Common Application Patterns: The Seven Dwarfs

Simon McIntosh-Smith

simonm@cs.bris.ac.uk

The Seven Dwarfs

The Landscape of Parallel Computing Research: A View from Berkeley

Krste Asanovic
Ras Bodik
Bryan Christopher Catanzaro
Joseph James Gebis
Parry Husbands
Kurt Keutzer
David A. Patterson
William Lester Plishker
John Shalf
Samuel Webb Williams
Katherine A. Yelick

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2006-183
http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.html

December 18, 2006

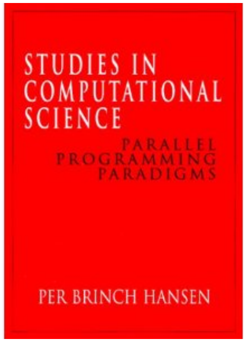
- First described by Phil Colella at LBNL in 2004

- Expanded to 13 dwarfs by a group of researchers at Berkeley in 2006

http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.html

The idea goes back even further

- Per Brinch Hansen
- Danish computer scientist
- Invented much of concurrency theory
- Seminal works:
  - The Architecture of Concurrent Programs (1977)
  - Studies in Computational Science: Parallel Programming Paradigms (1995)



- http://brinch-hansen.net/

Dwarfs – Disney (1937)

Bashful

Sleepy

Happy

Sneezy

Grumpy

Dopey

Doc

University of BRISTOL

Dwarfs – Colella (2004)

Structured

N-Body

Spectral

Unstructured

Sparse LA

Monte-Carlo

Dense LA

University of BRISTOL

What are the Seven Dwarfs?

A useful taxonomy describing key algorithmic kernels found in many scientific applications

1. Dense linear algebra
2. Sparse linear algebra
3. Spectral methods
4. N-body methods
5. Structured grids
6. Unstructured grids
7. Monte Carlo

1. Dense linear algebra

Data in the form of dense matrices or vectors.

BLAS – Basic Linear Algebra Subprograms (www.netlib.org):

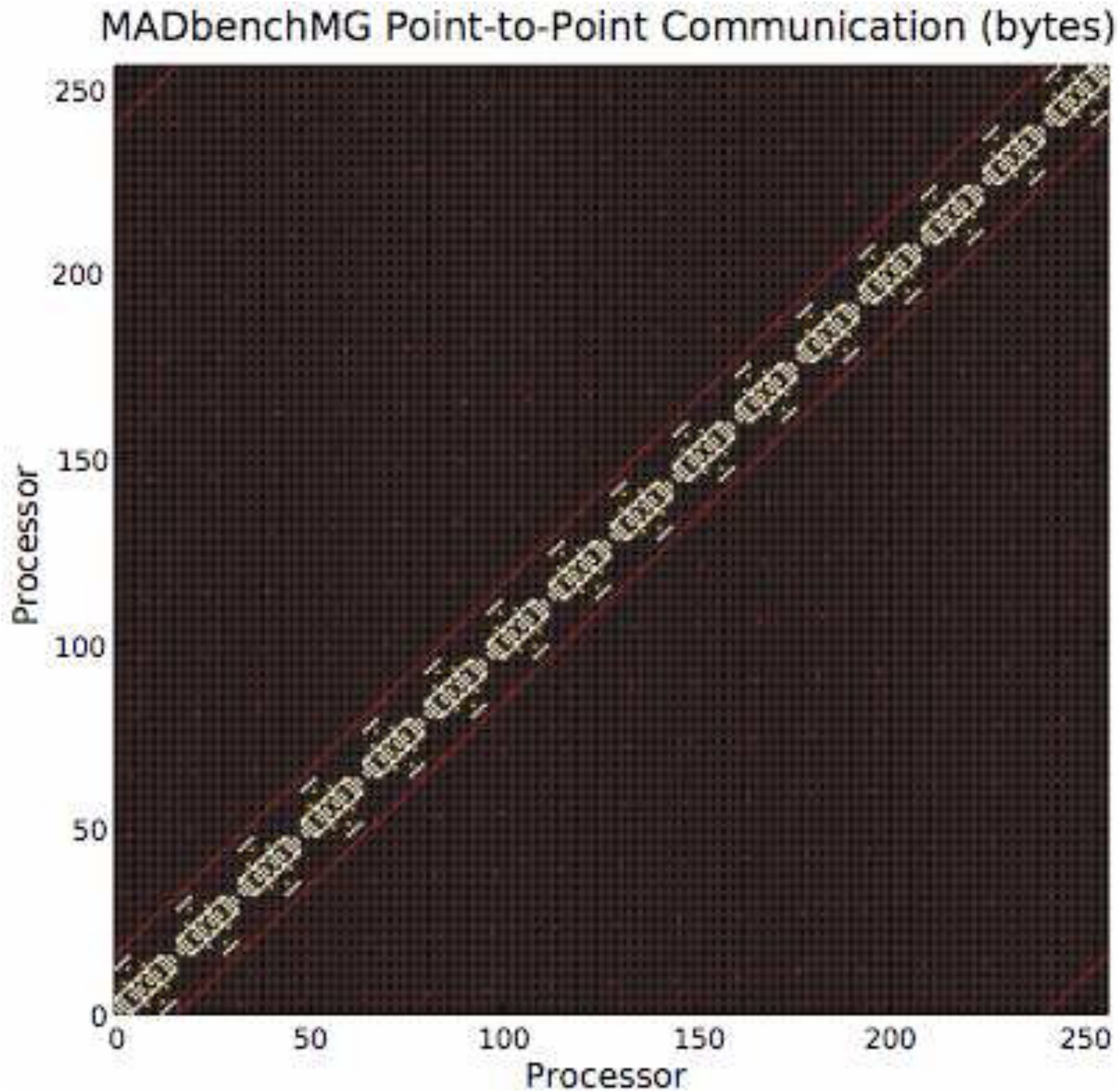
- Level 1 = vector-vector   **O(n)** data, **O(n)** compute
- Level 2 = matrix-vector   **O(n^2)** data, **O(n^2)** compute
- Level 3 = matrix-matrix   **O(n^2)** data, **O(n^3)** compute

Commonly used in solvers for systems of linear equations
- LU, QR, Cholesky et al

Fast implementations exploit ***caches*** via ***blocking.***
- Often achieve >85% of machine peak for L3 BLAS
- Can leverage Gustafson's Law for weak scaling

Examples include MATLAB, ScaLAPACK, **LINPACK Top500**, vendor libs.

