PRACE Application Enabling Work at EPCC

Xu Guo
Applications Consultant
EPCC, The University of Edinburgh
xguo@epcc.ed.ac.uk
+44 131 651 3530

Outline

- PRACE project overview
- Application enabling work @ EPCC:
  - NAMD
  - HELIUM
- Summary

Outline

- PRACE project overview
- Application enabling work @ EPCC:
  - NAMD
  - HELIUM
- Summary

PRACE

- Partnership for Advanced Computing in Europe
  - Aims to provide the European researches with a persistent pan-European HPC service to enable world-class science, consisting of several tier-0 centres
- EU approved the PRACE Project Preparatory Phase (Grant: INFSO-RI-211528)
  - Project duration: January 2008 – December 2009
  - Project budget: 20 M € , EC funding: 10 M €
  - Original 16 partners from 14 countries
  - By the end of 2009, 20 countries involved in and more are interested
- The Implementation Phase will start from June, 2010

PRACE Project Preparatory Phase

- Objective
  - Perform all legal, administrative, and technical work to create a legal entity and start providing Tier-0 HPC services in 2010

- PRACE project tasks in Preparatory Phase
  - Define the legal & administrative framework (WP2)
  - Dissemination, outreach & training (WP3)
  - Cooperate with the European HPC ecosystem (WP2/3)
  - Distributed computing (WP4)
  - Prototype system assessment (WP5)
  - **Software enabling for prototype systems (WP6)**
  - Procurement strategy: Petaflop/s systems for 2009/2010 (WP7)
  - Future Petaflop/s technologies, vendor cooperation (WP8)

PRACE WP6 in Preparatory Phase

- WP6: Software enabling for Petaflop/s systems
  - Worked closely with other work packages
- Primary goal
  - To identify and understand the software libraries, tools, benchmarks and skills required by users to ensure that their application can use a Petaflop/s system productively and efficiently.
- The largest technical activity in PRACE Preparatory Phase
  - Most of the PRACE partners involved in
  - EPCC carried the overall responsibility for WP6 and was heavily involved in the technical work

PABS

- PRACE Application Benchmark Suite
  - A set of representative applications benchmarks
  - To be used in the procurement process for Petaflop/s systems
  - ALYA, AVBP, BSIT, Code_Saturne, CPMD, CP2K, ELMER, GADGET, GPAW, GROMACS, ***HELIUM***, ***NAMD***, NEMO, NS3D, OCTOPUS, PEPC, QCD, Quantum_Espresso, SPECFEM3D, TORB/EUTERPE, TRIPOLI-4, and WRF
- Each application was ported to appropriate subset of prototypes
  - To capture the applications requirements for petascale systems
- Scalability and optimisation strategies were investigated

Prototypes for Petaflop/s Systems in 2009/2010

IBM BlueGene/P (FZJ)

0.85 GHz, PowerPC 450, 4 way

01-2008 / 06-2009

IBM Power6 (SARA)

4.7GHz Pwr6, 32 way, SMT, IB

07-2008

Cray XT5 (CSC)

2.3 GHz Barcelona, 8 way

2.7 GHz Shanghai, 8 way

11-2008

IBM Cell/Power (BSC)

12-2008

NEC SX9, vector part (HLRS)

02-2009

Intel Nehalem/Xeon (CEA/FZJ)

2.93 GHz Nehalem, dual socket quad-core Nodes, IB

06-2009

Outline

- PRACE project overview
- Application enabling work @ EPCC:
  - NAMD
    - BCO: Joachim Hein
    - Contributors: Martin Polak (ICA, Johannes Kepler University Linz, Austria), Paschalis Korosoglou (AUTH, Greece), Xu Guo
  - HELIUM
- Summary

NAMD Overview

- Molecular dynamics application from UIUC
- Emphasis on scalability
- Written in C++ using Charm++ parallel objects
- Domain decomposition
- Long range electro-static forces: Particle Mesh Ewald
- Load balance dynamically during first 300 steps
- Current version: NAMD 2.7 β2, released: November 2009
- For this study: 2.7β1, released: March 2009

Benchmark Test Cases

- Three test cases by: P. Coveney and Shunzhou Wan (UCL)
- Sizes: 1M, 2M and 9M atoms
  - TCR-pMHC-CD complexes in a membrane environment
  - Immune response research
  - Basic set of 1M atoms contains four complexes
  - Larger sets multiple copies of basic set in a box
- 2fs stepsize
- Trajectory of at least 10ns for scientific meaningful simulation (5,000,000 steps)

Resources for MPI Library on XT5

- During the load balance step, NAMD sends large number of unexpected message
- Rank 0 often fails, giving a clear error message that it has run out of buffer space for unexpected messages
- Setting (in pbs script) the following helps:

```
export MPICH_UNEX_BUFFER_SIZE=100M
export MPICH_PTL_SEND_CREDITS=-1
```

- Rem: 10% of available memory as MPI buffer space!

