Libraries

© Simon McIntosh-Smith,
Gethin Williams

Rationale

Newton: “If I have seen further it is by standing on the shoulders of giants”

- Saves time
- Less buggy
- More efficient implementation
- Access to parallel algorithms
- But...

  The only way to appreciate an algorithm and the challenges of implementing it in parallel is to code it.

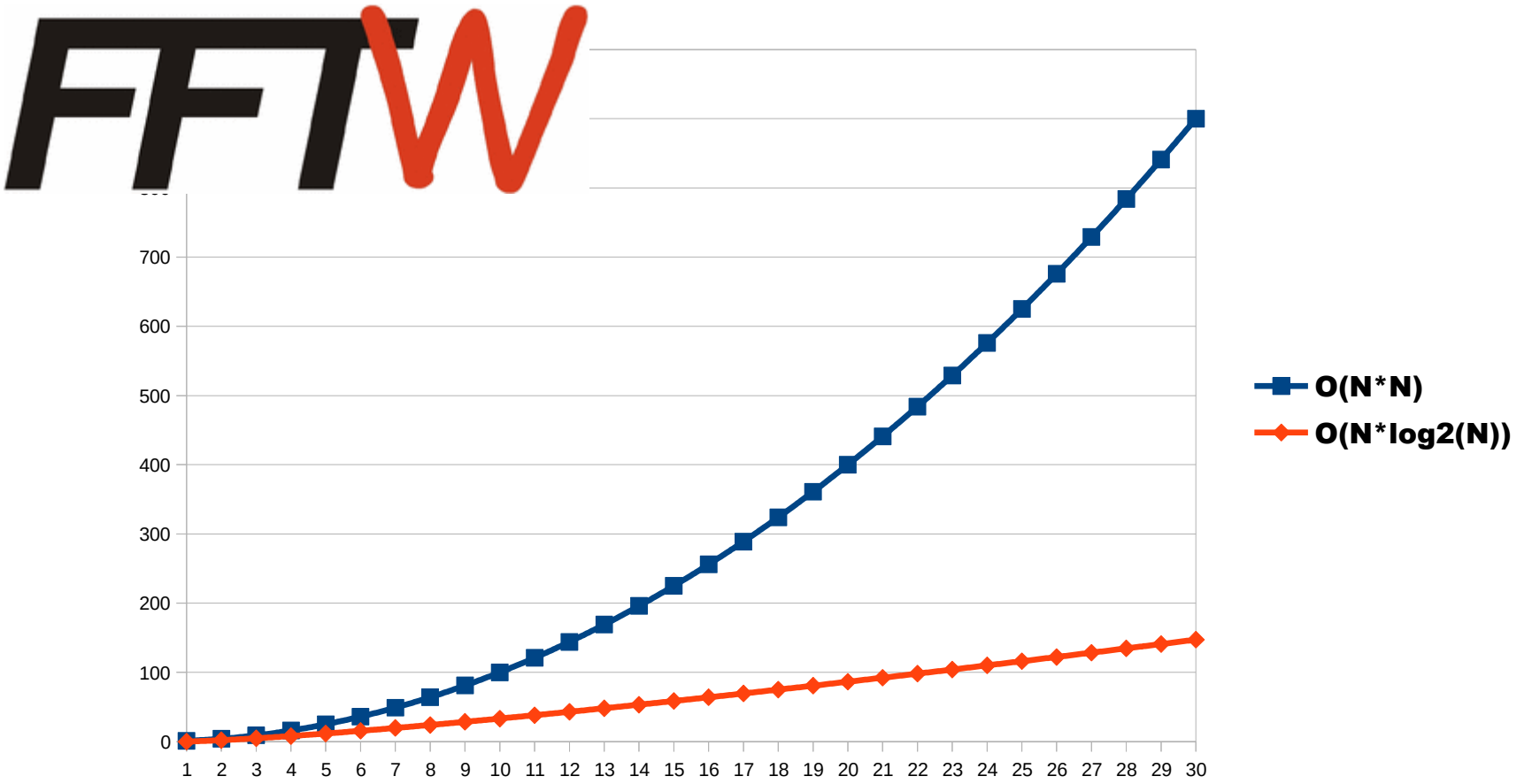
© Simon McIntosh-Smith,
Gethin Williams

Libraries and the 7 Dwarfs

Computational kernel	Example Library/Package
Spectral	FFTW
Dense linear algebra	BLAS + LAPACK, ATLAS, ScaLAPACK, PLASMA, MAGMA, Intel MKL
Sparse linear algebra	PetSc, Trilinos
N-Body	GRAPE, Fast Multipole Method, OpenMM
Structured grids	CHOMBO, OPS
Unstructured grids	ELMER (FEM), OP2
Map Reduce	Hadoop

© Simon McIntosh-Smith,
Gethin Williams

Spectral: Nlog(N) is worth having!

