AWE HPC Benchmark, 2005

Ron Bell, Neil Munday, and **Steve Hudson,** *AWE, Aldermaston, UK.*

ABSTRACT: *In December, 2005, the UK's Atomic Weapons Establishment (AWE) placed an order for a 3936-node Cray XT3, each node consisting of a 2.6 GHz dual-core Opteron chip. This paper describes the design and evaluation of the benchmark used during the procurement of this supercomputer and presents comparative results between some of the contending vendors.*

KEYWORDS: XT3, MPP, AWE, Benchmark

1. Introduction

In December, 2005, the UK Atomic Weapons Establishment (AWE), placed an order with Cray for a 3936 node XT3, each node to contain a 2.6 GHz dual-core Opteron chip. When fully operational (planned for late 2006), the 7872 processing elements (PEs) of this machine will deliver over 40 Tflops peak and it is expected to be the most powerful supercomputer in the UK.

As measured by the benchmark used in conjunction with the procurement, the XT3 will deliver over 20 times the throughput of AWE's existing supercomputer running unmodified applications. During the benchmark, Cray performed extensive source-code tuning of the major applications and the XT3 running tuned benchmark codes should deliver almost 27 times the throughput of the present machine running untuned codes.

Potential vendors were given versions of the benchmark codes as early as late 2004 and the full benchmark was supplied at the beginning of July, 2005 to accompany AWE's Invitation to Tender. Bids and benchmark results were received on 31st August, 2005 from six vendors: Bull, Cray, Dell, IBM, Linux Networx (LNXI), and SGI.

This paper describes the design and evaluation of the benchmark.

2. Benchmark Objectives and Design

AWE's existing HPC facility is an IBM SP with 120 16-way POWER3 nodes running at 375 MHz connected by a Colony switch. There are 1920 PEs, of which 1856 are available for user computation. This machine is named *Blue Oak*. Capacity requirements for the new machine are expressed relative to Blue Oak as measured by the benchmark, not in terms of peak Tflops.

The benchmark design was underpinned by three major objectives:

1. It should represent codes from the whole user community,
2. It should measure both capacity (throughput) and capability (parallel scalability at high PE counts), and
3. It should include a throughput benchmark requiring the running of a concurrent mix of jobs representing the expected realistic workload.

These are discussed in the three sub-sections which follow.

2.1 User Requirements

There are three main groups of High Performance Computing (HPC) users at AWE:

(i) Physicists
(ii) Engineers, and

(iii) Material Scientists.

All three groups carried out a User Requirement analysis of the workload expected to be run on the new machine when it is in full production in 2006/2007. This led to a total capacity requirement up to 25 times Blue Oak.

Physics codes

The Physics codes run at AWE are almost all written in-house using FORTRAN 90 and MPI. In addition, the new supercomputer will be used largely for new versions of codes that are currently under development – for example a move to 3D modelling from 2D.

The Physicists' User Requirement analysis covered the nature of the workload as well as the amount of supercomputing capacity required to meet it. For the benchmark, a series of codes were provided that represented as far as possible the anticipated workload. Many of these were developments of similar codes that had been used in previous benchmarks.

The Physicists then worked with HPC personnel at AWE to match the benchmark codes to the anticipated real workload to arrive at a representative set of codes with weighted priorities.

Engineering codes

The engineers plan to considerably increase the size of the models they analyse – maybe up to around 30 million elements. Both explicit and implicit analysis will be needed.

Explicit analysis was represented in the benchmark by MPP-Dyna from LSTC.

For implicit analysis, models of the size and nature required by AWE cannot be analysed today and it is currently unclear whether iterative solvers or direct solvers will be used. It is, therefore the intention in the short term to pursue development using both of these options. Implicit direct solvers require a large shared memory SMP whereas iterative solvers are well suited to MPP architecture.

Implicit iterative solver techniques were represented in the benchmark by Salinas from the US Sandia National Laboratory (SNL) and implicit direct techniques by LS-Dyna Implicit from LSTC.

Weightings were estimated to reflect the expected future usage but there was necessarily a degree of uncertainty in these.

Material science codes

The two main codes used by the Material Scientists could be made available without modification. These were both molecular dynamics codes: DL/Poly from the UK Daresbury Laboratory, and WARP from SNL.

Make-up of the overall benchmark

Almost the whole of the user requirements were well-suited to MPP (distributed memory) architecture. This results partly from the fact that AWE has been working with MPI parallelisation for MPP for many years – but also from the intrinsic nature of the applications.