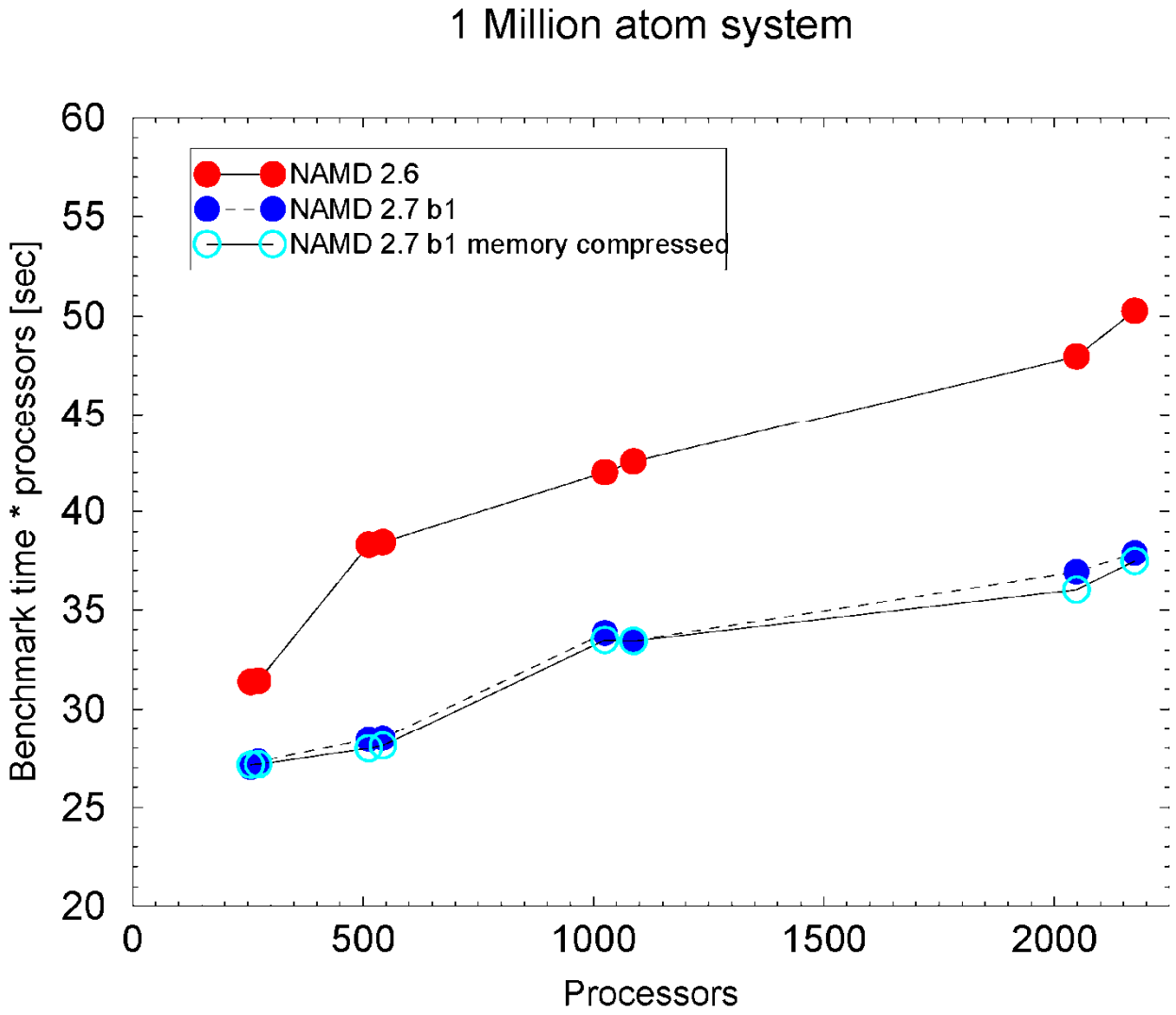
Memory Consumption

- Had memory problems on systems with 1GB/core, e.g. Louhi XT5@CSC/FI
- NAMD 2.7β1 offers simulation with reduced memory footprint
  - Specially build version for running, requires “vanilla” NAMD to compress input files
  - Essential for 9M atom benchmark and running 1M on 4 cores/node on BGP
- Below table for 2M atom system (CrayPat on XT5)

Number of tasks	Memory reduction	No Patch on Zero	Unload Zero	Footprint rank 0	Average footprint
256	No	Default	Default	1.58 GB	0.87 GB
512	No	Default	Default	1.58 GB	0.85 GB
1025	No	Default	Default	1.52 GB	0.85 GB
256	**Yes**	Default	Default	0.60 GB	0.44 GB
512	**Yes**	Default	Default	0.58 GB	0.44 GB
1025	**Yes**	Default	Default	0.60 GB	0.43 GB
256	**Yes**	**Yes**	Default	0.60 GB	0.43 GB
512	**Yes**	**Yes**	Default	0.58 GB	0.43 GB
1025	**Yes**	**Yes**	Default	0.59 GB	0.42 GB
256	**Yes**	**Yes**	**Yes**	0.56 GB	0.43 GB
512	**Yes**	**Yes**	**Yes**	0.60 GB	0.43 GB
1025	**Yes**	**Yes**	**Yes**	0.60 GB	0.42 GB

