Application performance on the UK's new HECToR service

Fiona Reid[1,2], Mike Ashworth[1,3], Thomas Edwards[2],
Alan Gray[1,2], Joachim Hein[1,2], Alan Simpson[1,2],
Peter Knight[4], Kevin Stratford[1,2], Michele Weiland[1,2]

[1]HPCX Consortium
[2]EPCC, The University of Edinburgh
[3]STFC Daresbury Laboratory
[4]EURATOM/UKAEA Fusion Association

CUG May 5-8th 2008

Acknowledgements

- STFC: Roderick Johnstone
- Colin Roach, UKAEA and Bill Dorland, University of Maryland for assistance porting GS2 to HECToR and supplying the NEV02 benchmark
- Jim Philips, UIUC and Ghengbin Zheng, UIUC for their assistance installing NAMD on HECToR

Overview

- System Introductions
- Synthetic Benchmark Results
- Application Benchmark Results
- Conclusions

Systems for comparison

- HPCx (Phase 3): 160 IBM e-Server p575 nodes
  - SMP cluster, 16 Power5 1.5 GHz cores per node
  - 32 GB of RAM per node (2 GB per core)
  - IBM HPS interconnect (aka Federation)
  - 12.9 TFLOP/s Linpack, No 101 on top500

- HECToR (Phase 1): Cray XT4
  - MPP, 5664 nodes, 2 Opteron 2.8 GHz cores per node
  - 6 GB of RAM per node (3 GB per core)
  - Cray Seastar2 torus network
  - 54.6 TFLOP/s Linpack, No 17 on top500

- Also included in some plots:
  - HECToR Test and Development system (TDS)
    - Cray XT4, 64 nodes: 2.6 GHz dual core, 4 GB RAM/node

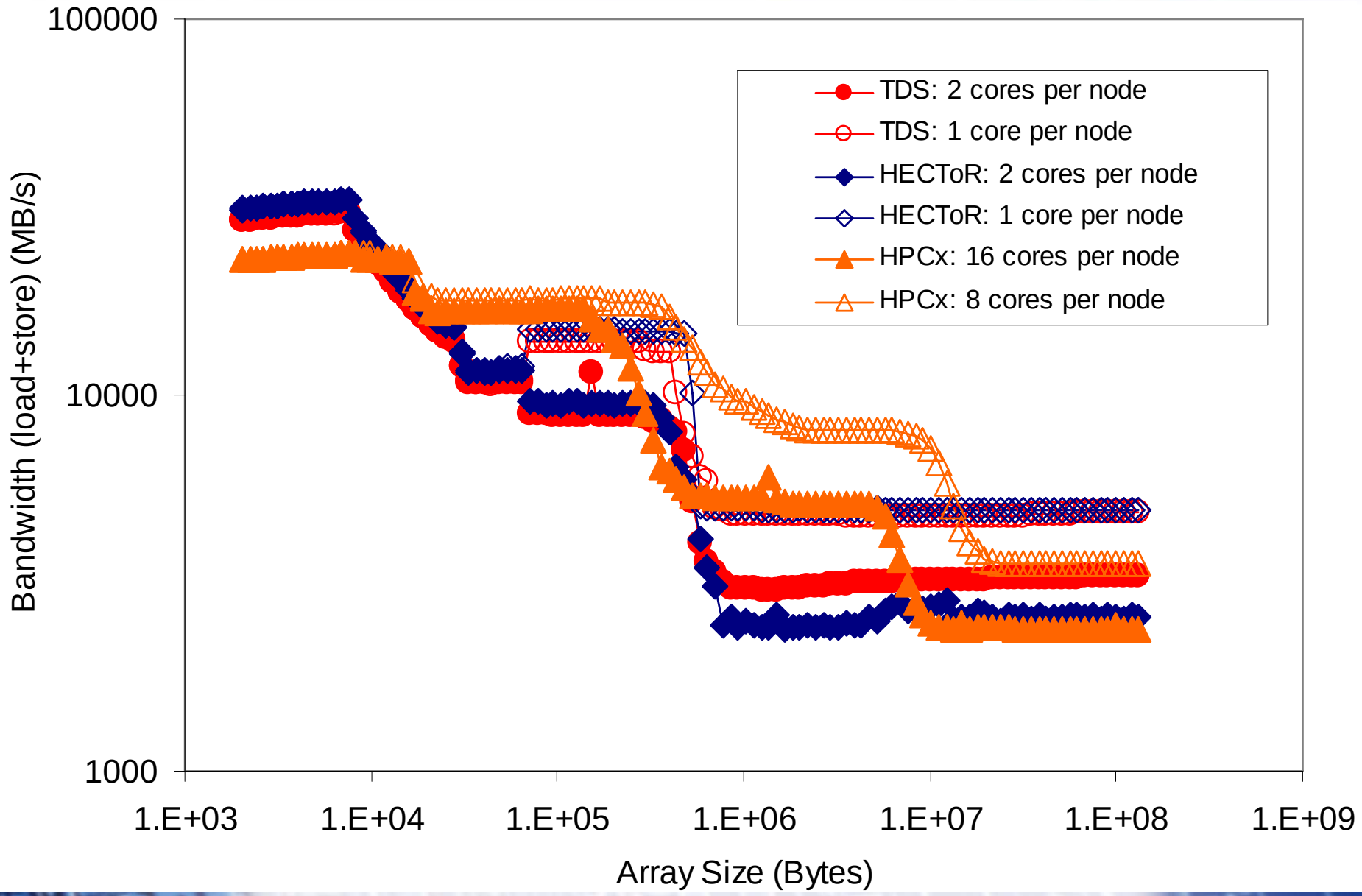
System Comparison (cont)

	HPCx	**HECToR**
Chip	IBM Power5 (dual core)	AMD Opteron (dual core)
Clock	1.5 GHz	2.8 GHz
FPUs	2 FMA	1 M, 1 A
Peak Perf/core	6.0 GFlop/s	5.6 GFlop/s
cores	2560	11328
Peak Perf	15.4 TFLOP/s	63.4 TFLOP/s
Linpack	12.9 TFLOP/s	54.6 TFLOP/s

Synthetic Benchmarks

- Memory Bandwidth
  - Streams
- MPI Bandwidth
  - Intel MPI Benchmarks
    - PingPing

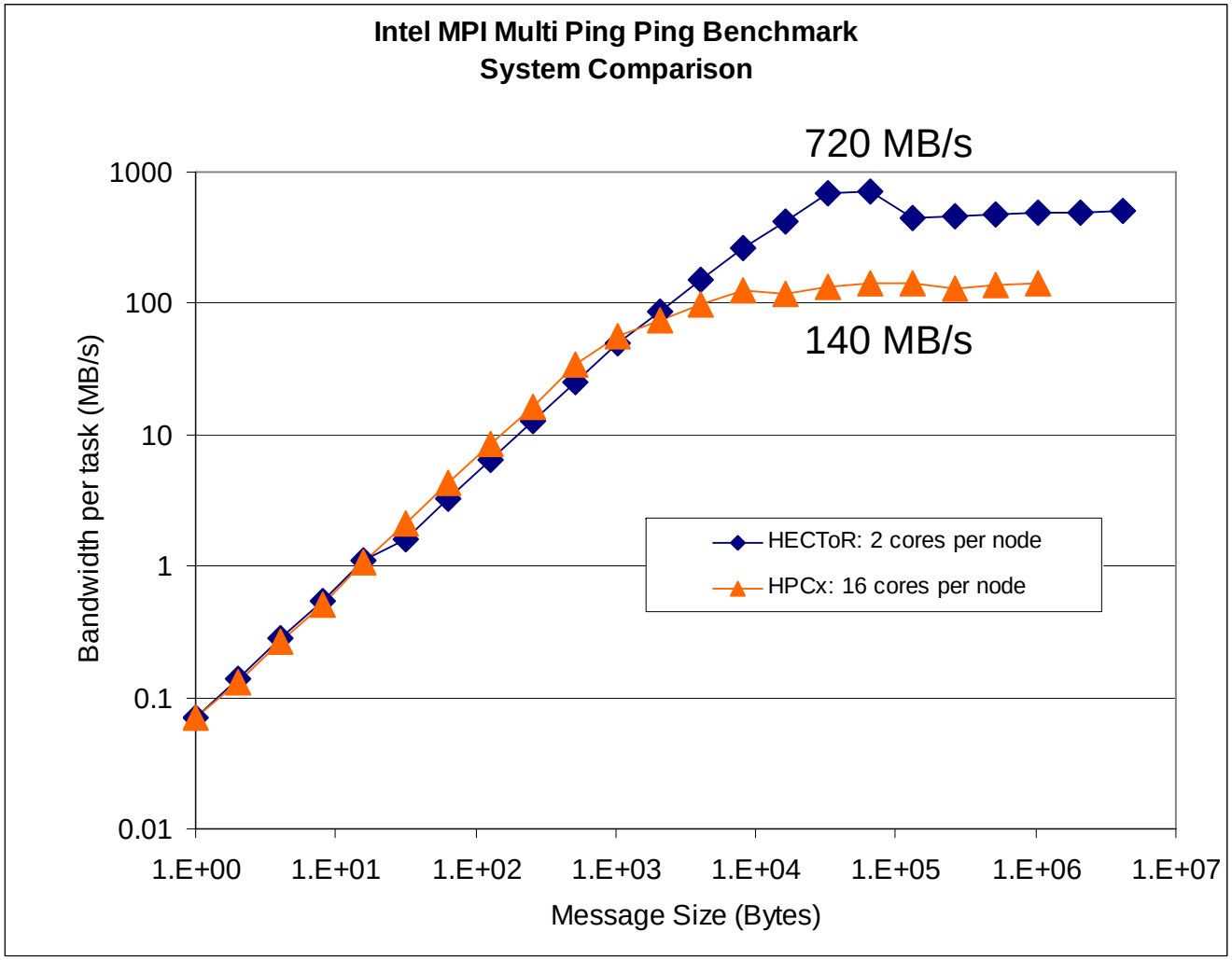
Memory bandwidth - Streams



Memory bandwidth - Streams

- Can clearly see caches
- HECToR better at L1, slightly better on main memory
  - HPCx has advantage for intermediate array sizes.
- Underpopulating nodes (1 core per chip) gives improvements on both systems
  - memory bandwidth cannot sustain 2 cores per chip
  - HECToR worse than HPCx, especially on main memory
  - Of course, 1 core/chip means double the resource for same no. tasks
- TDS has lower clock rate than HECToR, but has higher bandwidth from main memory!
  - 4=2+2 GB RAM on TDS is symmetric, interleaving possible
  - 6=4+2 GB RAM on HECToR only allows partial interleaving

