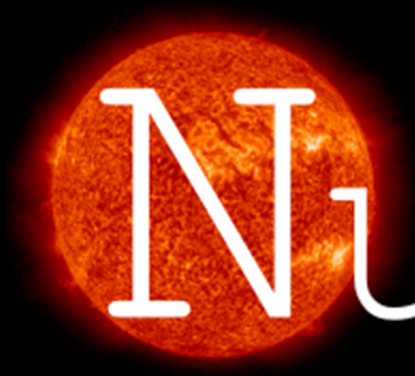
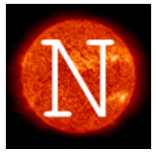


Nu-FuSE

HPC Issues for DFT Calculations

Adrian Jackson
EPCC

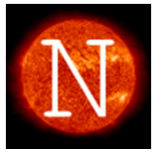
NSERC CRSNG ANR DFG РФФИ RESEARCH COUNCILS UK NSF



http://www.nu-fuse.com

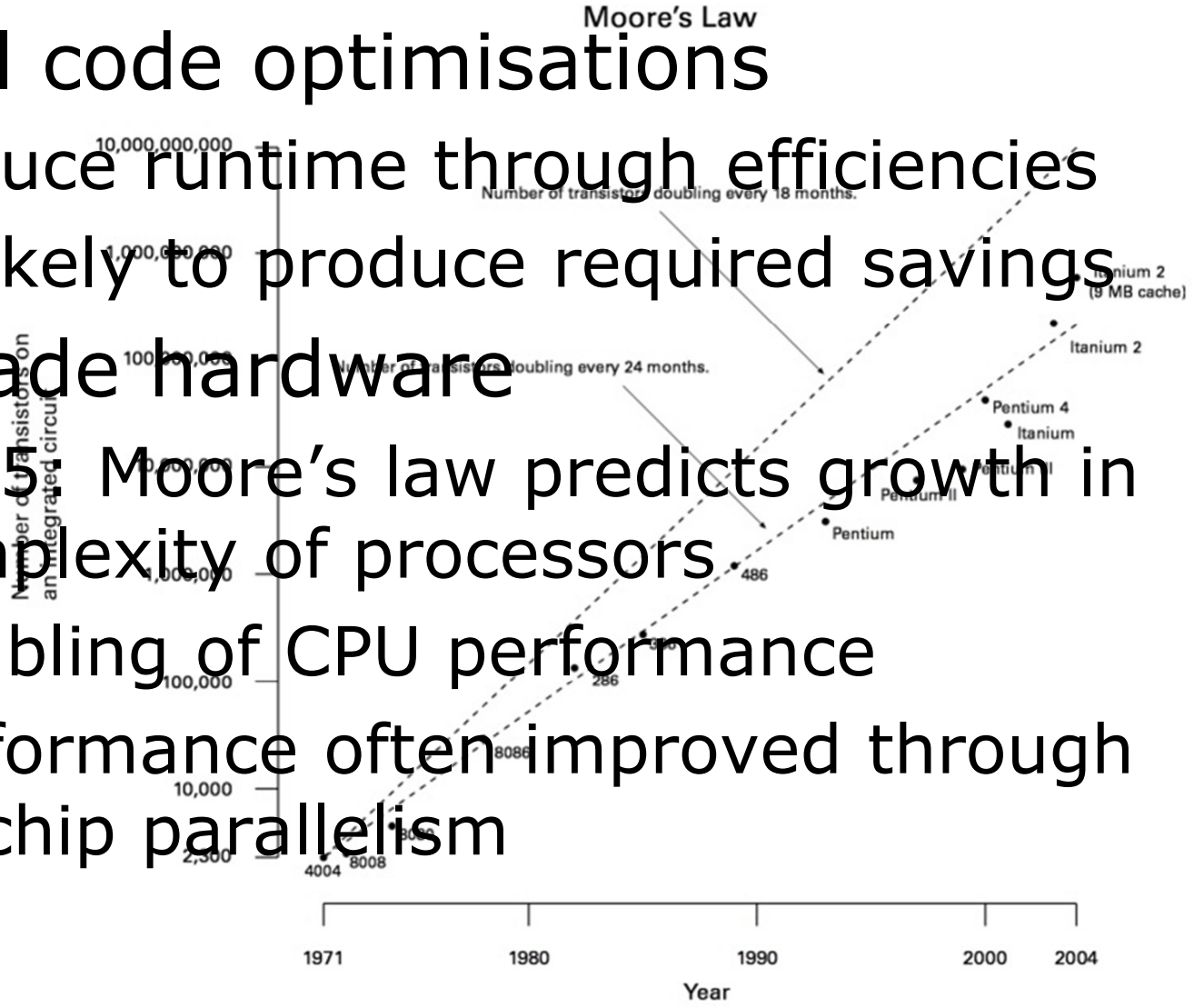
Scientific Simulation

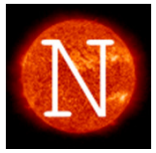
- Simulation fast becoming 4th pillar of science
  - Observation, Theory, Experimentation, Simulation
- Explore universe through simulation rather than experimentation
  - Test theories
  - Predict or validate experiments
  - Simulate "untestable" science
- Reproduce "real world" in computers
  - Generally simplified
  - Dimensions and timescales restricted
  - Simulation of scientific problem or environment
  - Input of real data
  - Output of simulated data
  - Parameter space studies
  - Wide range of approaches



Reduce runtime

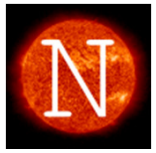
- Serial code optimisations
  - Reduce runtime through efficiencies
  - Unlikely to produce required savings
- Upgrade hardware
  - 1965: Moore's law predicts growth in complexity of processors
  - Doubling of CPU performance
  - Performance often improved through on chip parallelism





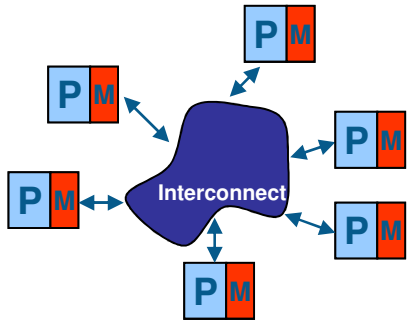
Parallel Background

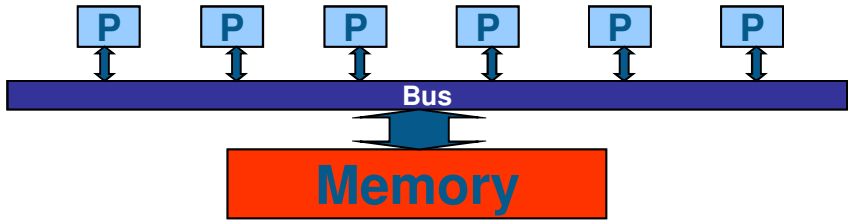
- Why not just make a faster chip?
  - Theoretical
    - Physical limitations to size and speed of a single chip
    - Capacitance increases with complexity
    - Speed of light, size of atoms, dissipation of heat
    - The power used by a CPU core is proportional to Clock Frequency x Voltage^2
    - Voltage reduction vs Clock speed for power requirements
      - Voltages become too small for “digital” differences
  - Practical
    - Developing new chips is incredibly expensive
    - Must make maximum use of existing technology