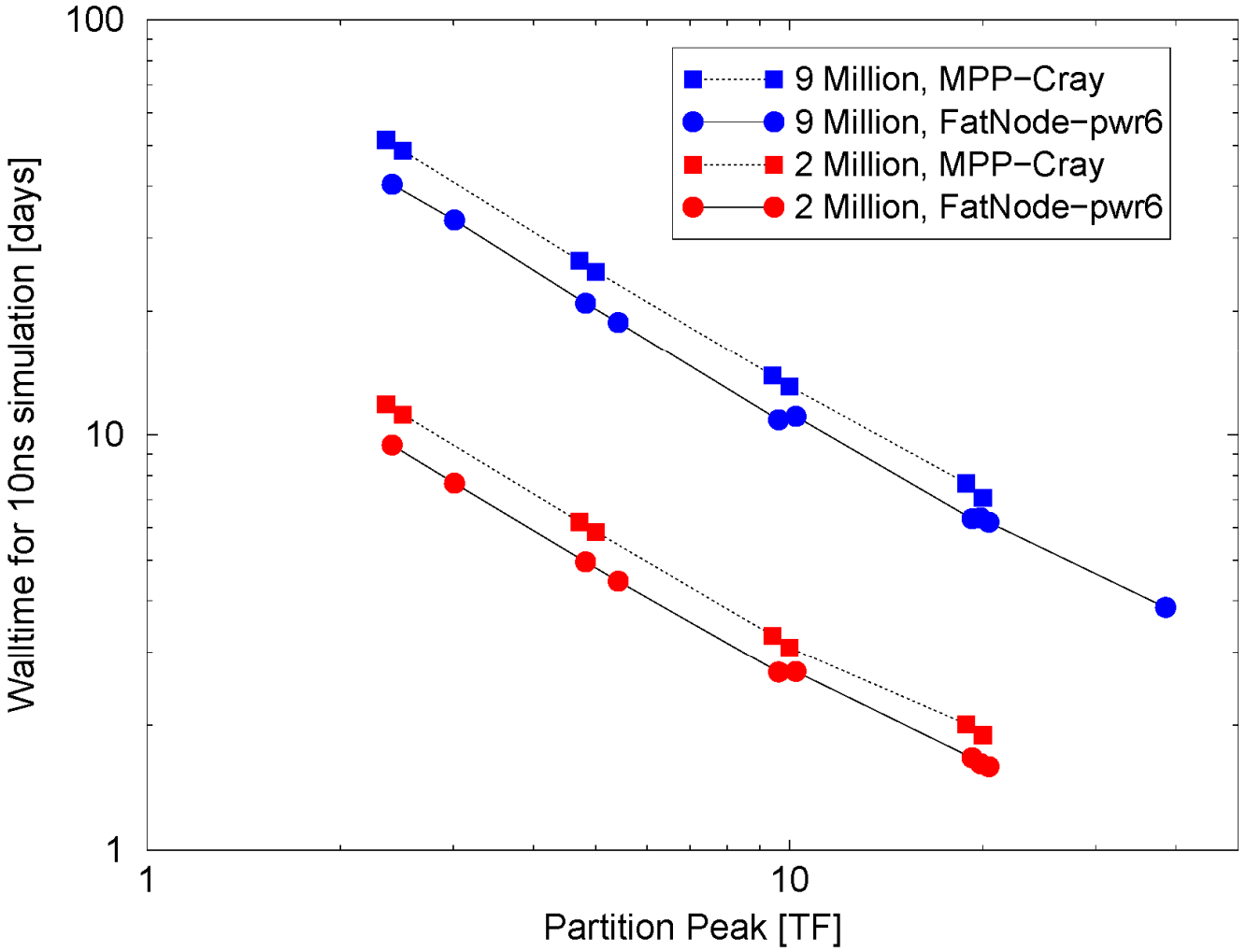
Version Performance on Cray XT5 Louhi

- NAMD 2.7b1 offers improved performance and scalability



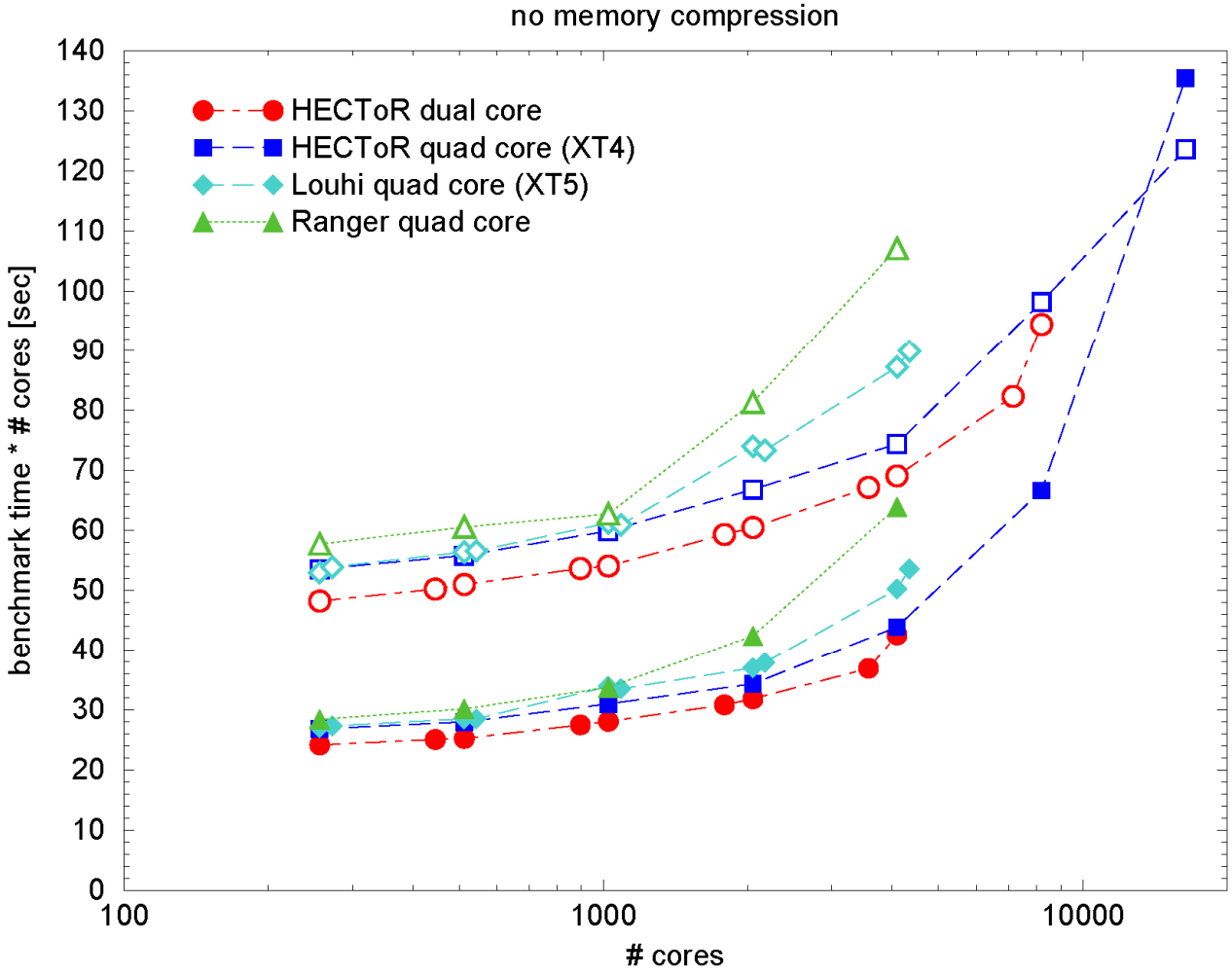
Feasibility of A Simulation on PF system

