AWE HPC Benchmark, 2005

DESIGN and EVALUATION

Ron Bell
ron.bell@awe.co.uk

HPC at AWE

- AWE is the Atomic Weapons Establishment, Aldermaston, UK
- Existing system is “Blue Oak”
  - IBM POWER3 (16-way Nighthawk nodes)
  - 1856 usable PEs at 375 MHz
  - 2.78 peak Tflops
- Procuring a system with a capacity of
  - Up to 25 x Blue Oak
  - As measured by benchmark
  - Not peak Tflops

New HPC System at AWE

- Order for Cray XT3 December, 2005
- Planned installation: June 2006
- 3936 nodes of dual-core 2.6 GHz Opteron (7872 PEs)
- > 40 Tflops peak
- Throughput vs Blue Oak
  - Weighted set of benchmark codes
  - 20 x Blue Oak (asis code)
  - 27 x Blue Oak (Cray-tuned code vs BO untuned)

AWE HPC BENCHMARK: Topics

- Benchmark objectives
- User requirements and codes
- Benchmark job mix
  - Proportions to represent workload
- Capability vs capacity
  - Turnround vs throughput
- Evaluation Issues
- Comparative results

Benchmark Objectives

- Represent codes from whole user community
  - Physicists
  - Engineers
  - Material Scientists
- Measure both capacity (throughput) and capability (parallel scalability)
- Include “Throughput Benchmark” to make whole system busy

User Requirements: Physics

- Users did thorough job defining requirements
  - Set of existing and planned codes
- Set of benchmark jobs
  - Many highly scalable
  - Up to 1024 PEs on Blue Oak
  - Planned to go to 4096 PEs and beyond
- Users worked with HPC to match planned workload to benchmark code

Engineering

- Engineering requirements
  - Up to 30M elements (100MDOFs) models
  - Both Explicit and Implicit (non-linear)
  - Such models cannot be analysed today
  - Implicit solvers can be iterative or direct
    - Currently pursuing both
- Three codes in benchmark
  - Explicit: MPP-Dyna from LSTC
  - Implicit iterative: Salinas from SNL
  - Implicit direct: LS-Dyna from LSTC

Material Physics

- Only two main codes, both Molecular Dynamics
  - DL-Poly (from UK Daresbury Lab.)
  - WARP (from Sandia)
  - Can be distributed to vendors
  - Can benchmark the real thing
  - Highly scalable - >1024 PEs

Overall Benchmark Job Mix

- Throughput Benchmark (WEIGHT)
  - Hydra 1
  - Corvus 1
  - PETSc 2
  - Chimaera 8
  - Serial1 4
  - Warp 2
  - Dlpoly 2
  - MPP-Dyna 5
  - Salinas 4
    - **TOTAL 29**
- Plus some extras:
  - LS/Dyna (SMP)
  - Visualisation
  - I/O
  - TYPHON/IO
  - FORTRAN 90
  - PALLAS comms test
  - MPI overlap test

Physics

Material Science

Engineering

Designed to sum to 29
for BlueOak throughput run.
29x64=1856
No. of usable PEs on BlueOak

Aside on CPUs, Cores, and PEs

- **How many "CPU"s are there in one dual-core chip?**
  - I (and a majority of my colleagues in a straw poll) say "two"
  - Chip vendors say "one"
- Moral: **AVOID the term "CPU"**
- I will use "PE" (Processing Element) instead
  - 1 PE = 1 core
  - 1 dual-core chip = 2 PEs

Capacity versus Capability

- Capacity vs Capability
  - That is: Throughput vs Turnaround
  - These CONFLICT
    - For example:- slow CPUs optimise throughput but not turnaround

If you don't measure it properly

Why slow CPUs may emphasize throughput

- Consider
  - System A has 64 fast PEs
  - System B has half-speed PEs – but 128 to compensate. Interconnect scales exactly as A.
  - 64-way parallel job takes time T on A
  - Therefore it takes time 2T on B
  - But you can run two concurrently on B
  - **SO the throughput is the same**
  - **RIGHT?**
- **WRONG!** See next slide.