The only requirement *not* suited to MPP was implicit analysis using direct solvers for the engineers. To meet this, a requirement for a high-memory SMP component was included in the ITT as optional and but it was subsequently excluded and is currently being acquired by a separate tender process.

The MPP applications were grouped together and weighted as shown in the following table. The weights served both to indicate the relative importance of the applications and to represent the contributions of each code to the *throughput benchmark* described below.

Code	Weight	User community
Hydra	1	**Physics**
Corvus	1	
PETSc	2	
Chimaera	8	
Serial1	4	
MPP Dyna	5	**Engineering**
Salinas	4	
Warp	2	**Material science**
DL/Poly	2	
	29	

Table 1. The Throughput Benchmark

In addition, the following codes, not in the MPP throughput suite, were included in the benchmark but are not further discussed in this paper:

- LS/Dyna (SMP engineering analysis)
- Visualisation
- I/O
- TyphonIO (AWE written parallel I/O)
- FORTRAN 90 (standards conformance)
- Pallas (Interconnect performance measurement)
- Overlap (test to overlap processing and MPI message passing)

2.2 Benchmarking Capacity vs. Capability

For a given workload, the capacity (or throughput) of a system is relatively easy to define and measure. It is the rate at which the system can continuously turn round repeating instances of the complete workload.

However, when comparing different systems, the question arises as to whether, if a given job in the workload is run N-ways parallel on one system, it should also be run N-ways parallel on the others. In the design of the AWE benchmark, the view was taken that the correct answer to this question is "**NO**".

What we did for our capacity (throughput) measures was to insist that the turnaround times of the jobs on all systems were equal to or better than specified values. If a particular system had slow PEs relative to the others, then the degree of parallelism had to be adjusted upwards until the turnaround time was OK.

This is because, if you compare a fixed N-way parallel job across multiple systems then the measured throughput figures favour slow PEs.

To see why this is so, consider two systems A and B. Suppose the PEs on System A are twice as fast for serial code as on B and that the interconnect is also twice as fast so that parallel scalability is the same on both. System A has N PEs, and, to compensate for each PE being half-speed, System B has 2N PEs, so that, at first sight, the total capacity is the same.

Suppose you run a single N-way parallel job on A and it takes T seconds. Then the same N-way parallel job on B will take 2T seconds. However, since there are 2N PEs, you can run two such jobs concurrently. In this scenario, the throughput (capacity) of the two systems is identical since A processes one job in T seconds and B processes two jobs in 2T seconds.

However, why was the job being run N-ways parallel on System A (rather than (N/2)-ways parallel, for example)? The answer must be "because the user needs the turnaround time given by running N-ways parallel". If a turnaround time of 2T is acceptable, then it would be better to run System A at (N/2)-ways parallel rather than N. This is because, in the usual case of sub-linear speedup, it is more efficient in throughput terms to run with the lowest degree of parallelism that gives acceptable turnaround.

If a turnaround time of 2T is unacceptable on System A, then it is also unacceptable on System B. To get comparable turnaround on B, let us suppose we run a single 2N-way parallel job. Now you are just running one job on B (as on A) and suppose it takes time T_B . Normally, T_B will be *greater* than T because of imperfect scalability so, now, the throughput of B is *less* than that of A.

In practice the scalability of different applications varies widely – but it would be quite typical for this effect to be a quite significant 20 or 30% – and, in an extreme case, where the turnaround time starts to *increase* with increasing N (passes the scalability limit) it may be impossible for System B to give the minimum acceptable turnaround time of T.

So for our hypothetical Systems A and B, the conclusions drawn from running jobs with same degree of parallelism on both are quite different from the conclusions from running jobs with similar turnaround times. As follows:

Compare N-way parallel on A with N-way parallel on B (INCORRECT)

System A gives much better turnaround but the total capacities of A and B are the same.

Adjust N to give comparable turnaround times on A and B (CORRECT)

Both the turnaround time AND the total capacity of B are worse than those of A.

2.3 Throughput benchmark design: the "4x Blue Oak Capability constraint"; vendor-committed capacity figures.

The design of the throughput benchmark was based on the following principles:

- Moderately sized test cases were chosen (as this is a capacity measure rather than capability). Degrees of parallelism were up to 64-way only.
- A reference run of the throughput benchmark as defined in Table 1 was performed on Blue Oak. To do this, the whole machine was dedicated. From this the base *turnaround times* for all jobs were measured.
- Vendors were advised that, if they won the business, they would be required to run this workload as an acceptance test across the whole machine and were asked to contractually commit to the throughput relative to Blue Oak that would be achieved.
- The "4x Blue Oak Capability Constraint". During the running of the acceptance test, the

turnaround times achieved on all jobs would be required to be equal to or less than one quarter of the time measured on Blue Oak. The figure of 4x was somewhat arbitrary but was designed to allow modern chips such as Opteron or POWER5 to meet the criterion with the *same* degree of parallelism as on Blue Oak. Vendors not meeting this requirement on some jobs needed to increase the degree of parallelism on these jobs so that it was met.