MPI bandwidth - PingPing



HPCx reaches saturation point earlier – HECToR may scale better

On both systems the latency (via IMB PingPong) ~5.5µs

AlltoAll - HPCx has the advantage for small (<100 bytes) messages, HECToR outperforms HPCx for larger messages

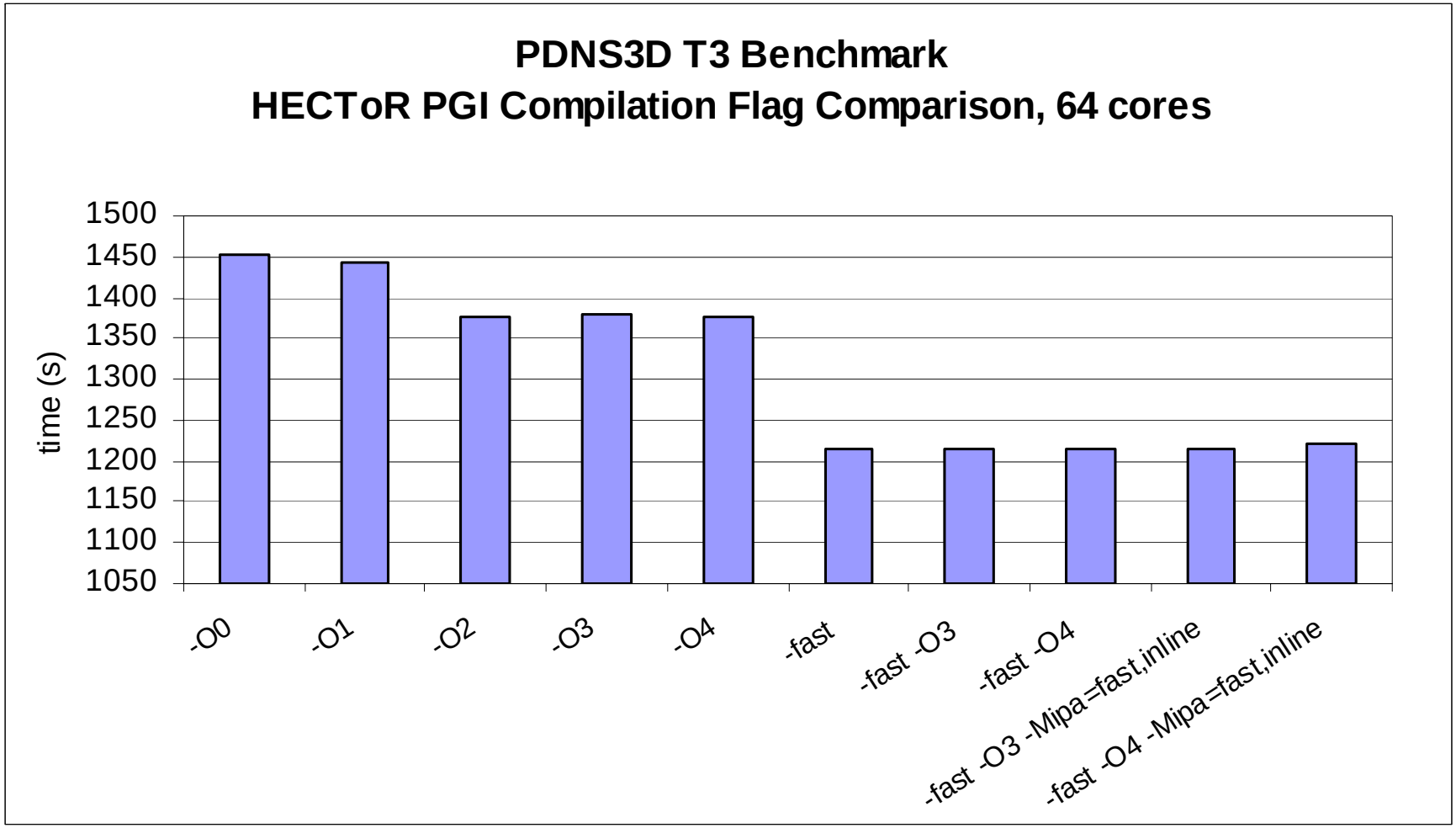
Applications

- Fluid Dynamics
  - PDNS3D
  - Ludwig
- Fusion
  - Centori
  - GS2
- Ocean Modelling
  - POLCOMS
- Molecular Dynamics
  - DL_POLY
  - LAMMPS
  - GROMACS (see paper)
  - NAMD
  - AMBER (see paper)

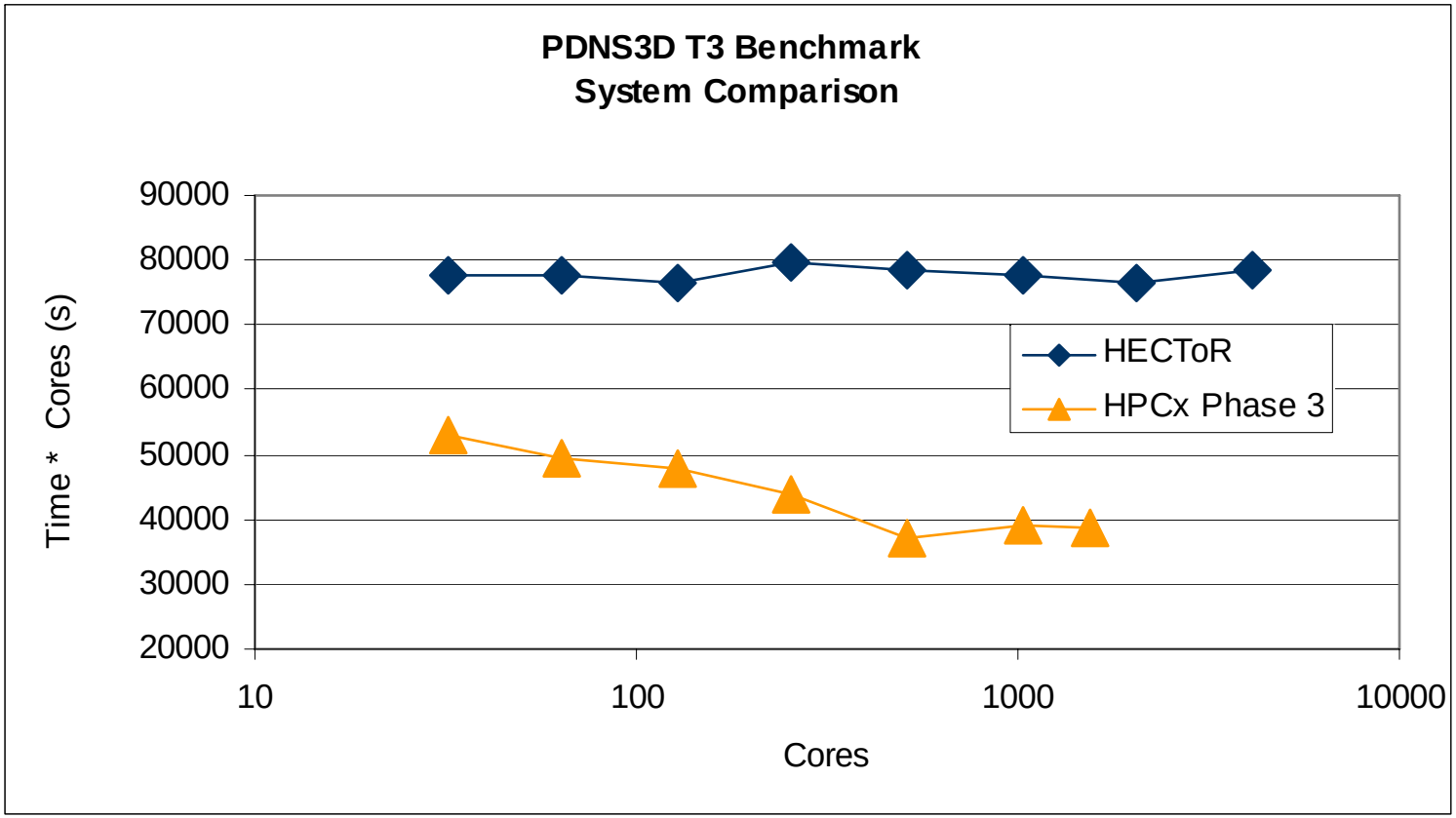
Fluid Dynamics: PDNS3D (PCHAN)

- Finite difference code for Turbulent Flow
  - shock/boundary layer interaction (SBLI)
- Simulates the flow of fluids to study turbulence
- T3 benchmark - Involves a 360x360x360 grid
- Developed by Neil Sandham, University of Southampton