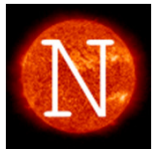


Parallel Systems

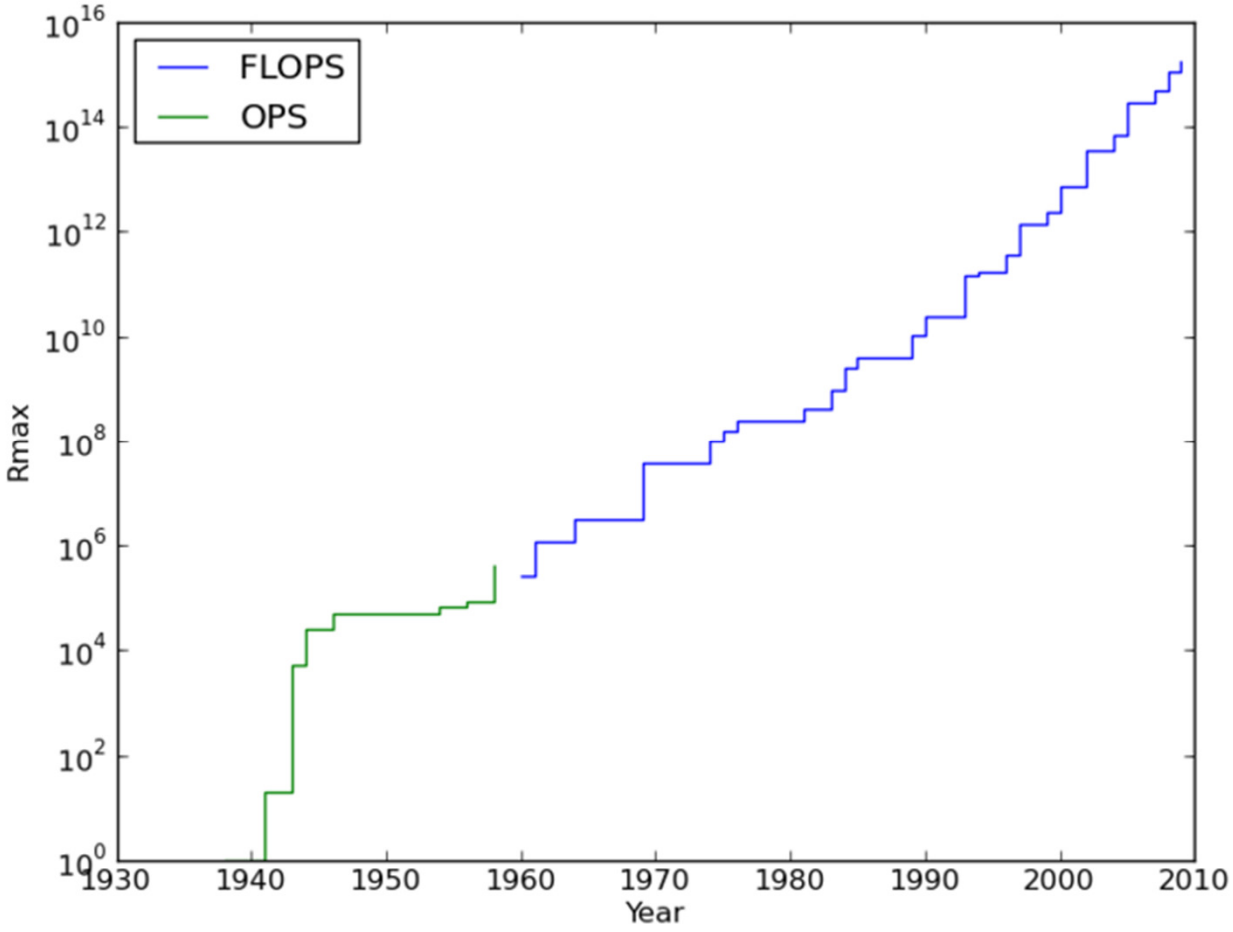
- Different types of parallel systems
  - Shared memory
  - Distributed memory
- Distributed memory: MPI
  - Each processor has its own local memory
  - Processors connected by some interconnect mechanism
  - Processors communicate via explicit message passing
  - Highly scalable architecture
- Shared memory: OpenMP
  - Each processor has access to a global memory
  - Communications via write/reads to memory
  - Caches are automatically kept up-to-date or *coherent*
  - Simple to program (no explicit communications)
  - Scaling is difficult because of memory access bottleneck
  - Usually modest numbers of processors





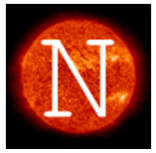


HPC Trends



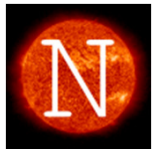
FLOPS

- **Yotta: 10^{24}**
- **Zetta: 10^{21}**
- **Exa: 10^{18}**
- **Peta: 10^{15}**
- **Tera: 10^{12}**
- **Giga: 10^{9}**
- **Mega: 10^{6}**
- **Kilo: 10^{3}**

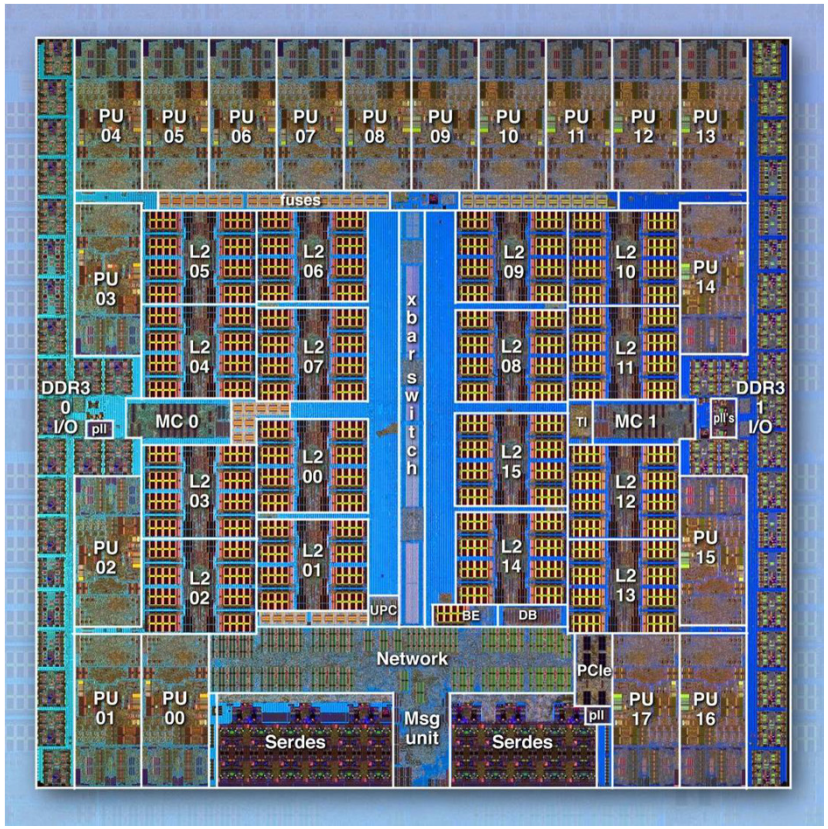


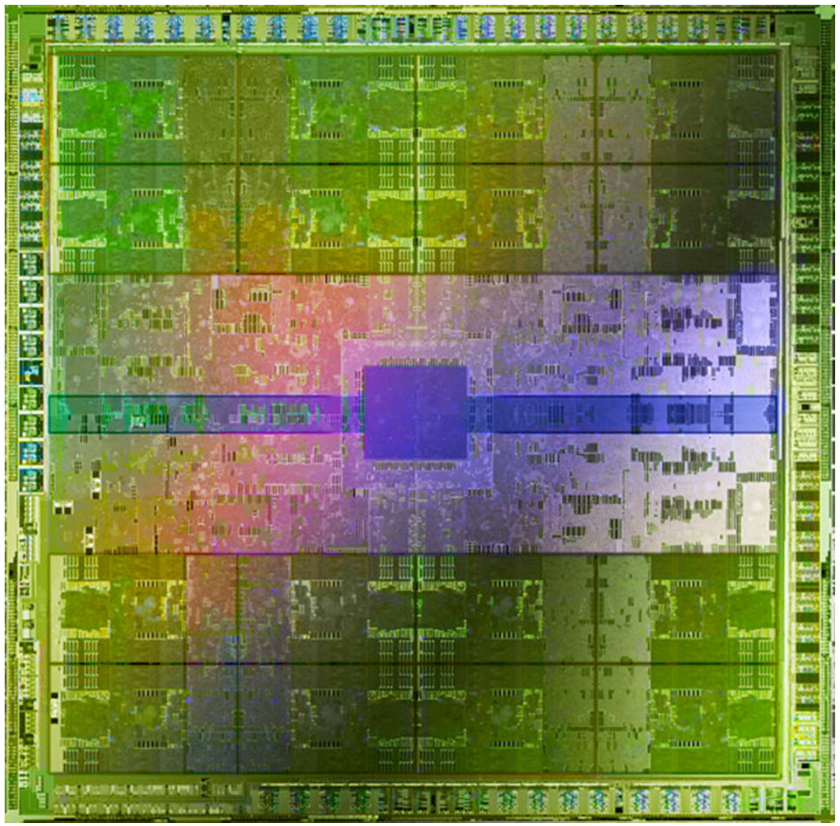
Parallel Background

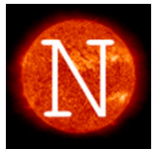
- Now parallelism explicit in chip design
  - Beyond implicit parallelism of pipelines, multi-issue and vector units
- Now possible (and economically desirable) to place multiple processors on a chip.
- From a programming perspective, this is largely irrelevant
  - simply a convenient way to build a small SMP
  - on-chip buses can have very high bandwidth
- Main difference is that processors may share caches
- Typically, each core has its own Level 1 and Level 2 caches, but the Level 3 cache is shared between cores
- May also share other functional units
  - i.e. FPU