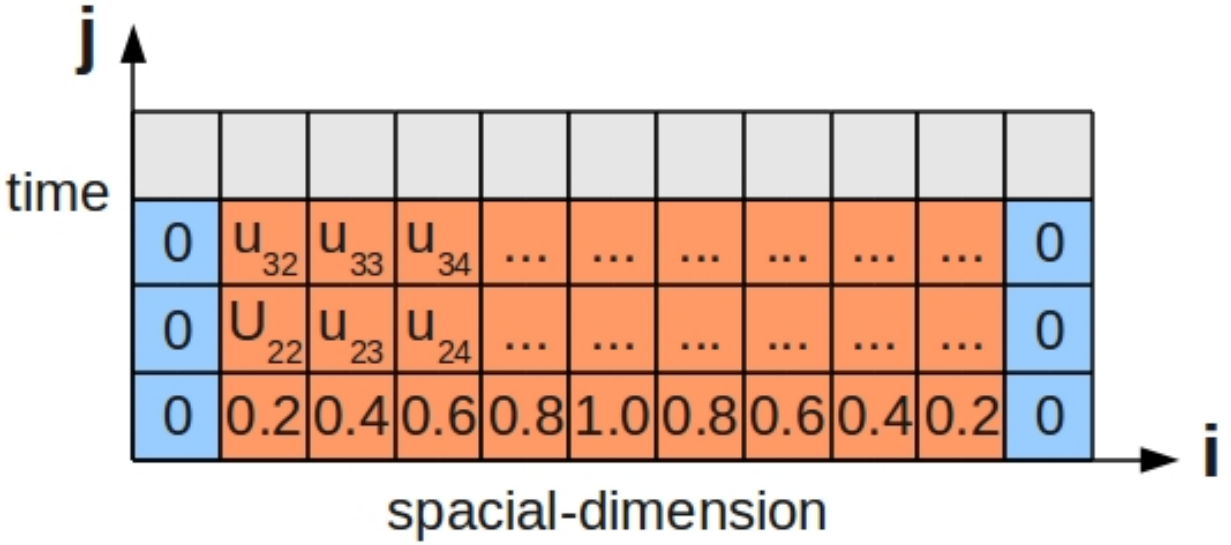


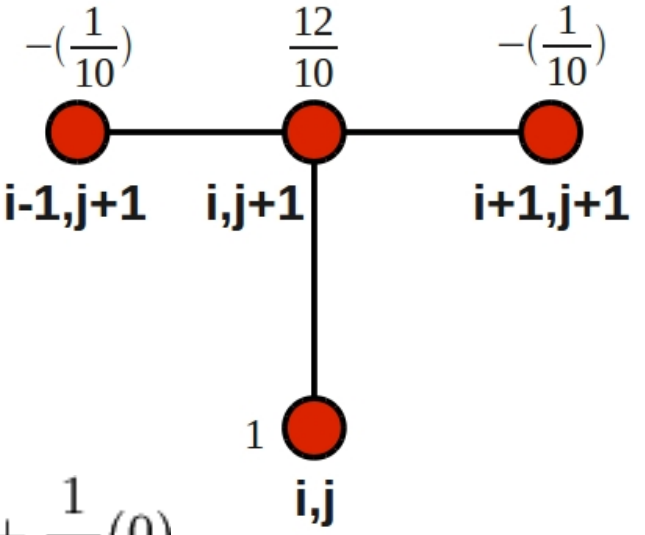
A Fourier transform is O(N*N) operations. An FFT is O(N*Log(N)).

© Simon McIntosh-Smith,
Gethin Williams

University of BRISTOL

Dense Linear Algebra: Solving PDEs





$$\frac{1}{10}(0) + \frac{12}{10}(u_{22}) - \frac{1}{10}(u_{23}) = 0.2 + \frac{1}{10}(0)$$

$$\frac{1}{10}(u_{22}) + \frac{12}{10}(u_{23}) - \frac{1}{10}(u_{24}) = 0.4$$

$$\begin{bmatrix} 1.2 & -0.1 & 0 & 0 & \cdots & 0 \\ -0.1 & 1.2 & -0.1 & 0 & \cdots & 0 \\ \vdots & & & & & \vdots \\ & & & \cdots & 0 & -0.1 & 1.2 \end{bmatrix} \times \begin{bmatrix} u_{22} \\ u_{23} \\ u_{24} \\ \vdots \end{bmatrix} = \begin{bmatrix} 0.2 + 0.1(0) \\ 0.4 \\ \vdots \\ 0.2 + 0.1(0) \end{bmatrix}$$

© Simon McIntosh-Smith,
Gethin Williams

University of BRISTOL

Dense Linear algebra: BLAS

complexity of routine	# of floating point operations	# of memory accesses	Ratio of memory access to flops (lower is better)
BLAS-1 (vector-vector)	2n	3n	3/2
BLAS-2 (matrix-vector)	$2n^2 + n$	$n^2 + 3n$	~1/2
BLAS-3 (matrix-matrix)	$2n^3$	$3n^2$	3/(2n)

- 3/(2n) is small for large n
- We know that memory accesses are expensive and limit performance
- BLAS-3 routines have the best chance of approaching theoretical peak performance of the CPU

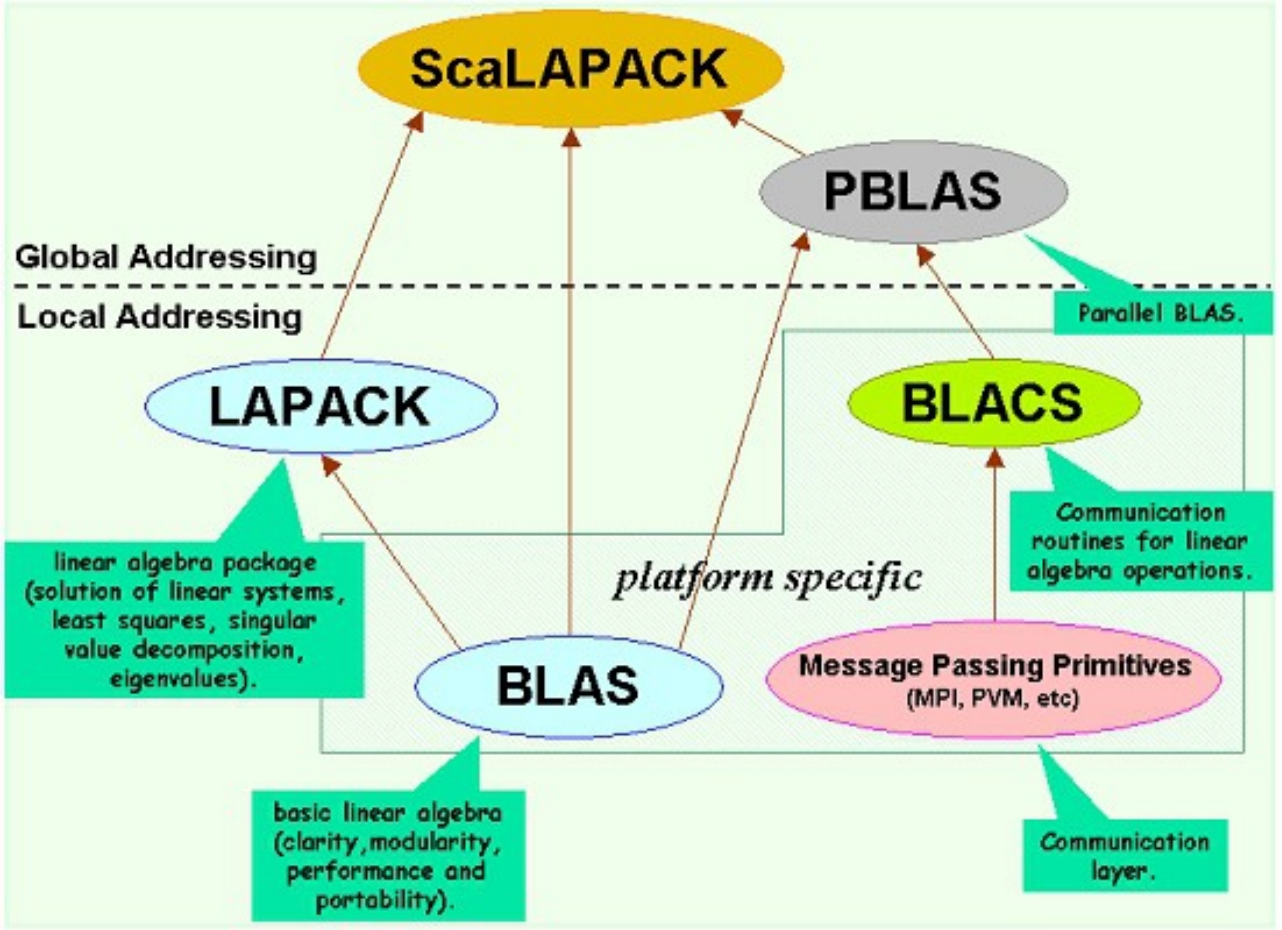
© Simon McIntosh-Smith,
Gethin Williams

Automatic Tuning of Linear Algebra Software (ATLAS)

- 'Blocked' algorithms are required to minimise the number of cache misses and hence gain best performance.
- The best block size is dependent upon the cache details for a particular microprocessor.
- ATLAS tries a number of experiments to empirically (and automatically) determine the best values for certain parameters, such as block size, length of loop to unroll etc.

