures of BlueSky and HPCx

The IBM Blue Gene/L is designed to be an extremely scalable system. The low clock rate of its processors (700MHz) allows many to be linked while keeping power consumption and floor space down. Each node features two processing cores, 3 cache levels (although level 2 cache is really just a memory buffer) and high speed interconnection networks all integrated onto a single ASIC. This enables extremely fast communication between MPI tasks [9].

BlueSky is a single rack IBM Blue Gene/L machine featuring 1024 nodes, i.e. 2048 processors. The processing cores have the PowerPC440 architecture and operate at 700 MHz. Each node has 512MB memory. The machine can operate in either Coprocessor (CO) mode, where only one processor in each node is utilised for computation and the other is reserved for communication, or Virtual-Node (VN) mode where all processors are able to act as virtual nodes and take care of both computation and communication. In VN mode, the resources of the node must be shared between the 2 processors, i.e. each process has access to only half the memory.

The HPCx system features 50 IBM p690+ compute nodes (frames), each containing 32 IBM Power4 64-bit RISC processors. Each processor operates at 1.7GHz.

The 32 processors in a compute node are packaged into 4 Multi-Chip Modules (MCMs), each containing 8 processors. Within a node, MCMs communicate via shared memory. Communication between nodes is provided by an IBM High Performance Switch (HPS). A total of 4 inter-node links are available: each node has 2 network adapters, each with 2 links.

The clock rate of an HPCx processor is 2.4 times higher than that of a BlueSky processor. Moreover, each HPCx processor has 2 independent floating point units<sup>1</sup>. Therefore, it is naively expected that HPCx should outperform BlueSky by a factor of 4.8 when comparing similar processor counts. It will be seen in this paper that BlueSky is generally performing much better than this expectation, and this could be explained by the fact that the observed main memory bandwidths of the BlueSky and HPCx processors are around 1.4GB/s and 2GB/s respectively [10], i.e. the BlueSky processors have much faster main memory access relative to their clock frequency.

It should be noted that the Blue Gene/L has been designed to scale to many more processors than clustered SMP systems such as HPCx, and is considerably cheaper in terms of hardware monetary cost, power consumption and floor space.

---

<sup>1</sup>The Blue Gene/L processors also have 2 floating point units, the so called *double hummer*, but these are not independent and currently the compilers have not been able to utilise the second unit to any observable degree for many real applications (see, e.g. [11] and Figure 13 in Appendix A)

Keeping the above points in mind, this report compares the performance of several applications on BlueSky to that on HPCx and examines the scalability to large numbers of processors.

### 3 Benchmarking of Applications

This section compares benchmarking results from applications run on BlueSky to those run on HPCx. The porting of applications to BlueSky is discussed in Appendix A.

Remember that in BlueSky CO Mode, the number of processors is equal to the number of nodes while in VN mode, the number of processors is double the number of nodes used. In this study, we compare BlueSky nodes with HPCx processors.

Unless otherwise stated, BlueSky should be assumed to be operating in CO mode. Also note that these results represent the current performance of these applications on BlueSky: further improvements may be possible.

#### 3.1 CASTEP

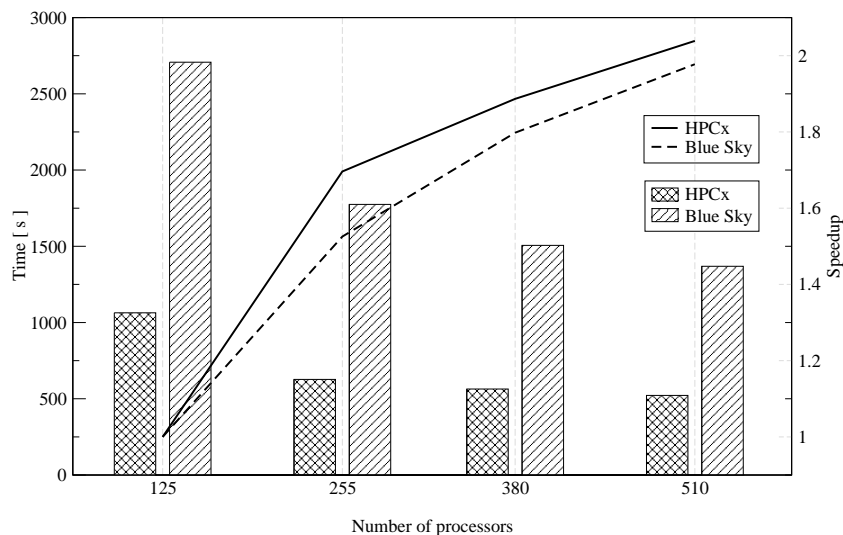


Figure 1: Comparison of the performance of CASTEP Al<sub>2</sub>O<sub>3</sub> simulations on BlueSky to those on HPCx: the dependence of the time taken (bars) and the speedup (lines) on the number of processors is represented for BlueSky as dashed lines and lined bars and for HPCx as solid lines and crossed bars.