Dense linear algebra



University of BRISTOL

2. Sparse linear algebra

Some data sets include many zero values.

- Dimensions may be $O(10^6)$
- Non zero values may be <<1%

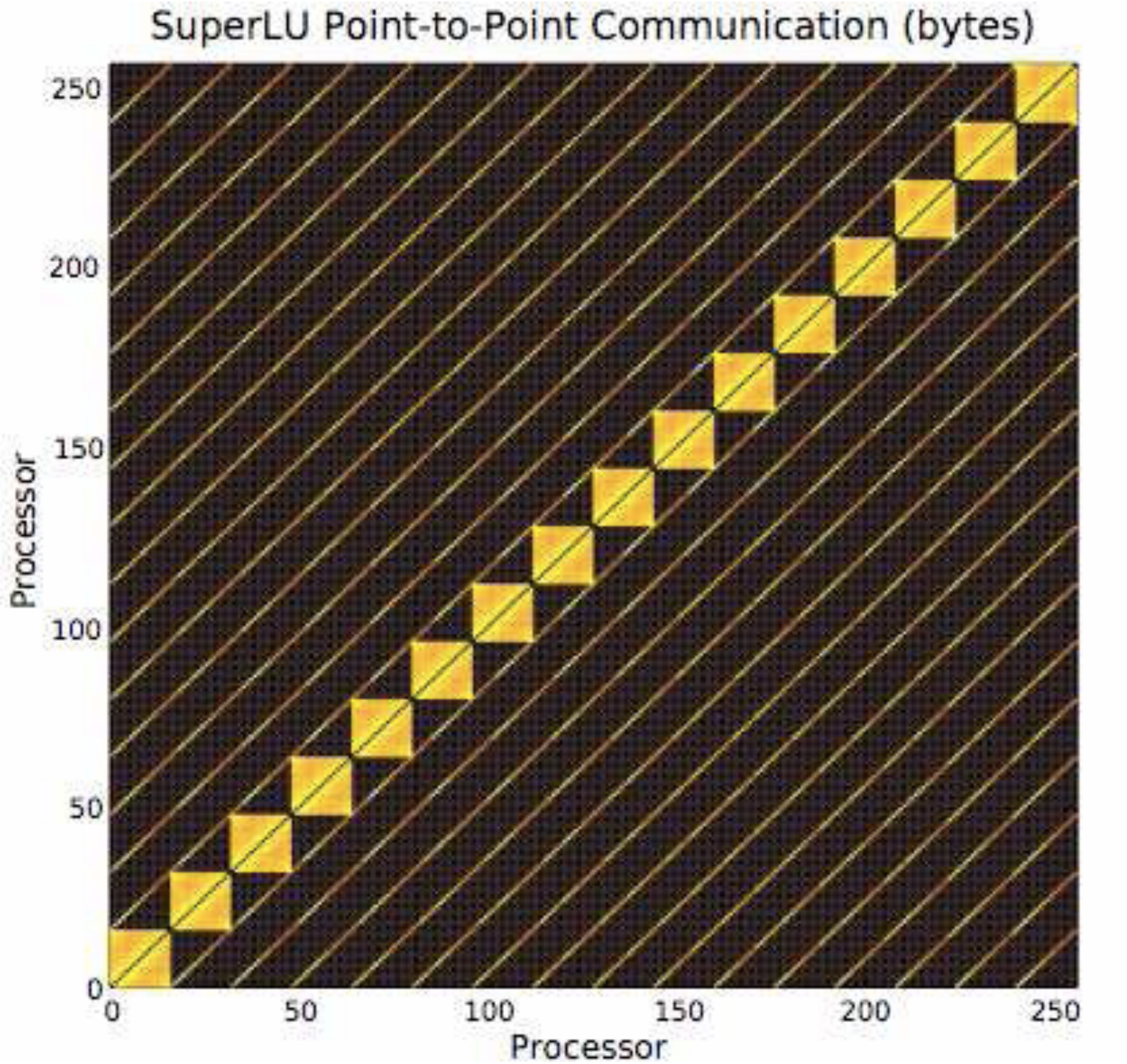
Use compressed data structures to reduce the storage and bandwidth requirements to access the nonzero values.

Indirection via compressed data structures means sparse linear algebra usually achieves a much lower fraction of machine peak performance

- <10% typical

Examples include SpMV, PetSc, SuperLU, many others

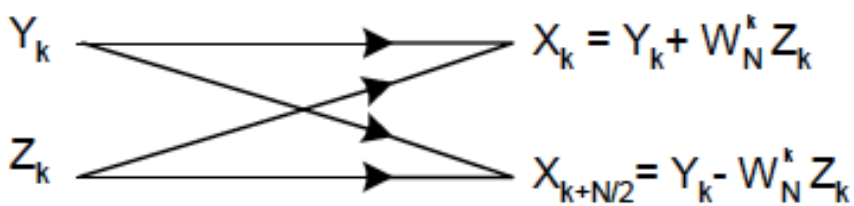
Sparse linear algebra



3. Spectral methods

Data is often from the frequency domain, as opposed to time or spatial domains.

Typically, spectral methods use multiple butterfly stages, which combine multiply-add operations and a specific pattern of data permutation, with all-to-all communication for some stages and strictly local for others.

