CP2K PERFORMANCE ON ARCHER

Fiona Reid

epcc | THE UNIVERSITY OF EDINBURGH

Overview

- ARCHER / HECToR architecture comparison
- Benchmarks and results
- Performance hints and tips
- Conclusions

Architecture comparison

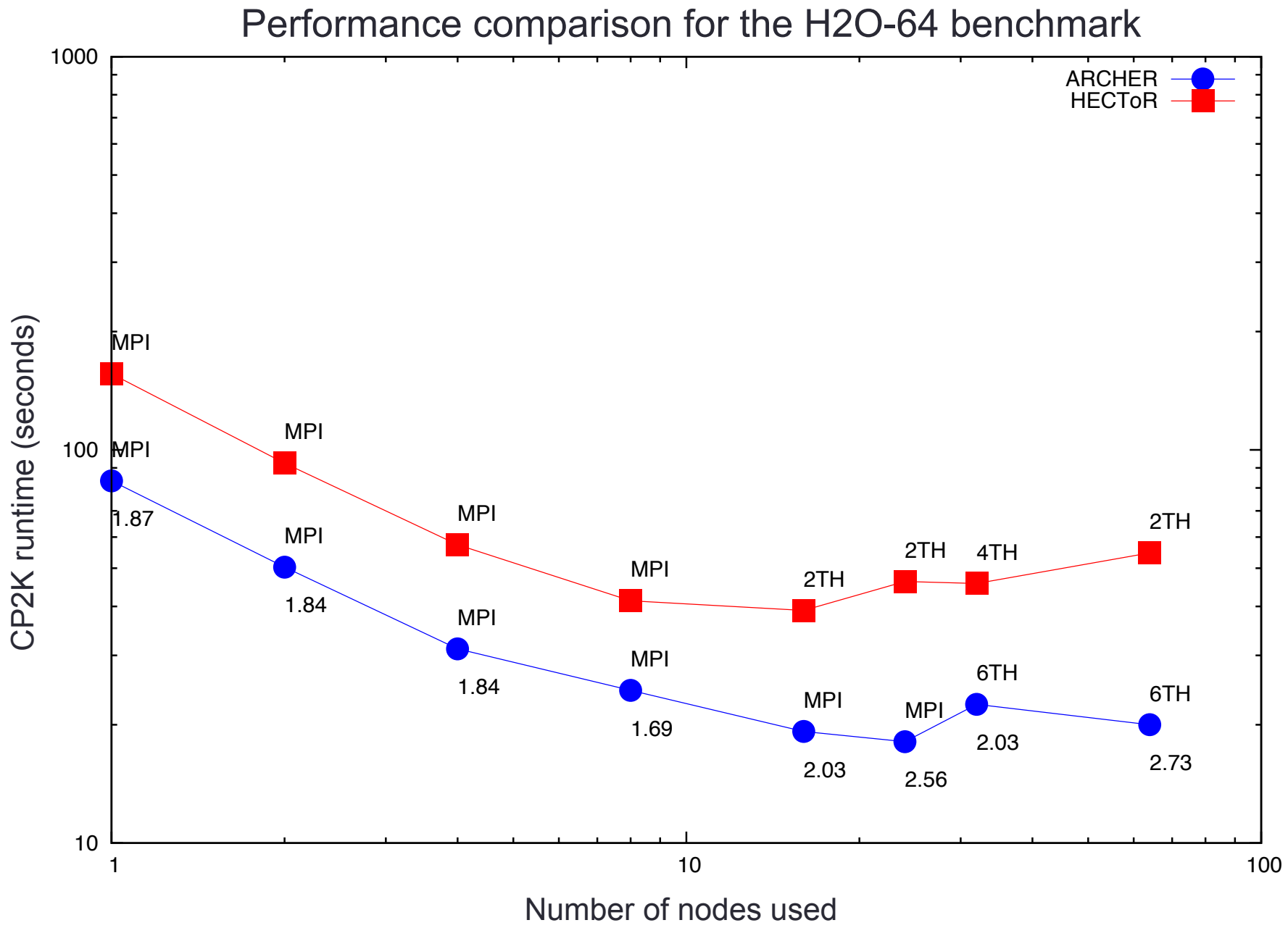
Feature	HECToR	ARCHER
Processors	AMD Interlagos 2.3GHz	Intel Ivy Bridge 2.7GHz
Cores per node	32 (4× 8-core NUMA)	24 (2× 12-core NUMA)
Memory per node	32 GB (1 GB/core)	64GB (2.66 GB/core) 128GB (5.33 GB/core)
Nodes	2816 (90,112 cores)	3008 (72,192 cores)
Interconnect	Cray Gemini	Cray Aries
Topology	3D Torus	Dragonfly
Post-processing Nodes	(None)	2 Nodes: 48 core SandyBridge 1TB Memory