Multi- and Many-core

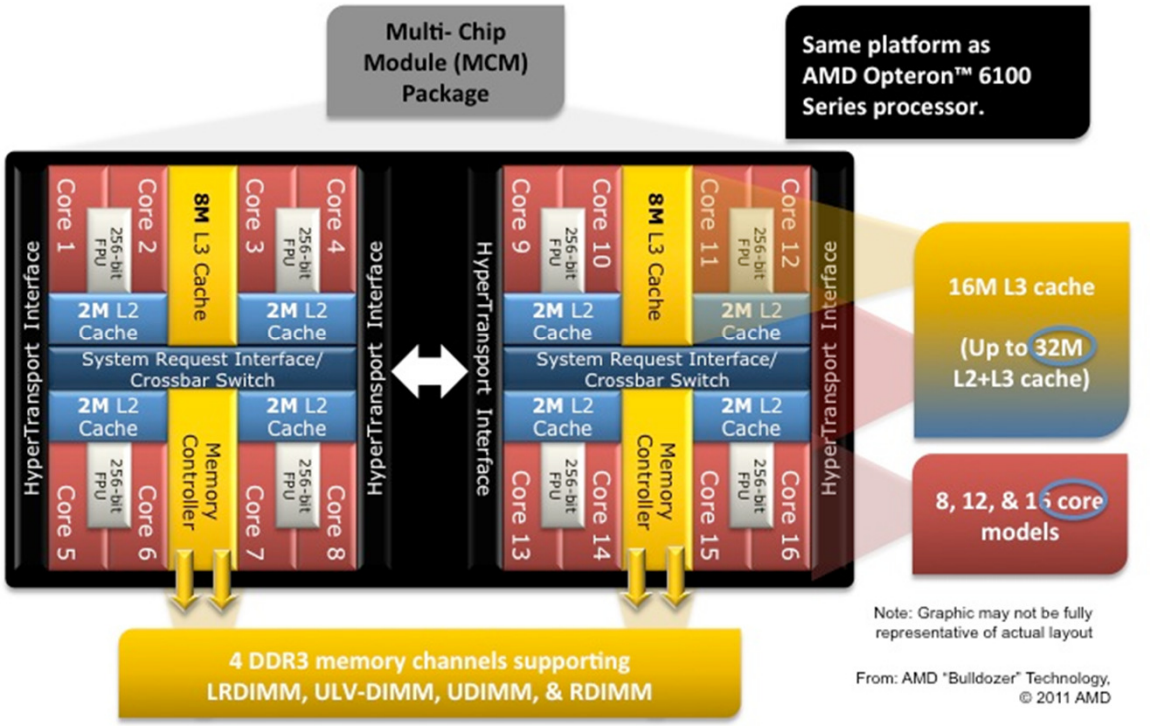






http://www.nu-fuse.com

HECToR XE6 Compute Node

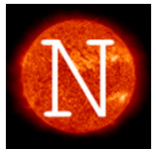


Node architecture:

- 2x 16-core 2.3GHz
  - Really 4x 8-core
- 32GB DDR3 RAM
- 85.3GB/s (3.6GB/s/core)

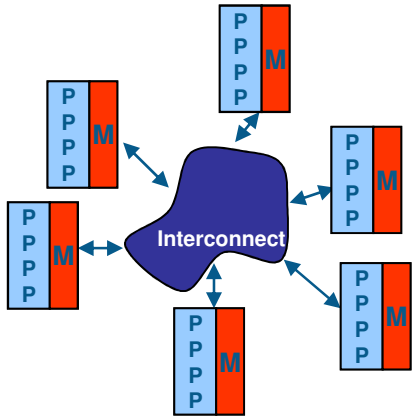
Gemini interconnect:

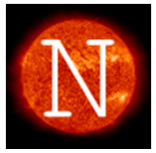
- 1 Gemini per 2 compute nodes
- Interconnect connected directly through HyperTransport
- Hardware support for:
  - MPI
  - Single-sided communications
  - Direct memory access
- Latency ~1-1.5µs;
- Bandwidth 5GB/s



Shared memory clusters

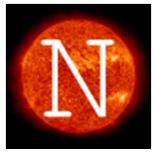
- Dominant HPC architecture
  - Shared memory clusters
  - Multi-core nodes
- Dominate processor architecture
  - Multi-core processors
  - Small shared memory systems on chip
  - 4-16 available per processor
  - Combine to give 48-64 cores per server
- Complex hierarchy of technologies
  - Efficient utilisation more challenging





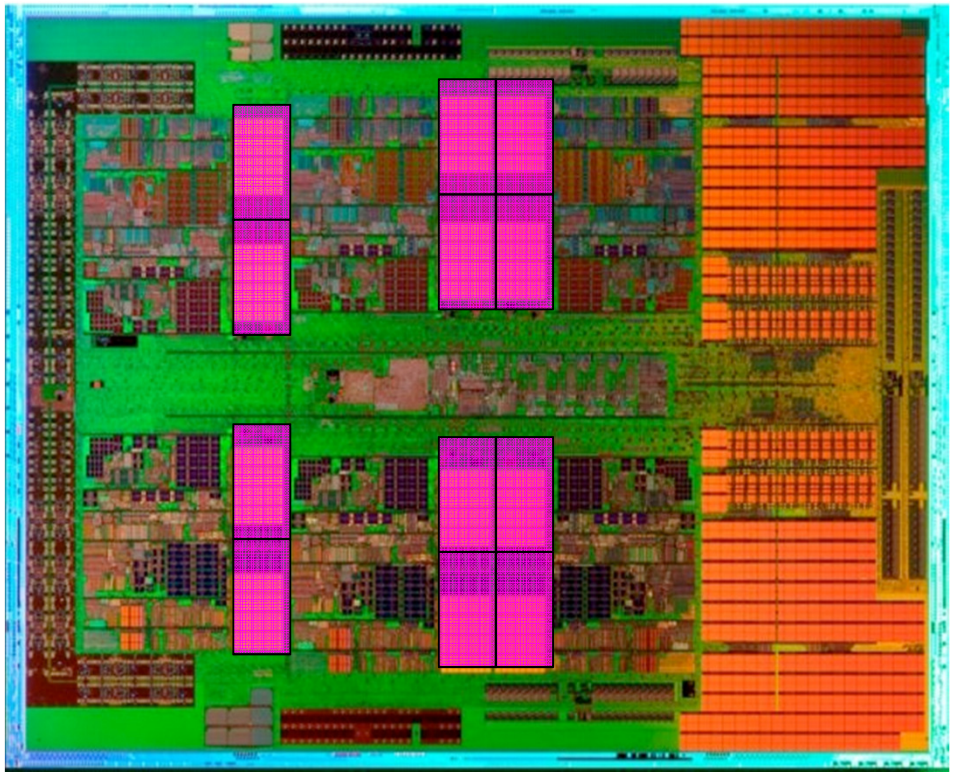
Accelerators

- Need a chip which can perform many parallel operations every clock cycle
  - Many cores and/or many operations per core
- Want to keep power/core as low as possible
- Much of the power expended by CPU cores is on functionality not generally that useful for HPC
  - Branch prediction, out-of-order execution etc
- However, still need to run complex programs (operating systems)
  - Solution is to add specialised processing elements to standard systems: Accelerators
  - GPGPU (Graphical Processing Units) most common

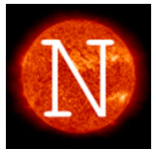


AMD 12-core CPU

- Not much space on CPU is dedicated to compute

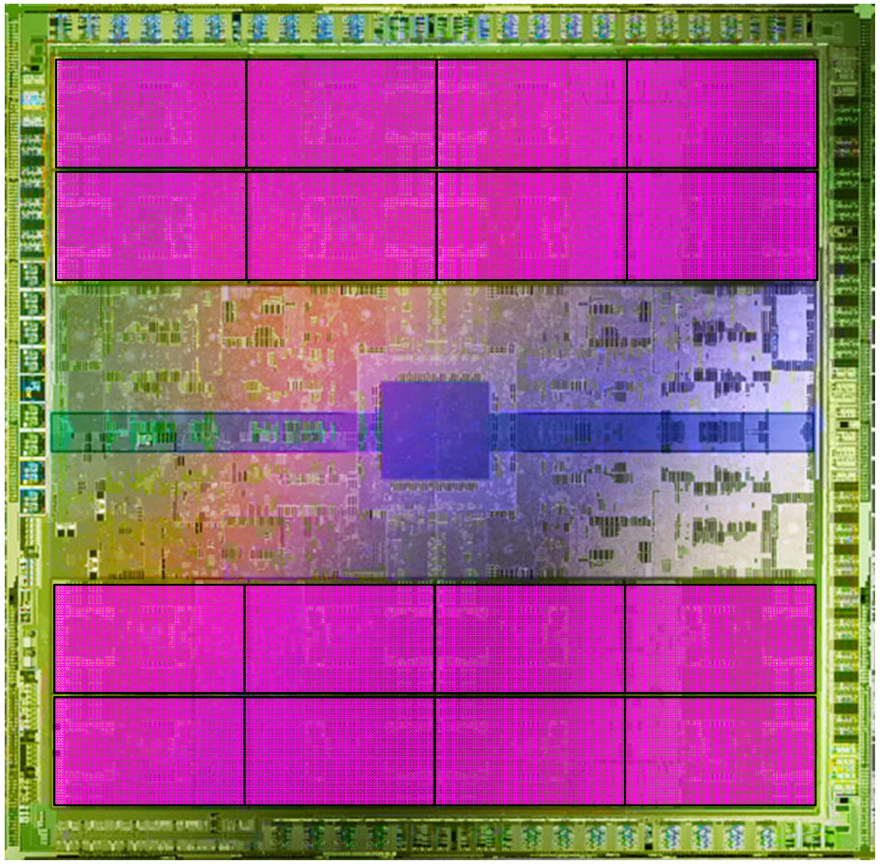


= compute unit
(= core)



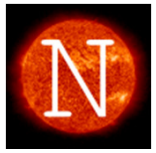
http://www.nu-fuse.com

NVIDIA Fermi GPU



= compute unit
(= SM
= 32 CUDA cores)