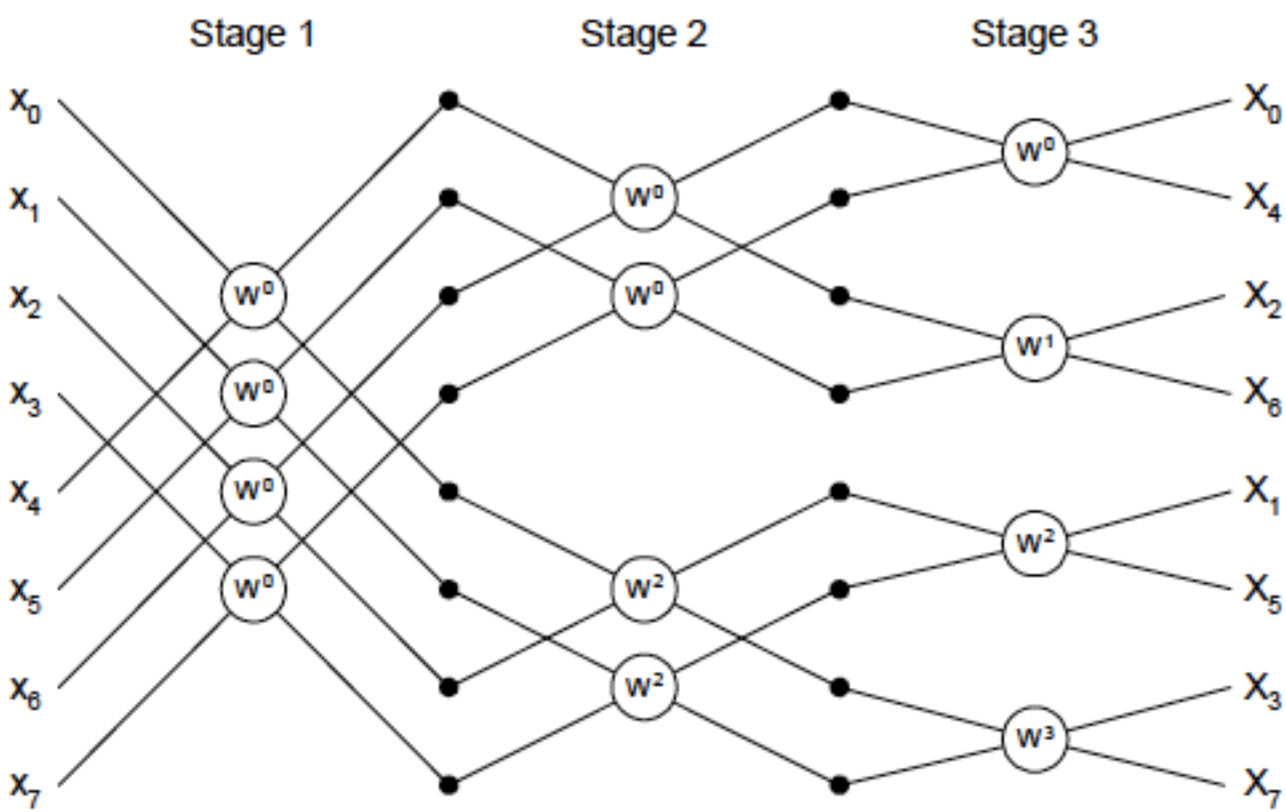


Harder to parallelise: O(n) data, O(n log n) compute

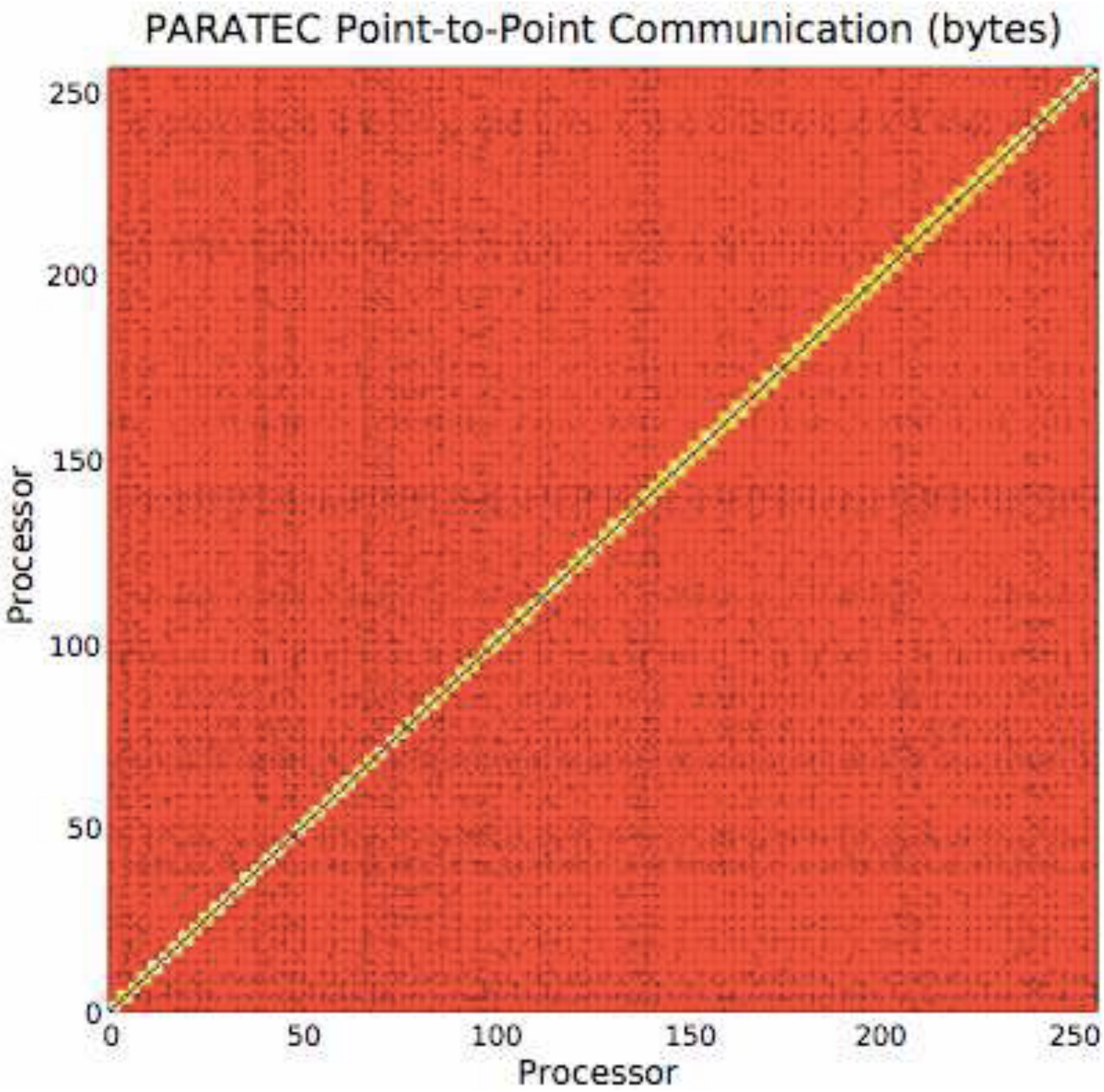
Examples include FFTW, vendor libraries such as Intel's MKL.