- There was a requirement to run a mini-throughput benchmark across 128 PEs only.
- There was no formal requirement to do any further throughput runs. Vendors could do whatever runs they deemed necessary to make the capacity commitment across the whole installed system.
- The computation of total throughput relative to Blue Oak was done in terms of the measured job turnaround times together with the weights in Table 1. *Speedup* for each job equals (turnaround on Blue Oak)/(turnaround on target platform). Then the throughput relative to Blue Oak is the weighted harmonic mean of the speedups scaled by the number of PEs required for the jobs and scaled by the total number of PEs bid. A spreadsheet that performed this calculation was supplied to vendors.

2.4 Benchmarking capability jobs

For capability jobs, AWE was interested in scalability up to 2048 PEs and beyond. The problem with this is that most vendors would not be able to find benchmark systems that big

As a partial solution, we requested vendors to project results to higher numbers of PEs than benchmarked and to provide details of industry-standard benchmarks such as LINPACK on high PE counts. However, vendors were reluctant to extrapolate beyond PE count measured and we had to make judgements on the limited amount of information that was available.

3. Benchmark Evaluation

Once the bids from the six vendors were received, the results were analysed at AWE by a small team consisting of the three authors of this paper.

There were two issues that arose during the evaluation. These were:

- Source code tuning, and
- How to reconcile and compare capacity and capability differences.

These issues are described in the following sub-sections on “Source code tuning”, “Capacity evaluation”, and “Capability evaluation”.

3.1 Source code tuning

It was a requirement of the benchmark that results be submitted from runs using identical source code to that used on Blue Oak (except for minor changes needed to get the application to run successfully). However, results from runs tuned via source code changes could also optionally be submitted.

The intention was to make like-for-like comparisons on asis code but to also evaluate whatever improvements vendors managed to get.

Credit was given to vendors who tuned but the basic system comparisons were done on the asis code figures – since, to do otherwise would have been measuring the skill of the benchmark teams rather than the capabilities of the systems.

Most vendors benchmarked on a system (typically single-core chips) different to that bid (typically dual core). The two vendors who tuned then projected the results onto the bid system – but only the tuned results. In order to do the like-for-like comparison on asis code, AWE used the same ratios as the vendors to also project the asis results.

3.2 Capacity evaluation

Again, throughput commitments were generally given only for tuned code and, again, AWE calculated the figure for asis code. Some vendors had not properly adhered to the 4x Blue Oak capability constraint and this was also corrected by AWE.

A single capacity figure for all bid systems running asis code was therefore derived with a reasonable degree of precision and confidence. These figures were directly comparable across vendors. Similar figures for tuned code were derived for those vendors that tuned. These were evaluated in the context of the asis figures but were not directly comparable with them.

3.3 Capability evaluation

As has been explained, it was possible to reliably quantify the *capacity* of the bid systems in terms of the throughput benchmark jobs. It was not so for *capability*. In particular,

it was impossible to effectively estimate the effects on throughput of capability differences.

The main tool used for investigating capability was a series of scalability charts for each test case and each application which plotted speedup compared with Blue Oak against the number of PEs used (ways parallel). These are described in detail in the next section on "Vendor Comparisons".

The main problems in trying to evaluate capability were:

- Lack of data. Many vendors did not provide measurements above 1024 PEs.
- For the big scalable applications of most interest to AWE, the scalability up to 1024 was very similar on all systems, almost certainly because we were seeing the intrinsic scalability of the application without a major overhead constraint from the interconnect or the system.
- In cases where the limit of scalability was reached on multiple vendor systems, whilst it was possible to say which was best, it was not possible to quantify it in terms of extra capacity. This is because, whenever one system reached the scalability limit (went past the "turnover point" where turnaround time starts to *increase* with more PEs), the throughput comparison with a system that was still scaling varied wildly and there was no rational ground to choose one value rather than another.