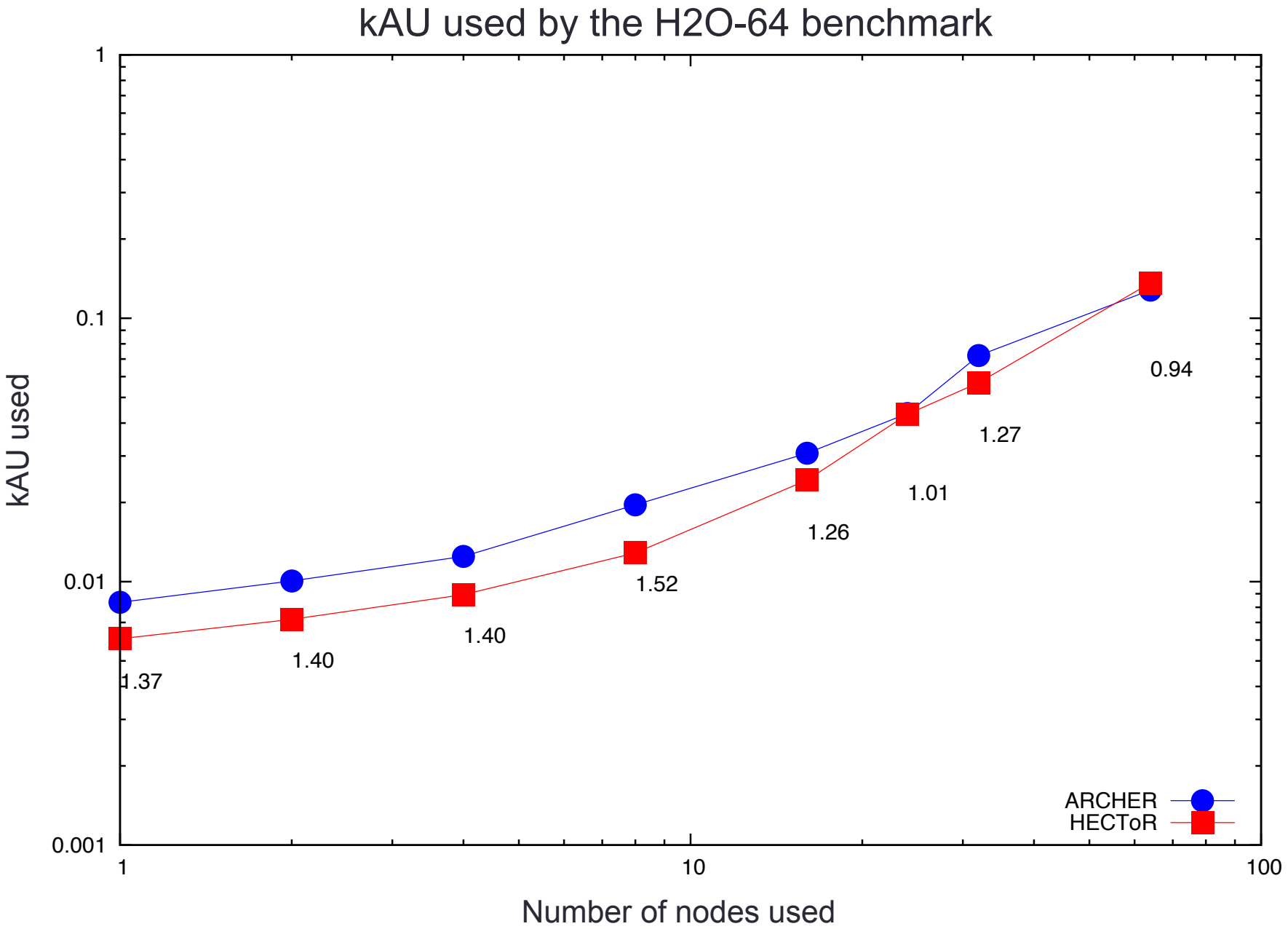
Benchmarks

- The performance of the code has been evaluated using five benchmarks which are indicative of the different types of calculations that can be run with CP2K.
- As both HECToR and ARCHER charge per node, our tests utilise full nodes.
- The PSMP (i.e. MPI/OpenMP) version of CP2K has been used with appropriate combinations of MPI processes and threads tested.

H2O-64

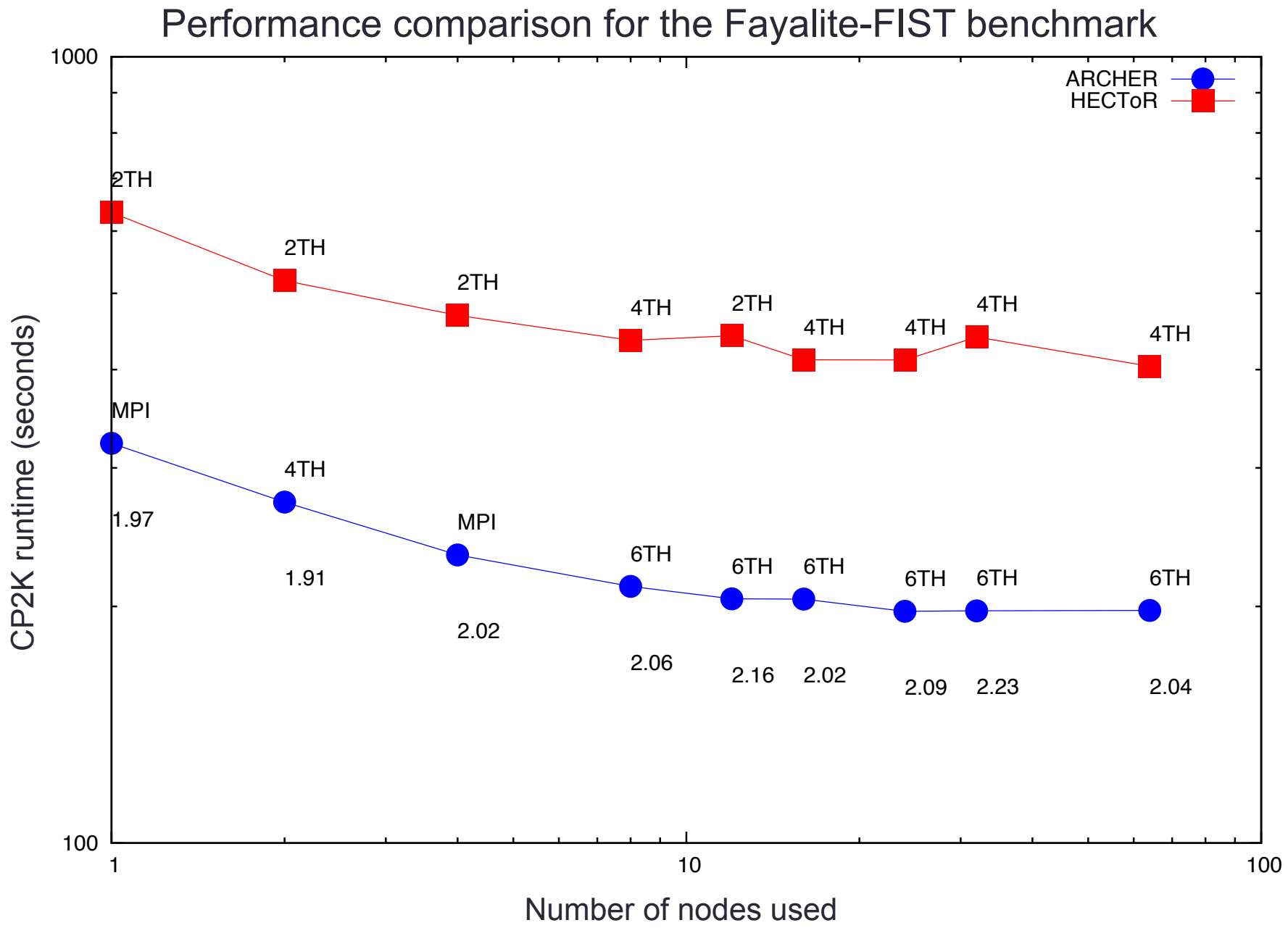
- Short molecular dynamics simulation (10 steps) in NVE ensemble at 300K
- 64 water molecules (192 atoms, 512 electrons) in 12.4 $Å^3$ cell
- Quickstep DFT, LDA functional, TZV2P basis set and 280 Ry cut-off