The CASTEP software package can be used to perform molecular dynamics simulations and provide an atomic-level description (including information regarding energies, forces, and stresses, and calculations of optimum geometries, structures and spectra) of a wide range of materials and molecules [1].

Here results are given for 2 systems: Al<sub>2</sub>O<sub>3</sub>, a 120 atom slab cell of aluminium oxide sampled with 5 k-points, and TiN, a 33 atom cell of titanium nitride with hydrogen defect sampled with 8 k-points [11].

Figure 1 compares the performance of an Al<sub>2</sub>O<sub>3</sub> CASTEP simulation on BlueSky to that on HPCx. It is seen from the computational time data that the HPCx runs were

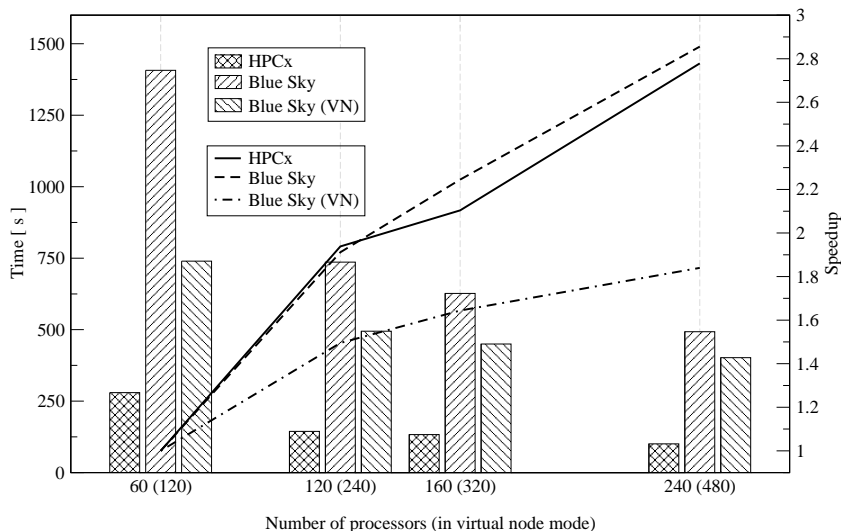


Figure 2: As in Figure 1, but for TiN simulations. Also shown is the performance of BlueSky in VN mode.

around 2.7 times faster than on BlueSky when comparing results on comparable numbers of processors. This is much lower than the expected factor of 4.8 (which assumes that the double hummer on BlueSky is showing no performance improvement). This could be due to better memory bandwidth on BlueSky. Note that better performance may be observed on HPCx for processor counts which are better suited to the architecture.

The speedup on  $n$  processors is defined here as the time taken on 125 nodes divided by the time taken on  $n$  nodes. From this data, it is seen that this simulation displays better scaling on HPCx than on BlueSky, although the gap closes as the number of processors increases. It would be interesting to compare data for higher processor counts.

Figure 2 shows results for a TiN simulation (where here the speedup is defined as the time taken on 60 nodes divided by the time taken on  $n$  nodes.). Here, HPCx is seen to be faster by a factor of 5. This factor is higher than expected, and is likely because the problem fits into the HPCx Level 2 cache (which is 1.5Mbyte shared between 2 processors) but not into that on BlueSky (which is 2Kbyte per processor). The simulation is seen to scale better on BlueSky.

It is also shown that significantly better performance is observed from using BlueSky VN mode. The fact that the code does not scale so well in VN mode can be explained, if communication is dominant at large numbers of processors, by a saturation of the communications bandwidth. Therefore the higher processing capability of VN mode within the node becomes less advantageous as the number of processors increases.

### 3.2 DL\_POLY

DL\_POLY is a classical molecular dynamics code which can be used to simulate systems with very large numbers of atoms [2]. Here, DL\_POLY3, which is parallelised by domain decomposition and is suitable for large numbers of processors (as opposed to DL\_POLY2 which uses a Replicated Data strategy and is suitable for of order 100 processors), was run for both a 216000 ion NaCl (sodium chloride) system and a system of gramicidin