© Simon McIntosh-Smith,
Gethin Williams

ScaLAPACK: MPI Parallel



© Simon McIntosh-Smith,
Gethin Williams

University of BRISTOL

Fast vendor BLAS libraries

- Intel's Math Kernel Library (MKL)
- Nvidia's CuBLAS
- AMD's ACML
- Open source clBLAS (on github, clFFT too!)

© Simon McIntosh-Smith,
Gethin Williams

Multi- and Many-core Versions

ICL UT INNOVATIVE COMPUTING LABORATORY THE UNIVERSITY of TENNESSEE

- http://icl.cs.utk.edu/plasma
- http://icl.cs.utk.edu/magma

TACC TEXAS ADVANCED COMPUTING CENTER

Powering Discoveries That Change The World

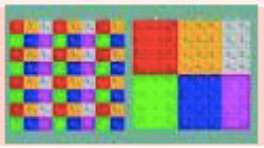
https://www.tacc.utexas.edu/research-development/tacc-software/gotoblas2

© Simon McIntosh-Smith,
Gethin Williams

University of BRISTOL

A New Generation of Software:

Parallel Linear Algebra Software for Multicore Architectures (PLASMA)

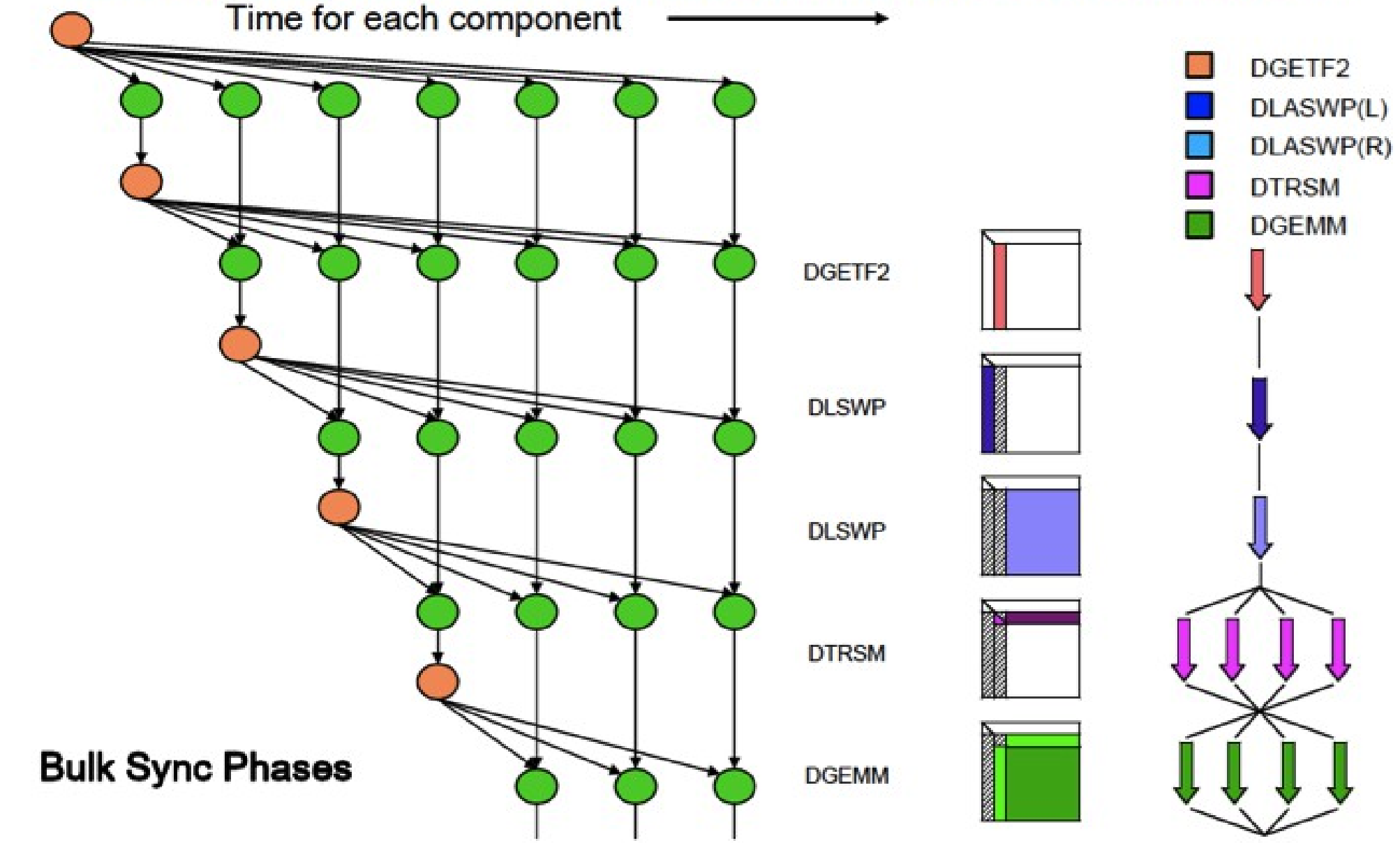
Software/Algorithms follow hardware evolution in time		
LINPACK (70's) (Vector operations)		Rely on - Level-1 BLAS operations
LAPACK (80's) (Blocking, cache friendly)		Rely on - Level-3 BLAS operations
ScaLAPACK (90's) (Distributed Memory)		Rely on - PBLAS Mess Passing
PLASMA (00's) New Algorithms (many-core friendly)		Rely on - a DAG/scheduler - block data layout - some extra kernels