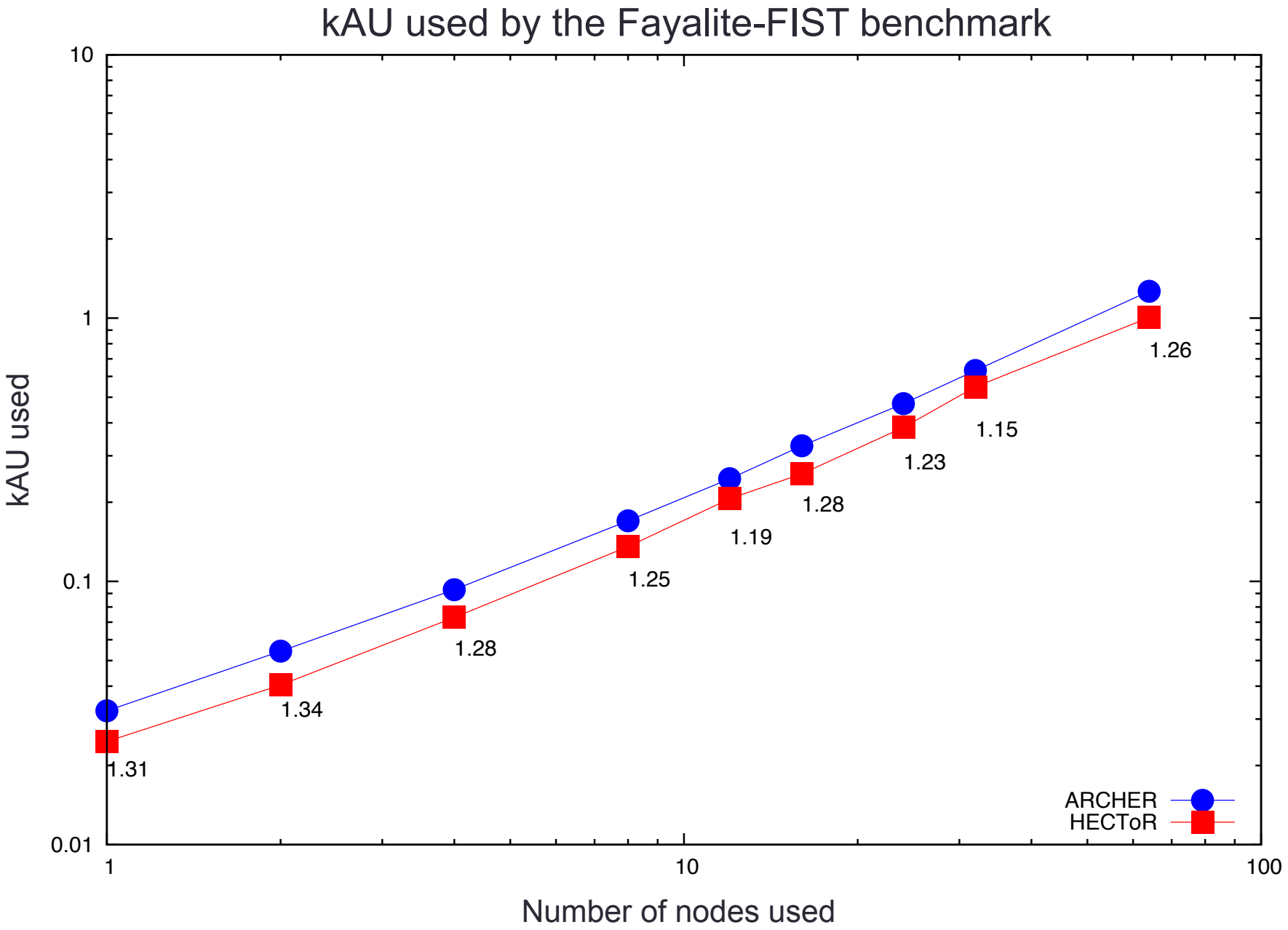

Performance comparison for the H2O-64 benchmark
ARCHER
HECToR
CP2K runtime (seconds)
Number of nodes used
1000
100
10
1
10
100
MPI
MPI
MPI
MPI
2TH
2TH
4TH
2TH
MPI
1.87
MPI
1.84
MPI
1.84
MPI
1.69
MPI
2.03
MPI
2.56
6TH
2.03
6TH
2.73


kAU used by the H2O-64 benchmark
kAU used
Number of nodes used
1
0.1
0.01
0.001
1
10
100
1.37
1.40
1.40
1.52
1.26
1.01
1.27
0.94
ARCHER
HECToR