- On a 20TF partition a 9M atom simulation is feasible (walltime of about a week)!



Opteron Performance (M1 and M2)

- Ranger executable build by NAMD authors
- Their own results (Preprint: PPL Paper: 10-03) see better scalability for Ranger over XT4 portion of Jaguar – presently reasons not understood

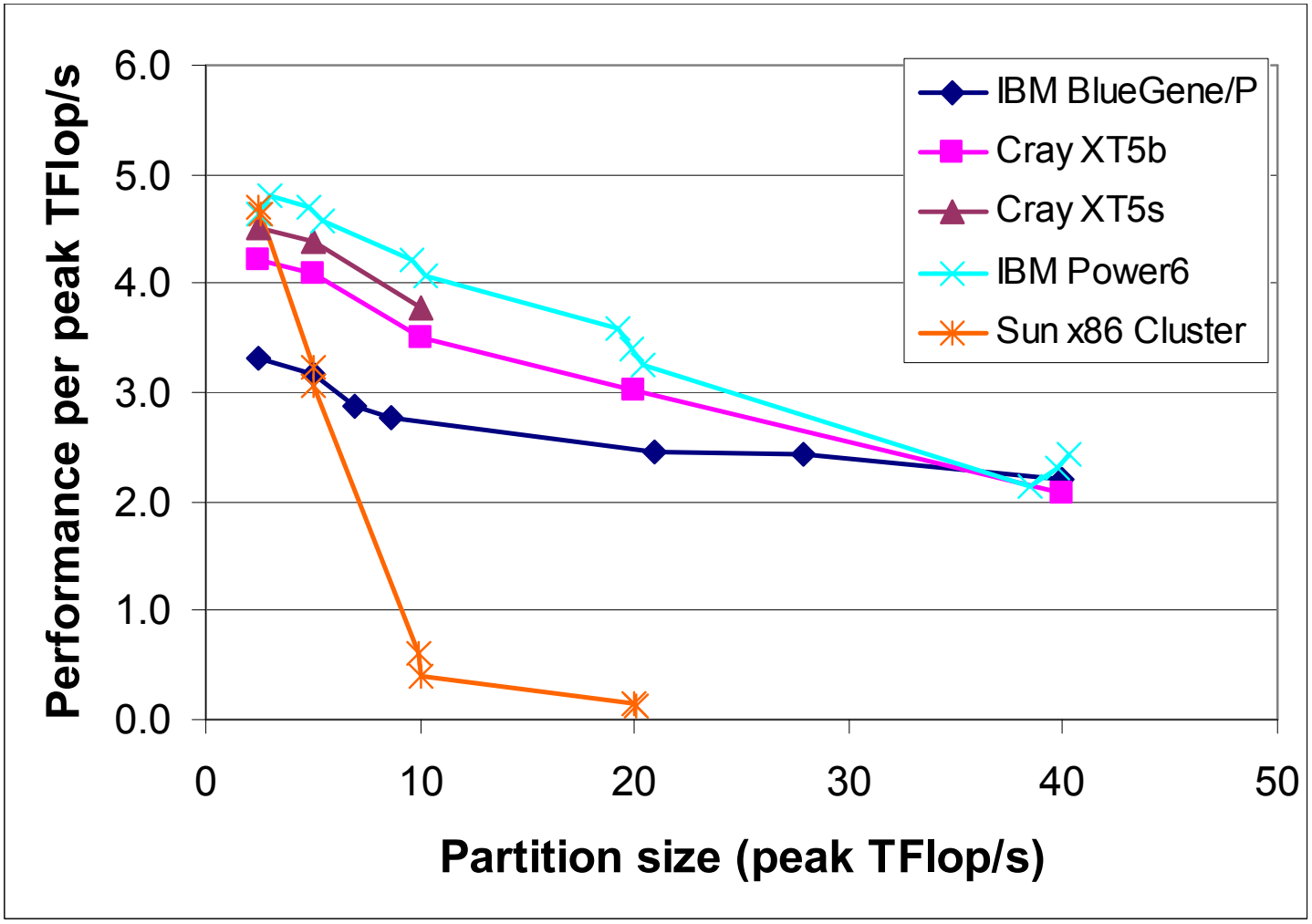


Compiler Optimisations on Cray XT5 Louhi

- Compiler: GCC 4.3.3
- Benchmark: 2 million atoms, 256 compute tasks
- Only modest sensibility to compiler optimisations