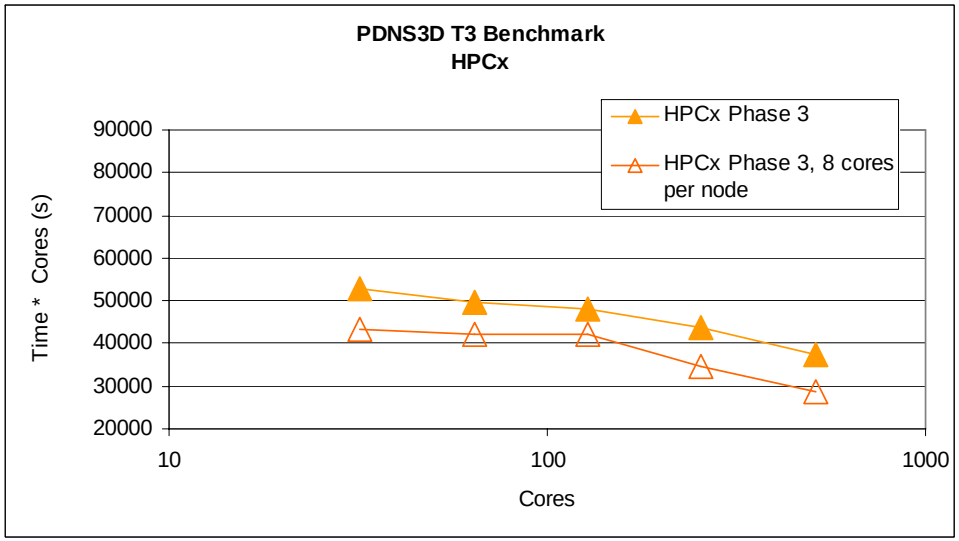
PDNS3D – compilation optimization

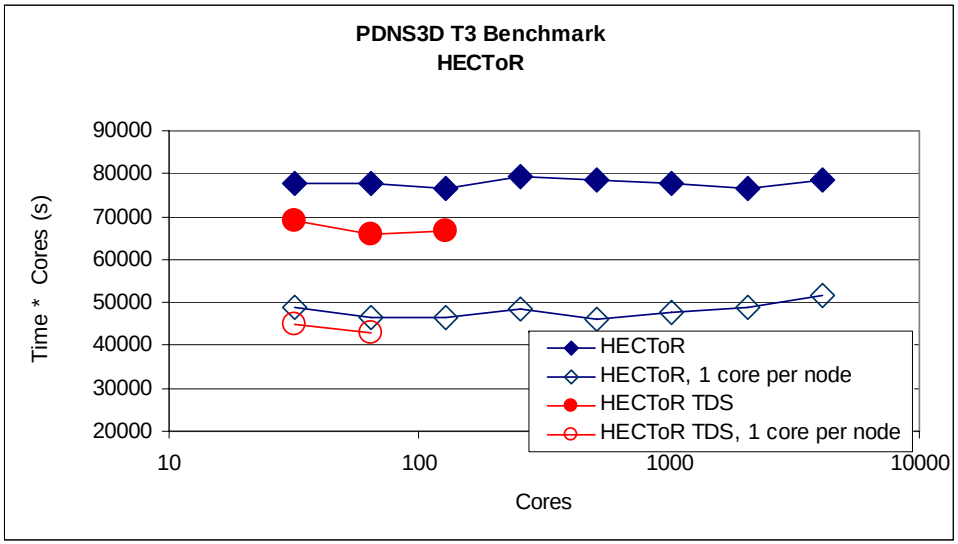


PDNS3D – system comparison



PDNS3D Memory Bandwidth sensitivity

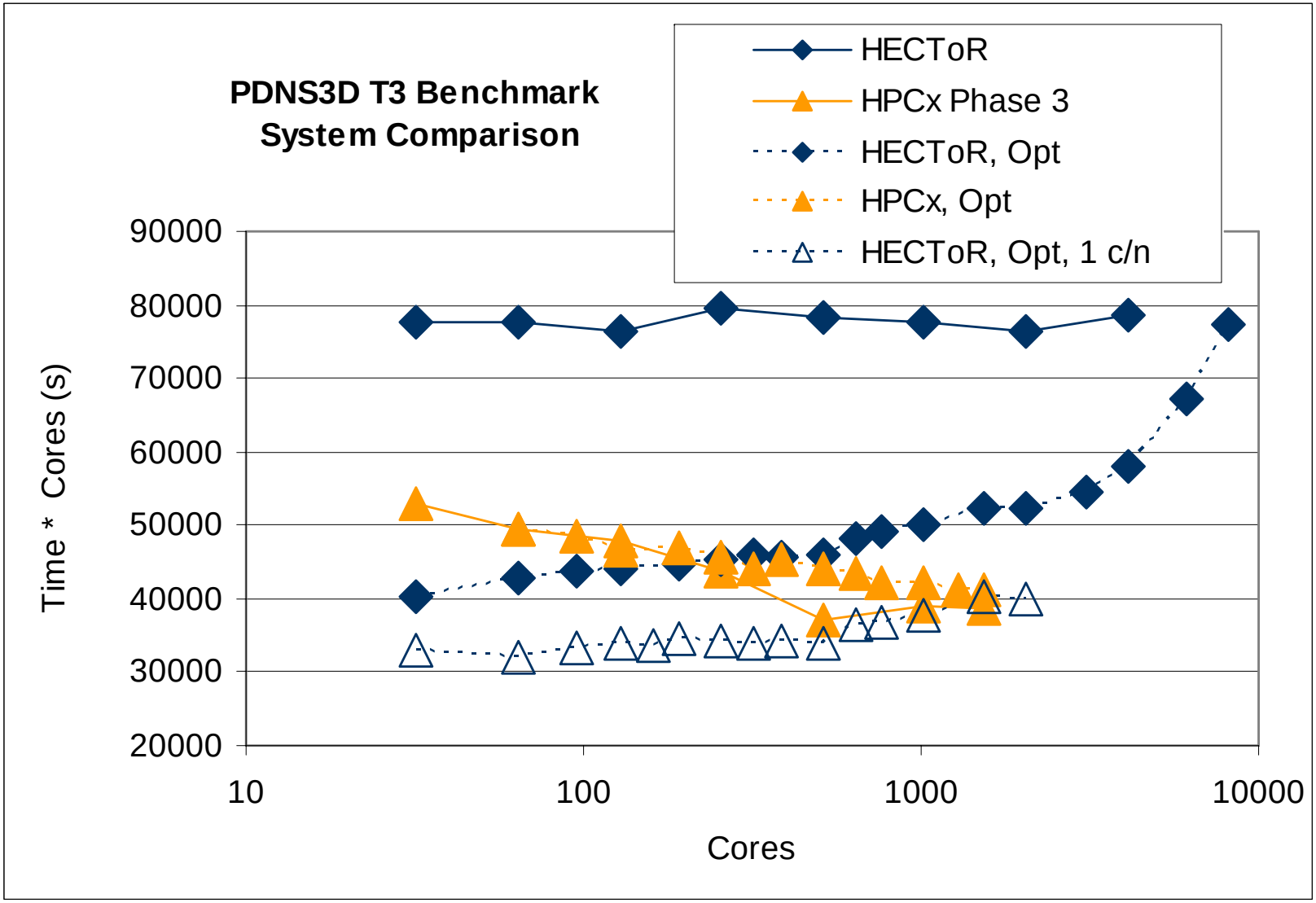




- Underpopulating nodes gives huge improvement (in terms of performance/core) on HECToR, slight improvement on HPCx
- TDS outperforms HECToR
- c.f. streams