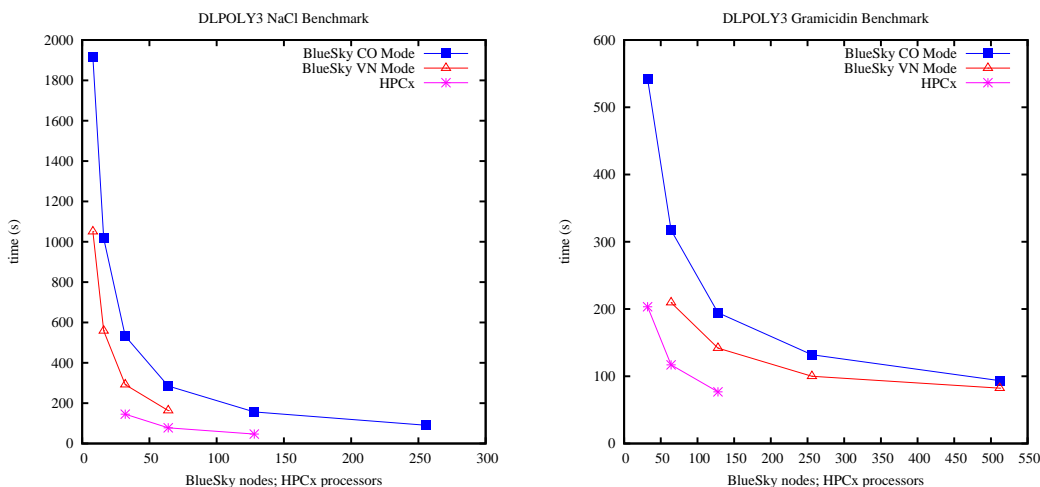


Figure 3: The time taken for NaCl (left) and Gramicidin (right) DLPOLY3 benchmarks. For BlueSky, data are plotted against the number of nodes where closed squares and open triangles represent CO and VN modes respectively. HPCx data are represented as bursts and are plotted against the number of processors.

molecules in water with a total of 792,960 atoms.

The time taken by the NaCl and gramicidin simulations are shown in Figure 3. Results are plotted against the number of nodes for BlueSky and the number of processors for HPCx. Some BlueSky VN runs were not possible due to memory limitations.

Performance is better on HPCx than BlueSky (CO mode) by factors of around 3.3 and 2.5 for the NaCl and Gramicidin benchmarks respectively. Both of these factors are lower than the expected 4.8, probably due to the better memory bandwidth on BlueSky. The performance gap is seen to close as the number of processors increases, indicating better scaling behaviour on BlueSky. Performance gains of a factor of 1.9 (on 60 nodes) to 1.2 (on 240 nodes) are observed by using BlueSky VN mode.

### 3.3 H2MOL

H2MOL calculates the redistribution of energy between electrons and nuclei when hydrogen molecules are heated by short intense laser pulses [3]. Such calculations can be compared with experiment to help to develop the understanding of such molecular behaviour in general.

Figure 4 plots the times taken by the H2MOL benchmark.<sup>2</sup> Only CO mode was used on BlueSky, due to memory limitations.

For this code, the number of grid points is directly proportional to the number of processors, so ideal scaling would result in a constant simulation time for different numbers of processors. The unusual low processor results on HPCx are explained in [12].

Good scaling is observed on both machines. The HPCx times are seen to be around a factor of 1.7 lower. Again this is much below the expected factor of 4.8 (remembering

<sup>2</sup>Note that for this benchmark 13 grid points were used in the Z direction to allow the problem to fit into BlueSky's memory. Note also that the writing of intermediate states was switched off, as currently BlueSky has an extremely inefficient IO setup.

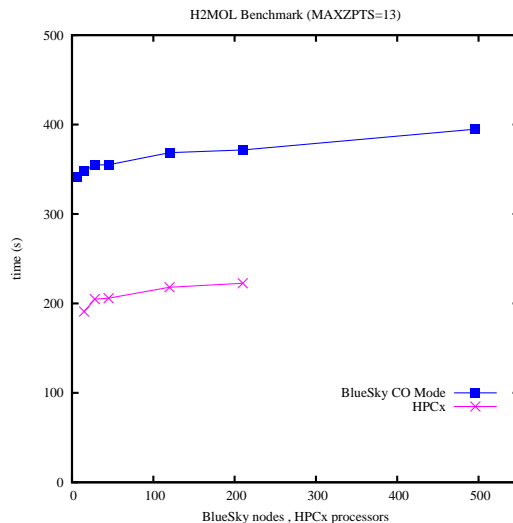


Figure 4: The time taken for the H2MOL simulation plotted against the number of nodes for runs on BlueSky (squares) and number of processors for HPCx (crosses).

that this factor is twice the clock frequency ratio because the double hammer on BlueSky is assumed to show no performance improvement) and likely due to relatively better memory bandwidth on BlueSky.

### 3.4 PCHAN

The PCHAN code is designed to simulate the flowing of fluids to study turbulence [4]. This is achieved by solving the governing equations of motion from first principles.

From Figure 5, the PCHAN T2 benchmark scales remarkably well on HPCx, and performs around 2-3 times faster than on BlueSky, again lower than the naively expected factor of 4.8. BlueSky VN mode is once more seen to give better performance than CO mode.

### 3.5 MDCASK

MDCASK is a molecular dynamics code which was originally developed to study radiation damage in metals [5]. It operates by calculating the energies of and forces on, and determining the motions of, the atoms in the system which is characterised by specific interatomic potentials and boundary conditions.