- GPU dedicates much more space to compute
  - At expense of caches, controllers, sophistication etc

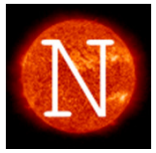


GPU Performance

- GPU vs CPU: Theoretical Peak capabilities

	NVIDIA Fermi	AMD Magny-Cours (6172)
Cores	448 (1.15GHz)	12 (2.1GHz)
Operations/cycle	1	4
DP Performance (peak)	515 GFlops	101 GFlops
Memory Bandwidth (peak)	144 GB/s	27.5 GB/s

- For these particular products, GPU theoretical advantage is ~5x for both compute and main memory bandwidth
- Application performance very much depends on application
  - Typically a small fraction of peak
  - Depends how well application is suited to/tuned for architecture



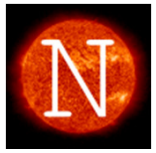
http://www.nu-fuse.com

Intel Xeon Phi

- 60 cores, 4 threads per code, 1.053 GHz
  - 1 Tflop/s DP performance

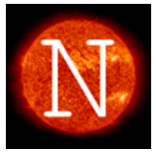
CPU-like
x86 cores: same instruction set as standard CPUs
Relatively small number cores/chip
Fully cache coherent
In principle could support an OS

GPU-like
Simple cores, lack sophistication e.g. no out-of-order execution
Each core contains 512-bit vector processing unit (16 SP or 8 DP numbers)
Supports multithreading
Not expected to run OS, but used as accelerator (at least initially)



Accelerators

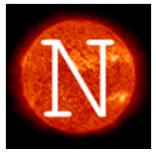
- Challenge to fully exploit both accelerator and processor
- Cost of data transfer
- Access to network
  - Currently through host
- Small amounts of memory per core



Exascale challenges

- Exascale challenge exacerbates issues
  - Very large shared memory nodes
  - Very expensive communications

	2013	2017	2022
System Perf.	20 PFlops	100-200 PFlops	1 EFlops
Memory	1 PB	5 PB	10 PB
Node Perf.	200 GFlops	400 GFlops	1-10 TFlops
Concurrency	32	O(100)	O(1000)
Interconnect BW	40 GB/s	100 GB/s	200-400 GB/s
Nodes	100,000	500,000	O(Million)
I/O	2 TB/s	10 TB/s	20 TB/s
MTTI	Days	Days	O(1 Day)
Power	10 MW	10 MW	20 MW



HECToR

http://www.nu-fuse.com

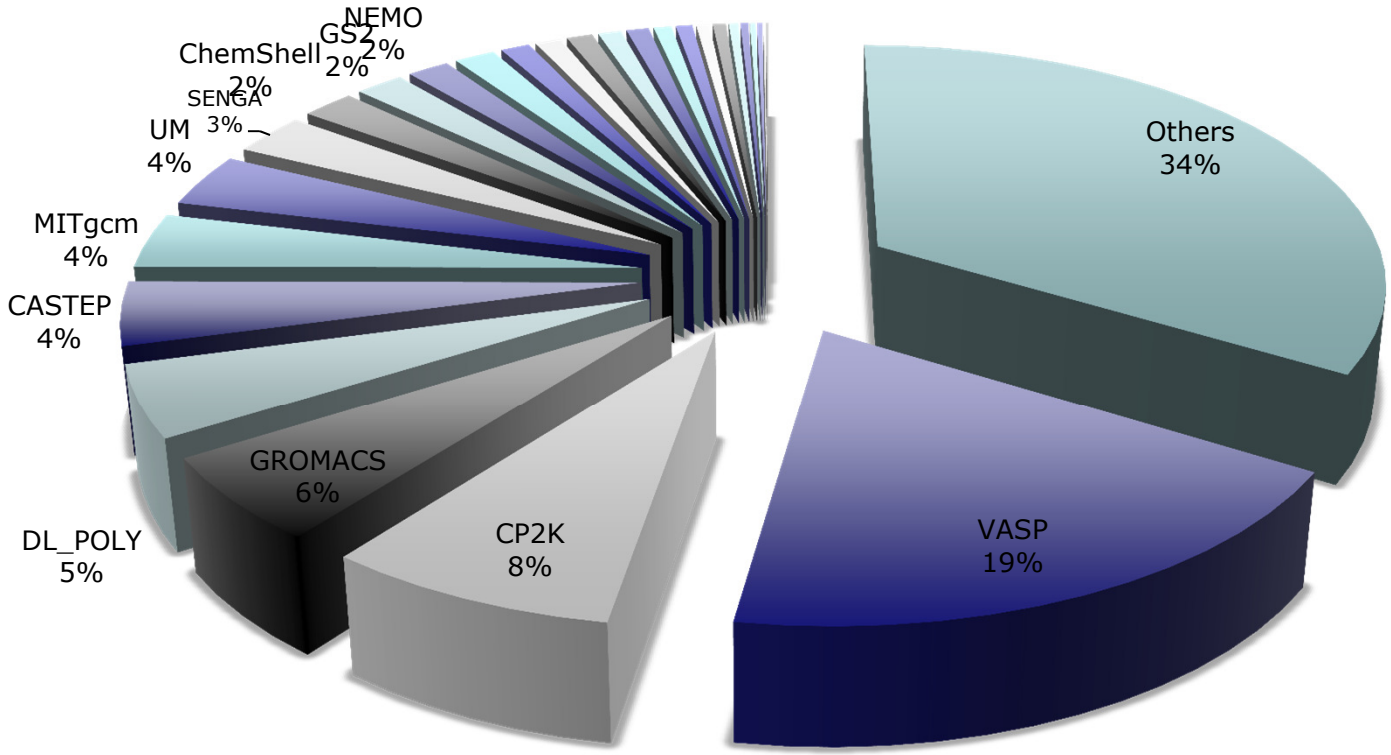
- UK National HPC Service
- Currently 30-cabinet Cray XE6 system
  - 90,112 cores
- Each node has
  - 2 × 16-core AMD Opterons (2.3GHz Interlagos)
  - 32 GB memory
- Peak of over 800 TF and 90 TB of memory

HECToR usage statistics

http://www.nu-fuse.com

Phase 3 statistics (Nov 2011 - Apr 2013)

Ab initio codes (VASP, CP2K, CASTEP, ONETEP, NWChem, Quantum Espresso, GAMESS-US, SIESTA, GAMESS-UK, MOLPRO)

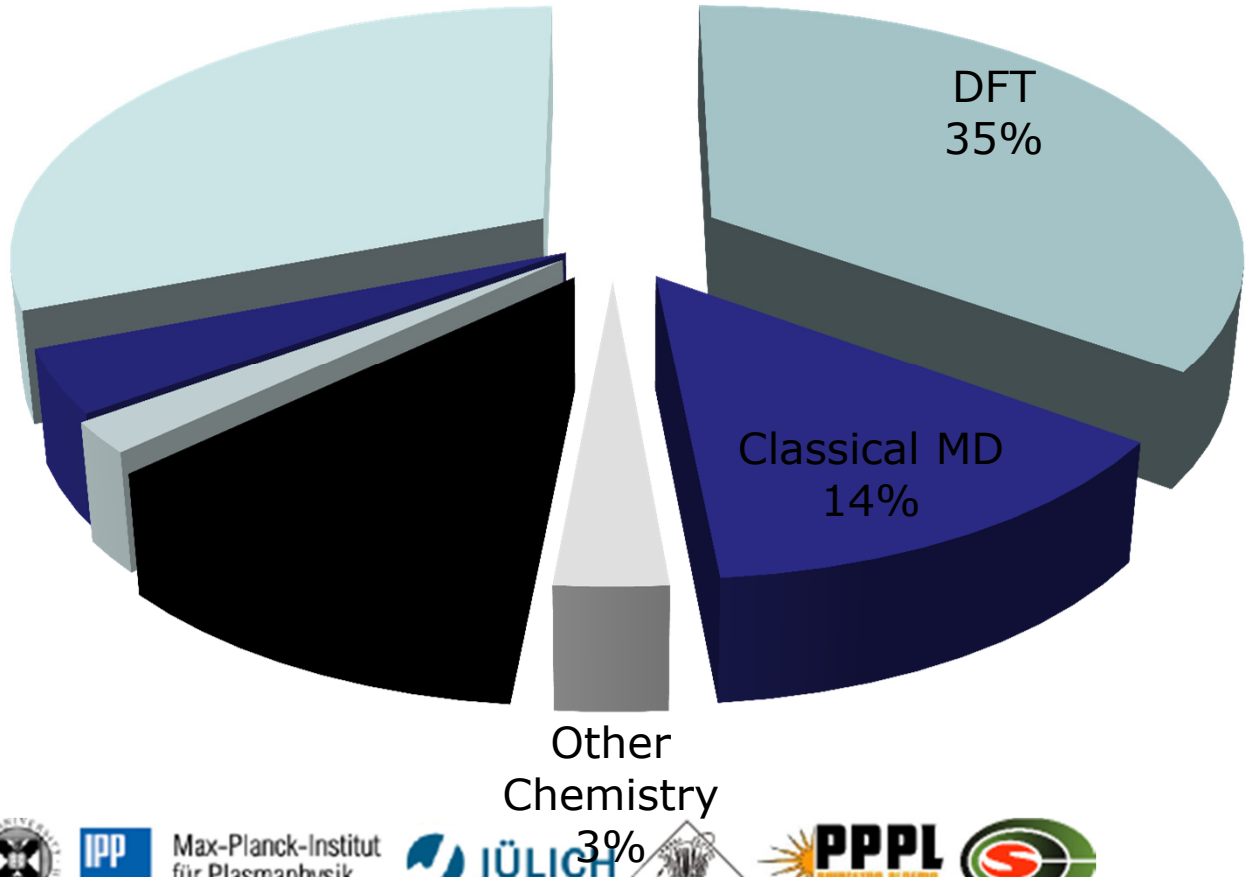


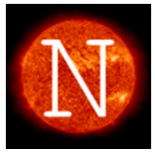
HECToR usage statistics

http://www.nu-fuse.com

Phase 3 statistics (Nov 2011 - Apr 2013)