University of BRISTOL

Spectral methods



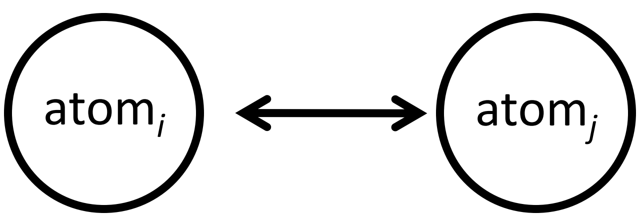
Spectral methods

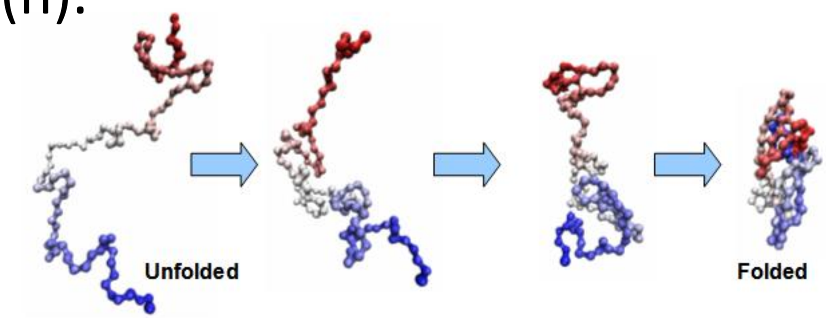


4. N-body methods

Interactions between many discrete points.

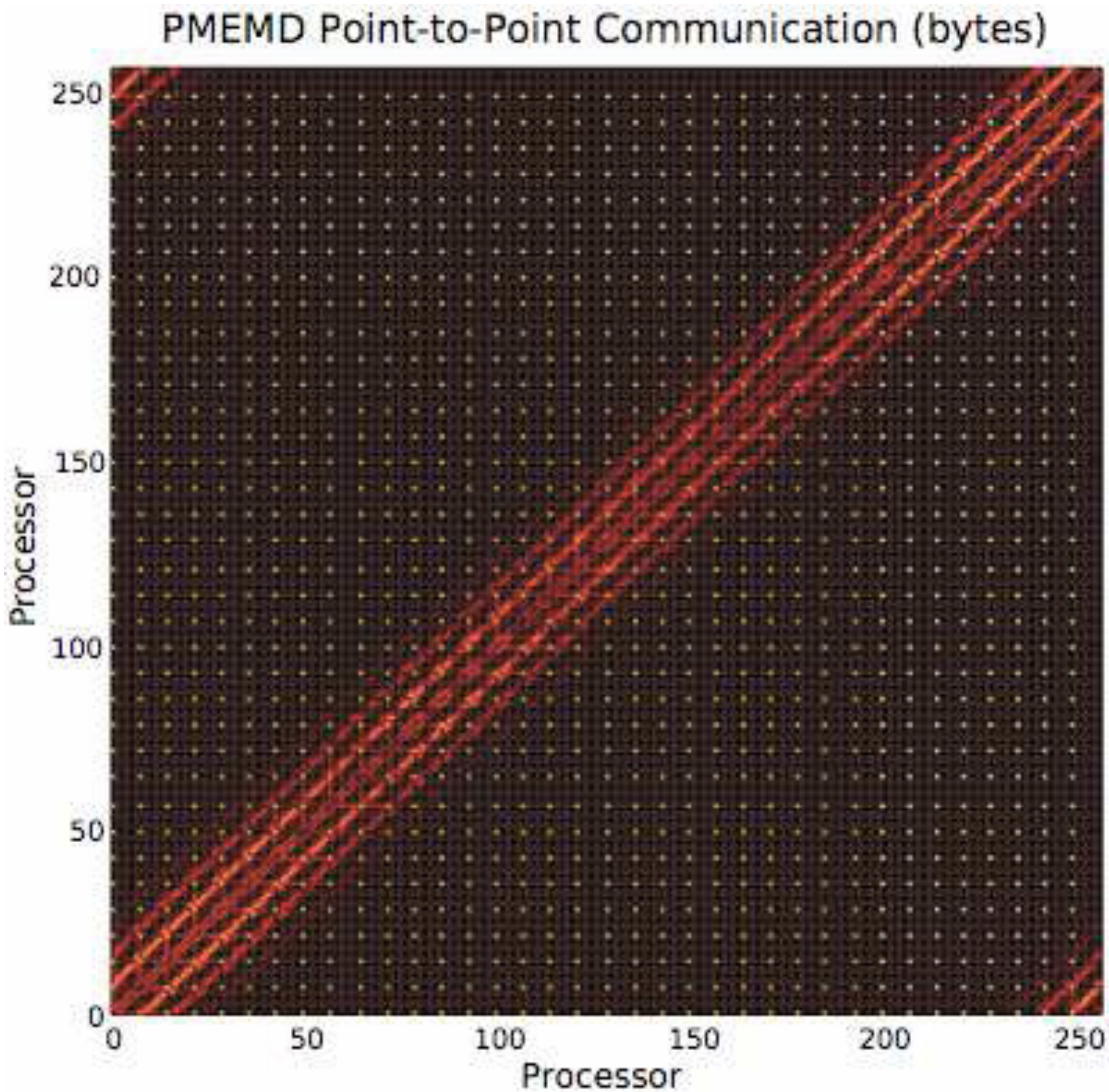
Variations include particle-particle methods, where every point depends on all others, leading to an $O(n^2)$ calculation, and hierarchical particle methods, which combine forces or potentials from multiple points to reduce the computational complexity to $O(n \log n)$ or $O(n)$.





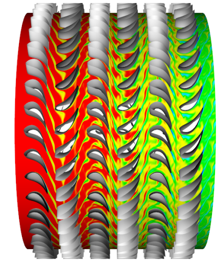
Examples include molecular dynamics, galaxy simulations, particle/smoke for visual effects, …