Fayalite-FIST

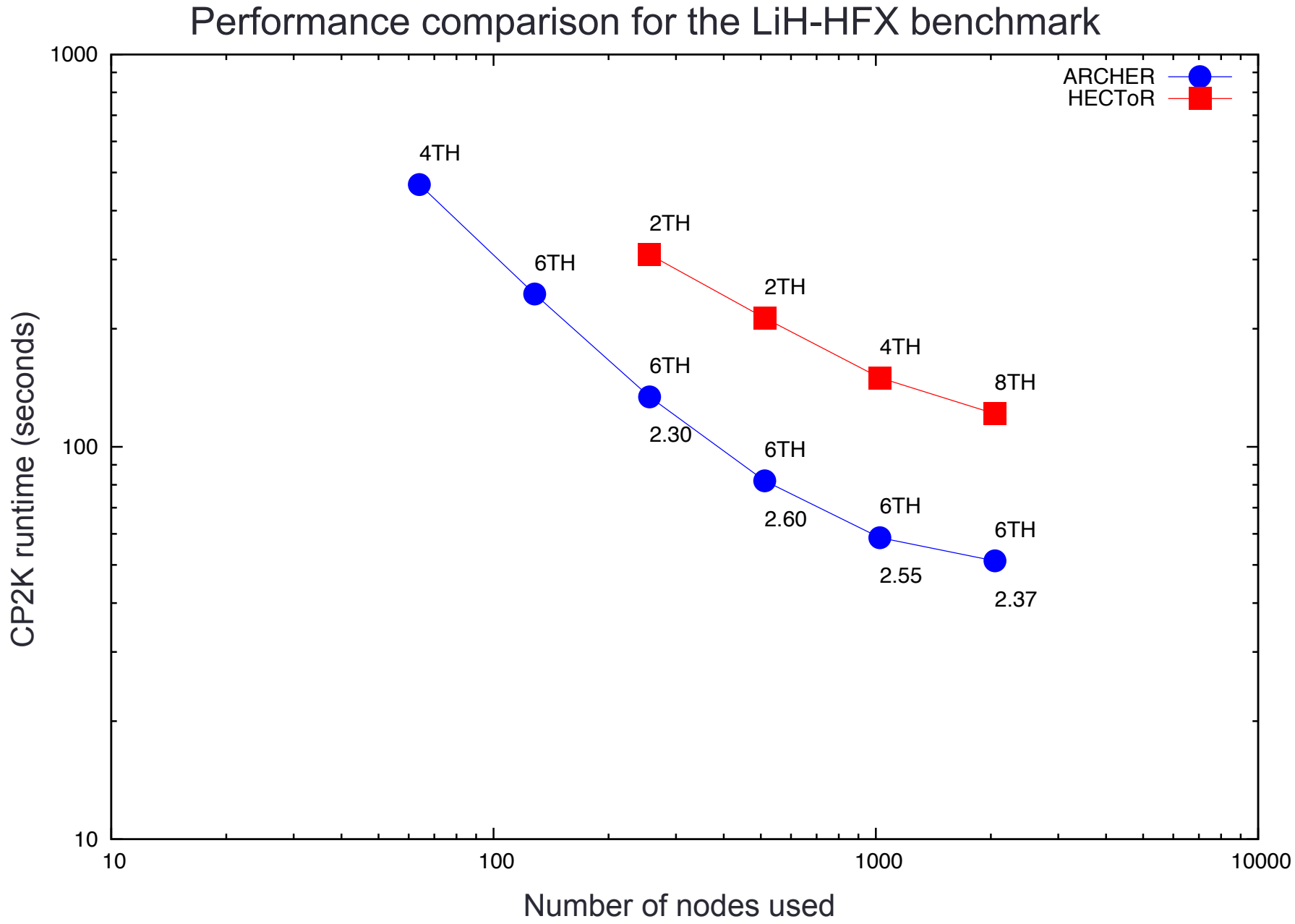
- Short molecular dynamics simulation (1000 steps) in NPT ensemble at 300K
- 28000 atoms (10^3 supercell, 28 atoms per unit cell)
- Fe_2SiO_4 (Iron silicate a.k.a. fayalite)
- Classical potential (Morse with hard-core repulsive term, cutoff 5.5 Å), plus long-range electrostatics with SPME summation (Smoothed Particle Mesh Ewald)