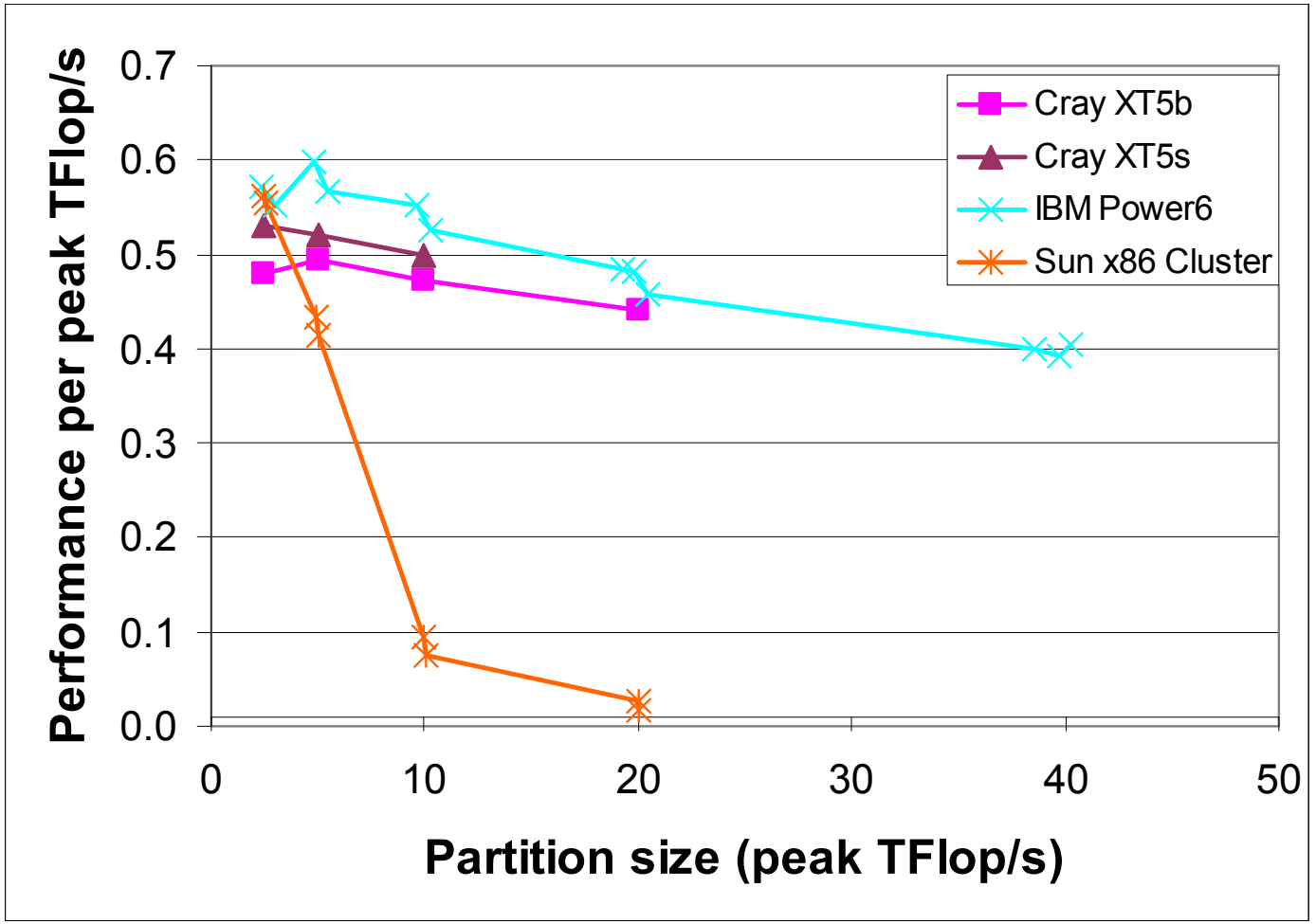
No	Compiler options	2.3GHz Barcelona processor	2.7GHz Shanghai processor
1	`-O2`	**0.204 s**	**0.165 s**
2	`-O3`	**0.203 s**	**0.163 s**
3	`-O3 -funroll-loops`	**0.196 s**	**0.157 s**
4	`-O3 -ffast-math -static -fexpensive-optimizations -fomit-frame-pointer`	**0.203 s**	**0.160 s**
5	`-O3 -ffast-math -static -fexpensive-optimizations -fomit-frame-pointer -funroll-loops`	**0.203 s**	**0.160 s**
6	`-O3 -funroll-loops -ffast-math`	**0.202 s**	**0.160 s**
7	`-O3 -static -fexpensive-optimizations -fomit-frame-pointer -funroll-loops`	**0.196 s**	**0.158 s**

