Architecture-aware Algorithms and Software for Peta and Exascale Computing

Jack Dongarra

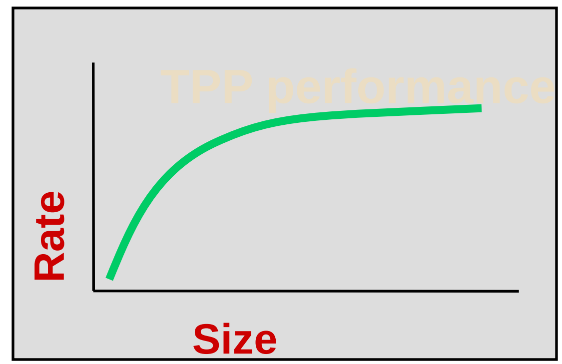
University of Tennessee
Oak Ridge National Laboratory
University of Manchester

TOP500 supercomputer

H. Meuer, H. Simon, E. Strohmaier, & JD

- Listing of the 500 most powerful Computers in the World
- Yardstick: Rmax from LINPACK MPP

  $Ax=b$, *dense problem*



- Updated twice a year
  SC'xy in the States in November
  Meeting in Germany in June
- All data available from **www.top500.org**

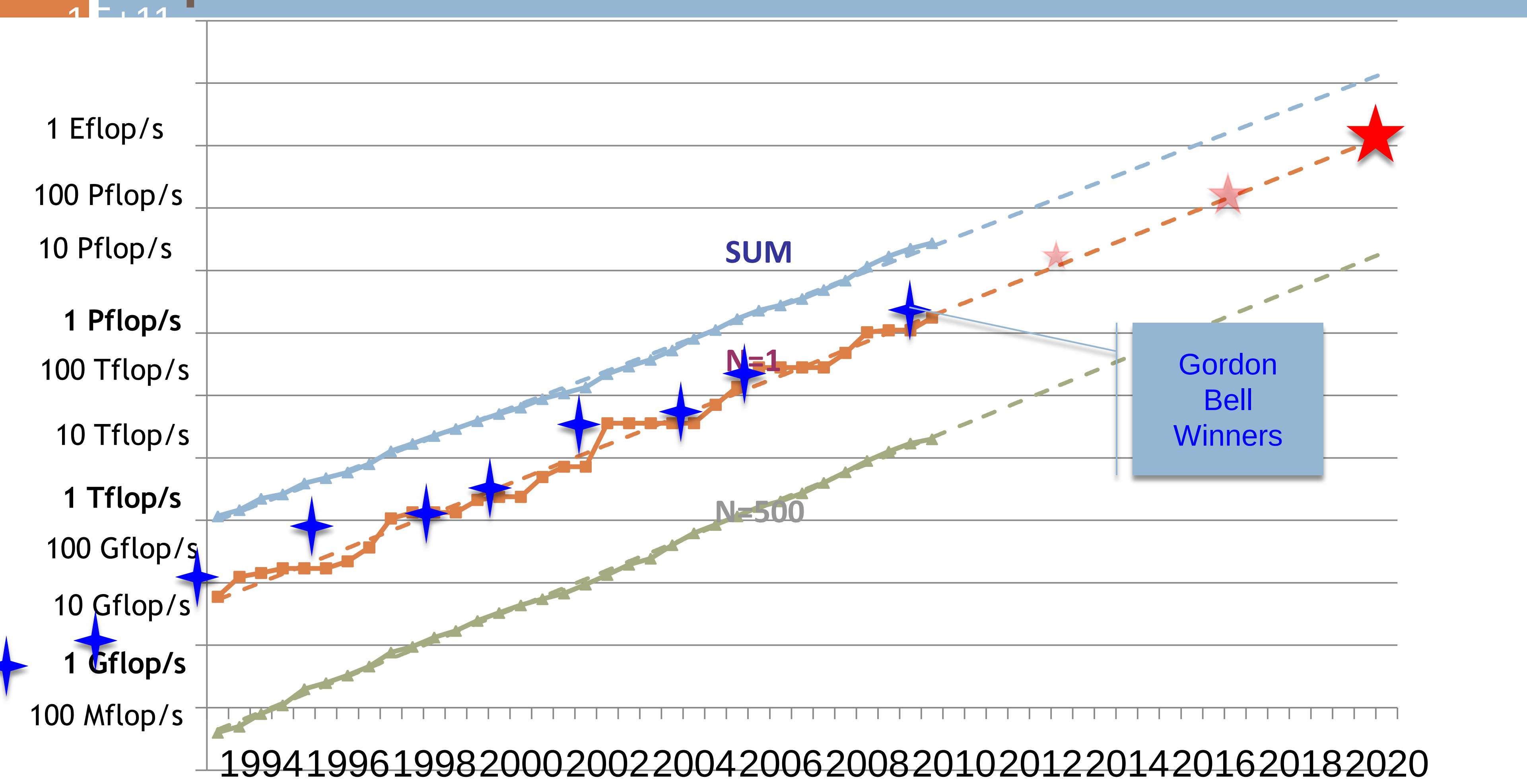
36rd List: The TOP10

Rank	Site	Computer	Country	Cores	Rmax [Pflops]	% of Peak
1	Nat. SuperComputer Center in Tianjin	NUDT YH Cluster, X5670 2.93Ghz 6C, NVIDIA GPU	China	186,368	2.57	55
2	DOE / OS Oak Ridge Nat Lab	Jaguar / Cray Cray XT5 sixCore 2.6 GHz	USA	224,162	1.76	75
3	Nat. Supercomputer Center in Shenzhen	Nebulea / Dawning / TC3600 Blade, Intel X5650, Nvidia C2050 GPU	China	120,640	1.27	43
4	GSIC Center, Tokyo Institute of Technology	Tusbame 2.0 HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU	Japan	73,278	1.19	52
5	DOE/SC/LBNL/NERSC	Hopper, Cray XE6 12-core 2.1 GHz	USA	153,408	1.054	82
6	Commissariat a l'Energie Atomique (CEA)	Tera-100 Bull bullx super-node S6010/S6030	France	138,368	1.050	84
7	DOE / NNSA Los Alamos Nat Lab	Roadrunner / IBM BladeCenter QS22/LS21	USA	122,400	1.04	76