The importance of turnaround

- Why are you running the job 64-way parallel on System A?
  - Because you need the turnaround of T
  - IF turnaround of 2T is OK, you SHOULD run it 32-way on A
    - to get more throughput, assuming imperfect application scalability
  - To get turnaround of about T, you must run 128-way parallel on System B.
    - Now time on System B is more than T (because of imperfect scalability) and
    - Throughput of B is lower than A.
      - Amount varies with application
      - 20 or 30% quite typical
      - In extreme case, it might be impossible for B to give a turnaround of T

Systems A and B: conclusion

- A: 64 fast PEs vs B: 128 half speed PEs
- Compare 64-way parallel on A with 64-way on B (INCORRECT)
  - A gives much better (2x) turnaround
  - A and B have equal capacities
- Compare 64-way job on A with similar turnaround (128-way) job on B (CORRECT)
  - B gives worse turnaround, AND
  - B has lower capacity
- CONCLUSION

  When measuring the capacity of different systems using parallel applications, the degree of parallelism should be adjusted so that all systems give similar turnaround times

Throughput Benchmark Design

- Reference Jobstream run on BlueOak
  - Divide 1856 CPUs into 29 Groups of 64 (moderate parallelism)
  - In each Group run repeating 64-way jobs (some exceptions)
- Vendors required to commit to capacity of installed system
  - Must be achieved across whole system as ACCEPTANCE TEST
- Vendors had to run:
  - 128-PE mini-throughput benchmark
  - Whatever further runs needed to make commitment
- On vendor platform
  - Apply "4x BO Capability Constraint"
    - Turnaround must be <= 0.25 x BO turnround
    - Adjust parallelism to achieve this if necessary
  - Run jobstreams similar to BO
  - Measure turnaround times and hence speedups
- Mean throughput increase is the weighted harmonic mean of speedups scaled by numbers of PEs

Benchmarking CAPABILITY

- Direct capability measures
  - E.g. compare 1024-way Chimaera job on each different target platform
    - PROBLEM:
      - Limited benchmark systems from most or all vendors
      - Very limited benchmark systems from some vendors
    - Partial Solutions
      - Ask that vendors estimate turnround for key capability jobs
      - Direct evaluation of interconnects (latency/bw etc.)
      - Draw scalability graphs and extrapolate
      - Ask for contracted scalability figures on industry-standard benchmarks

Evaluation Issues

- Tuning by modifying source code
- How to reconcile and compare capacity and capability

Source code tuning

- What we asked for
  - Asis results plus optionally tuned results
- What we got from different vendors was a mix of:
  - No tuning
  - Asis and tuned results on benchmark system. Only tuned results projected to (different) target system
  - Throughput commitments based on tuned code
  - Throughput commitments based on tuning not yet done!
- How we evaluated:
  - Main comparisons done on “asis” code (like for like)
    - Tuned projections back-projected by AWE to “asis”
  - Gave credit for fact that tuning demonstrated application skills