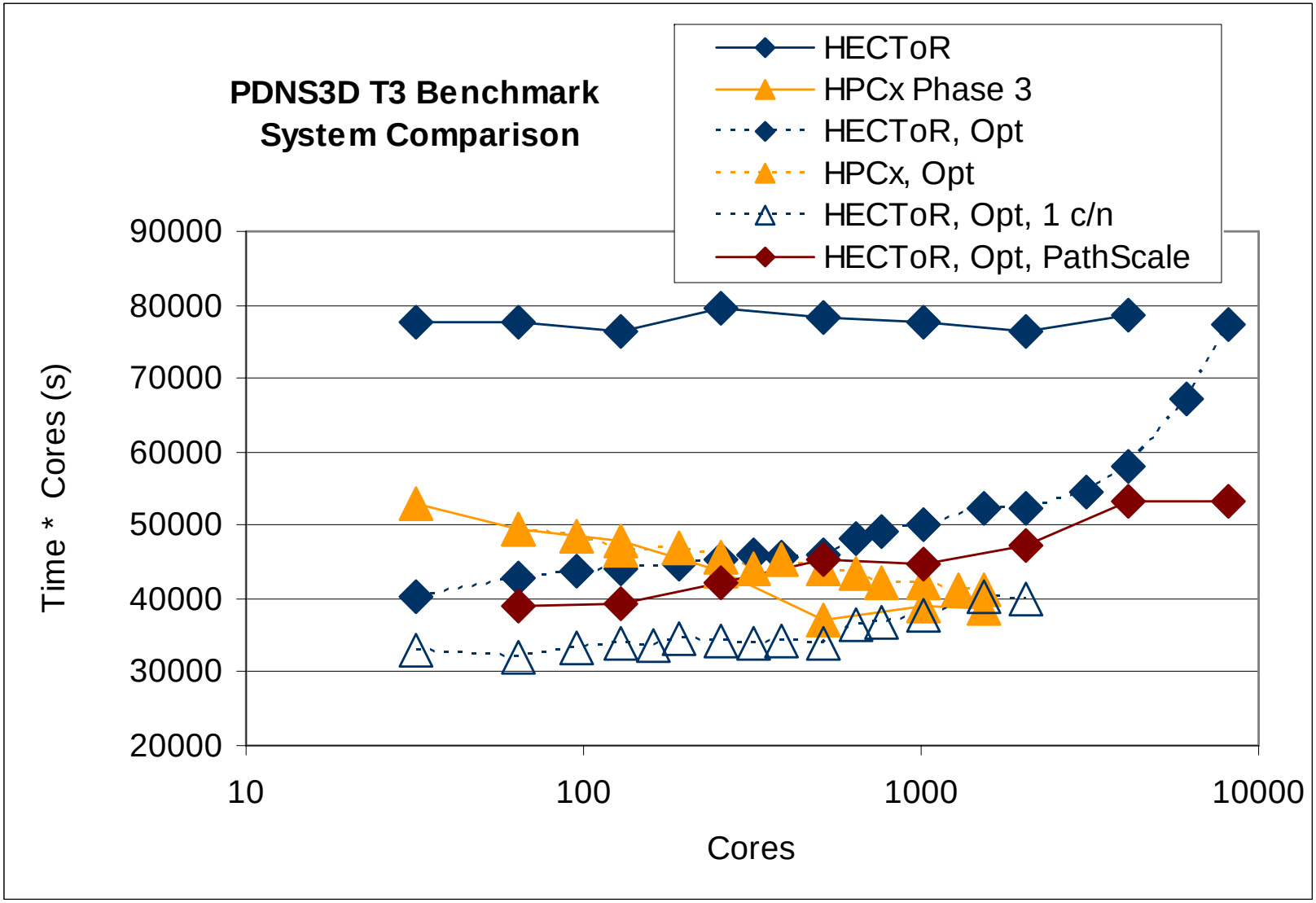
PDNS3D – Optimised version

- New optimised version less sensitive to memory bandwidth

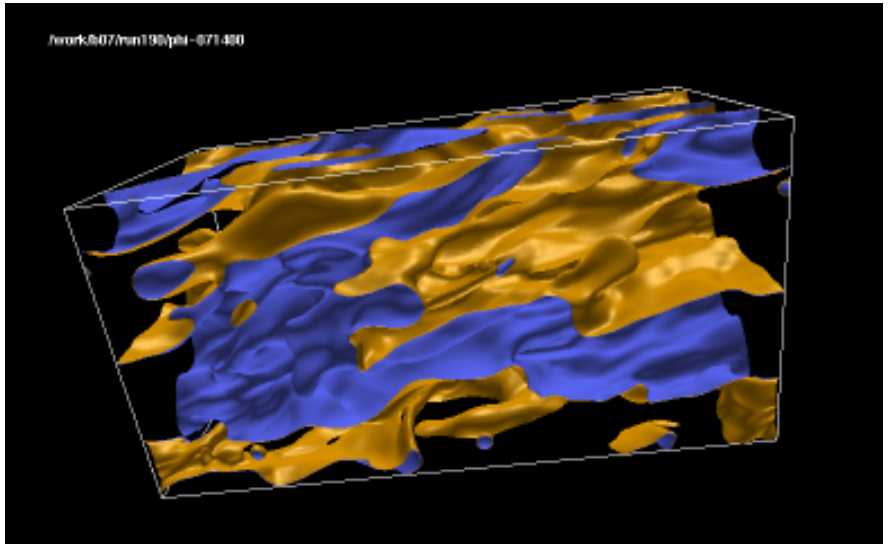


PDNS3D – Optimised version

- PathScale gives a further 10-15% improvement

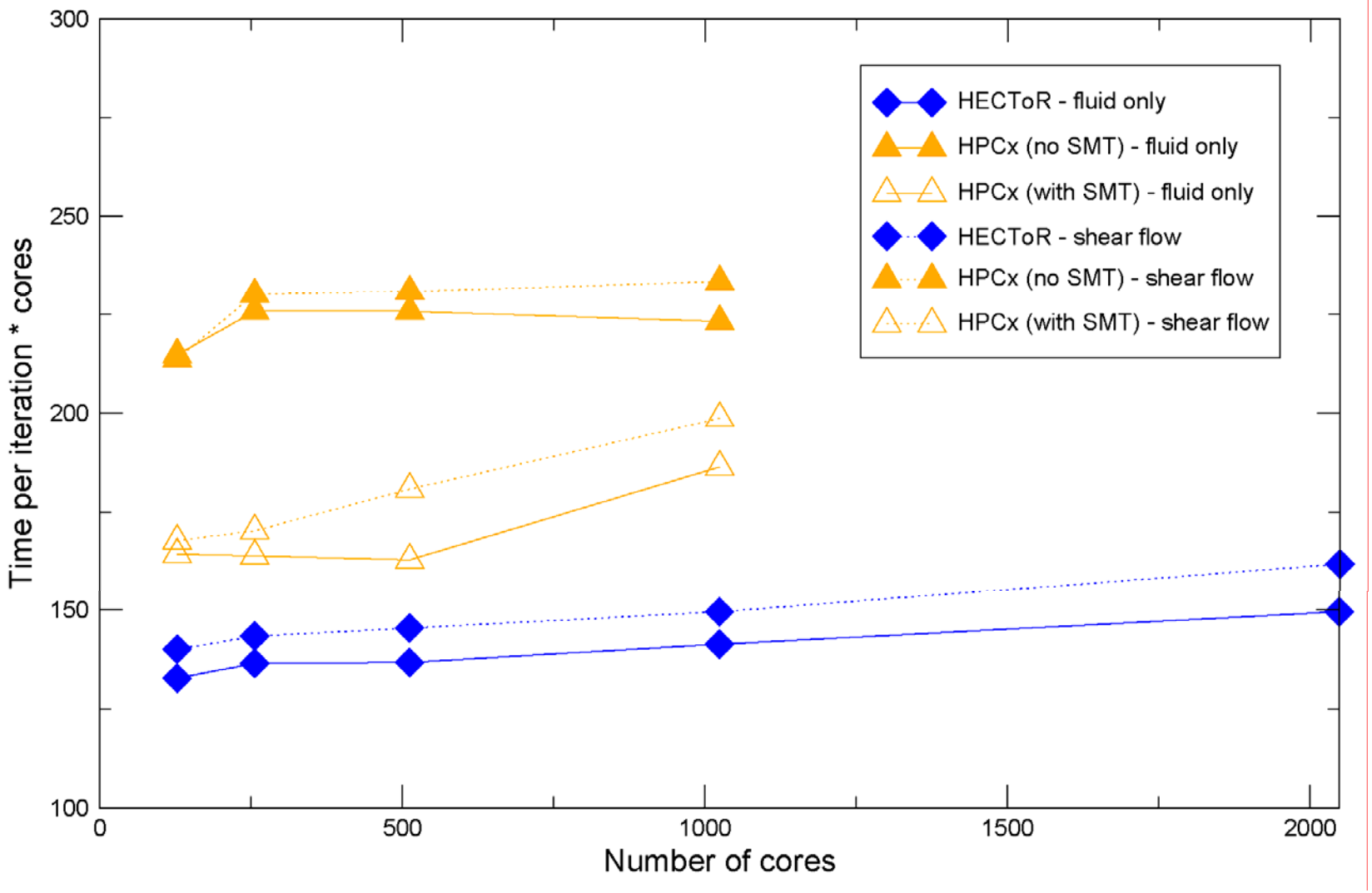


Fluid dynamics - Ludwig

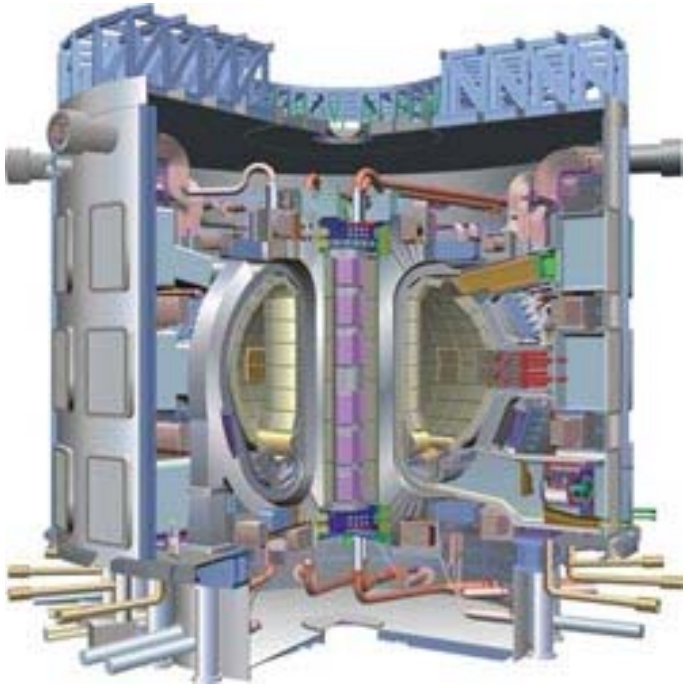


- Ludwig
  - Lattice Boltzmann code for solving the incompressible Navier-Stokes equations
  - Used to study complex fluids
  - Code uses a regular domain decomposition with local boundary exchanges between the subdomains
  - Two problems considered, one with a binary fluid mixture, the other with shear flow