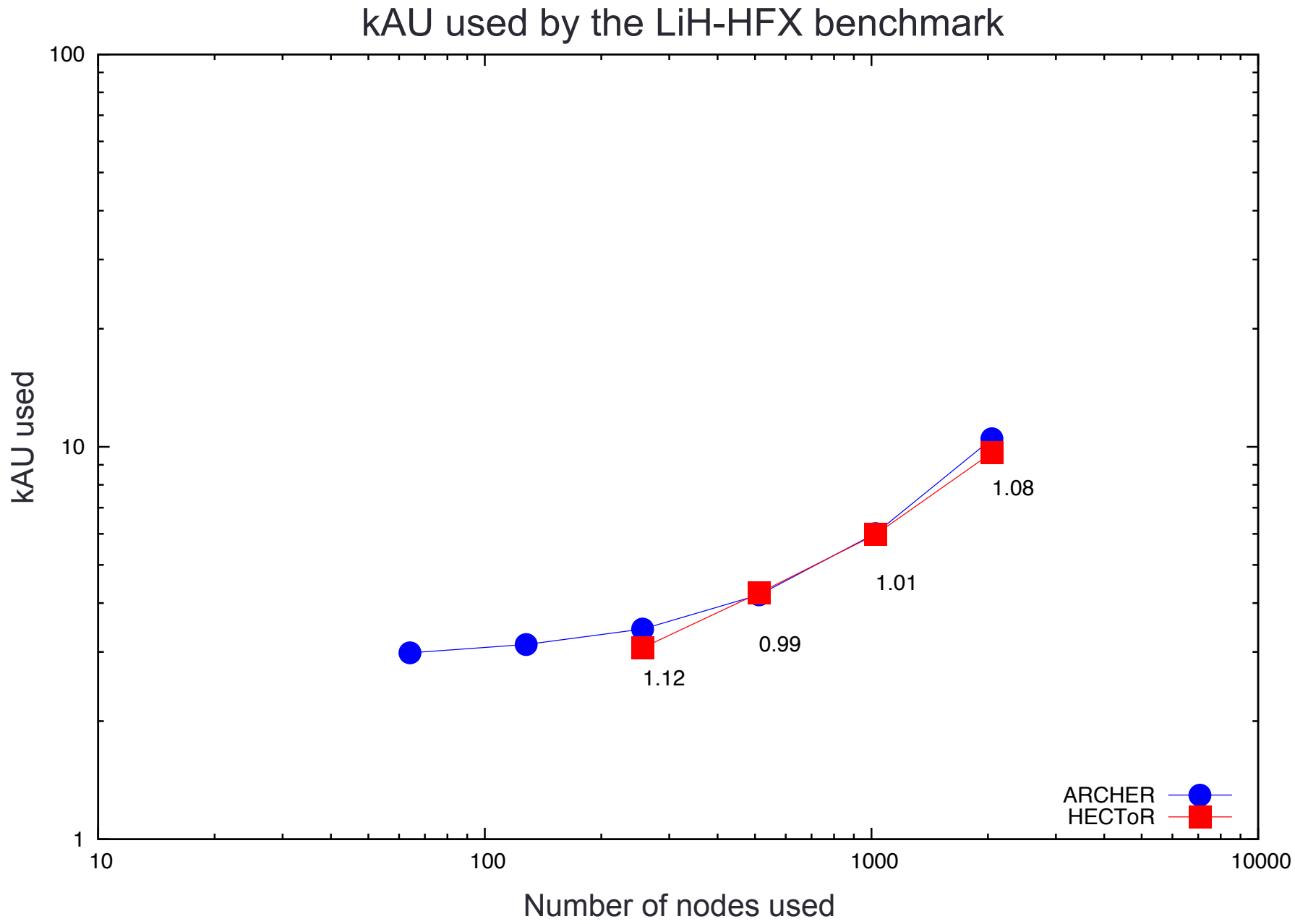

Performance comparison for the Fayalite-FIST benchmark
ARCHER
HECToR
CP2K runtime (seconds)
Number of nodes used
1000
100
1
10
100
2TH
2TH
2TH
4TH
2TH
4TH
4TH
4TH
4TH
MPI
4TH
MPI
6TH
6TH
6TH
6TH
6TH
6TH
1.97
1.91
2.02
2.06
2.16
2.02
2.09
2.23
2.04


kAU used by the Fayalite-FIST benchmark
kAU used
Number of nodes used
10
1
0.1
0.01
1
10
100
1.31
1.34
1.28
1.25
1.19
1.28
1.23
1.15
1.26
ARCHER
HECToR

LiH-HFX

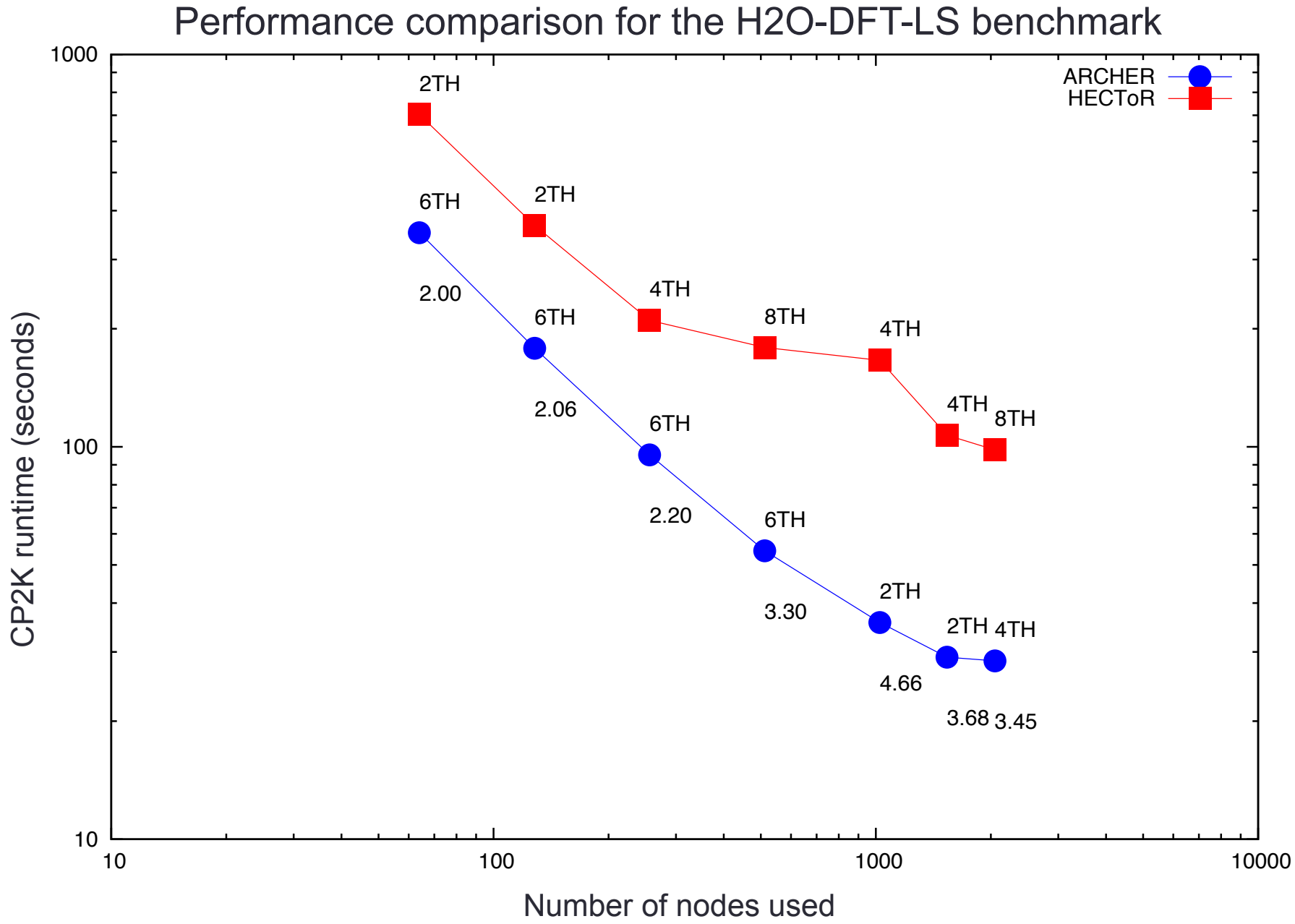
- Single-point energy calculation using Quickstep GAPW (Gaussian and augmented plane-waves) with hybrid Hartree-Fock Exchange
- 216-atom lithium Hydride crystal (432 electrons) in 12.3 $Å^3$ cell
- ~10x computational cost of standard DFT
- This benchmark was used as part of the ARCHER procurement process