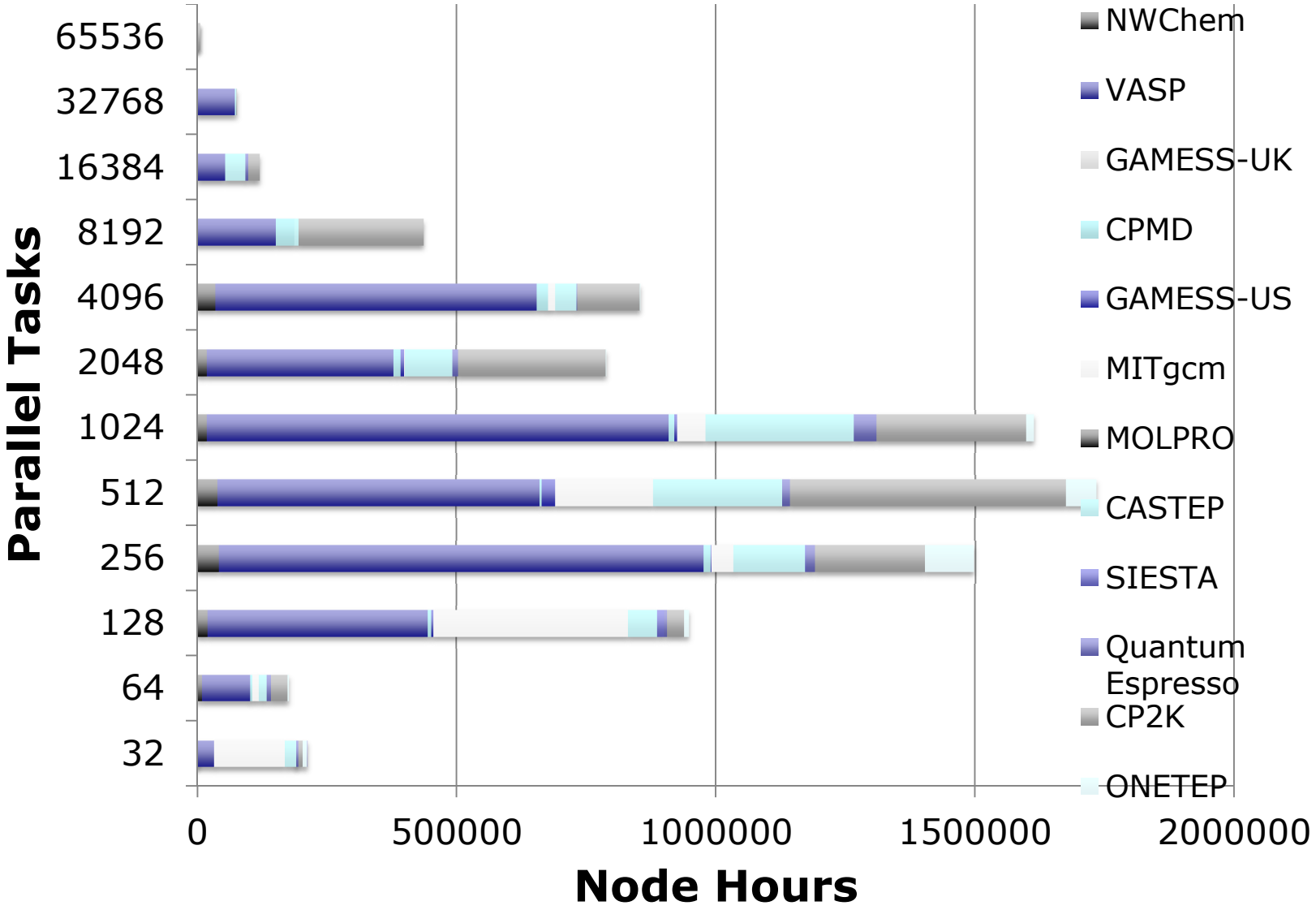
35% of the Chemistry software on HECToR is using DFT methods.

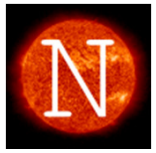




http://www.nu-fuse.com

HECToR Usage Statistics

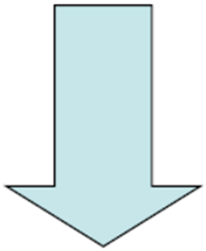




DFT Issues

Many Body Schrodinger Equation (exponential scaling)

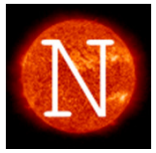
$$\{-\sum_i \frac{1}{2}\nabla_i^2 + \sum_{i,j} \frac{1}{|r_i - r_j|} + \sum_{i,I} \frac{Z}{|r_i - R_I|}\}\Psi(r_1,..r_N) = E\Psi(r_1,..r_N)$$



Kohn Sham Equation (65): The many body ground state problem can be mapped onto a single particle problem with the same electron density and a different effective potential (cubic scaling).

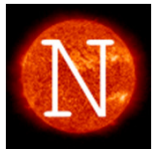
$$\{-\frac{1}{2}\nabla^2 + \int \frac{\rho(r')}{|r - r'|}dr' + \sum_I \frac{Z}{|r - R_I|} + V_{XC}\}\psi_i(r) = E_i\psi_i(r)$$