Figure 6 plots the total CPU time, which is the time of the run multiplied by the number of processors, against the number of processors for the MDCASK Ti benchmark. For perfect scaling, a flat straight line would be seen, and the near flatness of the data up to 512 processors demonstrates that this simulation scales well on BlueSky, slightly better than on HPCx. The HPCx runs were around 4 times faster than on BlueSky.

### 3.6 LAMMPS

LAMMPS is a molecular dynamics package which solves classical physics equations and is able to simulate a wide range of materials including atomic, molecular, metallic and

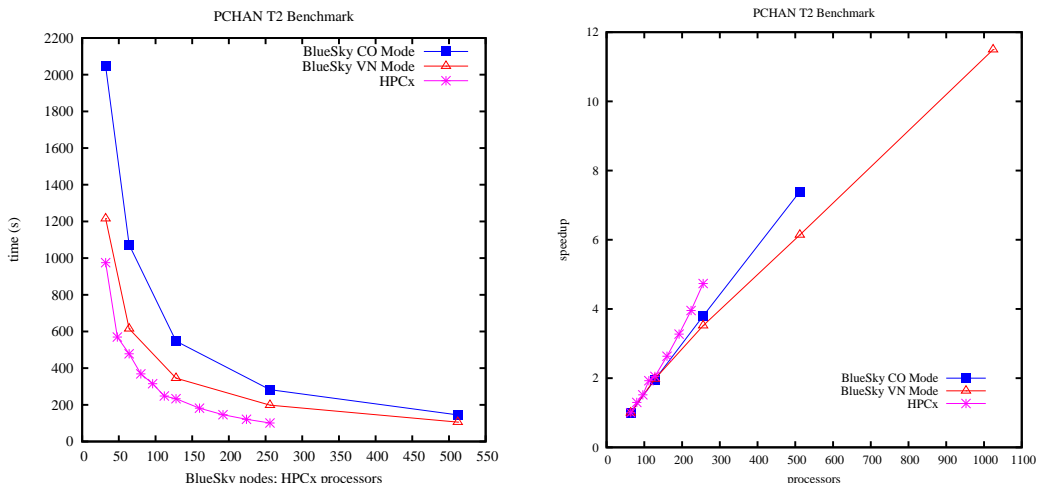


Figure 5: The dependence of the time taken on the number of BlueSky nodes or HPCx processors (left) and speedup on the number of processors (right) for the PCHAN T2 benchmark. Closed squares and open triangles denote BlueSky CO and VN operational modes respectively. Burst denote results from HPCx.

hybrid systems [6]. Here a clay polymer system, with 1012736 atoms, is benchmarked using LAMMPS 2001.

The time taken by the simulation is plotted against the number of processors on the left of Figure 7. The simulation is around 4.2 times faster on HPCx at 128 processors but this factor is seen to decrease at high processor counts demonstrating the good scaling behaviour on BlueSky. This is emphasised on the right hand side of Figure 7 by the BlueSky speedup data, where here speedup on  $n$  processors is defined as the time taken on 64 processors divided by the time taken on  $n$  processors.

The above LAMMPS BlueSky results were gained using CO mode. Figure 8, which plots the simulation time against the number of BlueSky *nodes*, shows that significant performance improvements can be gained by running in VN mode.

### 3.7 NAMD

The NAMD molecular dynamics code is designed to simulate biomolecular systems such as proteins [7].

Results are given for the ApoA1 benchmark, which involves 92000 atoms. The time taken by the simulation is plotted against the number of BlueSky nodes or HPCx processors on the left of Figure 9. Similarly to the LAMMPS benchmark results, the faster HPCx processors give it the performance edge of around a factor of 4.2. The right hand side of Figure 9 (where the speedup on  $n$  processors is defined as the time taken on 8 processors divided by the time taken on  $n$  processors) shows that in this case scaling is better on HPCx. In accordance with those other applications for which we have the relevant data, it is seen that significant performance improvements can be gained by running in VN mode.

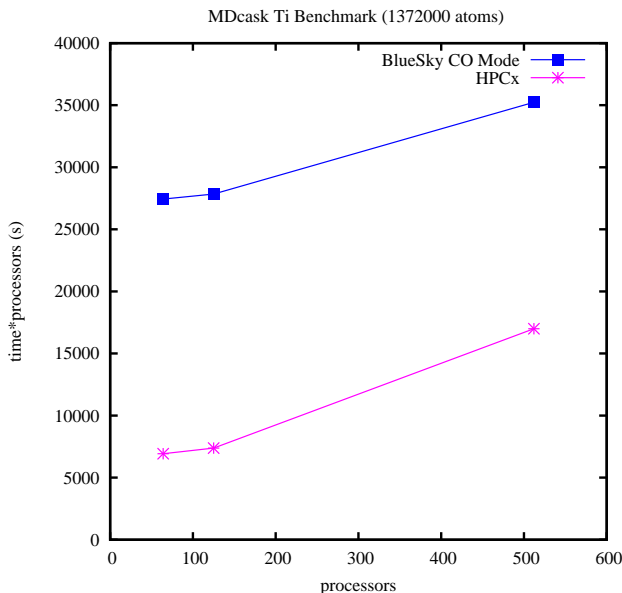


Figure 6: The dependence of total CPU time of an MDCASK simulation on the number of processors.