Performance Comparison M1 Benchmark



Performance Comparison M9 Benchmark

- Well rounded overall picture, apart from x86 Cluster (Nehalem/IB)
  - Machine new at the time of the test (Broken? Untuned?)
- Memory/task is still an issue (no BGP for M9, no 40TF point for Louhi)



Outline

- PRACE project overview
- Application enabling work @ EPCC:
  - NAMD
  - HELIUM
    - BCO: Xu Guo
    - Contributor: Andrew Sunderland (STFC Daresbury Laboratory, UK)
- Summary

HELIUM Overview

- Uses time-dependent solutions of the full-dimensional Schrodinger equation to simulate the behavior of helium atoms

- Developed by Queen's University Belfast
  - Has access restrictions

HELIUM Test Case

- A single Fortran 90 file with more than 14000 lines
- Using MPI parallelism
  - Total work in the whole grid space is divided into Blocks
  - Each processor works on one block
  - Multiple decomposition approaches for a fixed total problem size by changing block size and block number
- Particular parameters in source code
  - Control the simulation conditions and execution behaviour
  - E.g. the total problem size, block size, block count, required core number, I/O frequency, etc.
- Test case used in PRACE
  - Fixed problem size: 1540 grid units
  - Total time steps: 80
  - Output frequency: once every 20 time steps

Porting HELIUM to Cray XT5 Louhi

- Compiler selection
  - Pathscale Fortran 90 compiler
  - PGI: reallocation limit compiling issue with large parameter values
- Proper core numbers for execution
  - Depended on the total problem size and block number
  - Wrong core number will lead to execution failure
- Memory requirement
  - Memory limit is the key issue for porting HELIUM
  - Can be roughly estimated based on parameters in source code but no guarantee for upper requirement
  - Execution will freeze up and fail when reaching the memory limit
  - Half-populated or quad-populated may help for porting but performance could be low

Compiling Optimisations on Cray XT5 Louhi

- PathScale 3.2.0
- Higher optimisation level and tuning options
  - **`-O2 --> -O3 -OPT:Ofast`**
- PathScale compiler nested loop optimisations flags
  - **`-LNO:<suboptions>`**
  - Used options for loop fusion, loop unrolling, loop vectorisation
  - Others: loop fission, array padding, etc.
- Test case size of 1540 grid units on Barcelona processors

Cores	-O2	-O3 –OPT:Ofast	-O3 -OPT:Ofast:unroll_analysis=ON -LNO:fusion=2:fission=0:full_unroll_size=2000 :simd=2 -LIST:all_options=ON
630	936 s	731 s	729 s
1540	338 s	316 s	312 s

Reducing MPI Cost

- Removing unnecessary MPI debugging steps at the initialisation stage
  - The modifications has been merged into the latest version HELIUM by the developer

- The effect depends a lot on the system
  - MPI implementations

Cray XT5 Louhi (Barcelona)

Cores	Before	After
630	729 s	729 s
1540	312 s	311 s

IBM Power 6 Huygens

Cores	Before	After
630	714 s	712 s
1540	319 s	309 s

Merging Loops for HELIUM on Cray XT5 Louhi

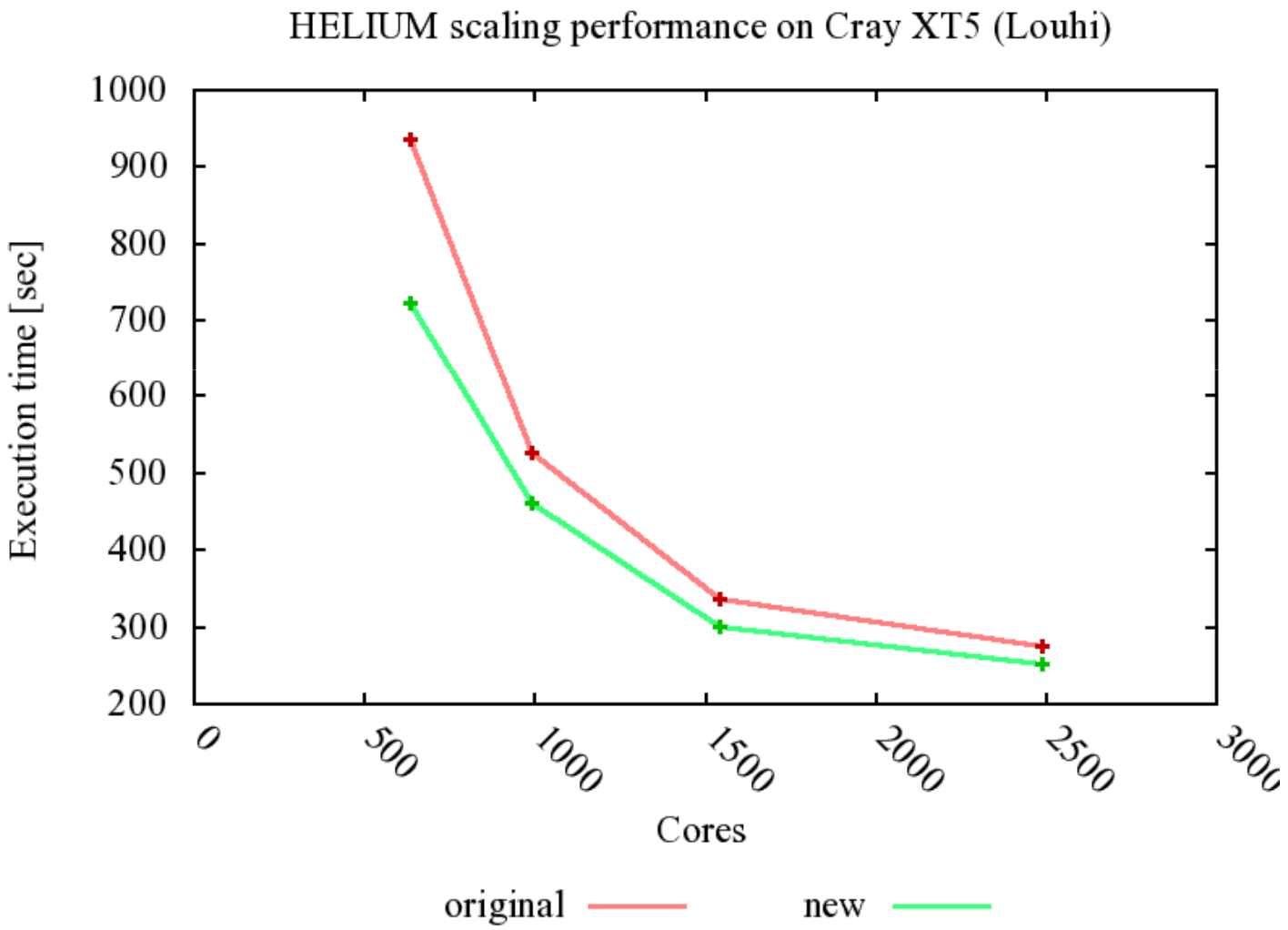
- Expensive routines with multiple calculation loops
  - CrayPat profiling
- Merge loops with the same boundaries together
- Loop through in a proper order
- Test case size of 1540 grid units on Barcelona processors
  - Performance improved
  - L1 / L2 cache misses reduced