N-body methods



University of BRISTOL

5. Structured grids

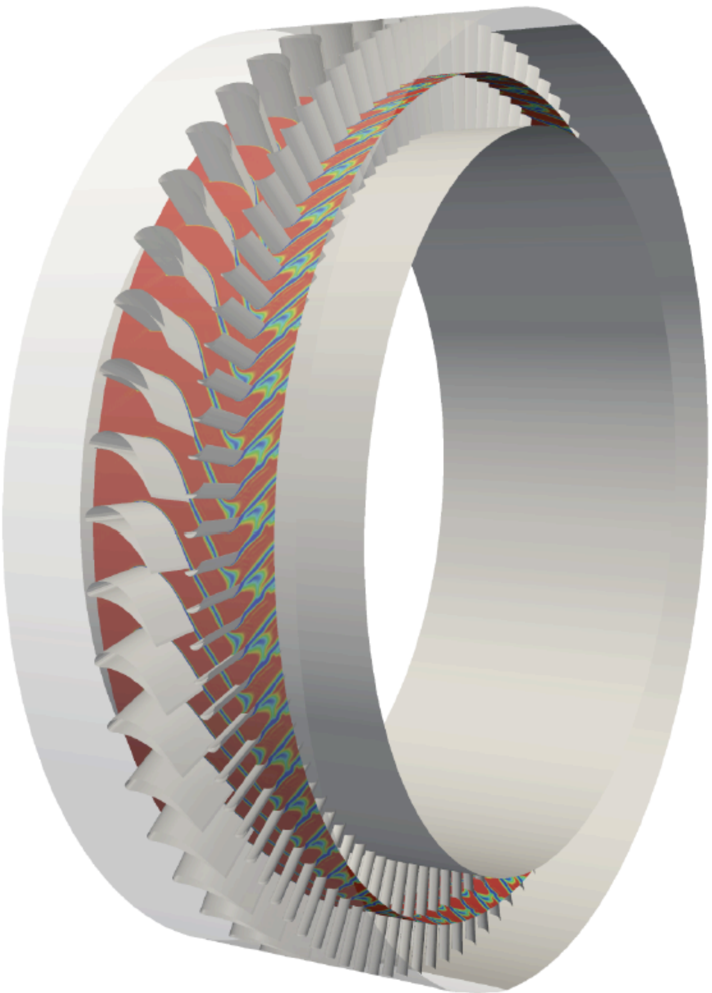


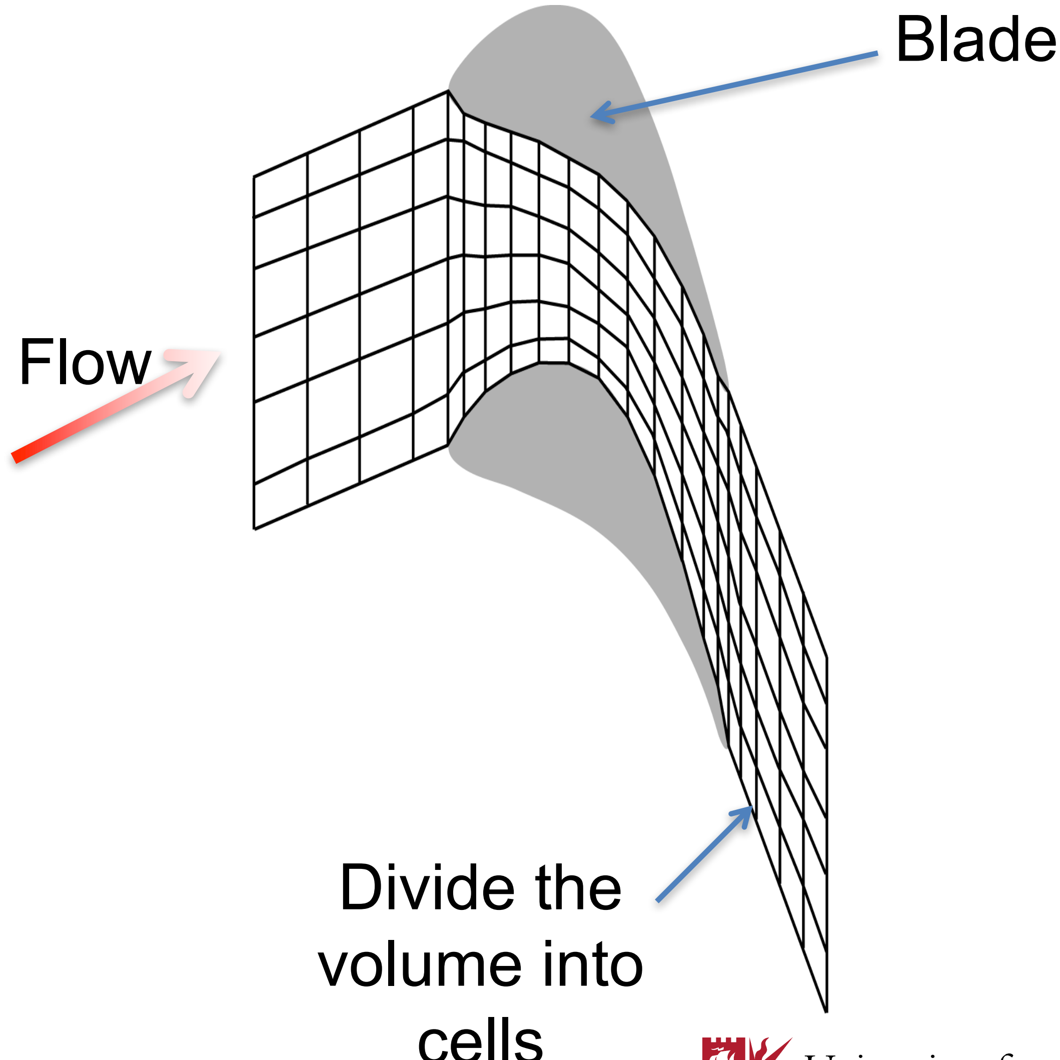
Characterised by a spatial grid discretised in a regular fashion

- Points on grid are conceptually updated together
- High spatial locality
- Updates may be in place or between two versions of the grid
- The grid may be subdivided into finer grids in areas of interest (“Adaptive Mesh Refinement”)
- Often referred to as “Stencil Operations” when updating cells based on values in surrounding cells