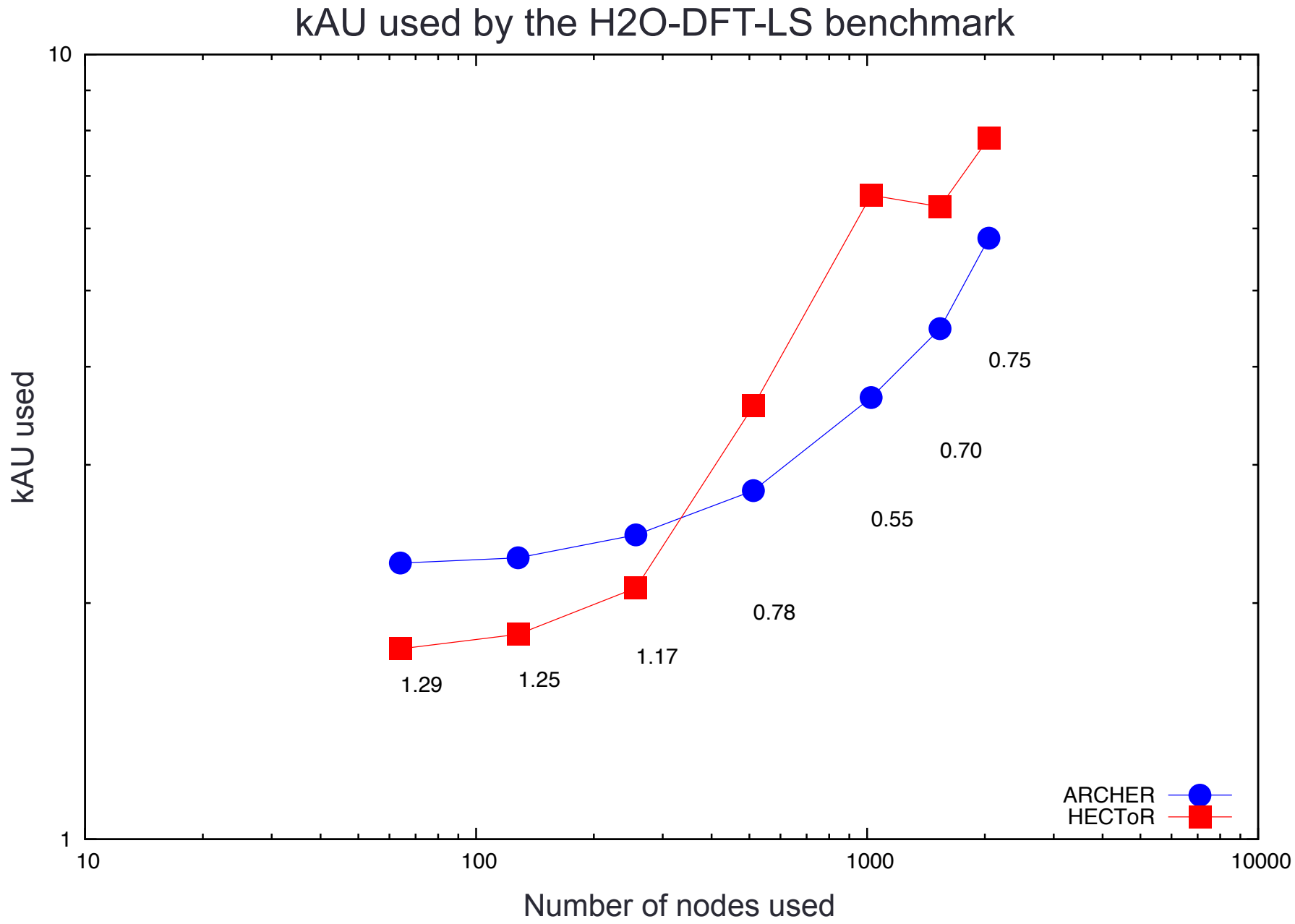

Performance comparison for the LiH-HFX benchmark
ARCHER
HECToR
4TH
6TH
6TH
2.30
6TH
2.60
6TH
2.55
6TH
2.37
2TH
2TH
4TH
8TH
1000
100
10
10
100
1000
10000
CP2K runtime (seconds)
Number of nodes used


kAU used by the LiH-HFX benchmark
kAU used
Number of nodes used
100
10
1
10
100
1000
10000
1.08
1.01
0.99
1.12
ARCHER
HECToR