Evaluating Capacity and Capability

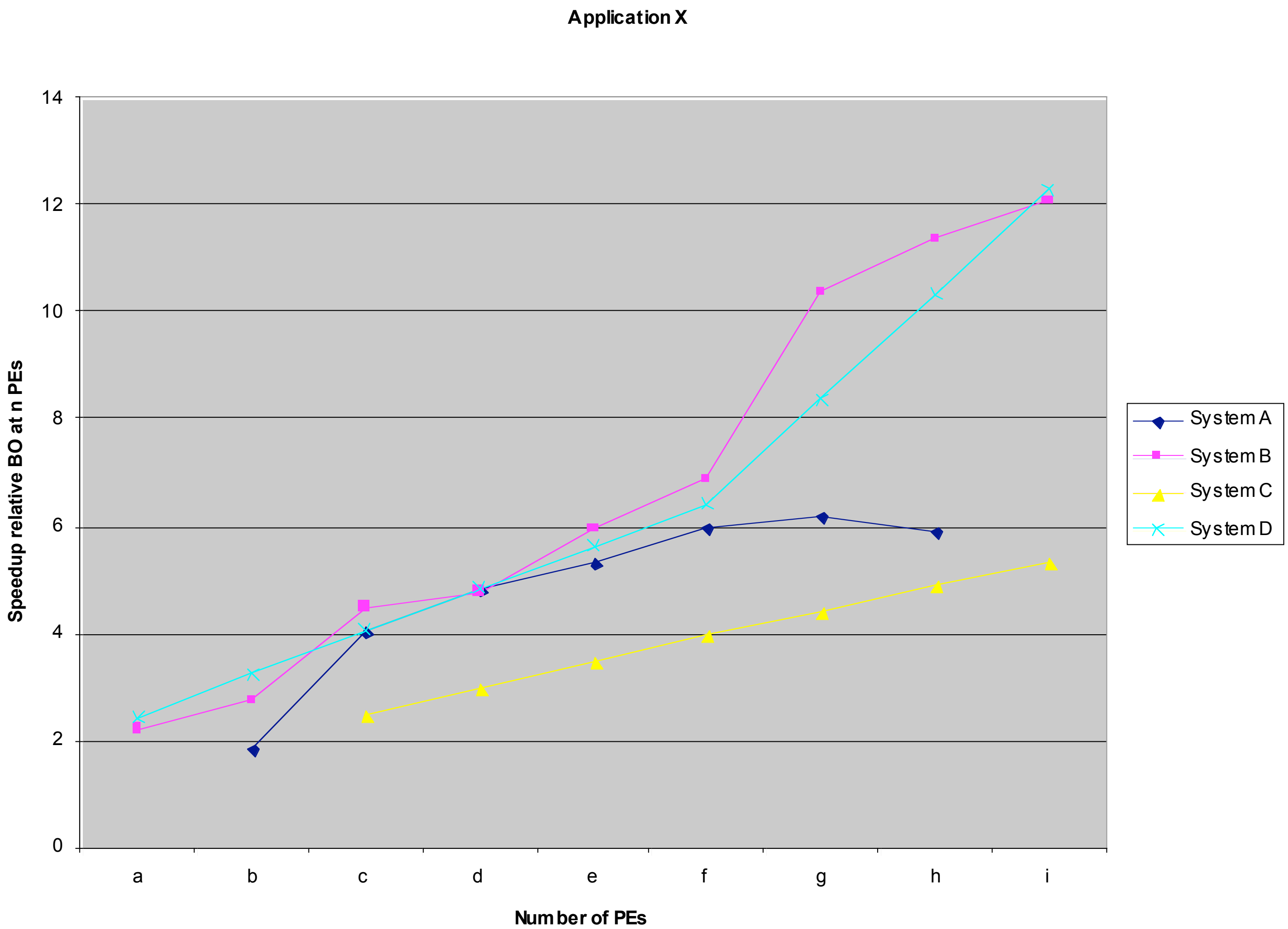
- Capacity
  - Single figure – easy to measure
  - Based on modest parallelism (64-way)
- Capability
  - Not possible to evaluate comprehensively because of limited data
  - Scalability differences showed up clearly in only a few cases
  - **Question**: Can you estimate the effect on throughput of capability differences?