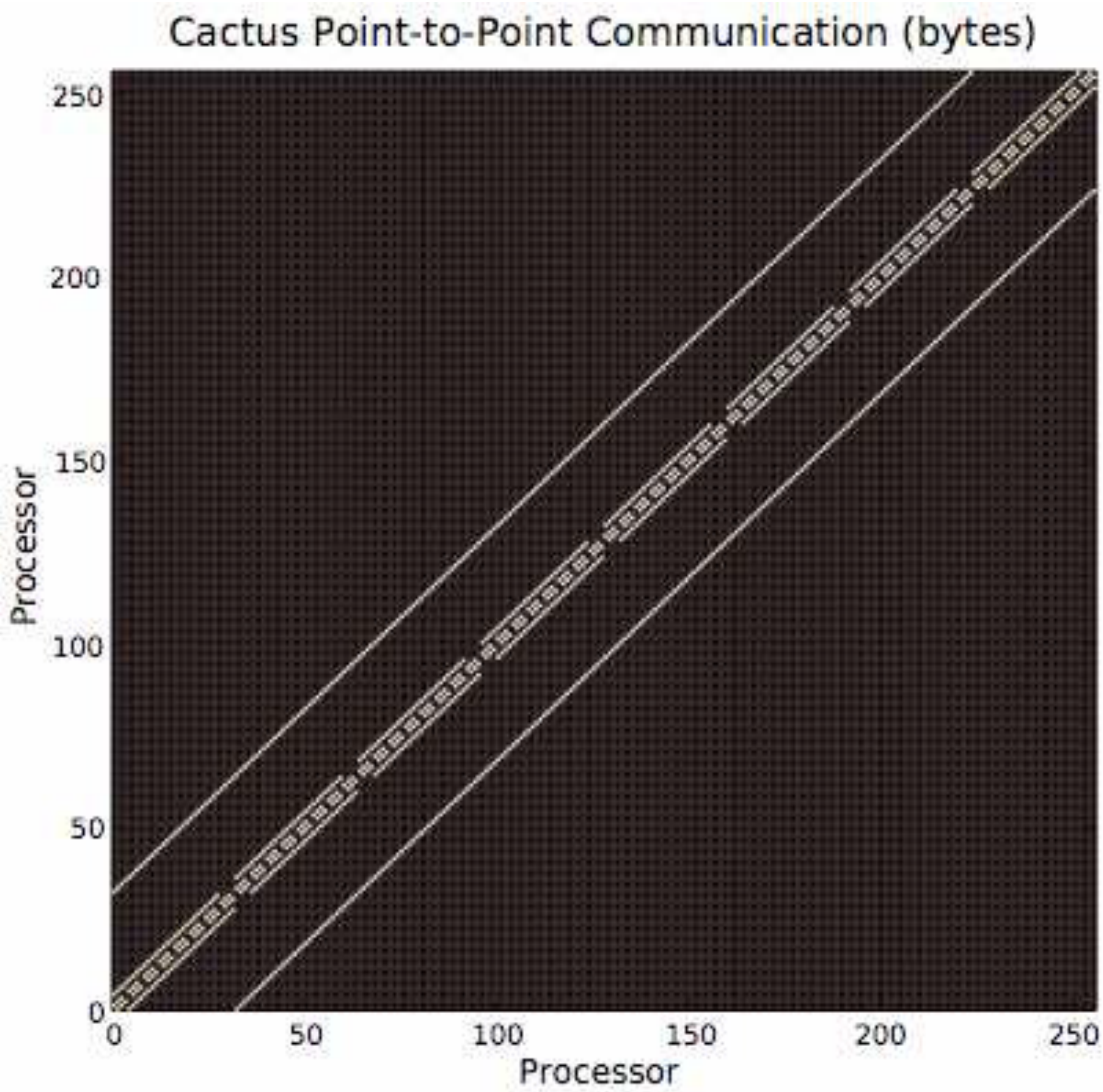
Examples include ***Lattice-Boltzmann***, PDE solvers, …

Structured grids





Structured grids



6. Unstructured grids

Characterised by an *irregular* grid.

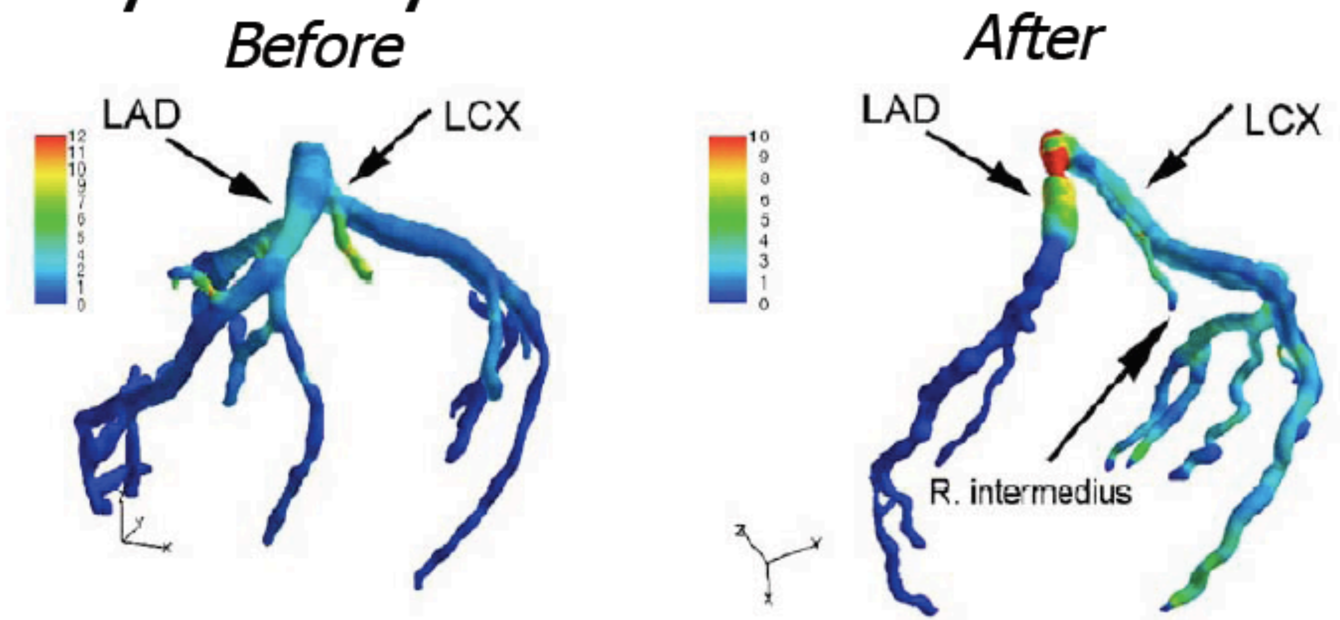
Data point location and connectivity of neighbouring points must be *explicit*.

The points on the grid are conceptually updated together.

Updates typically involve multiple levels of memory reference indirection, as an update to any point requires first determining a list of neighbouring points, and then loading values from those neighbours.

Examples include some CFD, physical structure codes, …

Coronary Artery Disease



- Modeling to help patient compliance?
  - 400k deaths/year, 16M w. symptom, 72M↑BP
- Massively parallel, Real-time variations
  - CFD FE solid (non-linear), fluid (Newtonian), pulsatile
  - Blood pressure, activity, habitus, cholesterol

7. Monte Carlo

A naturally massively parallel method used for numerical approximations to mathematical functions.