Those new algorithms

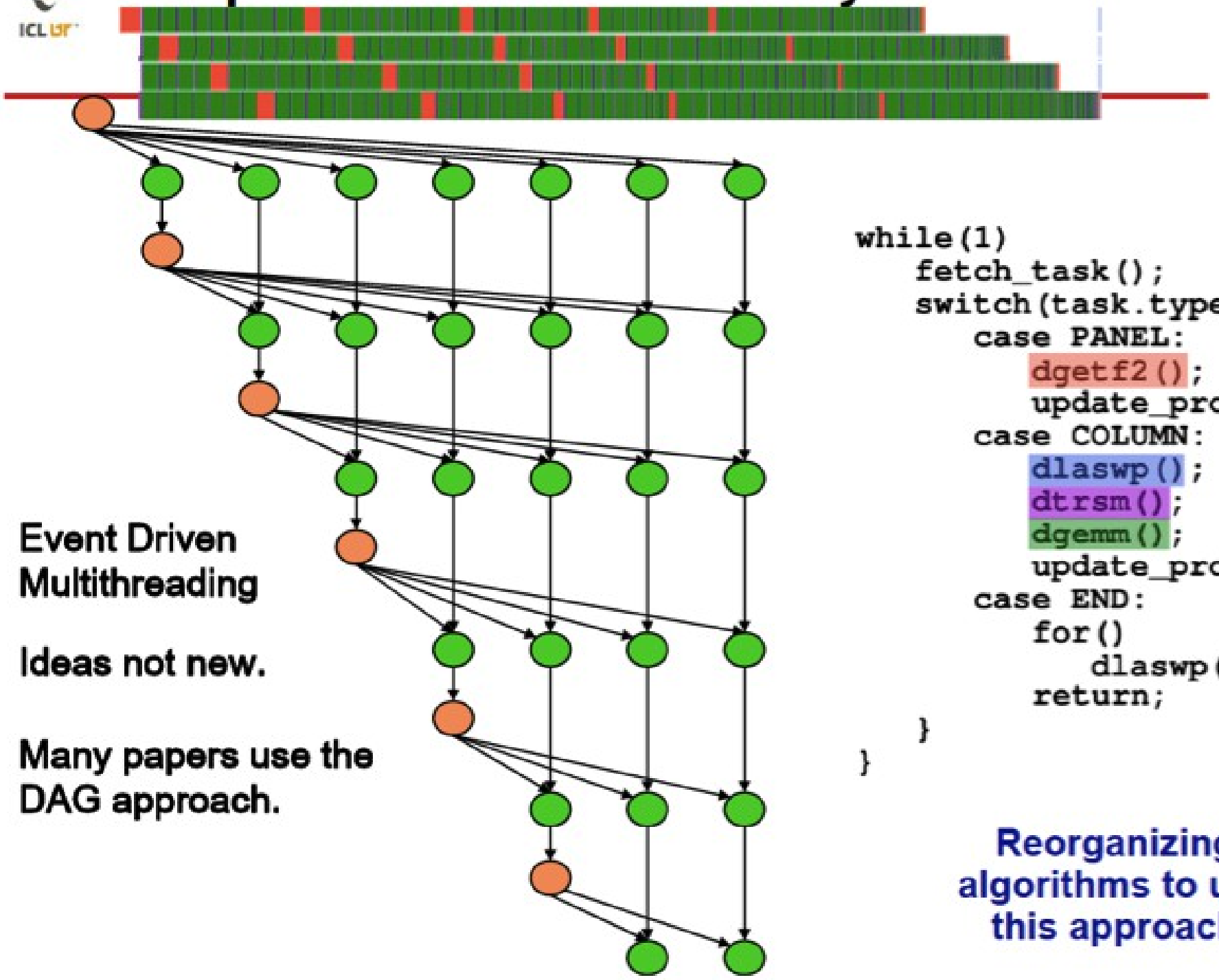
- have a very **low granularity**, they scale very well (multicore, petascale computing, …)
- **removes a lots of dependencies** among the tasks, (multicore, distributed computing)
- **avoid latency** (distributed computing, out-of-core)
- **rely on fast kernels**

Those new algorithms need new kernels and rely on efficient scheduling algorithms.

LU Timing Profile (4 processor system)



Adaptive Lookahead - Dynamic



Event Driven Multithreading

Ideas not new.

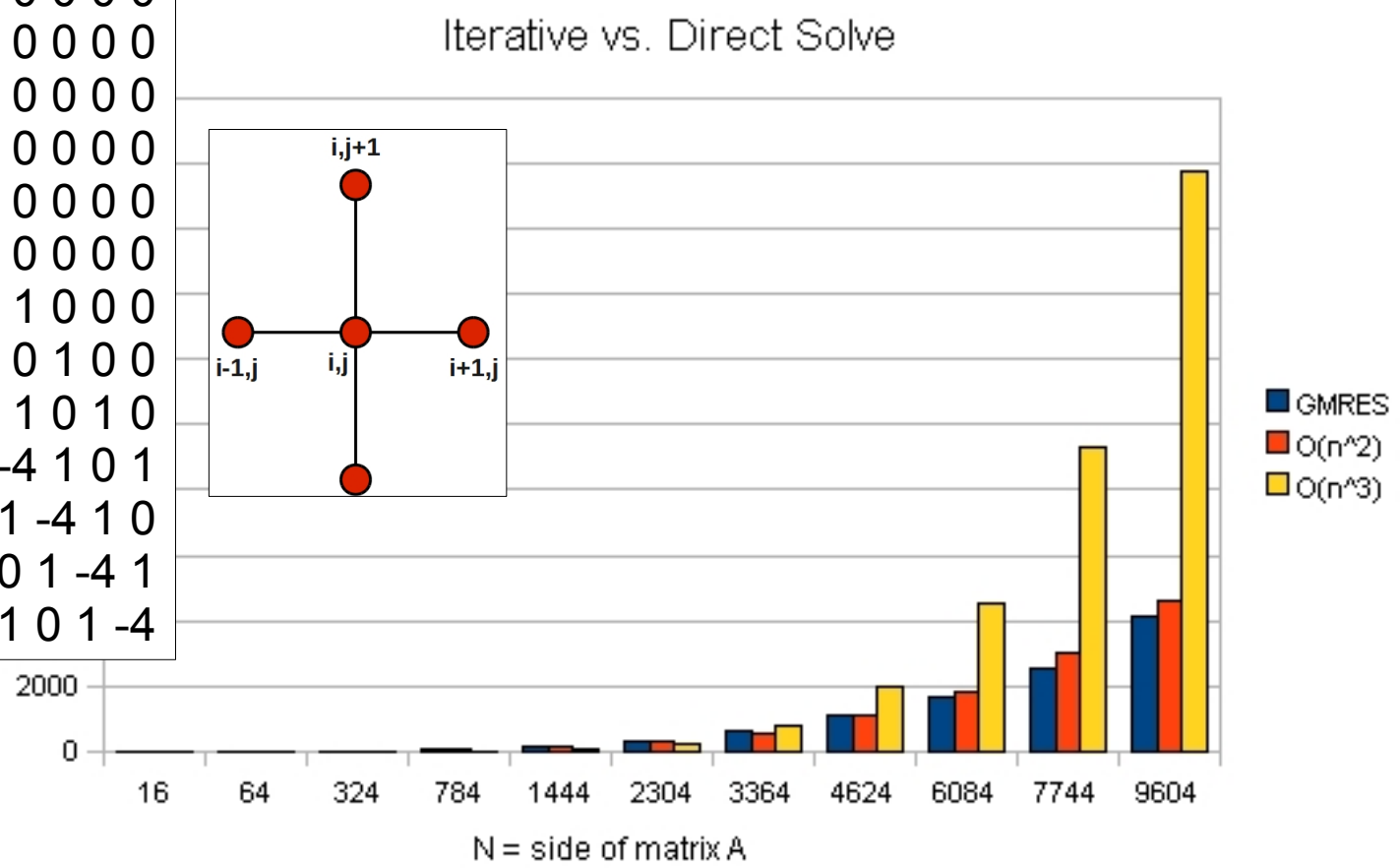
Many papers use the DAG approach.

```
while(1)
    fetch_task();
    switch(task.type) {
        case PANEL:
            dgetf2();
            update_progress();
        case COLUMN:
            dlaswp();
            dtrsm();
            dgemm();
            update_progress();
        case END:
            for()
                dlaswp();
            return;
    }
}
```