H2O-DFT-LS

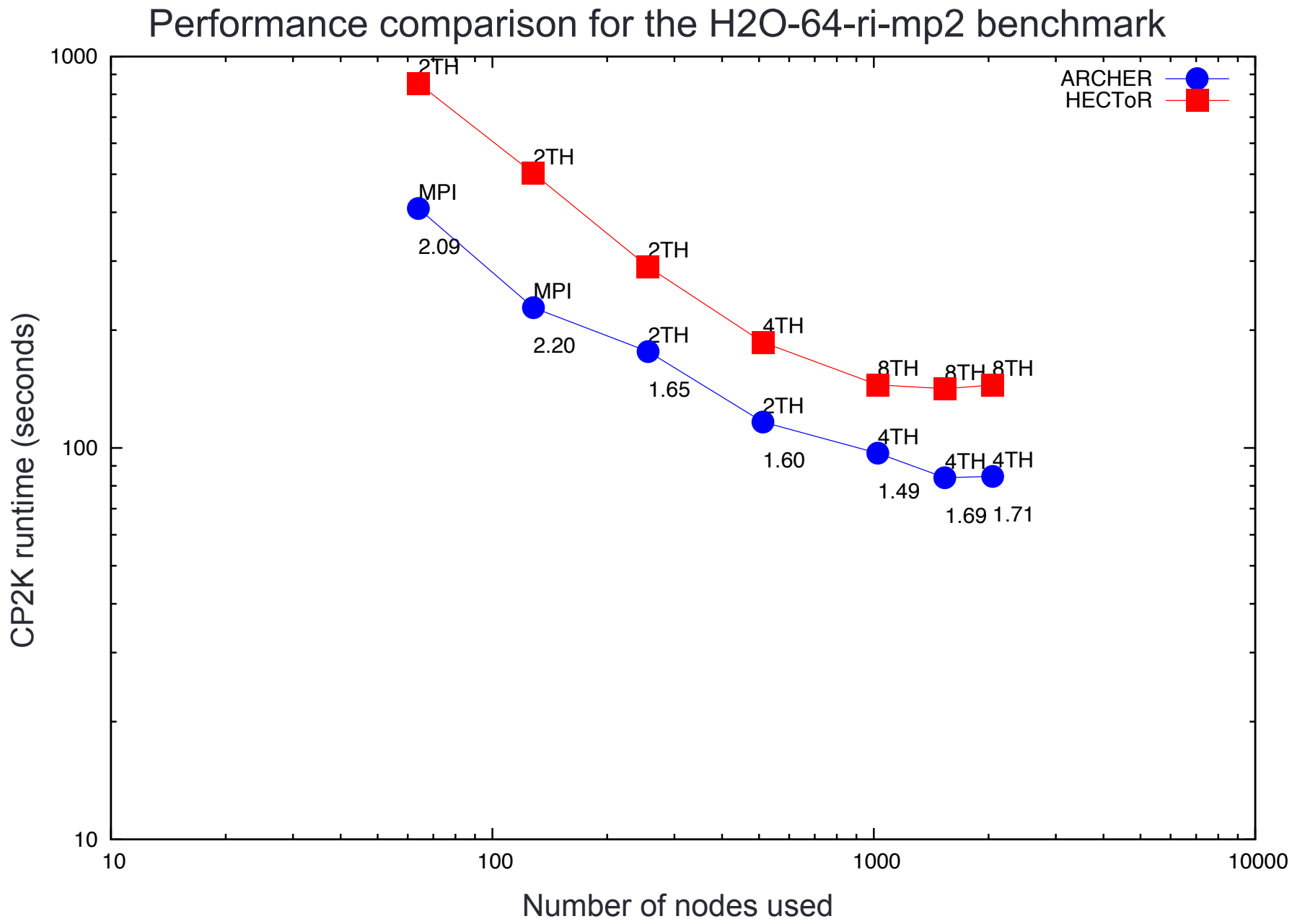
- Single-point energy calculation using linear-scaling DFT
- 6144 atoms in 39 $Å^3$ cell (32 water molecules in 4x4x4 supercell)
- LDA functional, DZVP MOLOPT (molecular optimised) basis set, 300 Ry cutoff
- Uses 5-level multigrid (finest grid is 420^3)
- Similar computational cost per atom to standard DFT (and scales to 100,000s+)