### 3.8 LUDWIG

The LUDWIG code uses Lattice-Boltzmann models to enable the simulation of the hydrodynamics of complex fluids in 3-D [8].

Figure 10 shows, for 64 and 256 processor runs, the times taken on HPCx and BlueSky (both CO and VN modes) for a  $384^3$  model size benchmark. HPCx performs only around 1.4 times faster than BlueSky CO mode, but BlueSky VN mode actually outperforms HPCx on a per chip basis! This is likely due to the fact that the run on HPCx is hampered by TLB misses caused by the many strided accesses to main memory in the key LUDWIG routines.

### 3.9 Summary

In general, the BlueSky application benchmark results have been seen to perform better, relative to those from HPCx, than the expectation from the clock frequencies of the processors, and this is likely due to better memory bandwidth on BlueSky. This is shown in Figure 11 which plots the BlueSky:HPCx benchmark time ratio for each application, where the number of nodes/processors was chosen to be as high as possible dependent on availability of data.

Such a direct comparison, however, does not take into account the fact that Blue Gene/L is much cheaper per processor in terms of hardware cost, power consumption and floor space<sup>3</sup>. Figure 12 takes into account each of the above cost factors demonstrating that BlueSky has a clear advantage over HPCx (note that the ratio here is HPCx:BlueSky). For example, a CASTEP 125 processor  $Al_2O_3$  run was seen to be a

<sup>3</sup>The hardware cost [13], power consumption [14] [15] and floor space [16] [17] per processor HPCx:BlueSky ratios were taken to be 19.2, 11.4 and 38.2 respectively

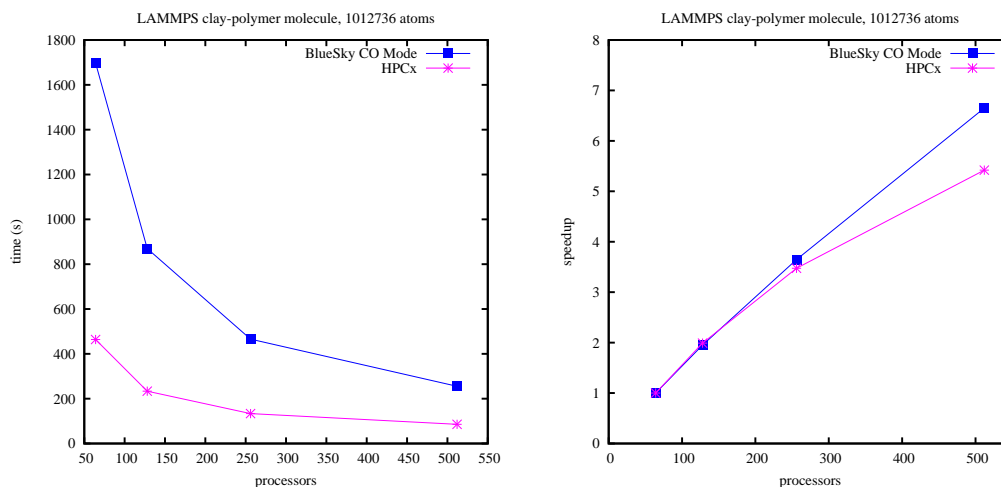


Figure 7: The dependence of the time taken (left) and speedup (right) for a LAMMPS simulation on the number of processors. Closed squares and bursts denote Blue Gene/L CO mode and HPCx results respectively.

factor of 7.5 more expensive on HPCx when defining the cost as the time taken by the run multiplied by the hardware monetary cost.

### 3.10 Conclusions

Benchmark results for a number of popular applications on BlueSky and HPCx have been presented and compared. The advantage due the superior clock frequencies and floating point architectures of the HPCx processors was found to be less than expected for several of the codes, and it is likely that this is due to the relatively good memory bandwidth of the processors on BlueSky. Where possible, the use of BlueSky VN mode has been seen to further improve performance.

### 3.11 Acknowledgements

I wish to thank Mike Ashworth and Kevin Stratford for providing results for this paper.