Effect of capability on throughput

- I wanted to be able to say things like:
  - System A has 10% higher throughput than system B for modestly parallel work
  - But system B has better scalability – so capability jobs show 20% higher throughput on B
  - If we assume half of the system will be dedicated to capability jobs, then System B gives more overall throughput
- **I concluded this could not be done**

Problem with measuring throughput of capability jobs

- At modest levels of parallelism, scalability largely unaffected by interconnect
  - Scalability is intrinsic to application
  - Ratio between systems constant with PE count
- At higher PE counts where performance “turns over”, relative throughput varies wildly and becomes meaningless
- **Sample scalability chart next …..**



Application X
14
12
10
8
6
4
2
0
Speedup relative BO at n PEs
a
b
c
d
e
f
g
h
i
Number of PEs
System A
System B
System C
System D

Capability: the best we could do

- For the few cases where we had benchmark data up to “turn over” point
  - Measure the job turnround at a point just before “turn over” became too serious
  - In other words: **The best the system can do irrespective of number of PEs**
- Generally, a system scoring BETTER on this measure would need more PEs to achieve it – so throughput per PE was lower
- Capacity (throughput) and capability figures then presented as separate measures
  - Warning about the large uncertainties on the capability figures

VENDOR COMPARISONS: Summary

- Quality of Cray's benchmark submission was quite outstanding
  - Benchmarked up to 4000 PEs
  - Impressively complete set of results
  - Extensive source code tuning
    - Majority of apps tuned
    - 2.5 x speedup on most important
      - Chimaera – tuning by Monika Wierse

Vendor Comparisons (contd.)

- Opterons faster than the Itaniums
- Shortlist was Cray and LNXI
- **Cray won** – on overall merit! - not necessarily best on all factors
  - Throughput
  - Scalability (demonstrated)
  - Support
    - Code tuning demonstrated
    - Established in UK
  - Price/performance
- LNXI were a close second

XT3 with 3936 nodes - 2.6 GHz DC Opteron
- 7872 PEs
- >40 Tflops peak
- to be installed June '06

Cray and LNXI - CAPACITY

- Moderate parallelism (64 PEs typically)
- Throughput **per PE** x Blue Oak
- Weighted average across all apps

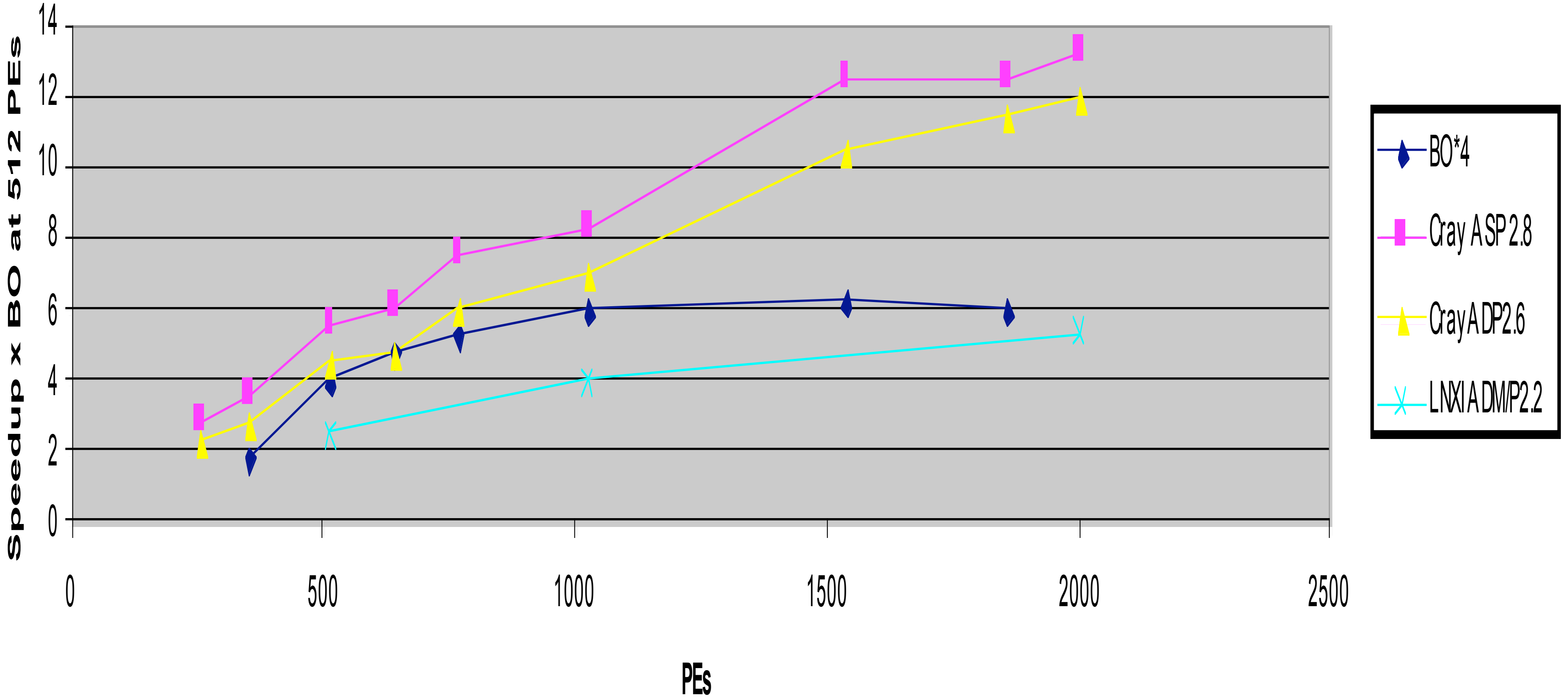
	Cray	**LNXI**
Asis code	4.77	4.88
Tuned code	6.33	No tuning