The main purpose of a capability system is to reduce turnaround times for huge parallel jobs. Therefore, we decided to take as a measure: *the best turnaround time that the system can deliver with reasonable efficiency irrespective of the number of PEs used.*

To measure this, it was necessary to take the point on the chart just before "turnover" became too serious.

Because of limited data above 1024 PEs, this was only possible for three test cases. The results for these is discussed in the next section on "Vendor Comparisons".

4. Benchmark Results and Vendor Comparisons

The following three sub-sections describe the benchmark results.

4.1 Short listing and the winning vendor

In terms of the quality of the benchmarking, Cray's submission was outstanding. They benchmarked on up to 4096 PEs and providing an impressively complete set of figures. In addition, they had extensively analysed and successfully tuned a majority of the applications.

Two vendors (Cray and LNXI) were short listed by AWE.

Cray, with the XT3, was finally selected on merit as the winning vendor for consistently good scores in the following areas. It was not necessarily best in all, but AWE judged it best overall.

- Throughput per PE (asis code)
  - Even better with tuned code
- Scalability
  - Demonstrated to 4096 PEs
- Support
  - Code tuning demonstrated application skills
  - Established in UK
- Price/performance

LNXI were a close second.

The more detailed capacity and capability measures in the following two sections are restricted to the two short listed vendors, Cray and LNXI.

4.2 Capacity results

The throughput per PE figures relative to Blue Oak are given in the following table:

	Cray	**LNXI**
Asis code	4.77	4.88
Tuned code	6.33	No tuning

Table 2. Capacity per PE relative to Blue Oak, dual-core chips.

For asis code, this shows Cray and LNXI almost neck and neck. At first sight this is not surprising since both bid dual-core Opteron chips – but, in fact, there were some significant differences which, it seems, nearly cancel out.

1. Cray bid 2.6 GHz chips whereas LNXI bid 2.2 GHz. This 18% difference is not apparent in the asis throughput figures.
2. LNXI measured on dual-core chips and found relatively little slowdown compared with single-core. Cray, on the other hand, measured single-core performance only and used a projection methodology to dual-core that predicted up to 30% slowdown on some of AWE's benchmark codes. It is possible that Cray's methodology

was cautious and that their figures might be higher when measured on the final installed system.

3. Fortran compiler differences (PGI from Cray and Pathscale from LNXI) might also have made a difference.

4.3. Capability results

As has been explained, it was difficult to see significant scalability differences for most applications because of a lack of data at PE counts above 1024 from many vendors.

Scalability was assessed using a series of charts. Two examples will be given: Chimaera – won by Cray – and PETSc – won by LNXI.

Figure 1 shows the scalability chart used for the highest weight application and the largest test case.

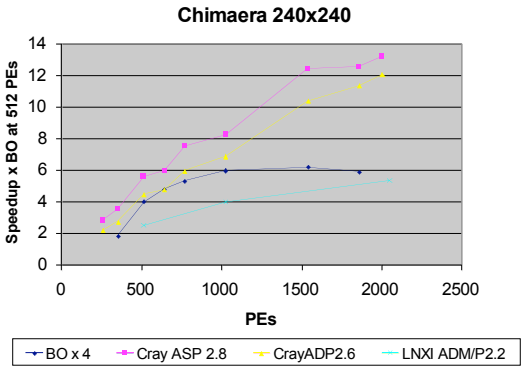


Figure 1. Scalability chart for Chimaera

Figure 1: key and explanation:

The x-axis is the PE count (number of ways parallel). The y-axis plots the speedup in absolute terms relative to Blue Oak at 512 PEs. The "BOx4" line is the reference line for Blue Oak itself. What is plotted here is four times the actual speedup. Therefore this line passes through y=4 at 512 PEs. The key to the other lines is as follows:

For example, the letters in Cray ASP2.8 and LNXI ADM/P mean as follows.
A = Asis code
S = Single-core chips.
M = Measured results
P = Projected from different clock rate
The number at the end is clock rate in GHz.

The following conclusions can be drawn from this chart for this application only.

- LNXI performs badly in this case. Looking at LNXI's other results, this is anomalous.
- Scalability up to 1024 PEs is good for both vendors.
- Cray scales well up to 2000 PEs. LNXI probably does also but the LNXI figure at 2048 PEs is an extrapolation from measurements up to 1024.

The second application shown here in Figures 2 and 3 is PETSc. This was a test driver for an iterative sparse matrix solver from the publicly available PETSc library being used to solve a realistic AWE matrix. This is the one application where we had measured data up to the limits of scalability for all vendors.