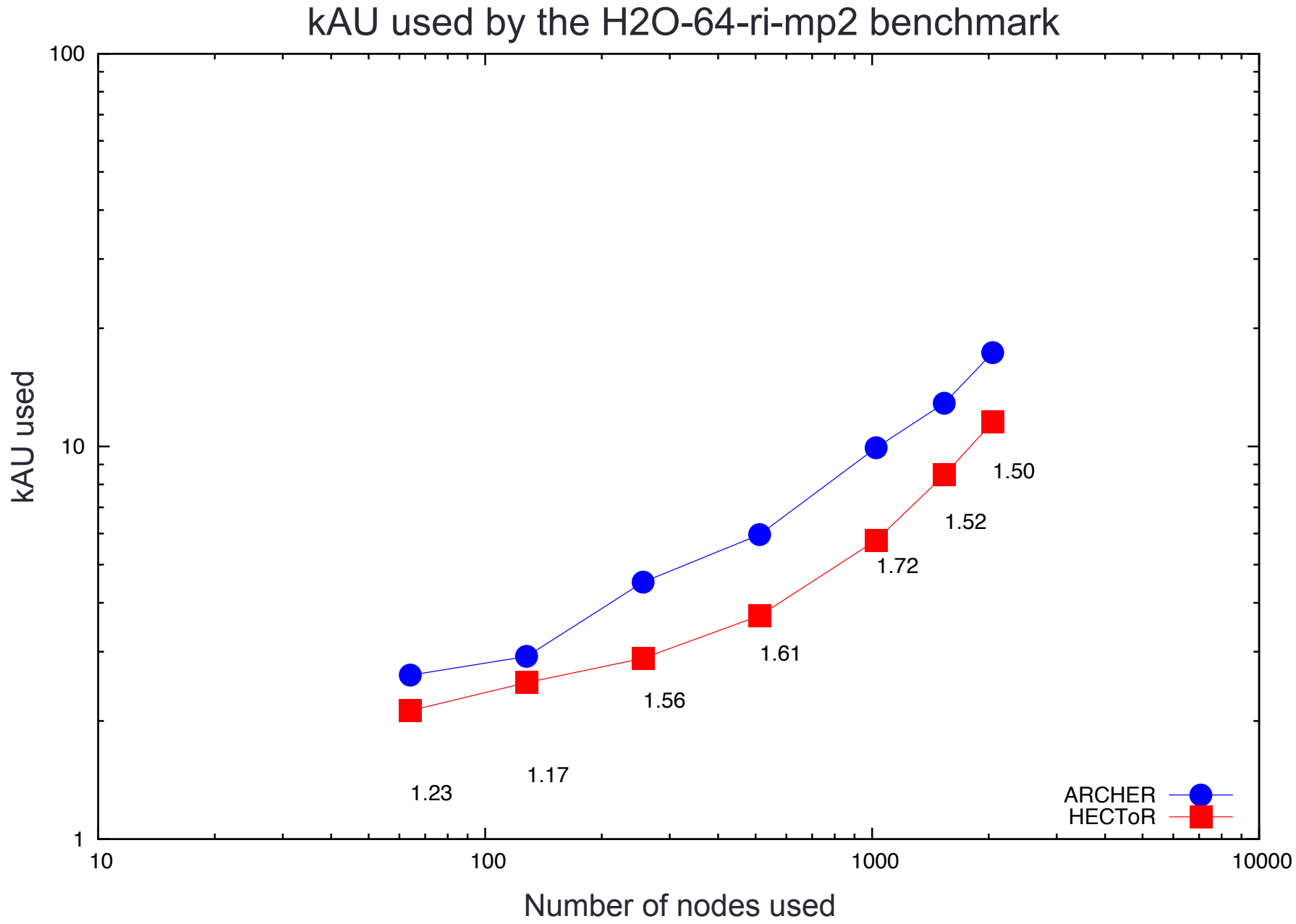

Performance comparison for the H2O-DFT-LS benchmark
CP2K runtime (seconds)
Number of nodes used
1000
100
10
10
100
1000
10000
ARCHER
HECToR
2TH
6TH
2TH
2.00
4TH
6TH
8TH
4TH
2.06
4TH
8TH
6TH
2.20
6TH
2TH
3.30
2TH
4TH
4.66
3.68
3.45


kAU used by the H2O-DFT-LS benchmark
kAU used
Number of nodes used
10
1
10
100
1000
10000
0.75
0.70
0.55
0.78
1.17
1.25
1.29
ARCHER
HECToR

H2O-64-ri-mp2

- H2O-64-ri-mp2 = resolution-of-the-identity MP2
- Single-point energy calculation using 2nd order Moller-Plesset Perturbation Theory (MP2) to calculate the exchange-correlation energy
- 64 water molecules in 12.4 $Å^3$ cell (same as H2O-64)
- ~100x computational cost to standard DFT


Performance comparison for the H2O-64-ri-mp2 benchmark
CP2K runtime (seconds)
Number of nodes used
1000
100
10
10
100
1000
10000
ARCHER
HECToR
2TH
2TH
2TH
4TH
8TH
8TH
8TH
MPI
2.09
MPI
2.20
2TH
1.65
2TH
1.60
4TH
1.49
4TH
1.69
4TH
1.71


kAU used by the H2O-64-ri-mp2 benchmark
kAU used
Number of nodes used
100
10
1
10
100
1000
10000
1.23
1.17
1.56
1.61
1.72
1.52
1.50
ARCHER
HECToR

Performance hints and tips

- Several additional tweaks to improve performance – suggested by Cray

- MPICH variables – may improve the MPI performance

```
export MPICH_GNI_MAX_EAGER_MSG_SIZE=131072
export MPICH_GNI_NDREG_MAXSIZE=16777216
export MPICH_GNI_RDMA_THRESHOLD=128
```

- Adding `-r 1` to `aprun` options, e.g.

```
aprun –r 1 –n 128 –N 4 –d 6 –S 2 cp2k.psmp
```

Performance hints and tips

- May get some benefit by using a specific rank reordering
- e.g. if you want to use 169 nodes with 4 MPI processes per node, you can run:

```
module load perftools
grid_order -R -H -m 676 -n 4 -g 26x26 -c 1x1 > MPICH_RANK_ORDER
```

- In your batch script add:

```
export MPICH_RANK_REORDER_METHOD=3
```

- Square grids are likely to give the best performance

- None of these gave any notable improvement for H2O-DFT-LS on ARCHER
  - But benchmark for your specific system!

Conclusion

- ARCHER is ~2 – 3 times faster (runtime) than HECToR
- ARCHER costs (kAU) more than HECToR but the improvements in runtime you'll get more than offset this
- To obtain good performance you should:
  - Use the package account version of CP2K
  - Benchmark your problem before starting any large runs
    - When benchmarking reduce the problem to a small number of time steps and run over a range of node counts
    - If you use a large number of MPI processes (> 1000) then using threads may well help you