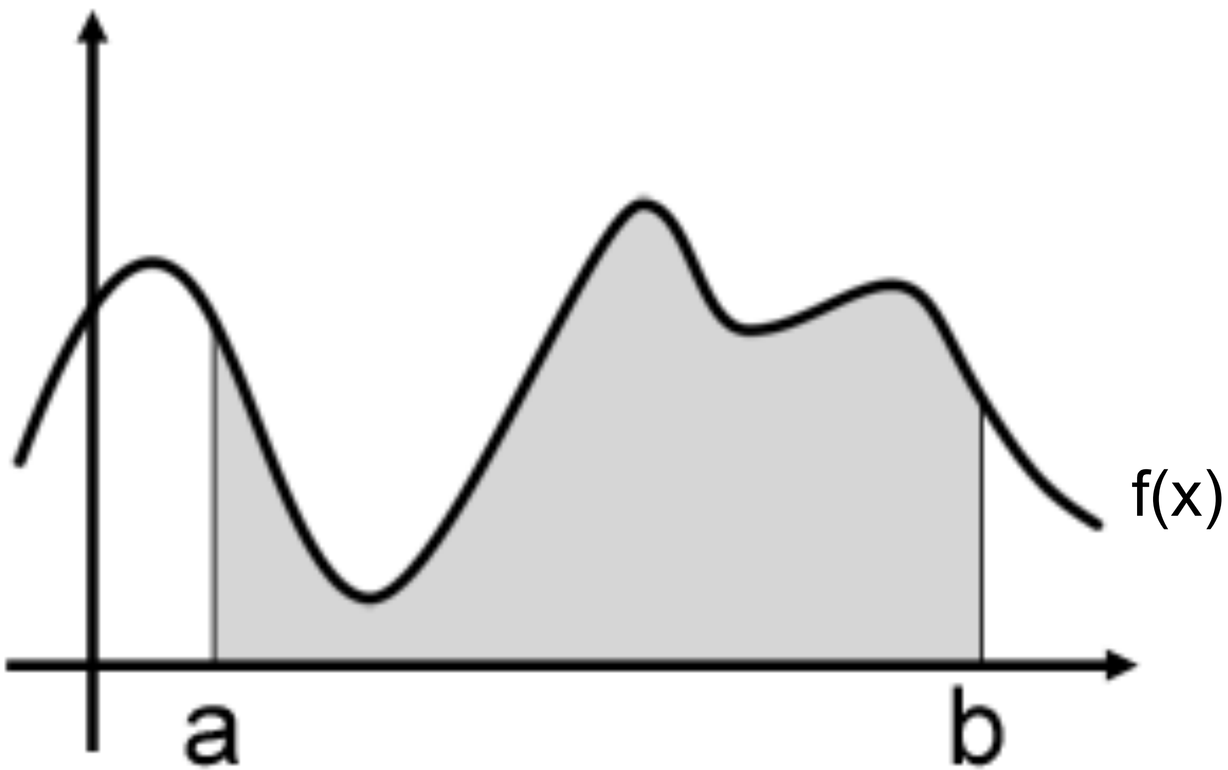
Involves the generation of many independent evaluations of the function under test with differing random input parameters.

Often used for numerical integration of complex functions – calculates the area under the curve.

Monte Carlo methods also have nice fault-tolerance properties.

Used to calculate, for example, the value of options in financial markets, as well as in many numerical methods.