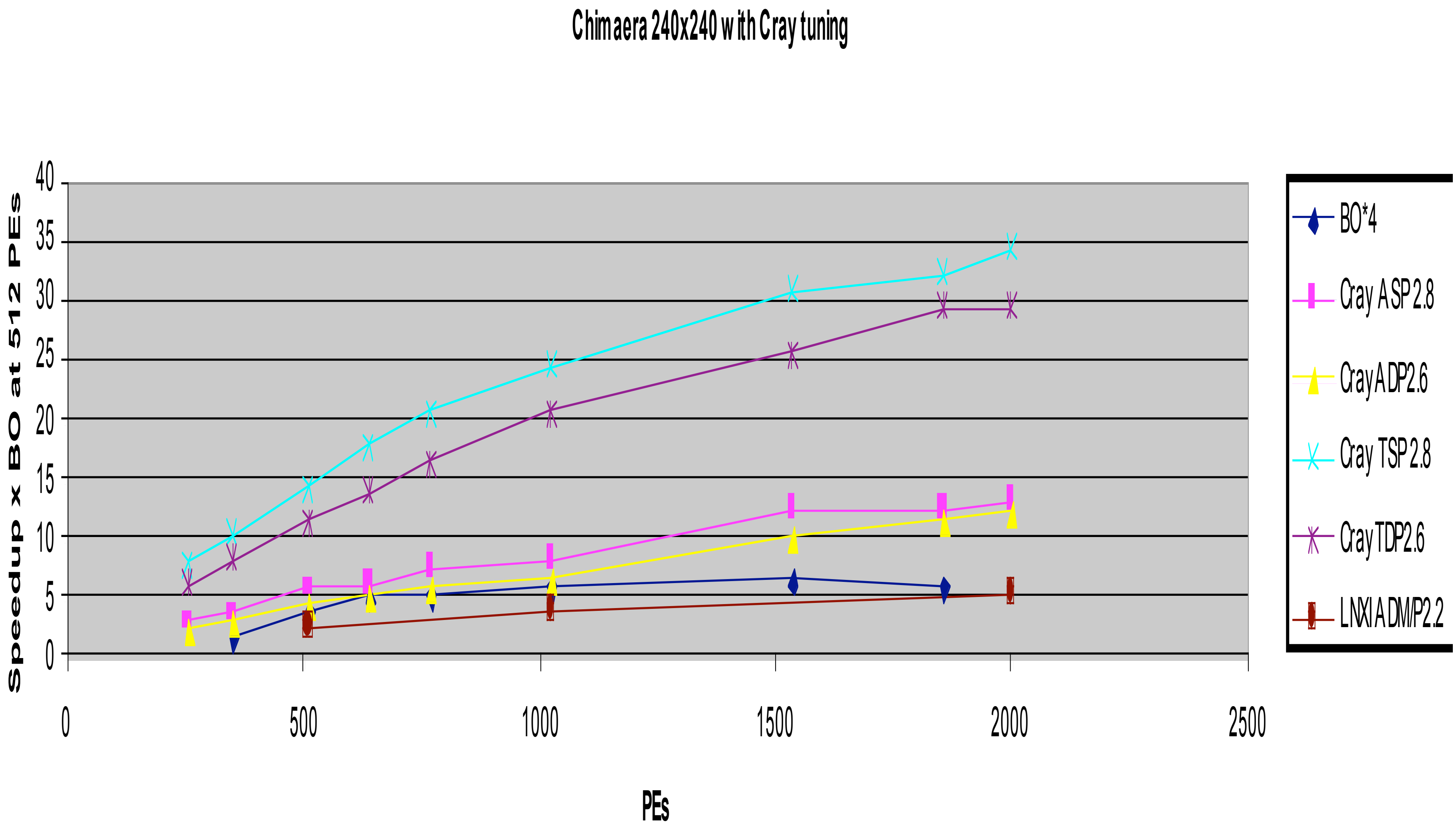
- Cray measured on 2.4 GHz SC – projected to 2.6 GHz DC
- LNXI measured on 2.2 GHz DC
- Cray's DC projections conservative (WE HOPE!)