Reorganizing algorithms to use this approach

Sparse Linear Algebra

-4	1	0	1	0	0	0	0	0	0	0	0	0	0
1	-4	1	0	1	0	0	0	0	0	0	0	0	0
0	1	-4	1	0	1	0	0	0	0	0	0	0	0
1	0	1	-4	1	0	1	0	0	0	0	0	0	0
0	1	0	1	-4	1	0	1	0	0	0	0	0	0
0	0	1	0	1	-4	1	0	1	0	0	0	0	0
0	0	0	1	0	1	-4	1	0	1	0	0	0	0
0	0	0	0	1	0	1	-4	1	0	1	0	0	0
0	0	0	0	0	1	0	1	-4	1	0	1	0	0
0	0	0	0	0	0	1	0	1	-4	1	0	1	0
0	0	0	0	0	0	0	1	0	1	-4	1	0	1
0	0	0	0	0	0	0	0	1	0	1	-4	1	0
0	0	0	0	0	0	0	0	0	1	0	1	-4	1
0	0	0	0	0	0	0	0	0	0	1	0	1	-4



- PETSc: http://www.mcs.anl.gov/petsc

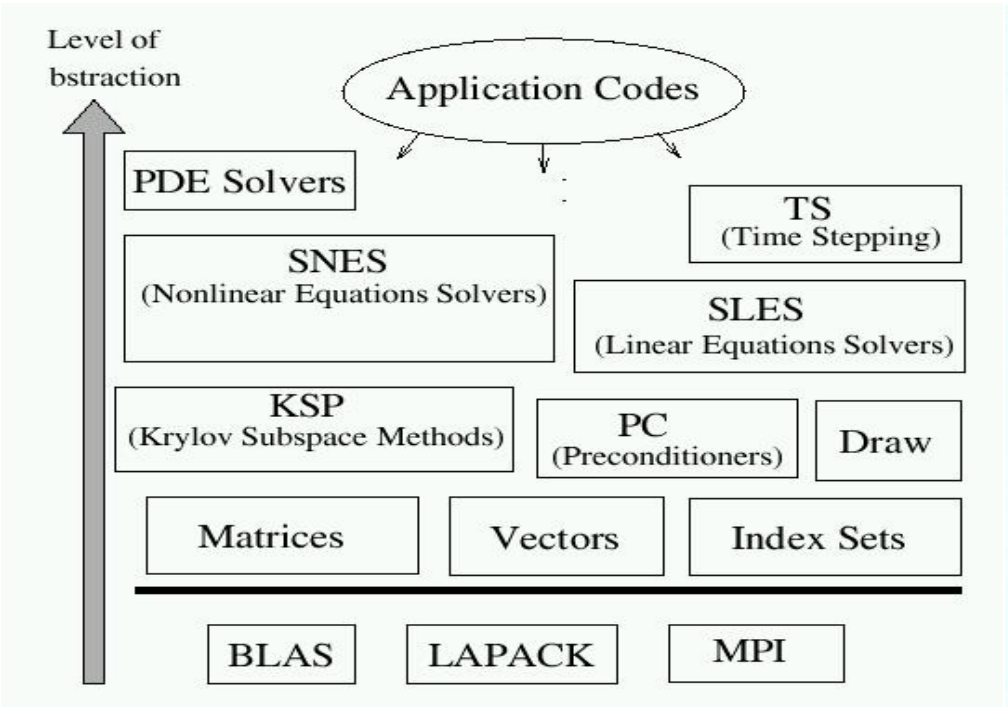
© Simon McIntosh-Smith,
Gethin Williams

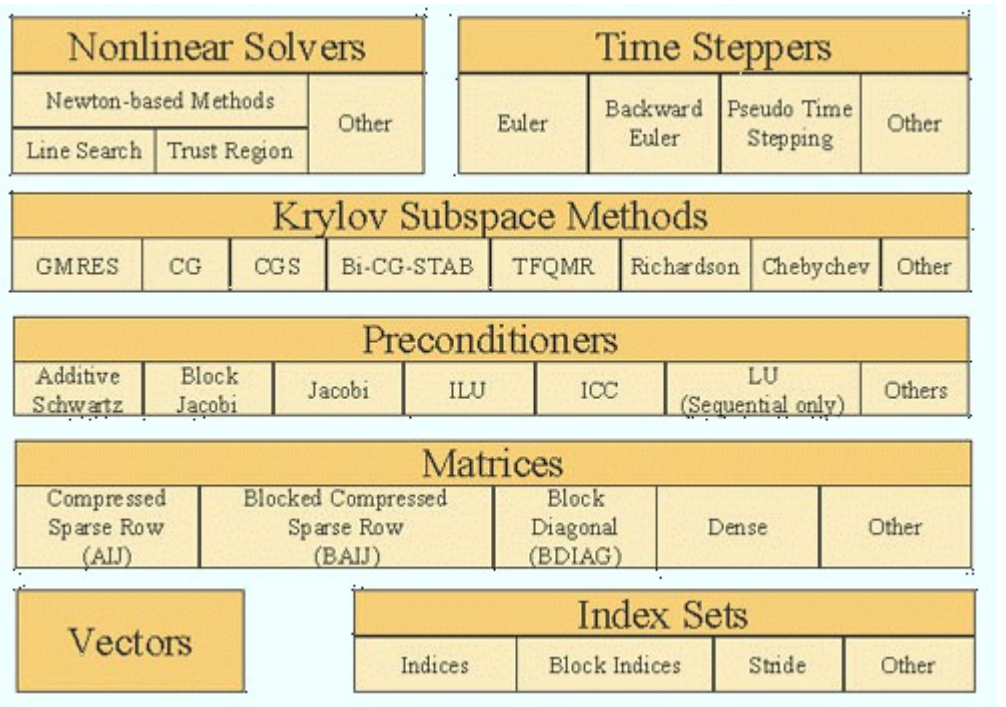
University of BRISTOL

Sparse Matrix Solver: PETSc

Argonne NATIONAL LABORATORY

- http://www.mcs.anl.gov/petsc/
- Threaded – multi-core
- MPI – distributed memory
- GPU – many-core