$$\rho(r) = \sum_i |\psi_i(r)|^2 = |\Psi(r_1,..r_N)|^2$$



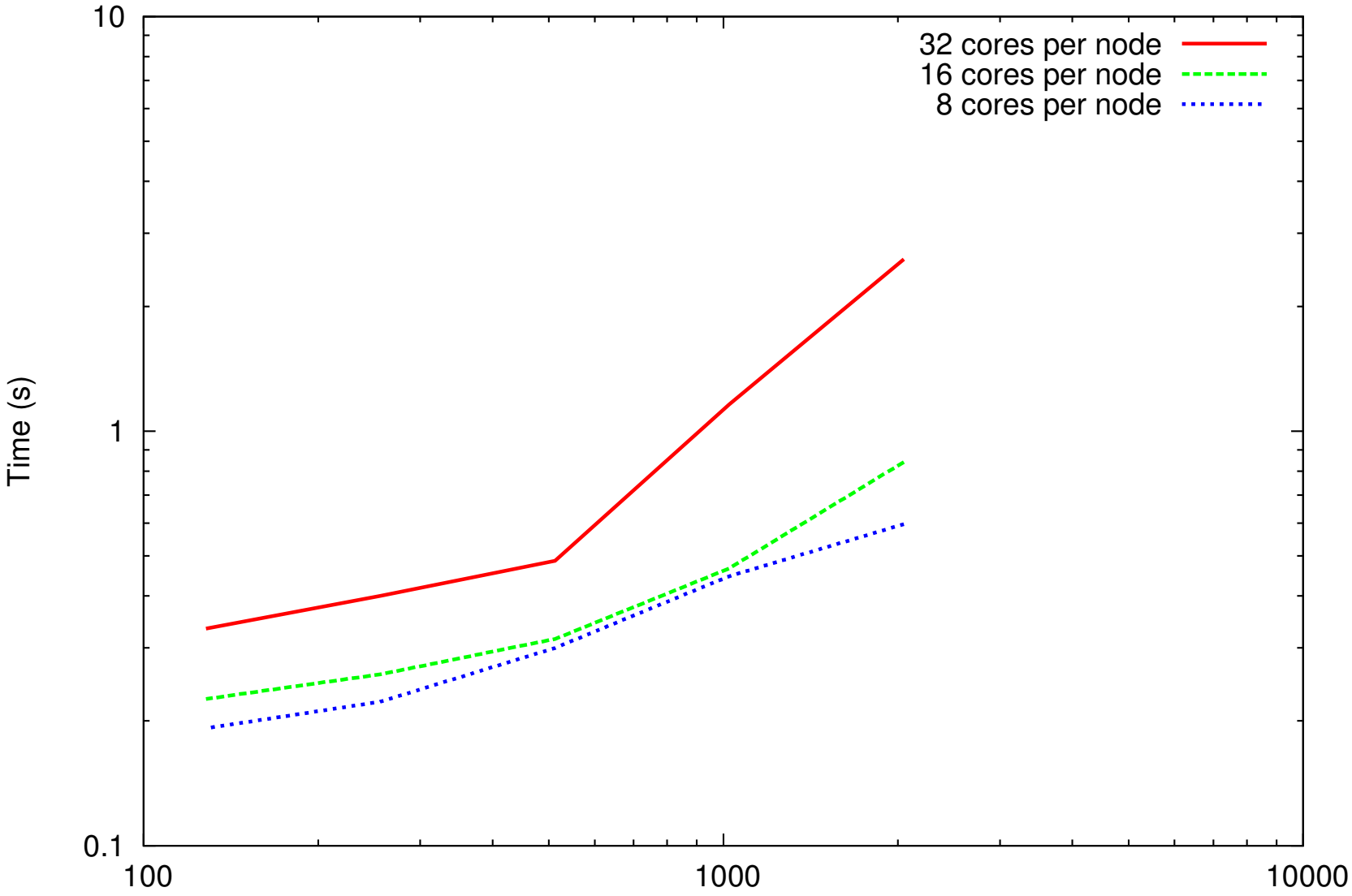
DFT Issues

- Pseudo-Potential Plane-wave
  - Good coverage, no region bias
  - Coverage where not needed (cubic scaling)
  - Relies heavily on 3D FFTs
    - Orthoganlisation
    - Diagonalisation
    - 3D FFTs
    - Pseudo-potentials
- CASTEP simulation of 64 atoms (Ti-Al-V): on 64 cores 13.5% of execution time is spent performing MPI_alltoallv for FFTs. Scaling 128 cores causes MPI to dominate, using 50% of the execution time

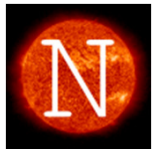


Scaling of 3D FFTs N=128^3

http://www.nu-fuse.com

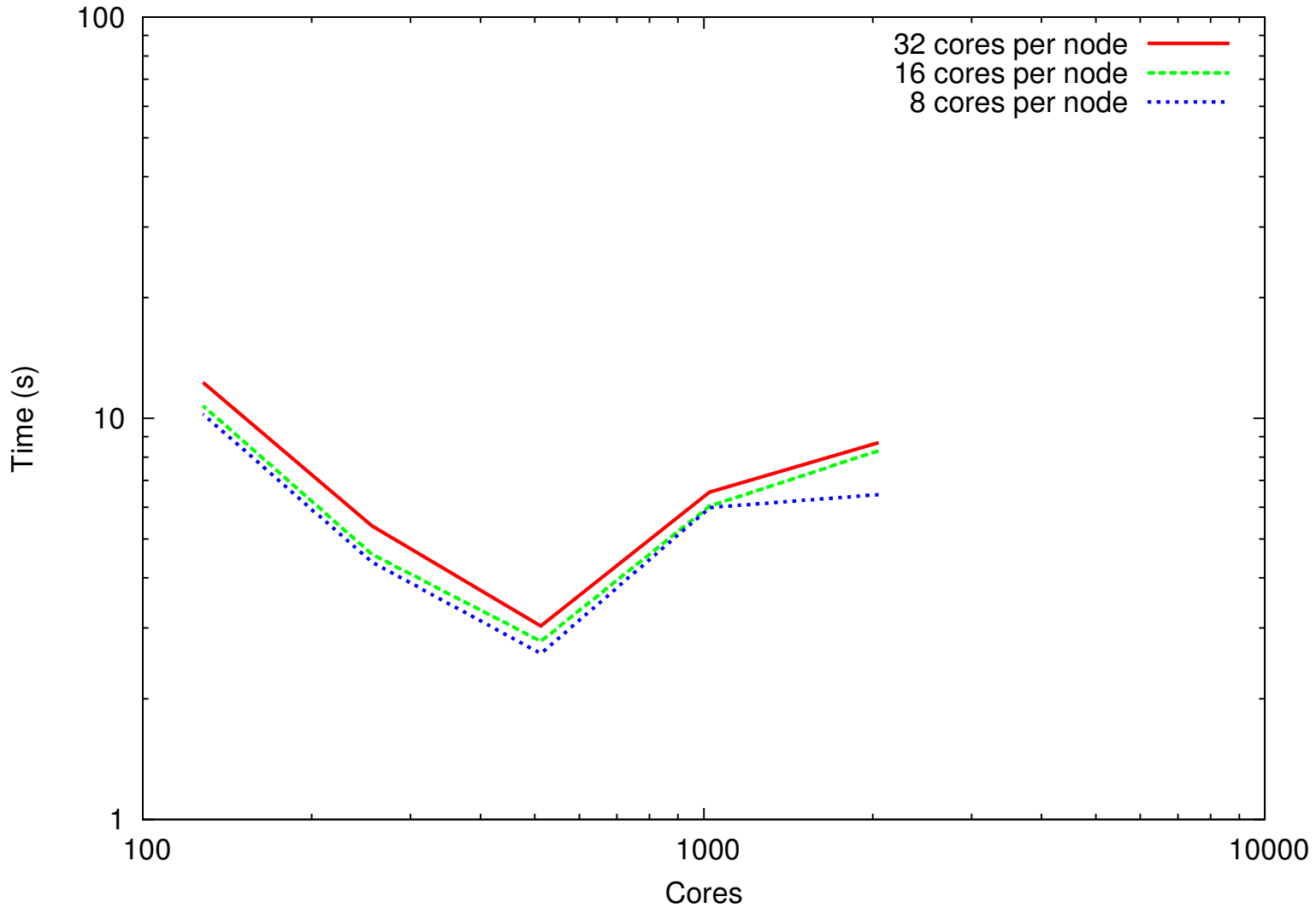


cea | epcc | IPP Max-Planck-Institut für Plasmaphysik | JÜLICH FORSCHUNGSZENTRUM | PPPL PRINCETON PLASMA PHYSICS LABORATORY

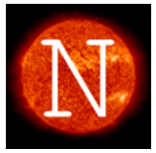


Scaling of 3D FFTs $N=1024^3$

http://www.nu-fuse.com

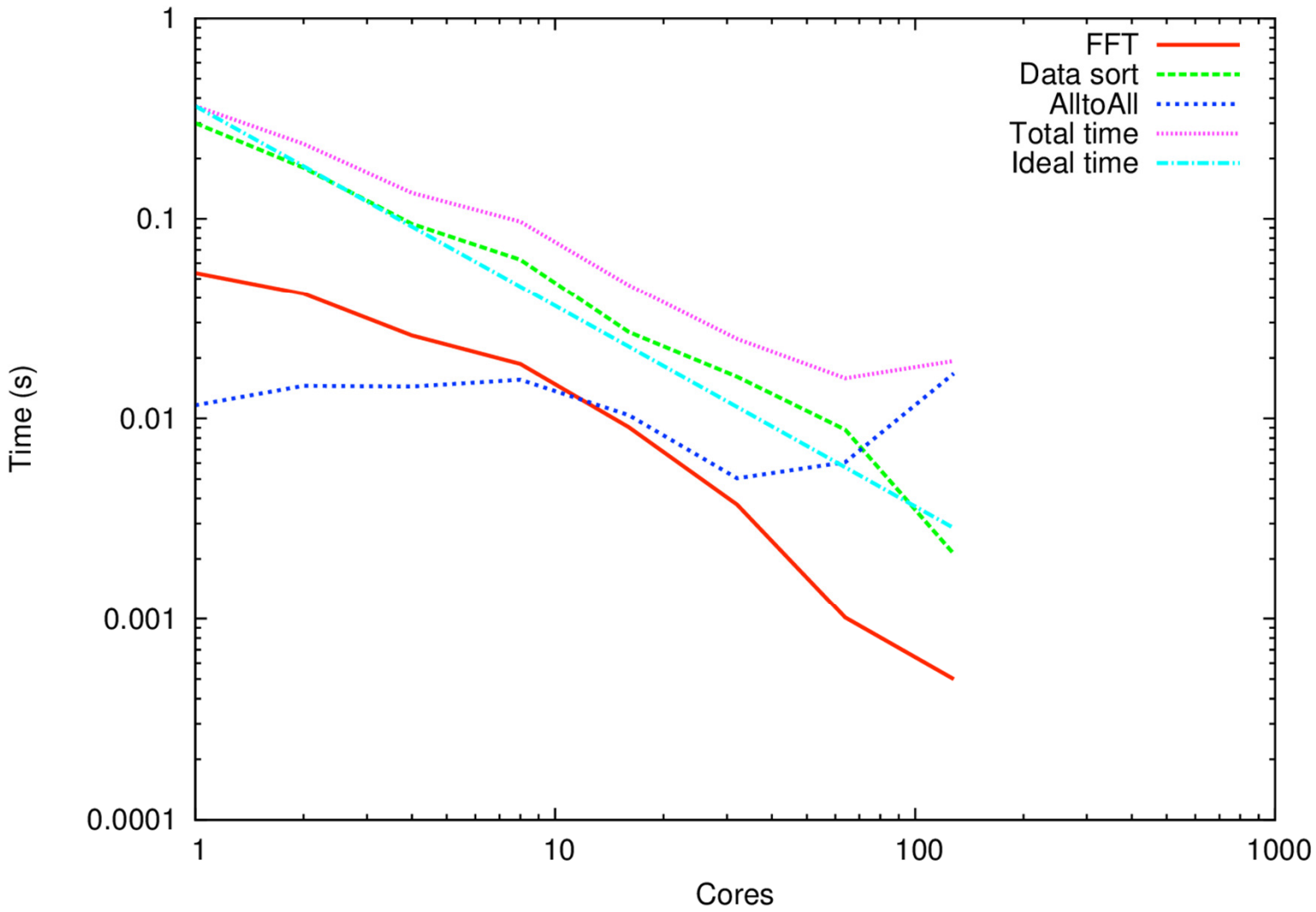


cea | epcc | IPP Max-Planck-Institut für Plasmaphysik | JÜLICH FORSCHUNGSZENTRUM | PPPL PRINCETON PLASMA PHYSICS LABORATORY