Ludwig 256x512x256 lattice



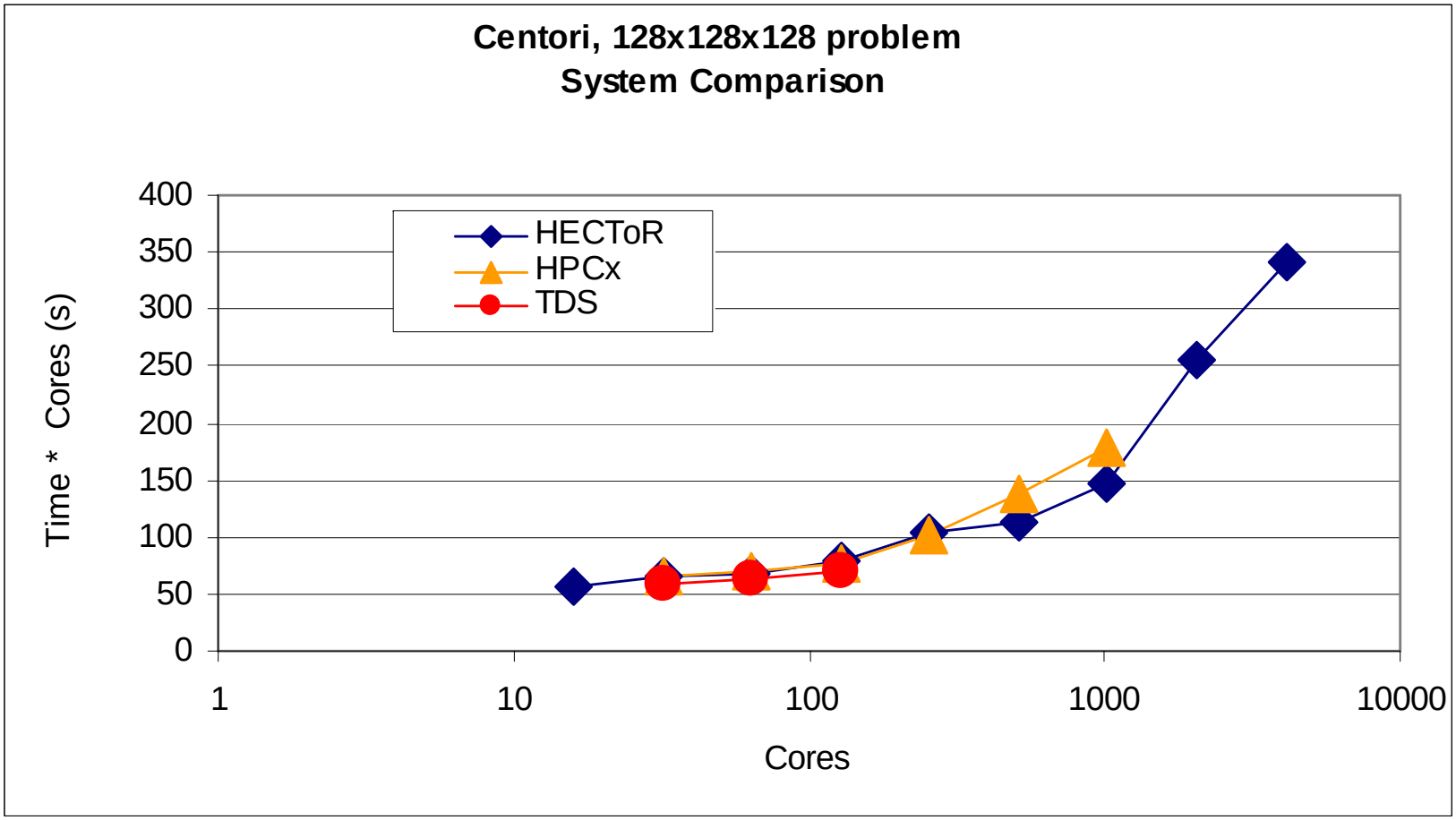
Fusion

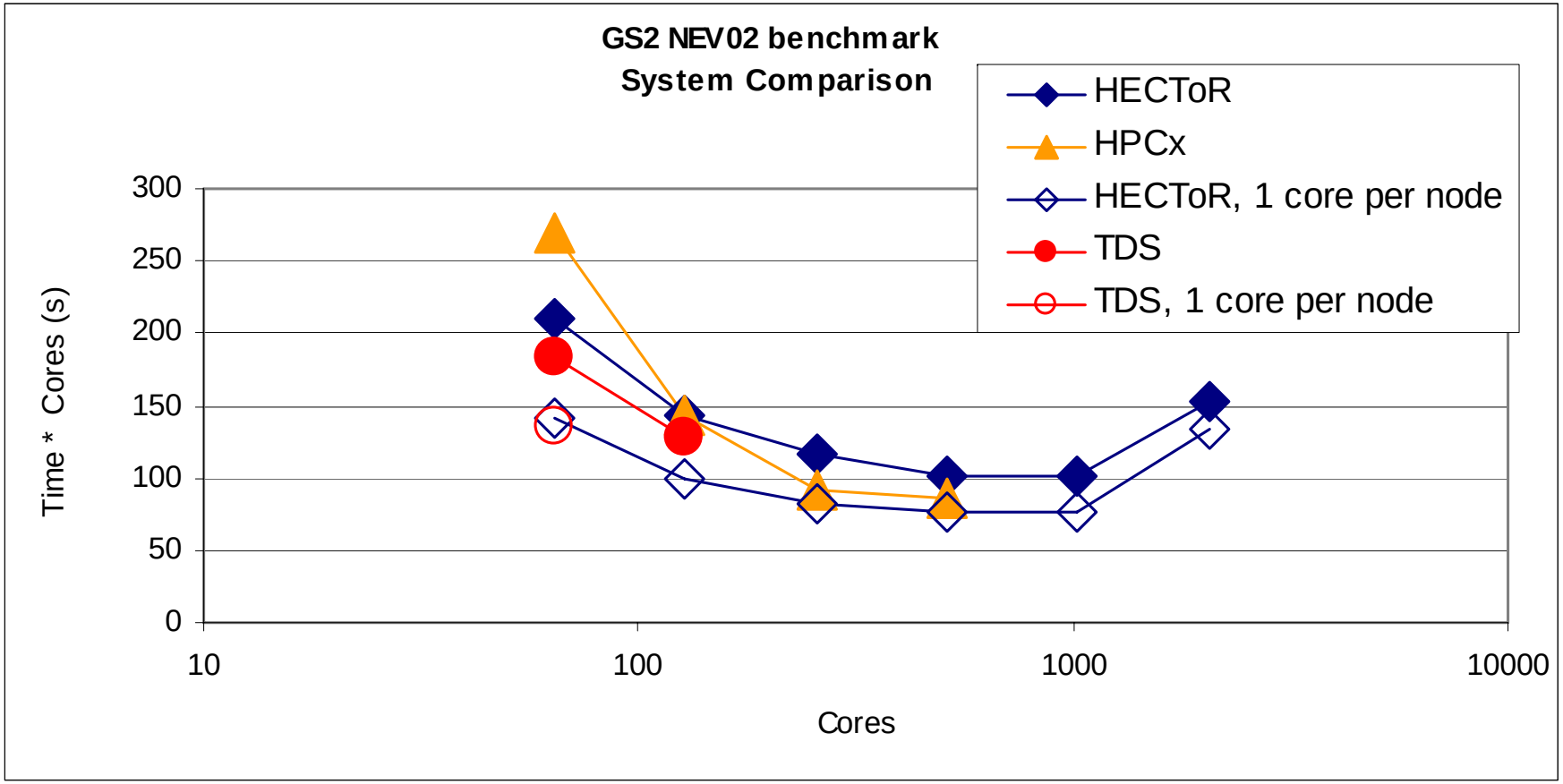


ITER tokamak reactor

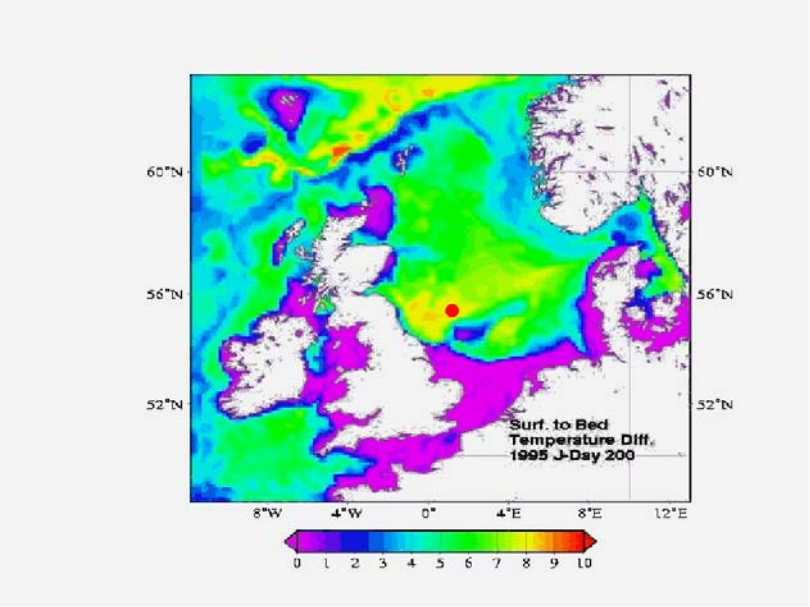
(www.iter.org)

- Centori
  - simulates the fluid flow inside a tokamak reactor developed by UKAEA Fusion in collaboration with EPCC
- GS2
  - Gyrokinetic simulations of low-frequency turbulence in tokamak developed by Bill Dorland et al.


Centori, 128x128x128 problem
System Comparison
Time * Cores (s)
400
350
300
250
200
150
100
50
0
HECToR
HPCx
TDS
1
10
100
1000
10000
Cores