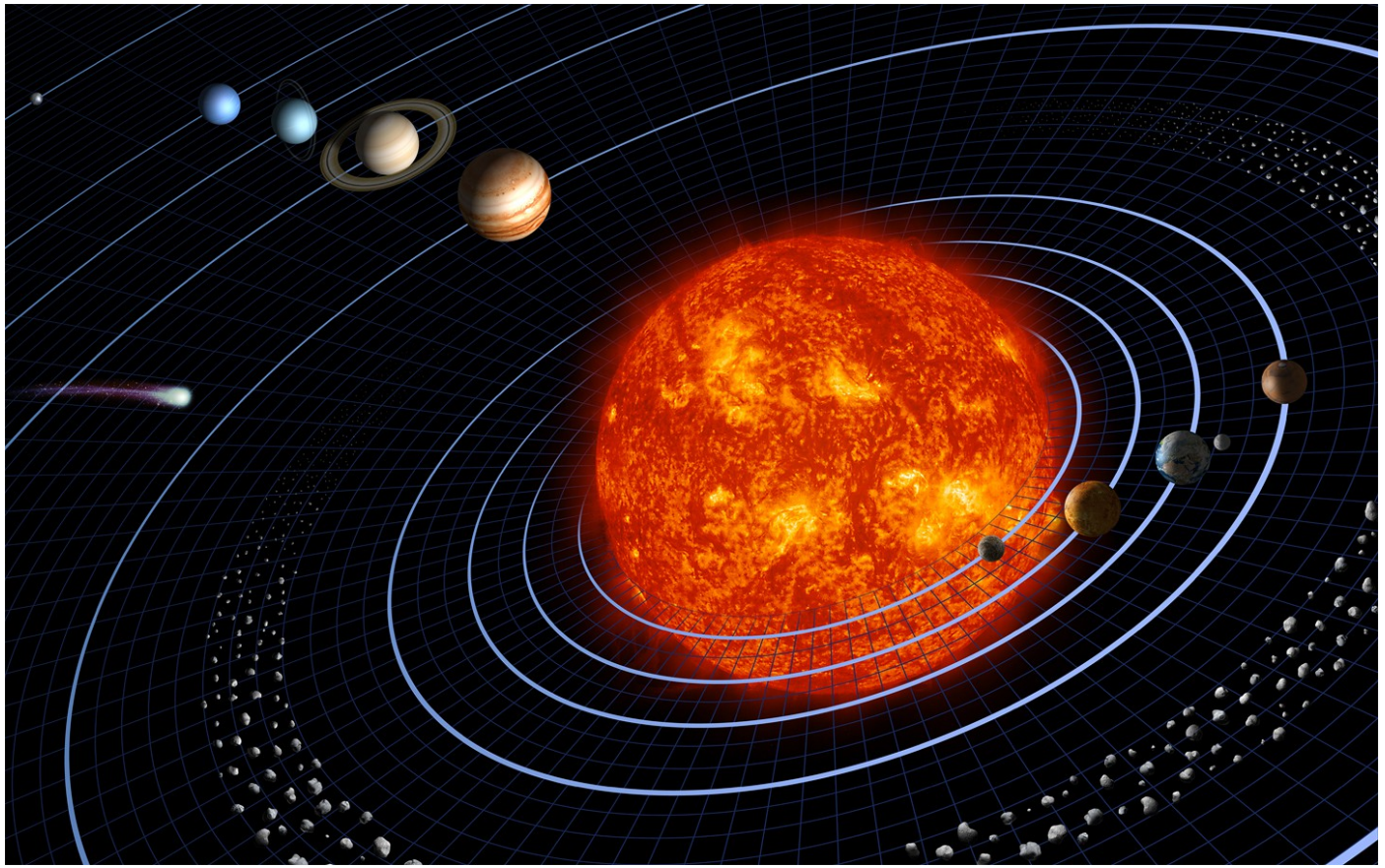




© Simon McIntosh-Smith,
Gethin Williams

University of BRISTOL

N-Body Problems



An N-body simulation numerically approximates the evolution of a system of bodies in which each body continuously interacts with every other body

© Simon McIntosh-Smith,
Gethin Williams

University of BRISTOL

OpenMM N-body library

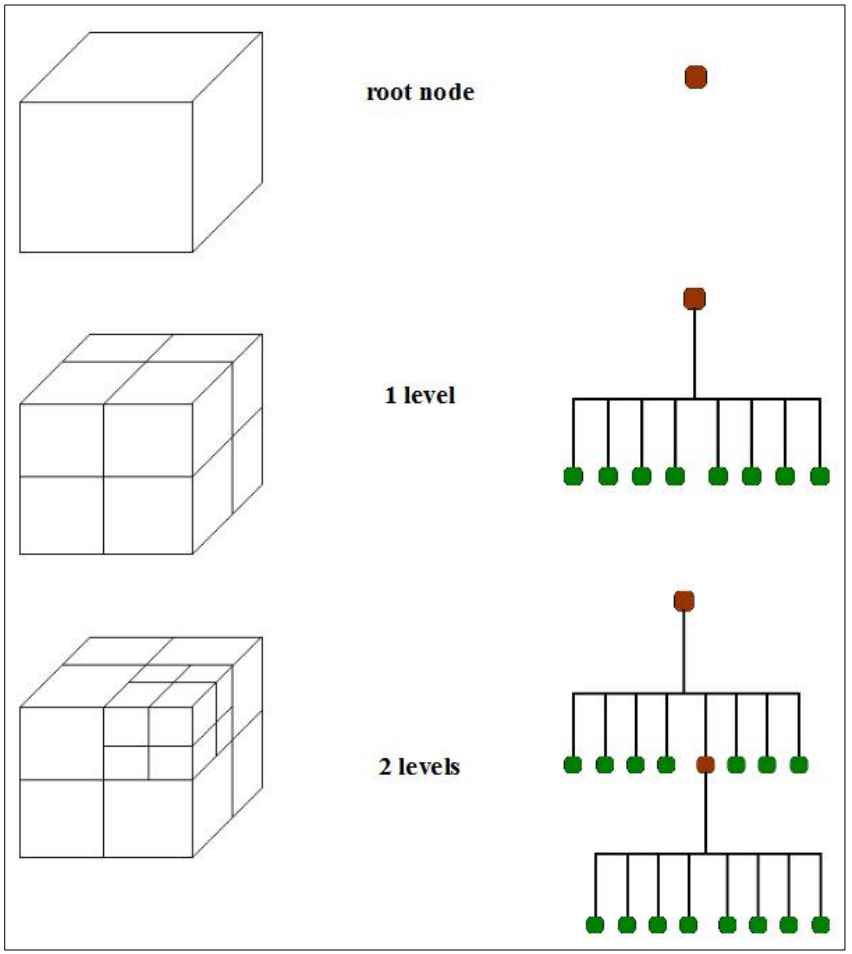
- Aimed at molecular mechanics (MM)
- Runs on CPUs, GPUs etc
- Easy to use from Python, C/C++, ...