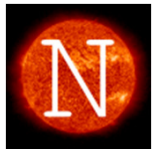


http://www.nu-fuse.com

3x1D FFT and MPI, N=128^3

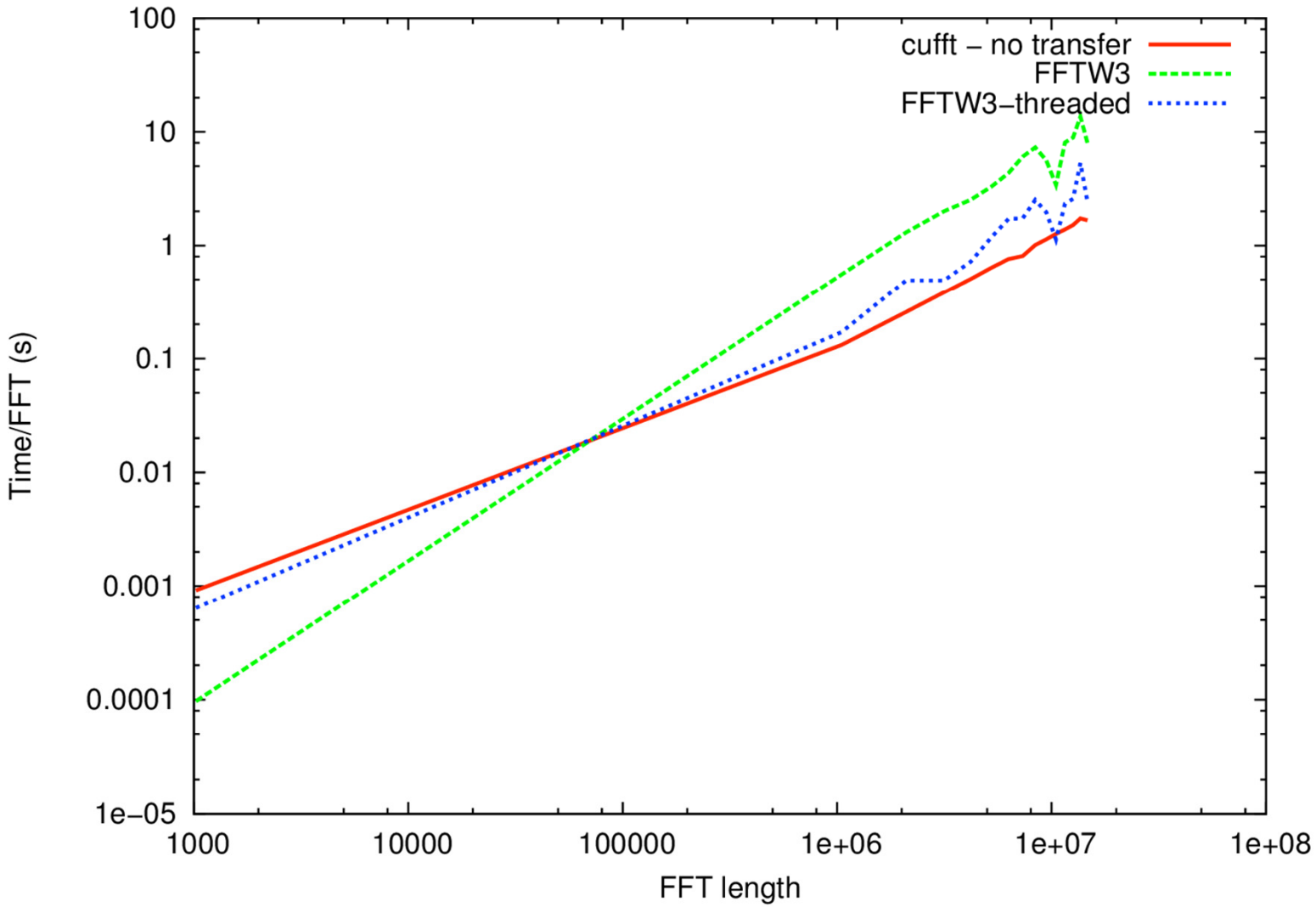


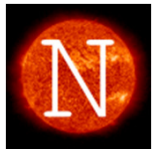
cea | epcc | IPP Max-Planck-Institut für Plasmaphysik | JÜLICH FORSCHUNGSZENTRUM | PPPL PRINCETON PLASMA PHYSICS LABORATORY