## References

- [1] M D Segall *et al* 2002, First-principles simulation: ideas, illustrations and the CASTEP code. *J. Phys.: Condens. Matter* **14** 2717-2744; <http://www.tcm.phy.cam.ac.uk/castep/>
- [2] Smith, W. & Forester, T. 1996, DL\_POLY: A General Purpose Parallel Molecular Dynamics Simulation Package. *J. Molec. Graphics* **14**, 136; Smith, W., Yong, C. & Rodger, M. 2002, DL\_POLY: Applications to Molecular Simulation. *Molecular Simulation* **28**, 385; Todorov, I. & Smith, W. 2004, DL\_POLY\_3: The CCP5 National UK Code for Molecular Dynamics Simulations. *Phil. Trans. R. Soc. Lond. A* **362**, 1835; [http://www.cse.clrc.ac.uk/msi/software/DL\\_POLY/](http://www.cse.clrc.ac.uk/msi/software/DL_POLY/)

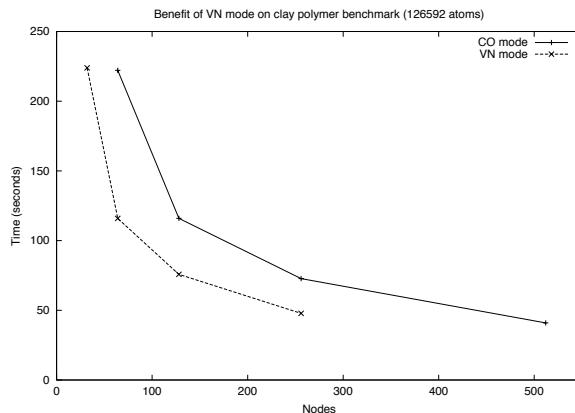


Figure 8: The dependence of the time taken (left) and speedup (right) for a LAMMPS simulation on the number of *nodes*. Vertical lines and crosses denote Blue Gene/L CO and VN operational modes respectively.

- [3] D. Dundas, K.J. Meharg, J.F. McCann and K.T. Taylor, 2003, Dissociative ionization of molecules in intense laser fields *Eur. Phys. J. D* **26**, 51-57;  
<http://www.hpcx.ac.uk/research/atomic/h2mol.html>
- [4] PCHAN Webpage,  
<http://www.cse.clrc.ac.uk/arc/pchan.shtml>
- [5] MDCASK README  
<http://www.llnl.gov/asci/platforms/purple/rfp/benchmarks/limited/mdcask/mdcask.readme.html>
- [6] Plimpton, S. 1995, Fast Parallel Algorithms for Short-Range Molecular Dynamics. *J. Comp. Phys.* **117**, 1–19;  
<http://www.cs.sandia.gov/~sjplimp/lammps.html>
- [7] Phillips, J., Zheng, G., Kumar S. & Kalé L. 2002, NAMD Biomolecular Simulation on Thousands of Processors. In *Proceedings of the SC2002 Conference*. IEEE Press;  
<http://www.ks.uiuc.edu/Research/namd/>
- [8] Ludwig - A general purpose Lattice-Boltzmann code on the Cray T3E  
<http://citeseer.ist.psu.edu/411388.html>
- [9] An Overview of the BlueGene/L Supercomputer, IBM and LLNL,  
<http://www.llnl.gov/asci/platforms/bluegenel/sc2002-pap207.pdf>
- [10] J. Hein 2005, Experiences on the Edinburgh Blue Gene System, available at  
<http://www.epcc.ed.ac.uk/BGworkshop/programme.html>
- [11] Bartosz Dobrzeleck, Portability and Performance of CASTEP Code on eServer Blue Gene, MSc Thesis available at  
<http://www.epcc.ed.ac.uk/msc/dissertations-0405.htm>

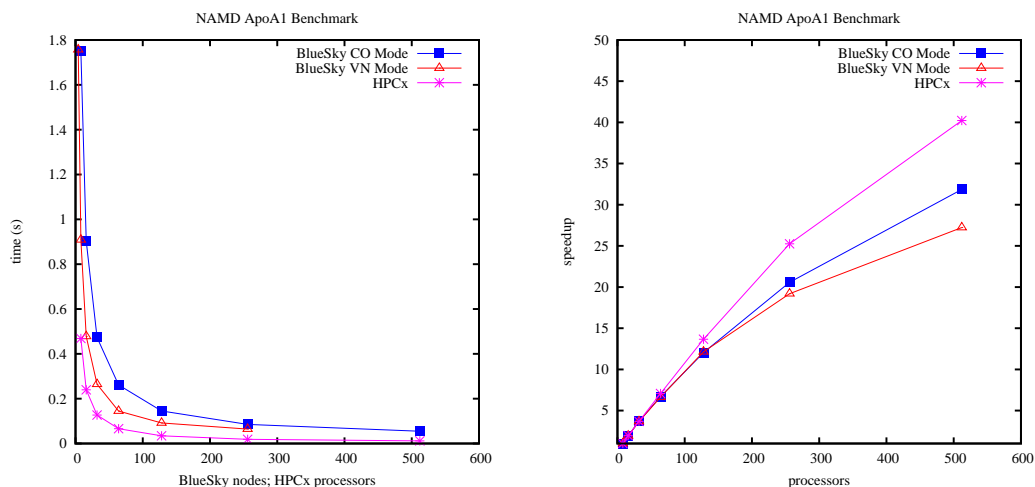


Figure 9: The dependence of the time taken on the number of BlueSky nodes or HPCx processors (left) and speedup on the number of processors (right) for the NAMD ApoA1 benchmark. Closed squares and open triangles denote BlueSky CO and VN operational modes respectively. Burst denote results from HPCx.