Monte Carlo



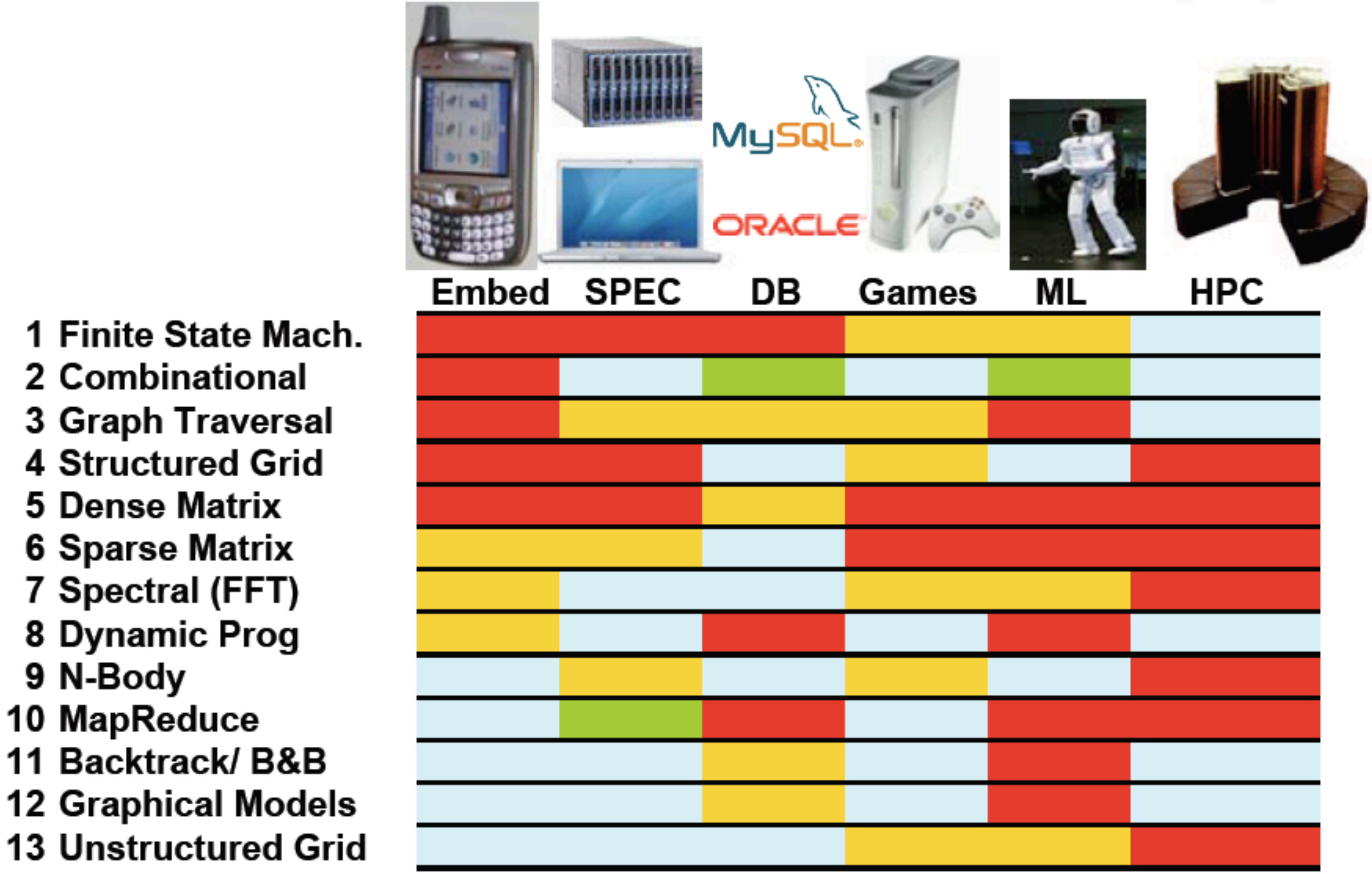
From Seven Dwarfs to Thirteen Motifs

A useful taxonomy describing key algorithmic kernels found in many scientific applications

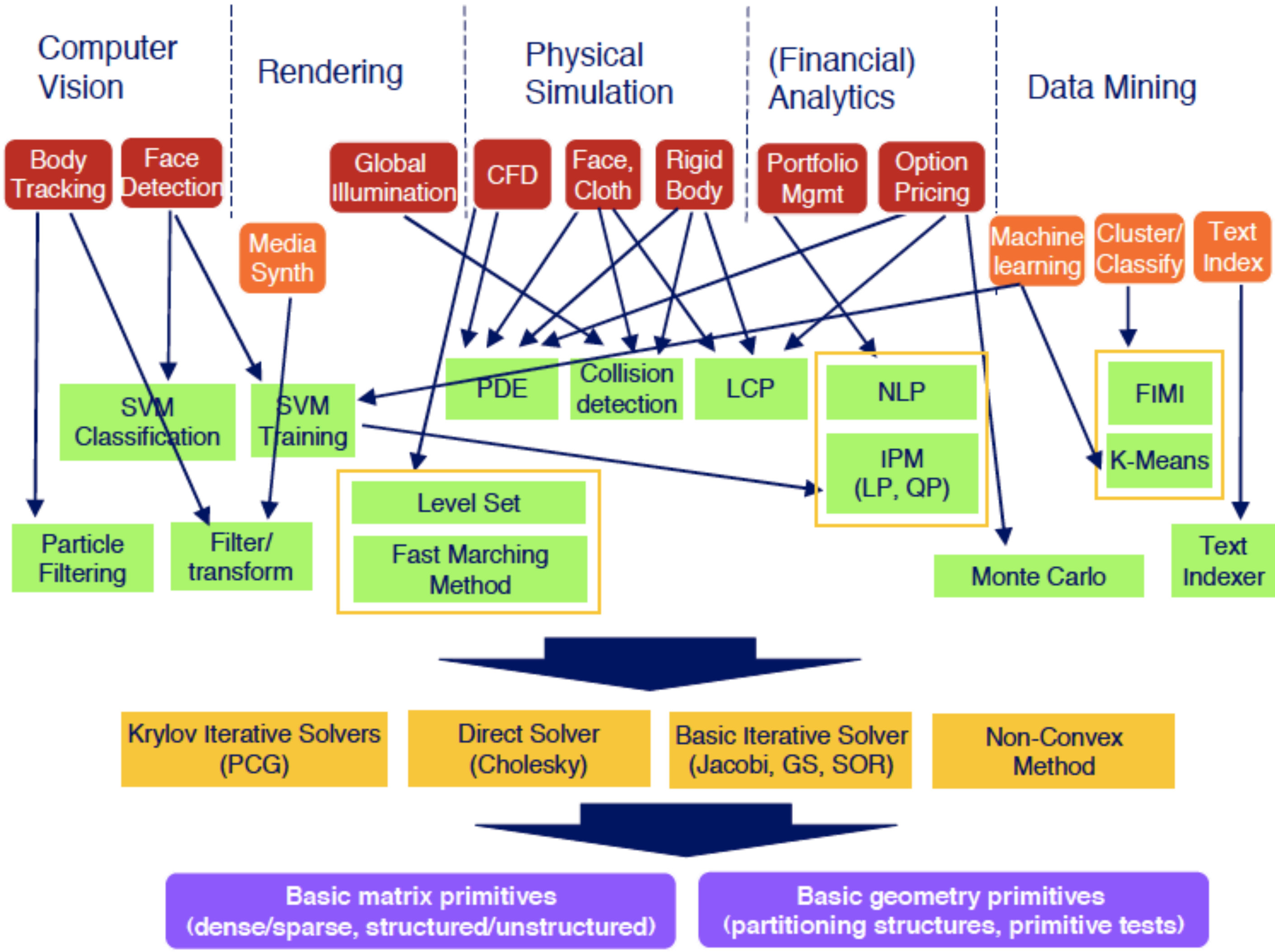
1. Dense linear algebra
2. Sparse linear algebra
3. Spectral methods
4. N-body methods
5. Structured grids
6. Unstructured grids
7. Monte Carlo
8. Combinational Logic
9. Graph traversal
10. Dynamic Programming
11. Backtrack and Branch+Bound
12. Construct Graphical Models
13. Finite State Machine

University of BRISTOL

Dwarf Popularity (Red Hot → Blue Cool)



Alternatives to the dwarfs:
Intel's "Recognition, Mining, and Synthesis"



University of BRISTOL

Conclusions

- There are common forms of algorithm
- There are common forms of parallelism
- Knowing how a dwarf has been solved in the past might help you when you come across one in the future
- Dwarfs tell you things about **relative complexities in**:
  - **Computation**,
  - **Communication** and
  - **Memory** requirement

Dwarfs – Colella (2004)

Structured

N-Body

Spectral

Unstructured

Sparse LA

Monte-Carlo

Dense LA

University of BRISTOL

Further reading

- Reading list available on the website
- READ BOTH SEVEN DWARFS PAPERS!!!
  - Both linked from the bottom of the unit webpage
- P. Colella, "Defining Software Requirements for Scientific Computing," presentation, 2004
- Per Brinch Hansen, "Studies in Computational Science: Parallel Programming Paradigms", 1995, ISBN 0-13-439324-4