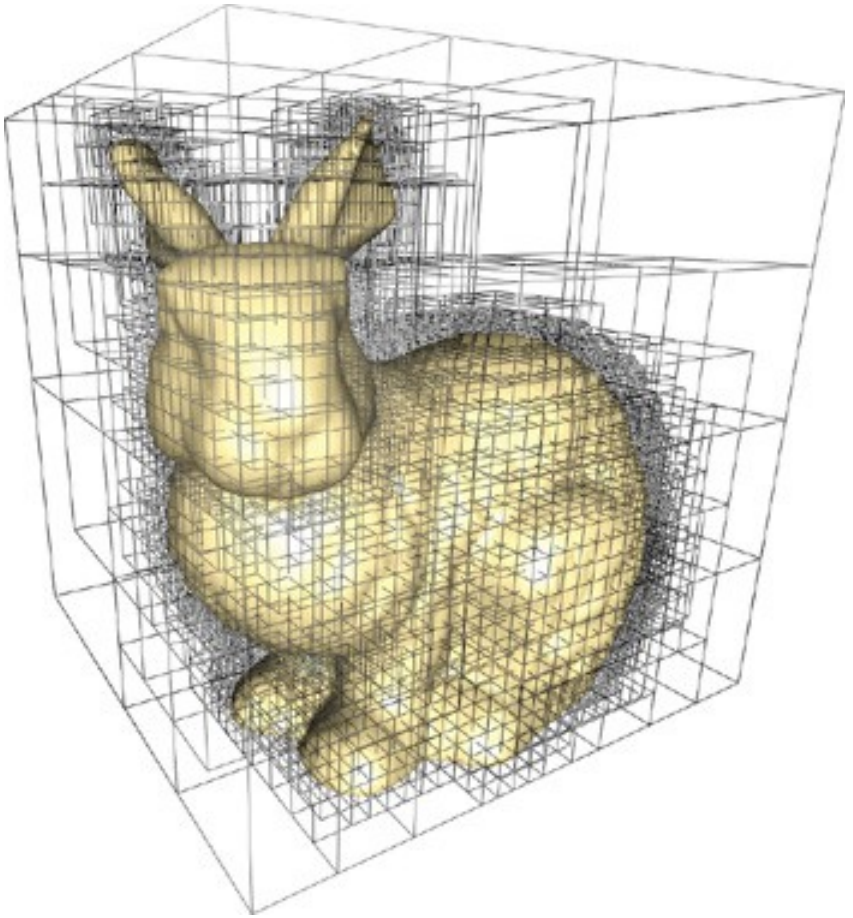
```
pdb = PDBFile('input.pdb')

forcefield = ForceField('amber99sb.xml', 'tip3p.xml')

system = forcefield.createSystem(pdb.topology, nonbondedMethod=PME,

nonbondedCutoff=1*nanometer, constraints=HBonds)

integrator = LangevinIntegrator(300*kelvin, 1/picosecond, 0.002*picoseconds) simulation =
Simulation(pdb.topology, system, integrator) simulation.context.setPositions(pdb.positions)

simulation.minimizeEnergy() simulation.reporters.append(PDBReporter('output.pdb', 1000))
simulation.step(10000)
```

© Simon McIntosh-Smith,
Gethin Williams

University of BRISTOL

Tree methods: Fast Multipole Method (FMM)



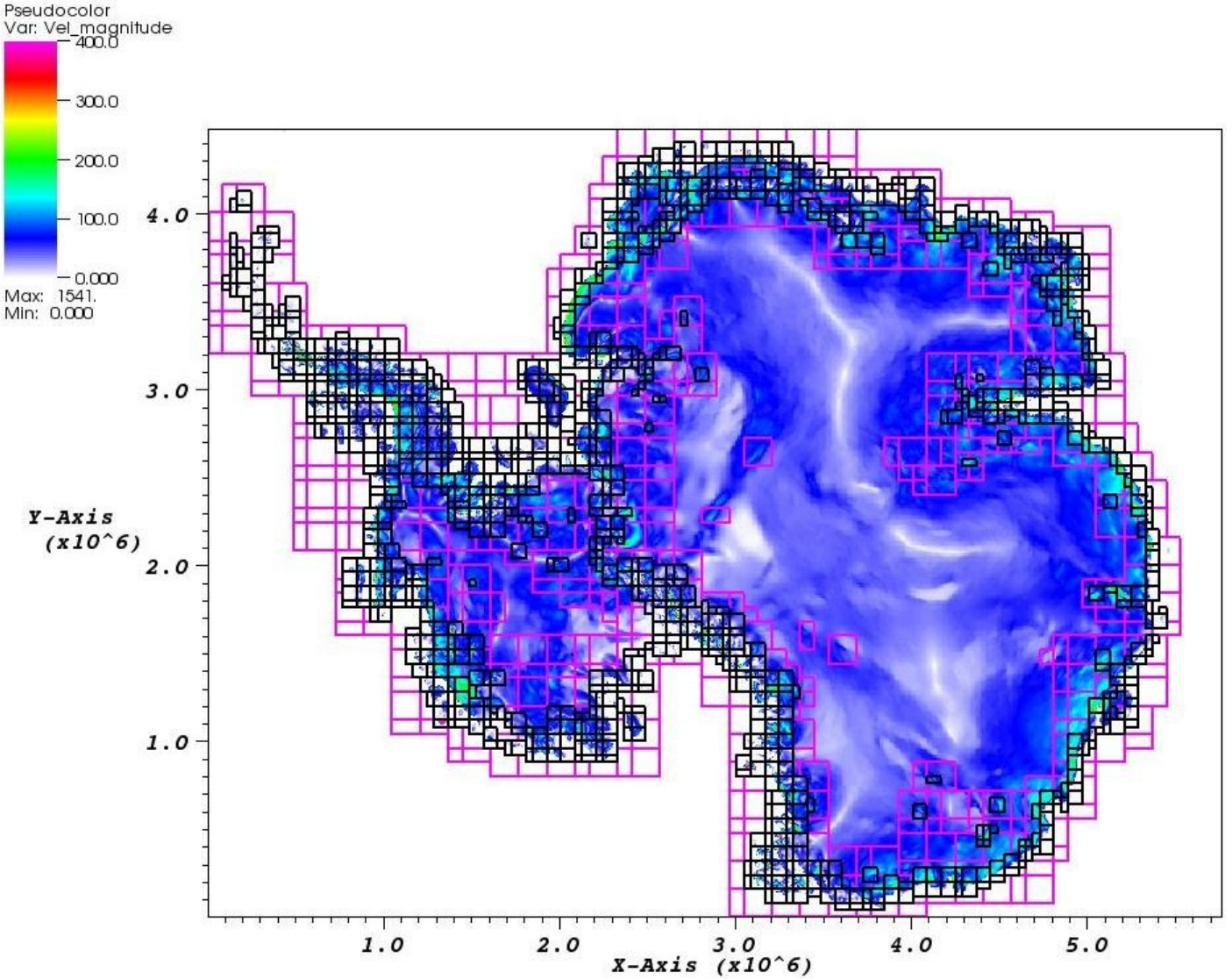


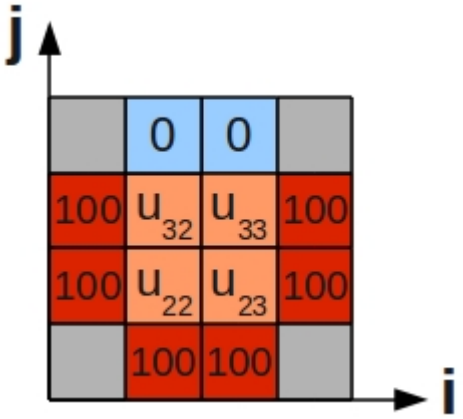
FMM in PETSc: http://barbagroup.bu.edu/Barba_group/PetFMM.html