[12] Lorna Smith, Mark Bull, Andrew Sunderland, Profiling H2MOL on an IBM p690+ Cluster, *HPCx Technical Report 0413*, [http://www.hpcx.ac.uk/research/hpc/technical\\_reports/HPCxTR0413.pdf](http://www.hpcx.ac.uk/research/hpc/technical_reports/HPCxTR0413.pdf)

[13] Derrick Byford, IBM, private communication

[14] <http://www.ctwatch.org/quarterly/print.php?p=13>

[15] HPCx Systems Administration staff, private communication.

[16] <http://www.linuxhpc.org/stories.php?story=04/11/08/6743151>

[17] [http://www-03.ibm.com/servers/eserver/pseries/hardware/highend/p690\\_specs.html](http://www-03.ibm.com/servers/eserver/pseries/hardware/highend/p690_specs.html)

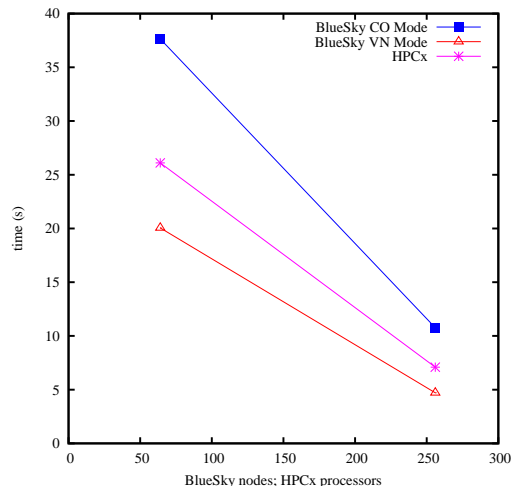


Figure 10: The dependence of the time taken on the number of BlueSky nodes or HPCx processors for a  $384^3$  LUDWIG benchmark. Closed squares and open triangles denote BlueSky CO and VN operational modes respectively, and bursts denote results from HPCx.

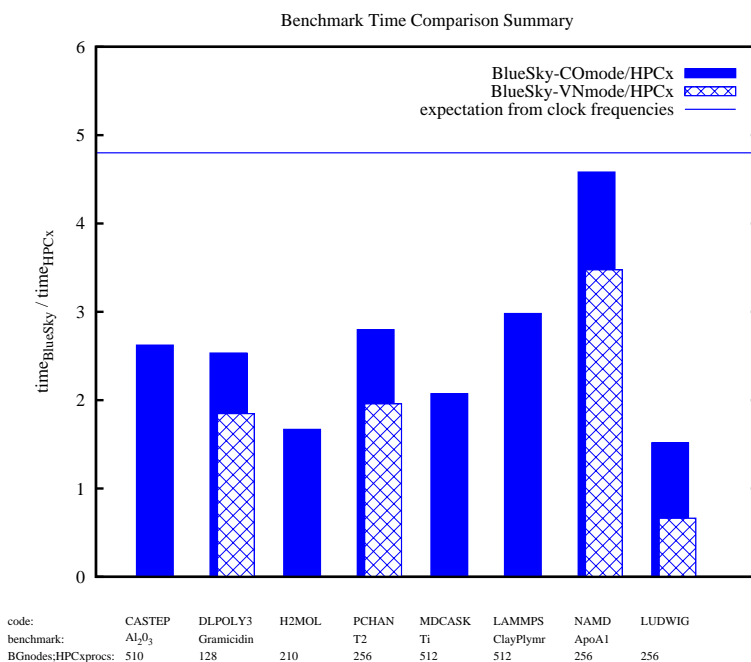


Figure 11: The ratio of benchmark time on BlueSky to HPCx. Results are shown for each application. Solid and patterned bars correspond to BlueSky CO and VN modes respectively. The line corresponds to the naive expectation from the clock frequencies of the processors (assuming that use of the double hummer has no performance advantage on BlueSky).