Cores	Before	After
630	729 s	723 s
1540	311 s	302 s

Cores	L1 Cache misses		L2 Cache misses	
	Before	After	Before	After
630	6.62E+09	2.30E+09	2.97E+09	7.05E+08
1540	1.69E+08	1.50+E08	1.63E+08	9.02E+07

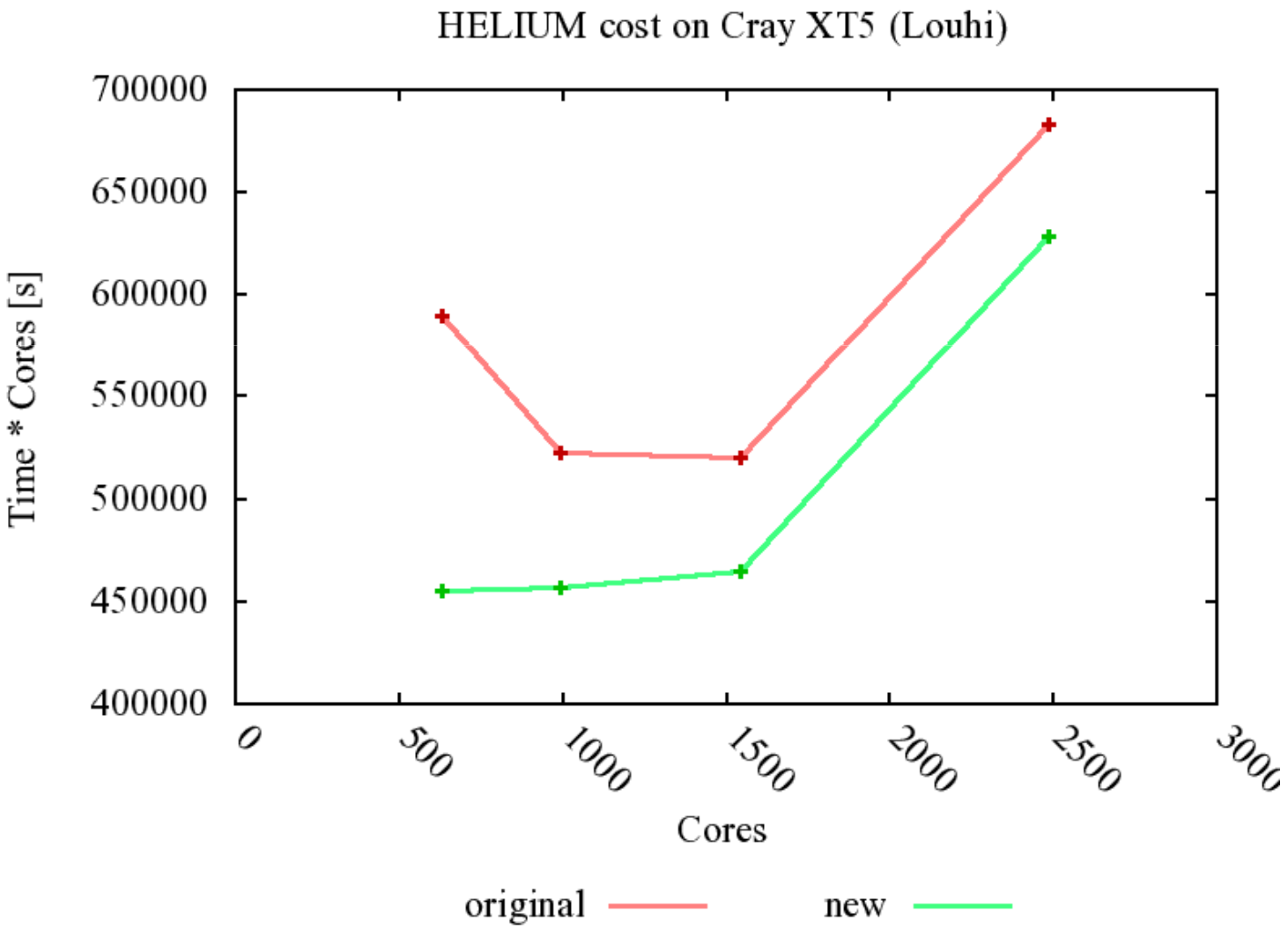
HELIUM Performance on Cray XT5 Louhi

- Test case of fixed problem size 1540 grid units



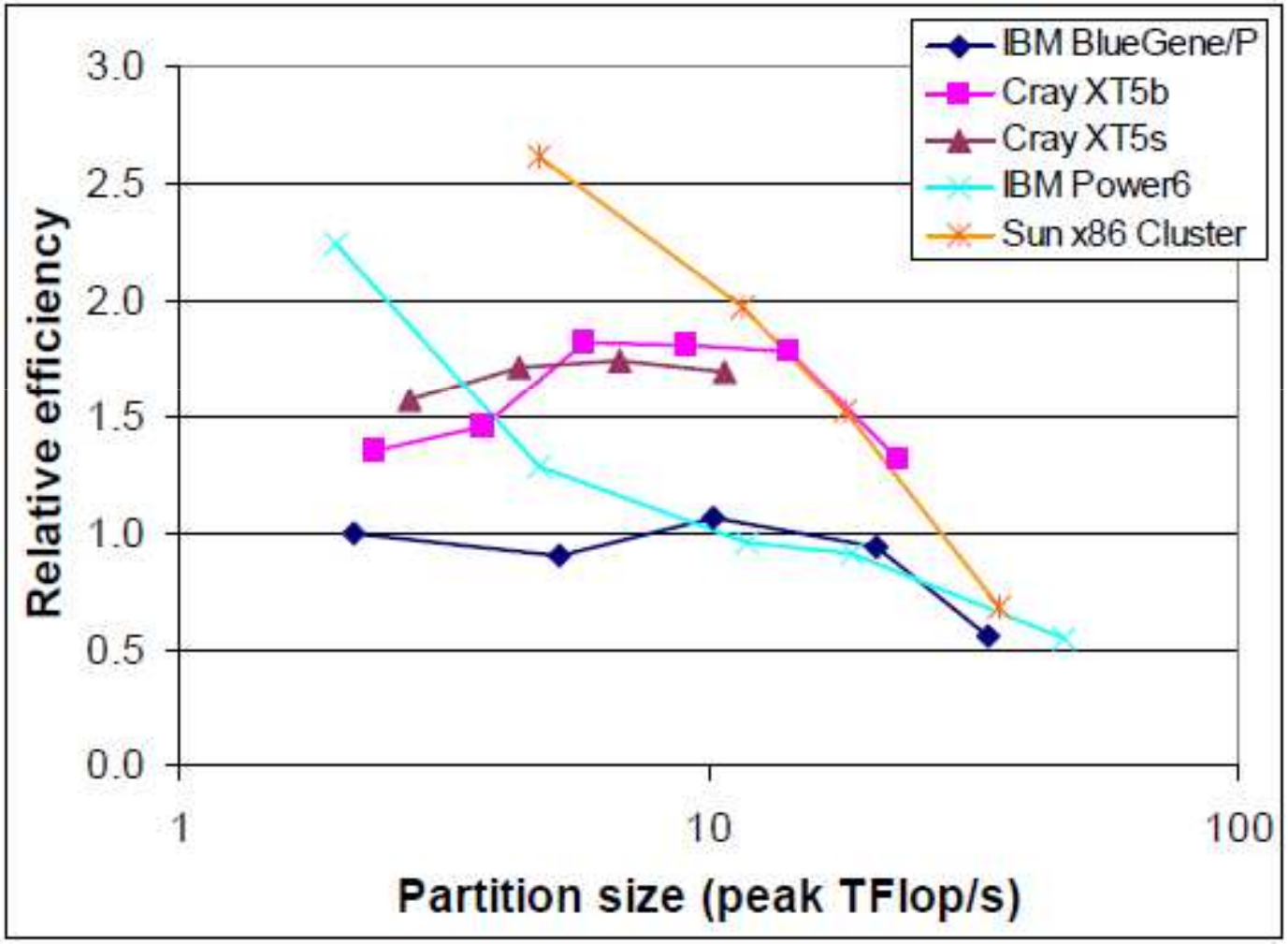
HELIUM Scaling Cost on Cray XT5 Louhi

- Scaling Cost = Execution time * Core number
  - A horizontal cost curve implies a linear scaling



Comparison on PRACE Prototypes

- Test case of fixed problem size 1540 grid units



Summary

- PRACE
  - In 2008/2009, “The progress of the preparatory phase project is satisfactory in all areas” – Project review 28/10/2009, Brussels
  - PRACE is ready to start the Implementation Phase in 2010
- Application enabling work @ EPCC
  - NAMD & HELIUM
  - Latest versions were ported to PRACE prototypes, including Cray XT5
  - Improved the performance and scalability
  - Both applications fit for the future petaflop systems of the forth coming phase
- EPCC is looking forward to continuing the contributions in the PRACE Implementation Phase