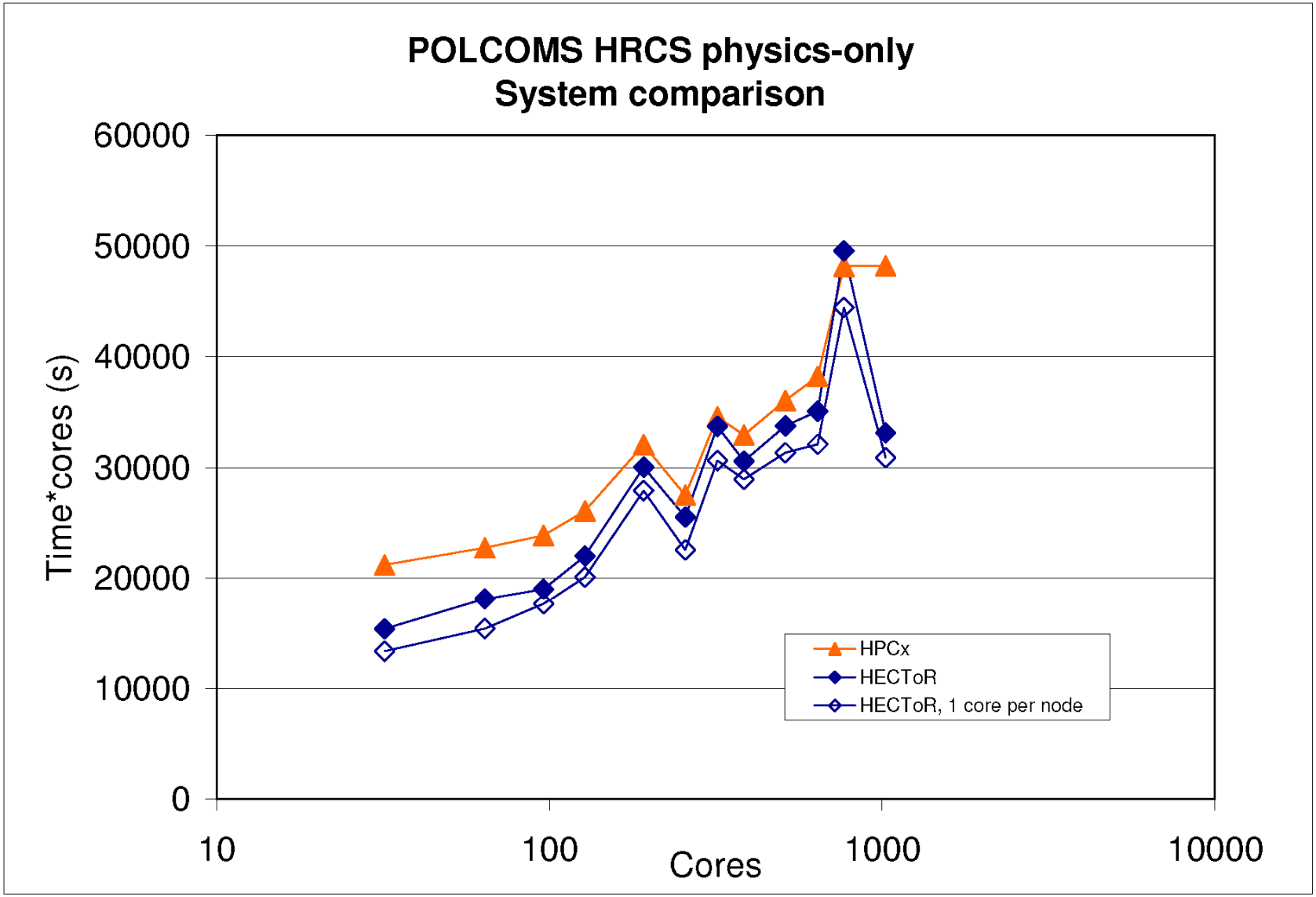

GS2 NEV02 benchmark
System Comparison
HECToR
HPCx
HECToR, 1 core per node
TDS
TDS, 1 core per node
Time * Cores (s)
300
250
200
150
100
50
0
10
100
1000
10000
Cores

Ocean Modelling: POLCOMS

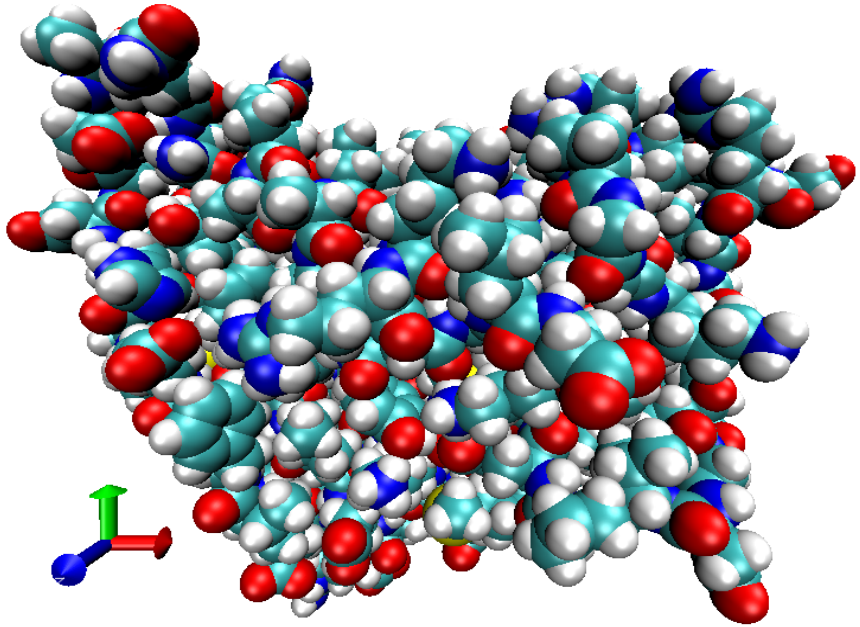


- Proudman Oceanographic Laboratory Coastal Ocean Modelling System (POLCOMS)
  - Simulation of the marine environment
  - Applications include coastal engineering, offshore industries, fisheries management, marine pollution monitoring, weather forecasting and climate research
  - Uses 3-dimensional hydrodynamic model