In Figure 2, the speedup is relative to Blue Oak at 1 PE. The main difference in type from Figure 1 is that the speedup has been divided by the number of PEs. Each line is therefore proportional to a parallel efficiency chart. As such, perfect scalability would yield a horizontal line. A real parallel efficiency chart always passes through the value "1" at 1 PE. The lines in Figure 2 are simply scaled relative to Blue Oak.

From the steep downwards slope of most of the lines, it can be seen that PETSc scales poorly with this matrix. Note that, for some unknown reason, LNXI shows super-linear speedup between 4 and 16 PEs.

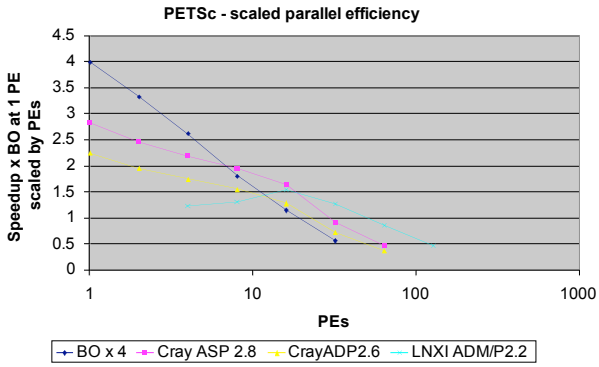
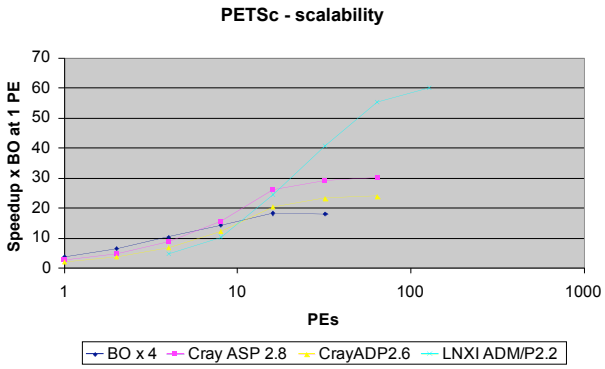


Figure 2. Scaled parallel efficiency chart for PETSc

The *same* data are presented in Figure 3 with the "efficiency" scaling by PEs removed. This is now the same type of scalability chart as Figure 1: it is the simple absolute speedup (relative to BO at 1 PE) that is charted.



The following conclusions can be drawn from Figures 2 and 3 for the PETSc application only.

- LNXI wins overwhelmingly on scalability
  - scales to 128 PEs
  - other vendors to 32 or 64
- Cray is faster than LNXI at low PE counts
- Cray scales better than Blue Oak.

The reason why LNXI scales so much better here is almost certainly because of the low MPI latency across the interconnect (about 2 microseconds). PETSc near the scalability limit does a huge amount of message passing of tiny messages.

Finally, some additional factors which supported the choice of Cray as winning vendor were:

- Cray had easily the best performance of all vendors if tuned code is taken into account (most vendors did not tune the code)
- Cray demonstrated scalability up to 4096 PEs
- Cray's lightweight kernel (Catamount) was regarded as a significant technical benefit
- Cray had the best price/performance (in terms of capacity)

Quantified capability measures

As was explained above in *"3.3 Capability Evaluation"* , the only way we found to quantify capability was to measure the best turnaround that the system could deliver irrespective of the number of PEs. This was expressed as a *speedup* relative to the same figure for Blue Oak. To measure this, it was necessary to have data up to the scalability limit (turnover point) and we had this for only three test cases:

- Chimaera 240x240
- PETSc, and
- DLPoly Large

The results were as follows:

Test case	Cray asis	Cray tuned	LNXI
Chimaera 240x240	8.0 at about 2000 PEs	19.5 at about 2000 PEs	Anomalously low
PETSc	5.2	5.8	13.0
DLPoly Large	7.0	7.3	5.2

Table 3. Maximum speedup relative to BO.

Note: there results are subject to a large uncertainty because:

- The figures are not fully representative since they are based on just three test cases
- These are NOT throughput figures since the numbers of PEs varied. They cannot be quantitatively combined with capacity figures.

5. Overall conclusions

The benchmarking carried out during the recent supercomputer procurement at AWE was successful in representing the expected workload and at differentiating between the submission of rival vendors.

AWE looks forward to working with Cray to make the installed XT3 system successful.

Acknowledgments

The authors wish to express their heart-felt admiration and thanks to all benchmarking staff from all the vendors who took part in this procurement.