8	NSF / NICS / U of Tennessee	kyaken/ Cray Cray XT5 sixCore 2.6 GHz	USA	98,928	.831	81
9	Forschungszentrum Juelich (FZJ)	Jugene / IBM Blue Gene/P Solution	Germany	294,912	.825	82
10	DOE/ NNSA / Los Alamos Nat Lab	Cray XE6 8-core 2.4 GHz	USA	107,152	.817	79

36rd List: The TOP10

Rank	Site	Computer	Country	Cores	Rmax [Pflops]	% of Peak	Power [MW]	Flops/ Watt
1	Nat. SuperComputer Center in Tianjin	NUDT YH Cluster, X5670 2.93Ghz 6C, NVIDIA GPU	China	186,368	2.57	55	4.04	636
2	DOE / OS Oak Ridge Nat Lab	Jaguar / Cray Cray XT5 sixCore 2.6 GHz	USA	224,162	1.76	75	7.0	251
3	Nat. Supercomputer Center in Shenzhen	Nebulea / Dawning / TC3600 Blade, Intel X5650, Nvidia C2050 GPU	China	120,640	1.27	43	2.58	493
4	GSIC Center, Tokyo Institute of Technology	Tusbame 2.0 HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU	Japan	73,278	1.19	52	1.40	850
5	DOE/SC/LBNL/NERSC	Hopper, Cray XE6 12-core 2.1 GHz	USA	153,408	1.054	82	2.91	362
6	Commissariat a l'Energie Atomique (CEA)	Tera-100 Bull bullx super-node S6010/S6030	France	138,368	1.050	84	4.59	229
7	DOE / NNSA Los Alamos Nat Lab	Roadrunner / IBM BladeCenter QS22/LS21	USA	122,400	1.04	76	2.35	446
8	NSF / NICS / U of Tennessee	kraken/ Cray Cray XT5 sixCore 2.6 GHz	USA	98,928	.831	81	3.09	269
9	Forschungszentrum Juelich (FZJ)	Jugene / IBM Blue Gene/P Solution	Germany	294,912	.825	82	2.26	365
10	DOE/ NNSA / Los Alamos Nat Lab	Cray XE6 8-core 2.4 GHz	USA	107,152	.817	79	2.95	277