Scalability graphs

- Two examples given
- One won by Cray and one by LNXI

Chimaera 240x240 - Scalability



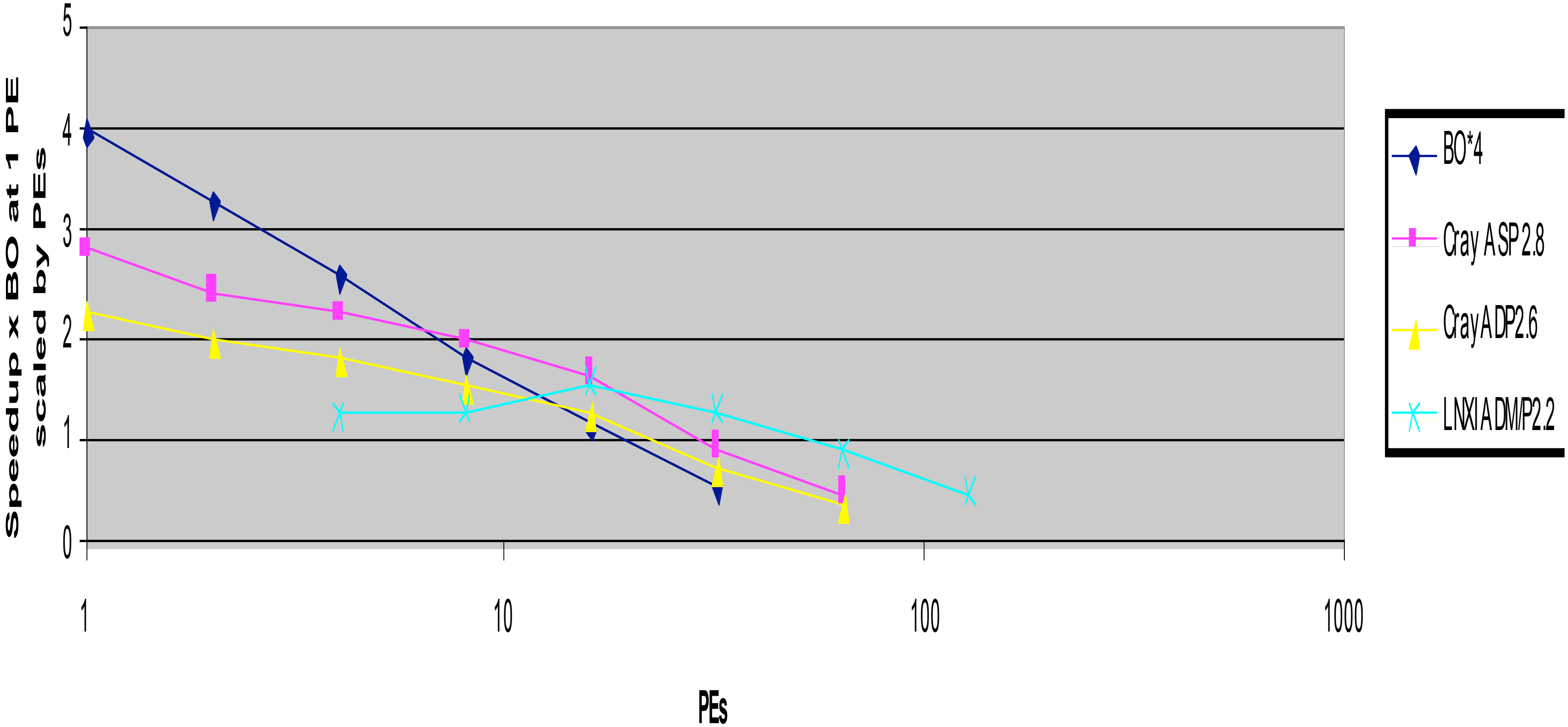


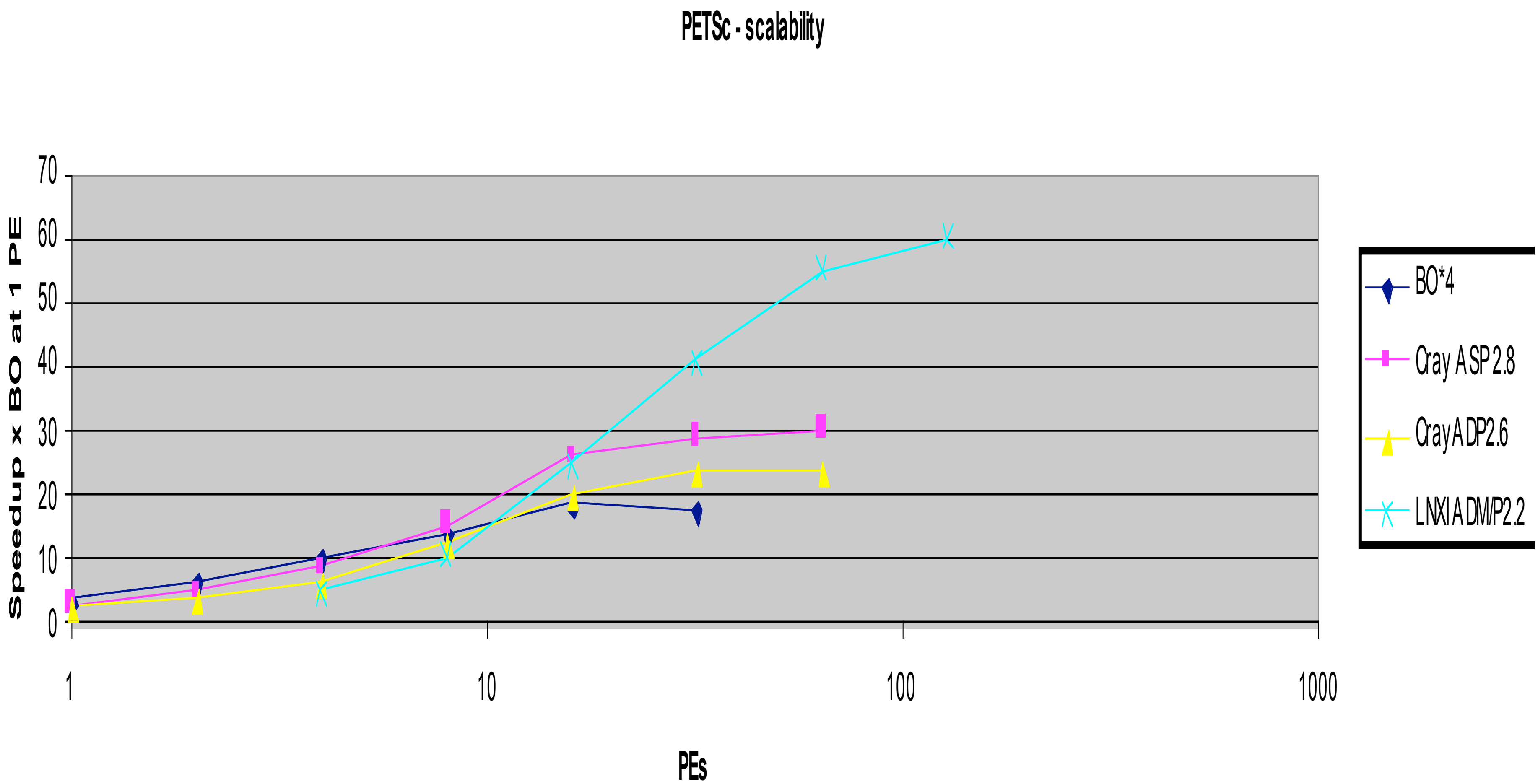
Chimaera 240x240 with Cray tuning
Speedup x BO at 512 PEs
0
5
10
15
20
25
30
35
40
0
500
1000
1500
2000
2500
PEs
BO*4
Cray ASP2.8
Cray ADP2.6
Cray TSP2.8
Cray TDP2.6
LNXI ADMP2.2