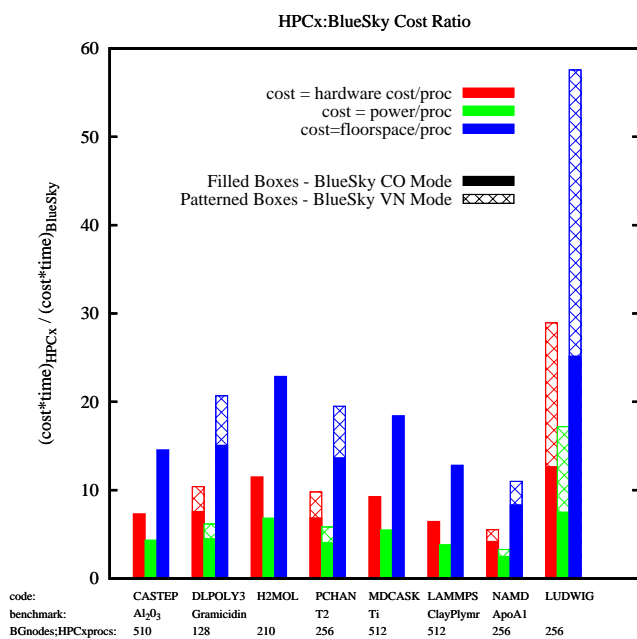


Figure 12: The ratio of the real cost of applications on HPCx to that on BlueSky (CO mode). Shown are results incorporating hardware cost (red), power consumption (green) and floor space (blue). Solid and patterned bars correspond to BlueSky CO and VN modes respectively.

## A Porting Applications

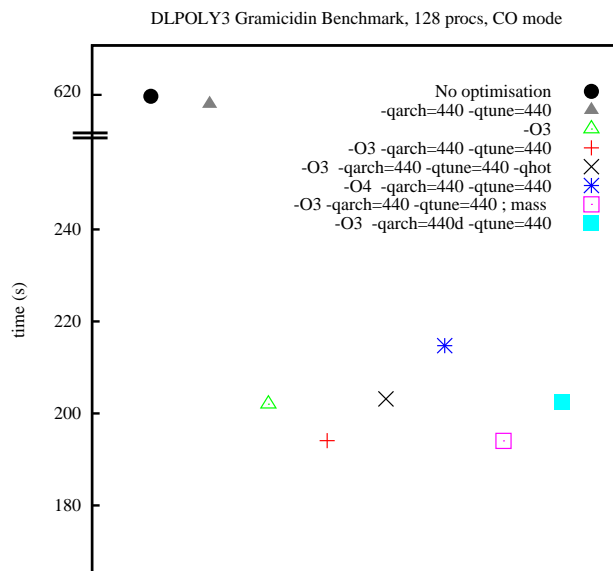


Figure 13: Comparison of different optimisation options on a DL\_POLY benchmark. With `-qarch=440d` the double hummer is utilised [9] and 'mass' means that the mass library has been used.

Some of the issues encountered in the porting of applications to BlueSky are documented in this appendix.

In general, the `mpixl*` compilers (which are actually wrapper scripts which use the `xl*` compilers and link to the `mpi` libraries) were used along with the optimisation flags

```
-O3 -qarch=440 -qtune=440
```

as these were seen to give the best performance, as is shown for DL\_POLY in Figure 13.

Below, some application specific issues are documented.

### A.1 DL\_POLY

The Fortran files are pre-processed by the C pre-processor `mpixlc -P`. At this stage, any comment lines which had an odd number of apostrophes (') raised the error

```
"coul2.f", line 8.59: 1506-209 (S) Character constants must  
end before the end of a line.
```

This was resolved by removing (or pairing) such apostrophes.

A conflict between the DL\_POLY `makefile` and the `mpixlf` script was discovered. The `makefile` sets the `FFLAGS` variable as

```
FFLAGS="-c -O3 -qarch=440 -qtune=440"
```

The same `FFLAGS` variable is used in the linking stage of the `mpixlf` script, and the `-c` argument caused no linking to be done, i.e. the application appeared to compile but no executable was created. This issue was resolved by using an amended `mpixlf` script which removed the `-c` argument at the linking stage. This amended script has been installed as

```
/bgl/local/bin/mpixlf_dlpoly
```

## A.2 H2MOL

The file `MPI_global.F` in the source code sub-directory

```
MACHINE/IBM/comms/MPI/MPI_global.F
```

was found to contain C style references to `MPI_DOUBLE` instead of the Fortran equivalent `MPI_DOUBLE_PRECISION`. These were fixed to avoid errors in compilation.

The same C pre-processor problem as for `DL_POLY` was encountered. In this case, many files contained problematic comment lines. However, it was found that there were actually no pre-processor directives in the source code, so the pre-processing stage was simply bypassed.

## A.3 PCHAN

The `Makefile` contained the rule

```
.F.f:  
$(CPP) $(CPPFLAGS) $(OPTIONS) $(OPTIONS2) $*.F > $*.f
```

However on BlueSky, with `mpixlc -P`, output is written to `<file>.i` instead of `stdout`, so `<file>.f` is empty. This was resolved by changing the rule to

```
.F.f:  
$(CPP) $(CPPFLAGS) $(OPTIONS) $(OPTIONS2) $*.F  
mv $*.i $*.f
```

The above `OPTIONS` variables were left unaltered from compilation on HPCx, including a variable `-D_HPCX_`. It was found that removing this resulted in a compilation error.