Performance Development in Top500

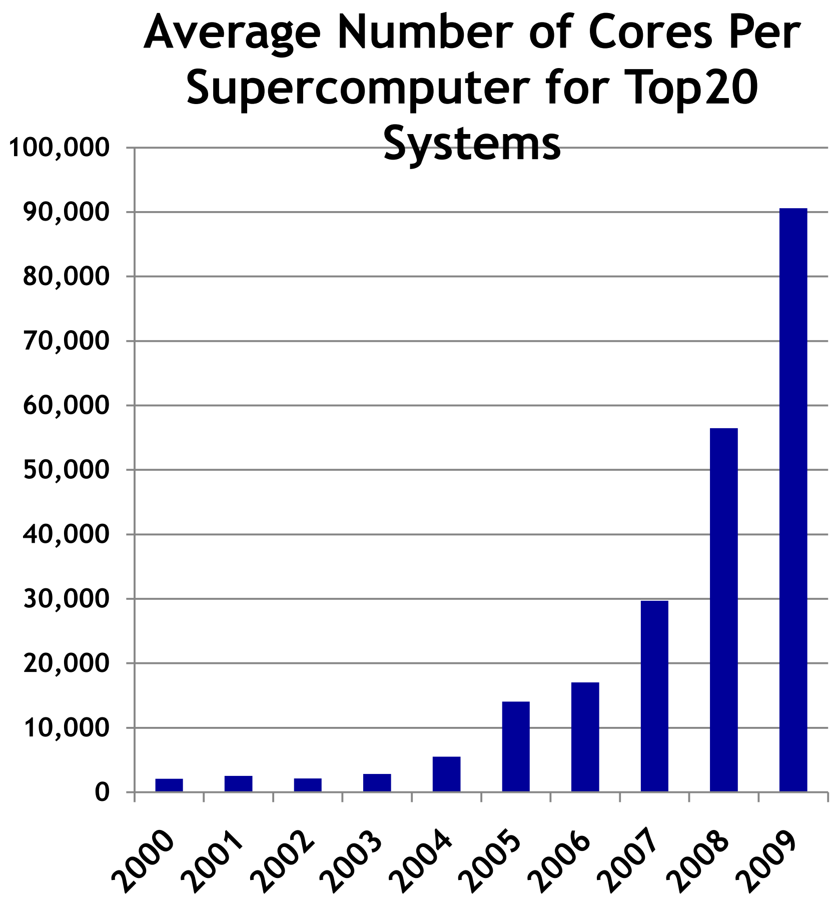


Pflop/s Club (11 systems; Peak)

Name	Peak Pflop/s	"Linpack" Pflop/s	Country	
Tianhe-1A	4.70	2.57	China	NUDT: Hybrid Intel/Nvidia/Self
Nebula	2.98	1.27	China	Dawning: Hybrid Intel/Nvidia/IB
Jaguar	2.33	1.76	US	Cray: AMD/Self
Tsubame 2.0	2.29	1.19	Japan	HP: Hybrid Intel/Nvidia/IB
RoadRunner	1.38	1.04	US	IBM: Hybrid AMD/Cell/IB
Hopper	1.29	1.054	US	Cray: AMD/Self
Tera-100	1.25	1.050	France	Bull: Intel/IB
Mole-8.5	1.14	.207	China	CAS: Hybrid Intel/Nvidia/IB
Kraken	1.02	.831	US	Cray: AMD/Self
Cielo	1.02	.817	US	Cray: AMD/Self
JuGene	1.00	.825	Germany	IBM: BG-P/Self