http://www.nu-fuse.com

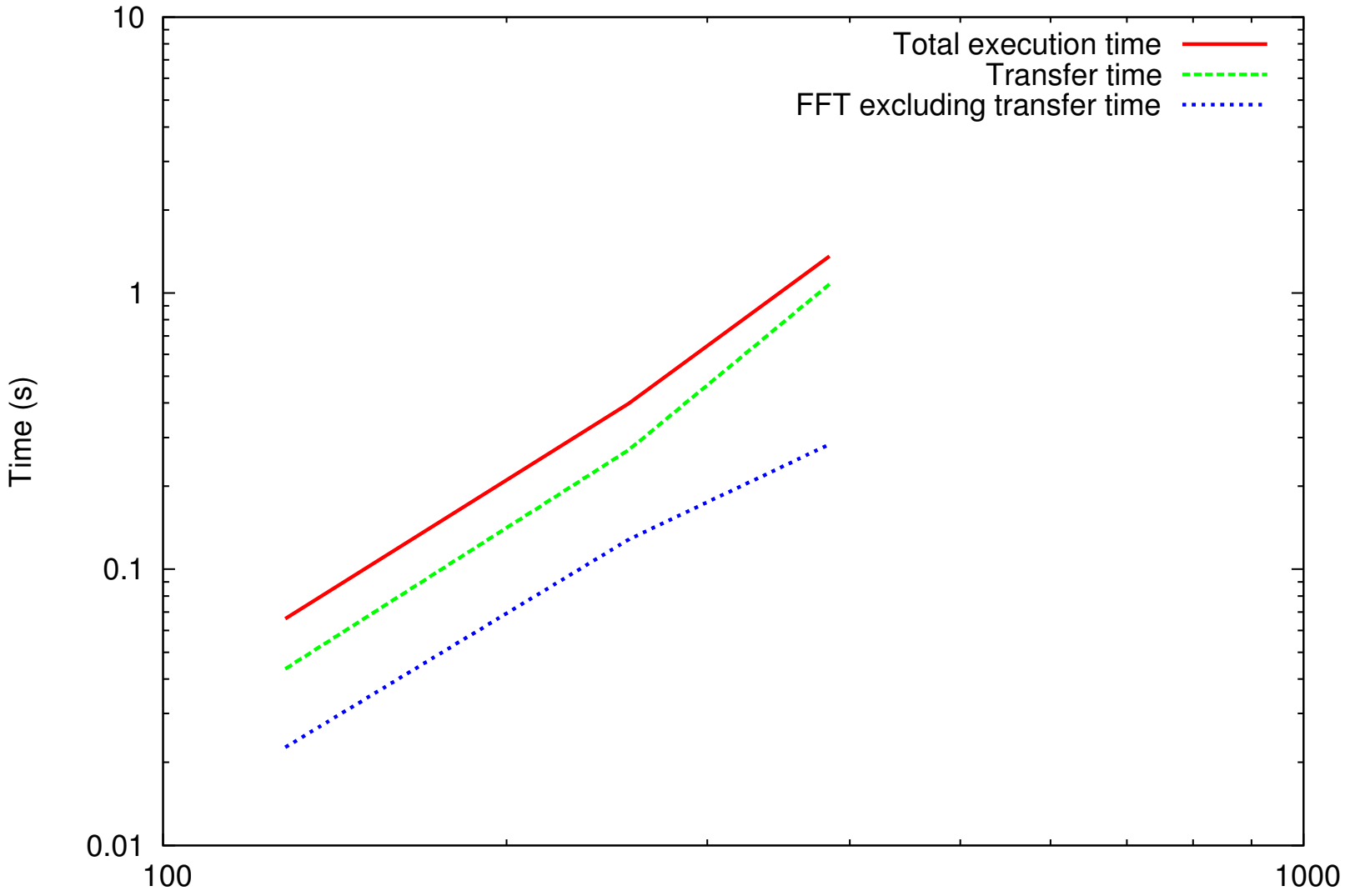
GPU 1D FFT



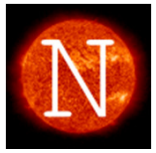


GPGPU: 3D FFT using cu-FFT

http://www.nu-fuse.com

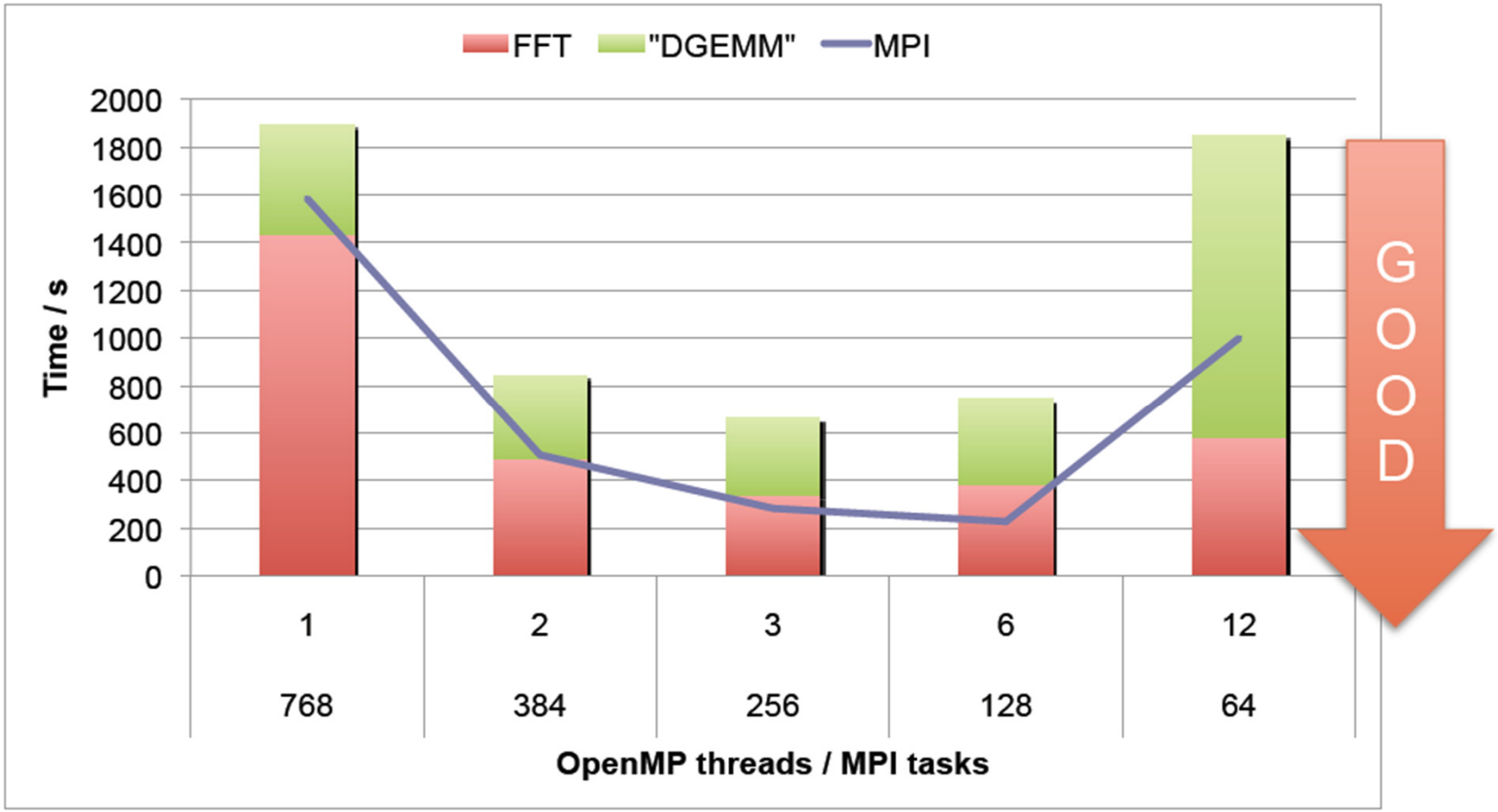


cea | epcc | IPP Max-Planck-Institut für Plasmaphysik | JÜLICH FORSCHUNGSZENTRUM | PPPL PRINCETON PLASMA PHYSICS LABORATORY

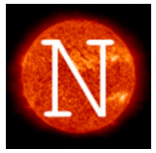


Optimisations

Work by Berekley Labs



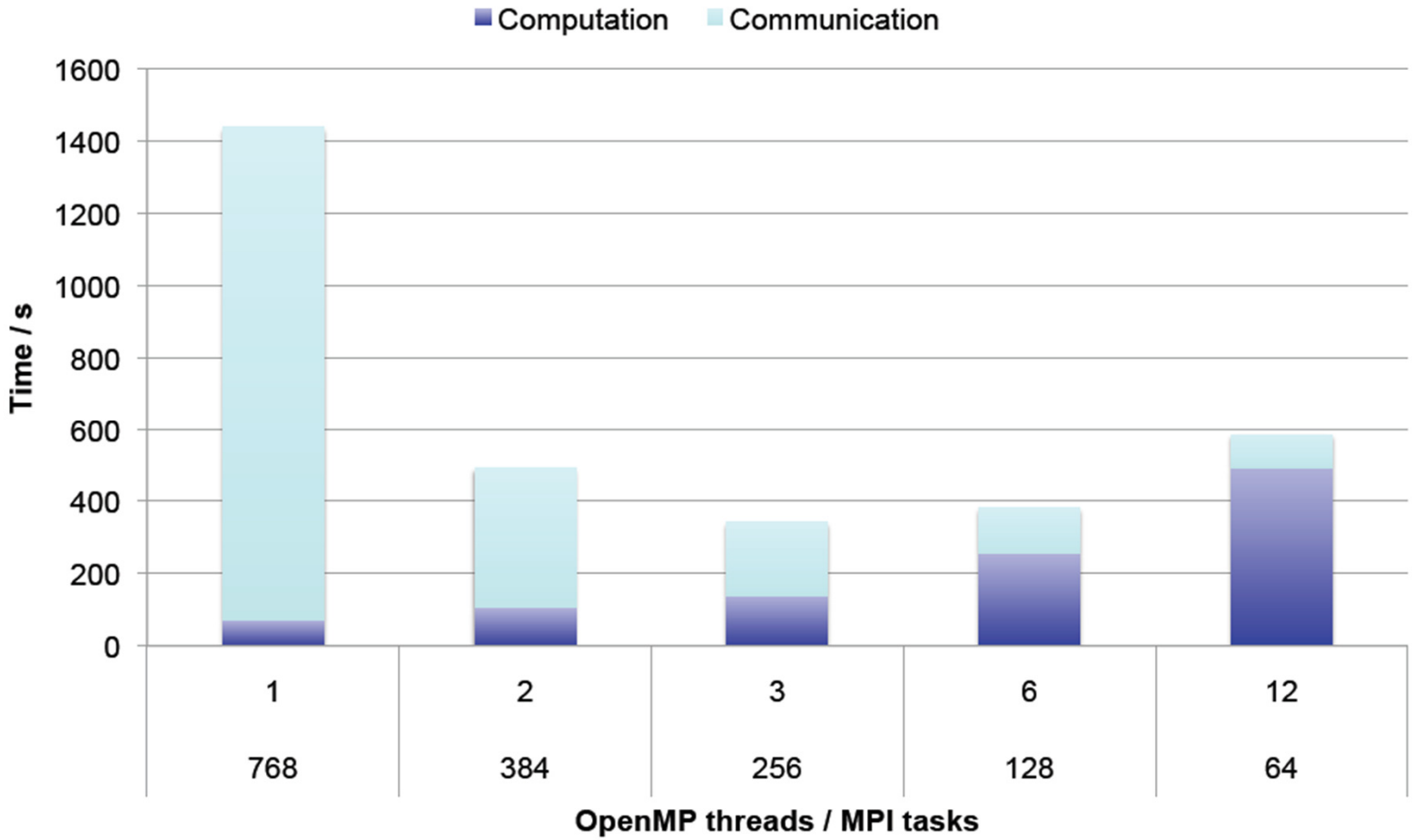
- Hybrid parallelisation
- Reduce FFT processes



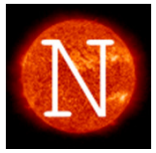
http://www.nu-fuse.com

FFT part

Work by Berekley Labs



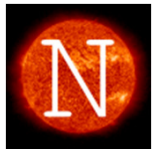
cea | epcc | IPP Max-Planck-Institut für Plasmaphysik | JÜLICH FORSCHUNGSZENTRUM | PPPL PRINCETON PLASMA PHYSICS LABORATORY



http://www.nu-fuse.com

Optimisation

- Optimisations – linear scaling
  - Real-space-grid algorithms (high-order finite difference method to calculate derivatives such as the kinetic-energy operator + localised wave functions)
  - Localised Basis sets
  - Not clear beneficial for metals



Conclusions

- DFT generally performed through packages
  - This means the optimisations should be done for you
- Accelerators and complex hardware becoming more common