© Simon McIntosh-Smith,
Gethin Williams

University of BRISTOL

Structured Grids





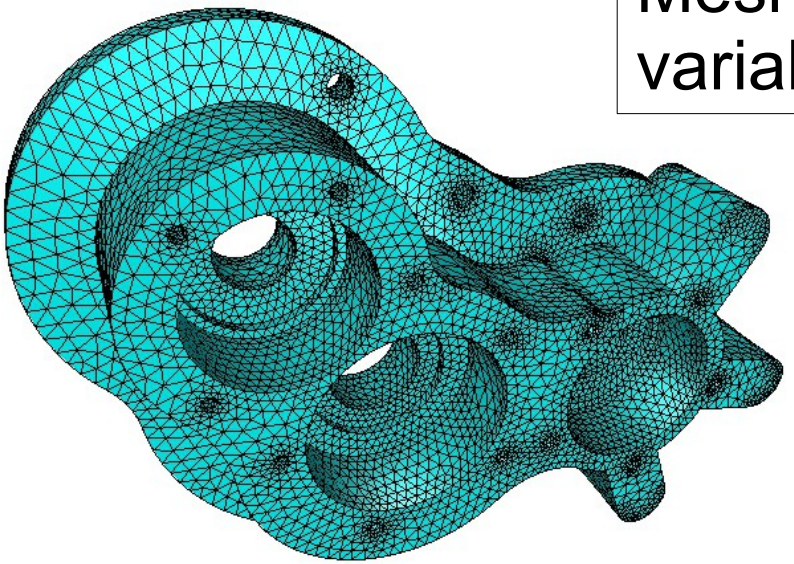
search: CHOMBO, OPS

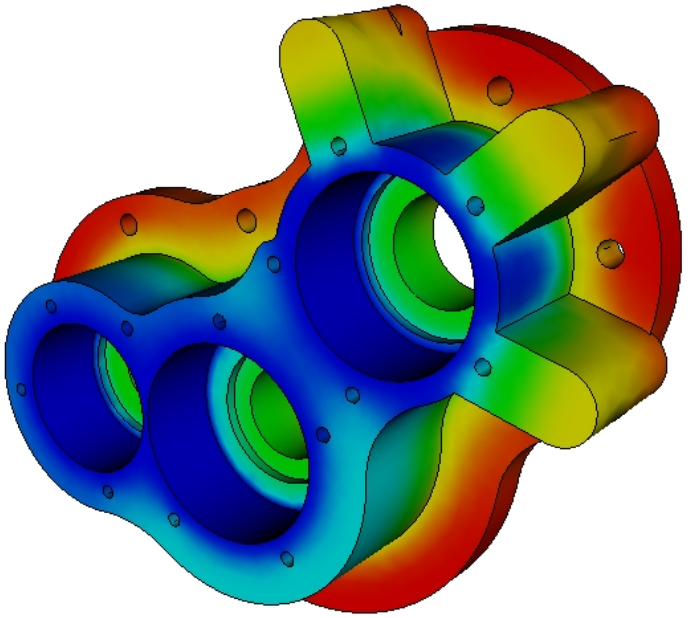
© Simon McIntosh-Smith,
Gethin Williams

University of BRISTOL

Unstructured Grids

Mesh assembled from variable sized polygons





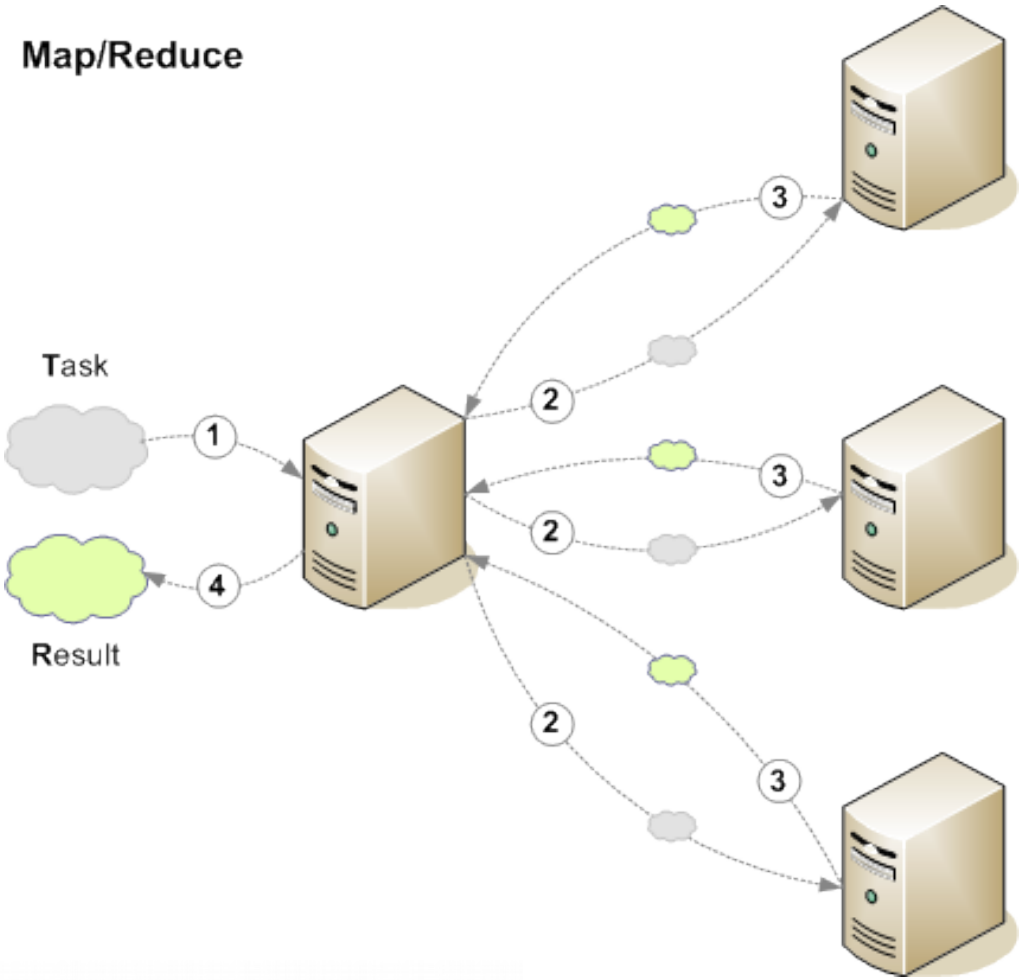
Computational expense can be focused on areas of interest

ELMER: http://www.csc.fi/english/pages/elmer
OP2: http://www.oerc.ox.ac.uk/projects/op2

© Simon McIntosh-Smith,
Gethin Williams

University of BRISTOL

MapReduce



http://hadoop.apache.org/

© Simon McIntosh-Smith,
Gethin Williams

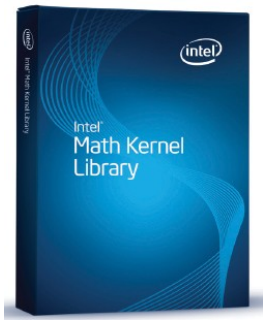
University of BRISTOL

Others

QuantLib

nag Results Matter. Trust NAG.

GSL: http://www.gnu.org/software/gsl/



AMD
ACML

NVIDIA®

http://developer.nvidia.com/

© Simon McIntosh-Smith,
Gethin Williams

University of BRISTOL

Recap: Libraries..

- Are typically well implemented & tested.
- Offer us parallel implementations of common computational kernels.
- Can fast-track us to real-world applications.
- Provide an abstraction between your application and the details of the implementation.
- Hard (impossible?) to beat for performance!

© Simon McIntosh-Smith,
Gethin Williams

Don't reinvent the wheel!

© Simon McIntosh-Smith,
Gethin Williams

University of BRISTOL