Chimaera 240x240

- The highest weight application
- Cray wins against LNXI
  - Overwhelmingly against all vendors if tuned code allowed
- Cray scales well up to 2000 PEs
- LNXI has (anomalously) poor result

PETSc - scaled parallel efficiency




PETSc - scalability
Speedup x BO at 1 PE
PEs
BO*4
Cray ASP2.8
Cray ADP2.6
LNXIADMP2.2
0
10
20
30
40
50
60
70
1
10
100
1000

PETSc

- Iterative sparse equation solver
  - "Difficult" matrix
  - Poor scalability (fairly small problem)
  - Huge number of tiny MPI messages at high PE counts
- Neither does well against BO at low PEs
- LNXI faster than Cray at low PEs
- LNXI super-linear speedup to 16 PEs
- LNXI wins scalability overwhelmingly above 16 PEs
  - LNXI scales to 128
  - Huge number of tiny messages
  - Highly latency sensitive

CAPABILITY measures

- Maximum speedups x BO irrespective of number of PEs

Test case	Cray asis	Cray tuned	LNXI asis
Chimaera 240x240	8.0	19.5	Anom-alously low
PETSc	5.2	5.8	13.0
DLPoly large	7.0	7.3	5.2

- Indicative only
  - Small No. cases – not representative

Other factors favouring Cray

- Cray won easily on tuned code
  - Gave us confidence in application skills
- Cray <u>demonstrated</u> scalability to 4000 PEs
- Cray lightweight kernel regarded as significant technical benefit
- **Best price/performance**

Acknowledgements and Thanks

- Heartfelt admiration and thanks to ALL benchmarking personnel from ALL vendors who took part in this procurement

In Conclusion

- WE LOOK FORWARD TO WORKING WITH CRAY TO MAKE AWE's XT3 SUCCESSFUL