Ocean Modelling: POLCOMS



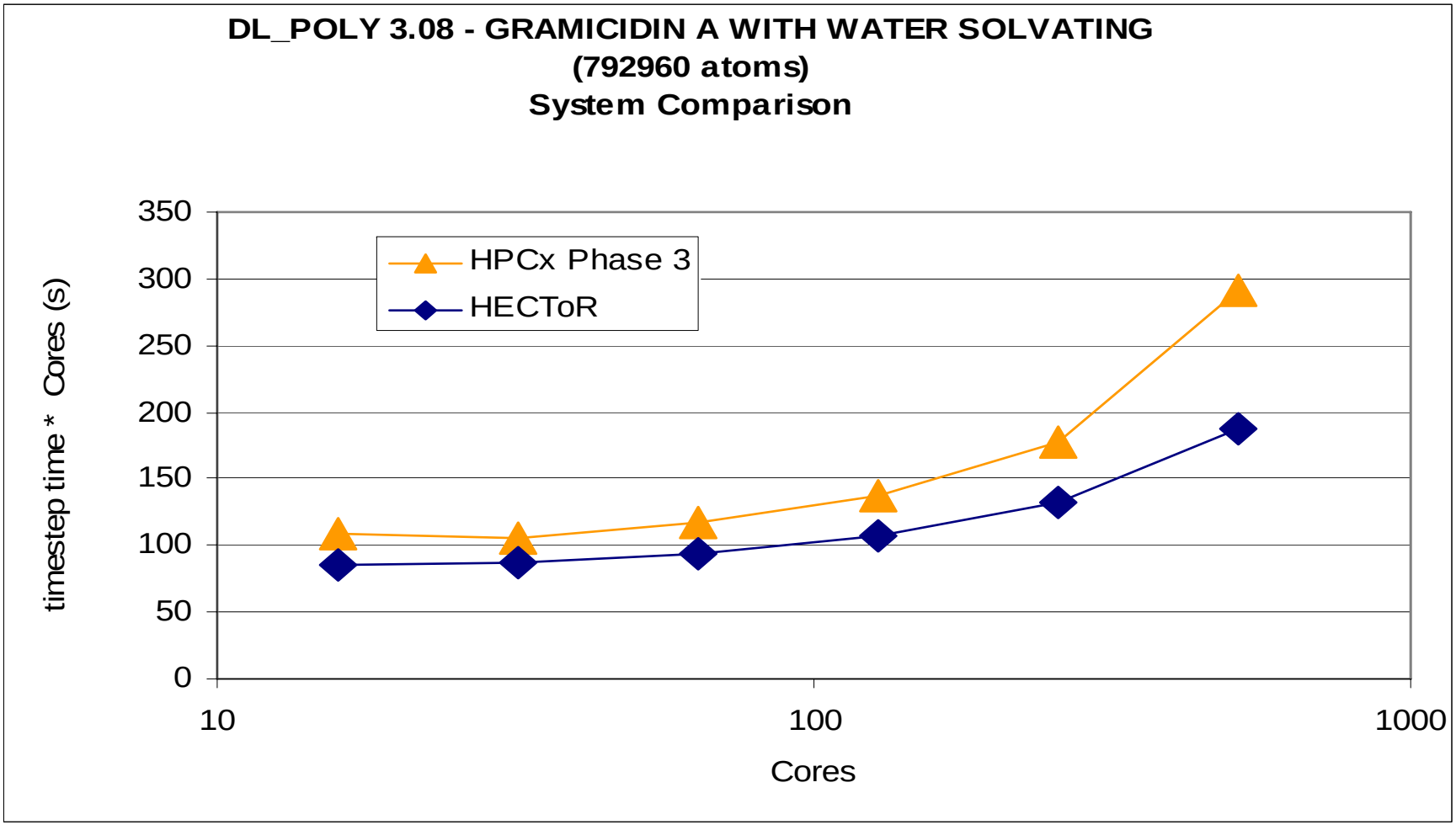
Molecular dynamics



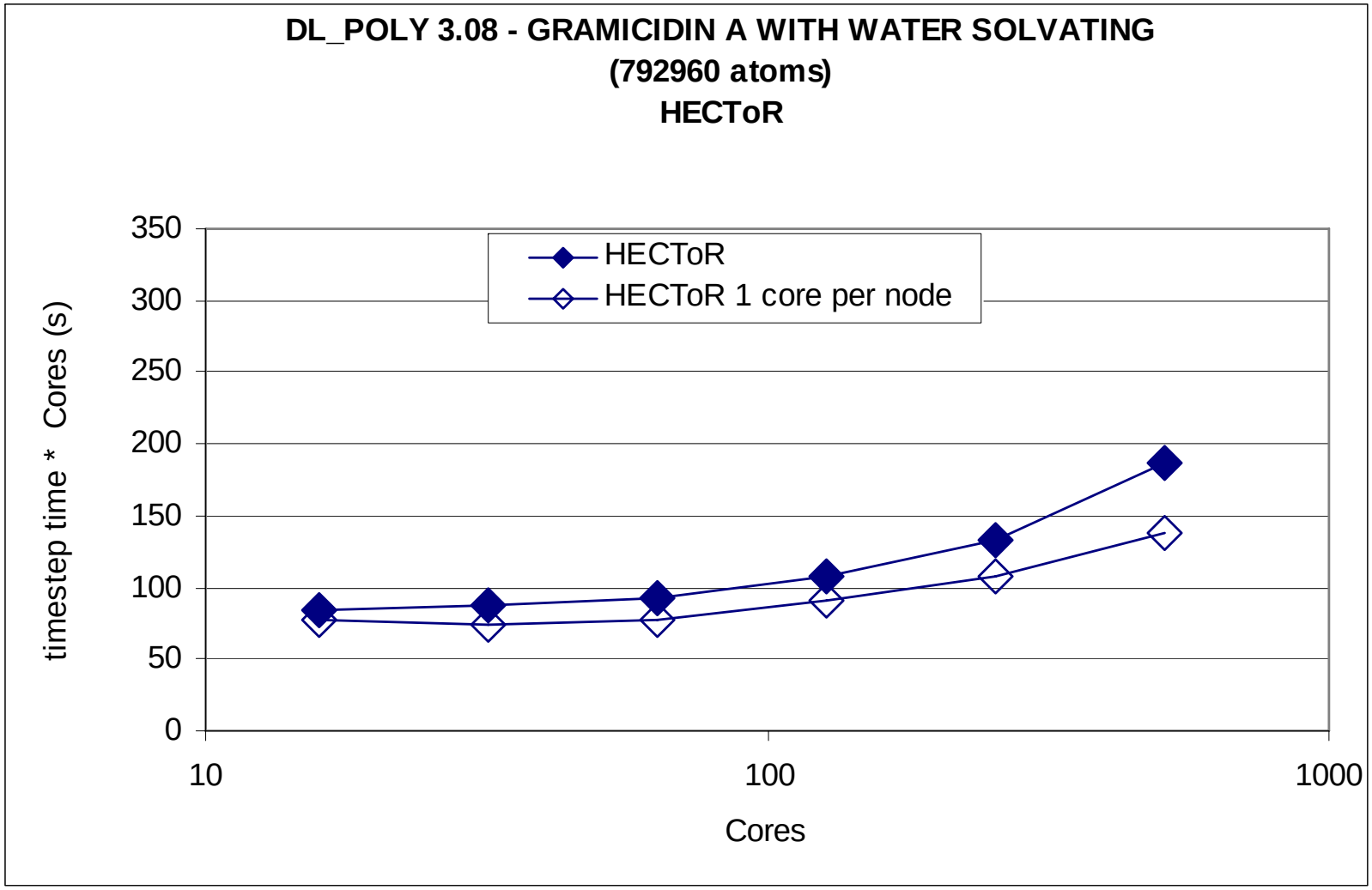
Protein Dihydrofolate Reductase

- DL_POLY
  - general purpose molecular dynamics package which can be used to simulate systems with very large numbers of atoms
- LAMMPS
  - Classical Molecular Dynamics - can simulate wide range of materials
- NAMD
  - classical molecular dynamics code designed for high-performance simulation of large biomolecular systems
- AMBER
  - General purpose biomolecular simulation package
- GROMACS
  - General purpose MD package - specialises in biochemical systems, e.g. protiens, lipids etc

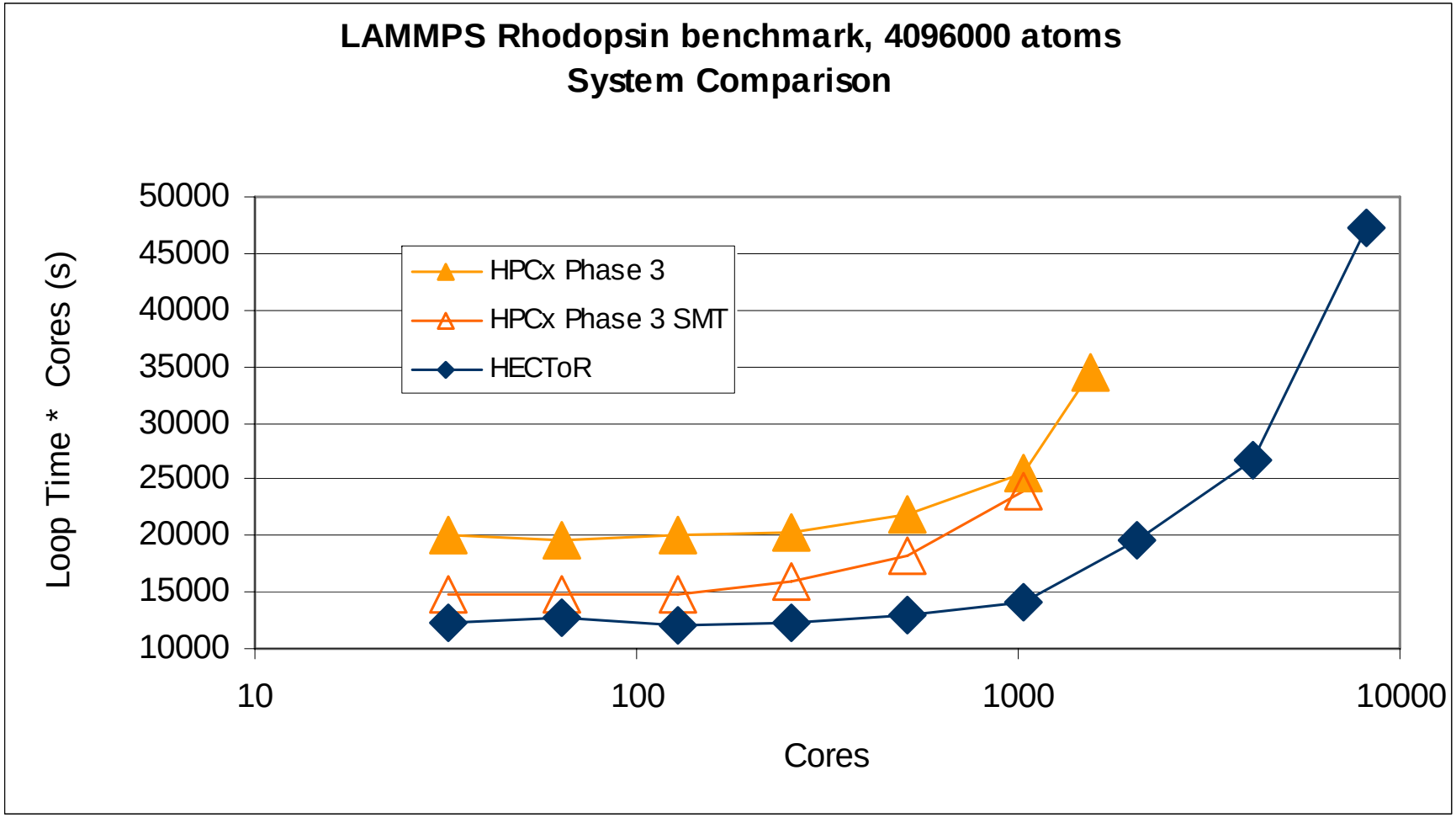
DL_POLY – system comparison



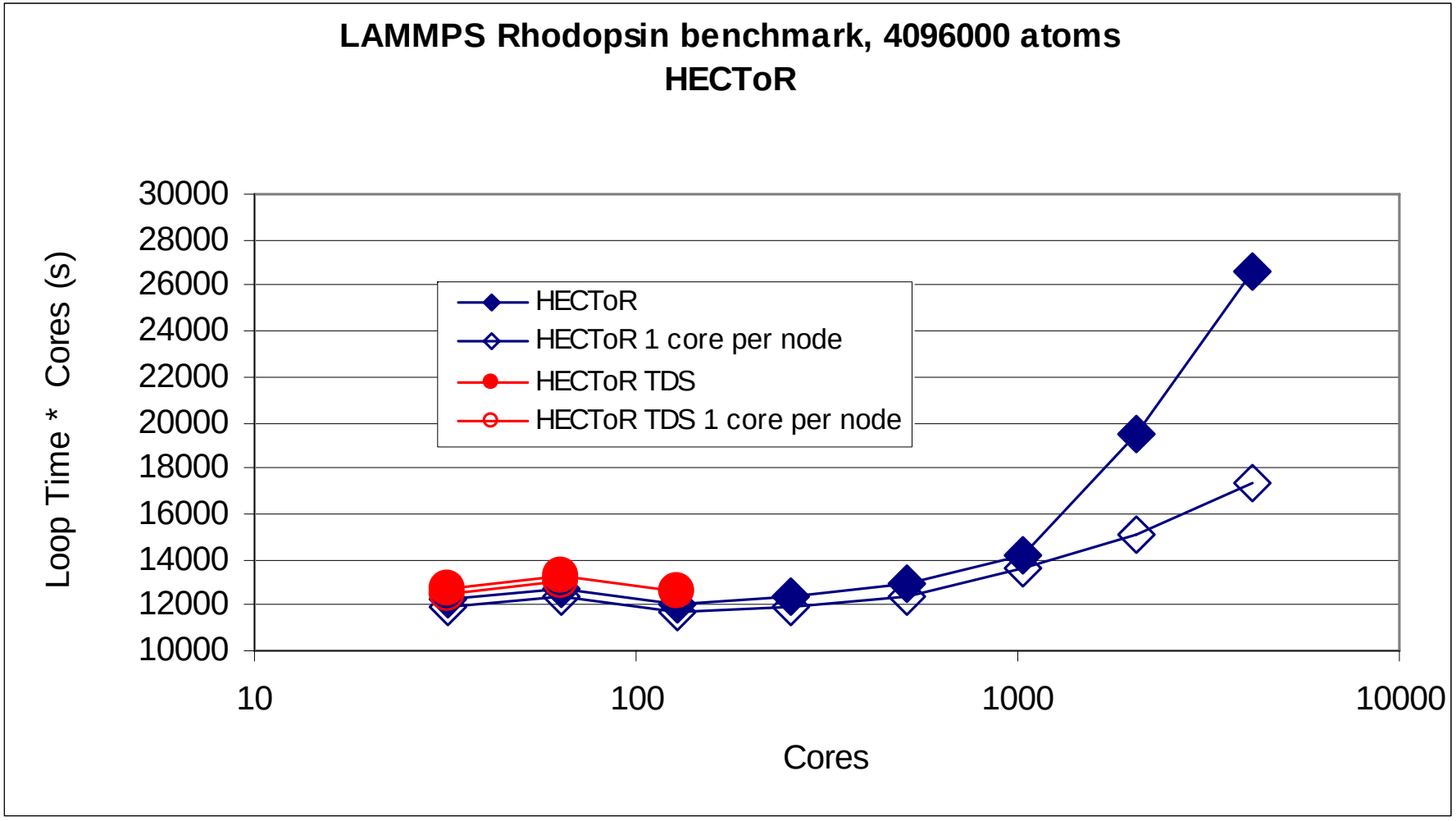
DL_POLY – system comparison



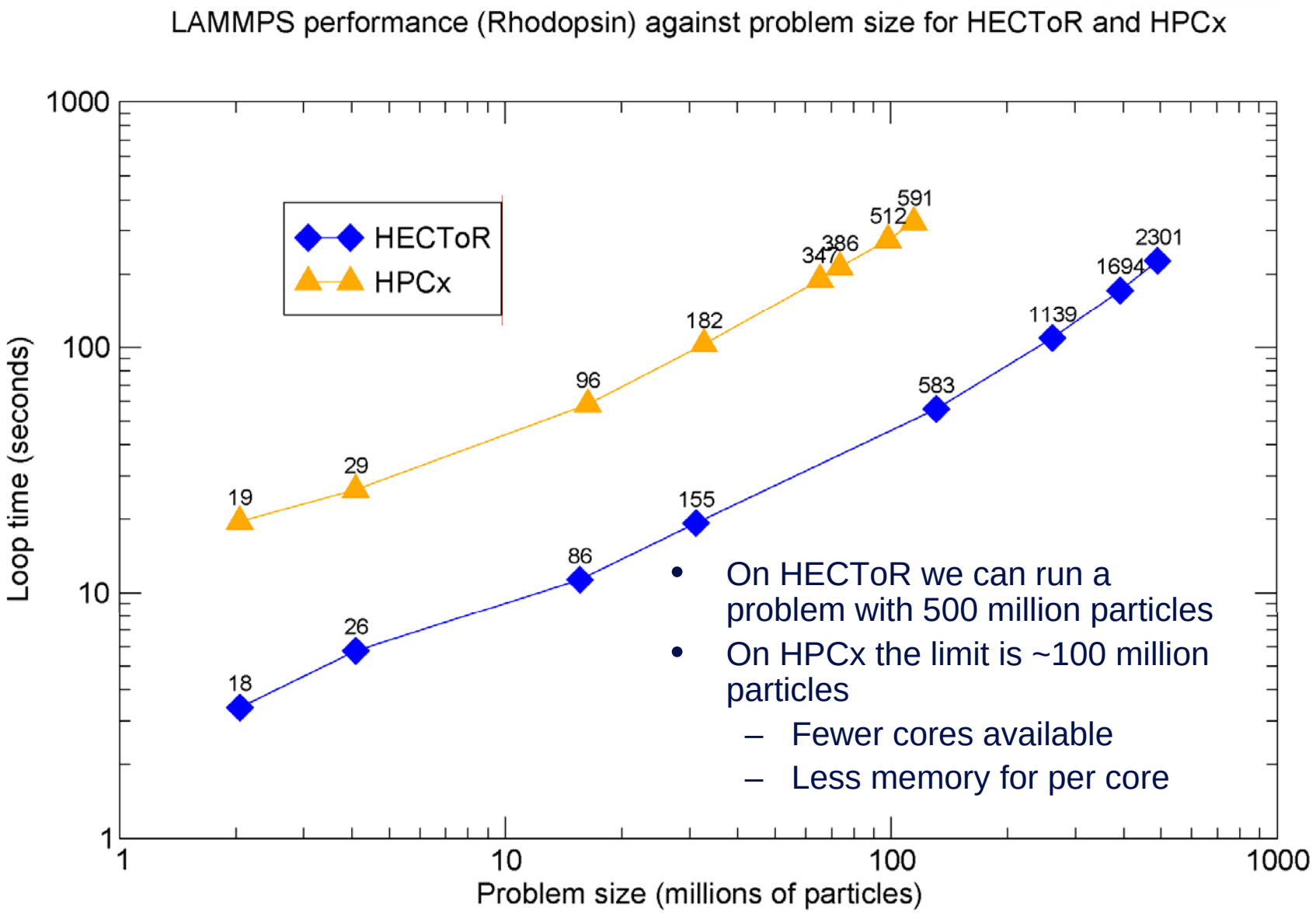
LAMMPS


LAMMPS Rhodopsin benchmark, 4096000 atoms
System Comparison
Loop Time * Cores (s)
50000
45000
40000
35000
30000
25000
20000
15000
10000
10
100
1000
10000
Cores
HPCx Phase 3
HPCx Phase 3 SMT
HECToR

LAMMPS

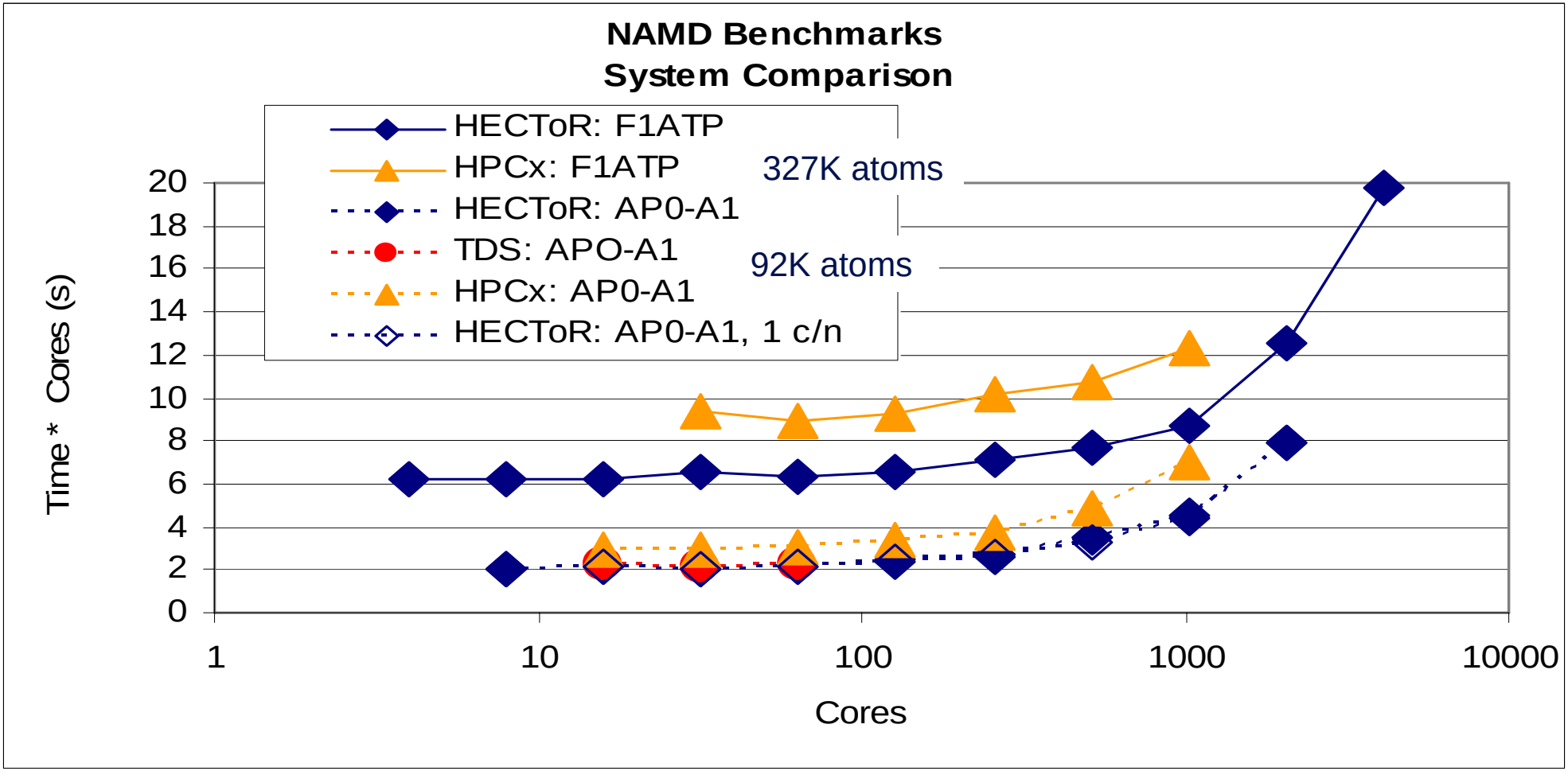

LAMMPS Rhodopsin benchmark, 4096000 atoms
HECToR
Loop Time * Cores (s)
30000
28000
26000
24000
22000
20000
18000
16000
14000
12000
10000
10
100
1000
10000
Cores
HECToR
HECToR 1 core per node
HECToR TDS
HECToR TDS 1 core per node

LAMMPS



- On HECToR we can run a problem with 500 million particles
- On HPCx the limit is ~100 million particles
  - Fewer cores available
  - Less memory for per core

NAMD


NAMD Benchmarks
System Comparison
HECToR: F1ATP
HPCx: F1ATP
HECToR: AP0-A1
TDS: APO-A1
HPCx: AP0-A1
HECToR: AP0-A1, 1 c/n
327K atoms
92K atoms
Time * Cores (s)
0
2
4
6
8
10
12
14
16
18
20
1
10
100
1000
10000
Cores

Conclusions

- On a core by core basis, not much difference between HECToR and HPCx in terms of application performance
  - But HECToR has many more cores and a more scalable interconnect
- Scaling better at high core counts on HECToR
  - HECToR can also run much bigger problems, e.g. LAMMPS
- Memory bandwidth cannot sustain fully populated nodes for both systems
  - general problem for HPC systems these days
  - This is seen in performance of memory bandwidth sensitive applications
  - Problem is worse for HECToR, especially with current non-symmetric memory setup.