Factors that Necessitate Redesign of Our Software

- Steepness of the ascent from terascale to petascale to exascale
- Extreme parallelism and hybrid design
  - Preparing for million/billion way parallelism
- Tightening memory/bandwidth bottleneck
  - Limits on power/clock speed implication on multicore
  - Reducing communication will become much more intense
  - Memory per core changes, byte-to-flop ratio will change
- Necessary Fault Tolerance
  - MTTF will drop
  - Checkpoint/restart has limitations



Software infrastructure does not exist today

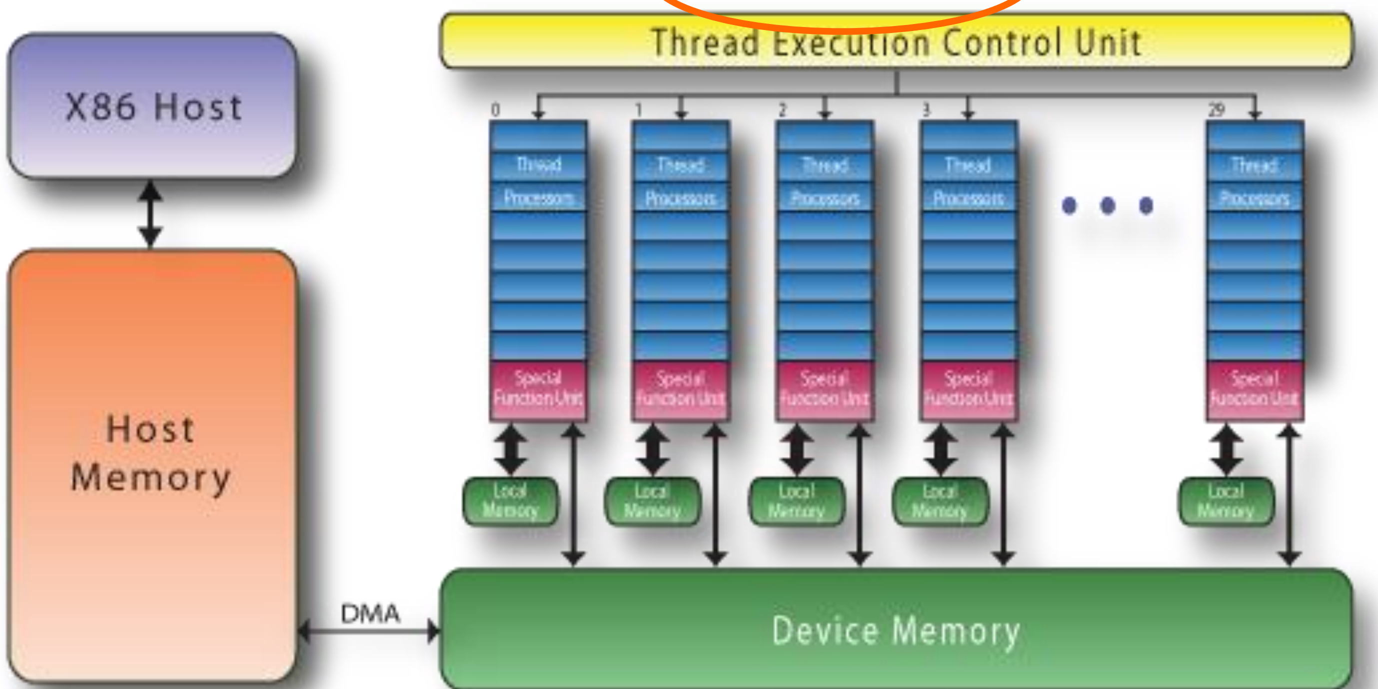
Commodity plus Accelerators

Commodity

Intel Xeon
8 cores
3 GHz
8*4 ops/cycle
96 Gflop/s (DP)

Accelerator (GPU)

Nvidia C2050 “Fermi”
448 “Cuda cores”
1.15 GHz
448 ops/cycle
515 Gflop/s (DP)



Interconnect
PCI Express
512 MB/s to 32GB/s
8 MW – 512 MW

07

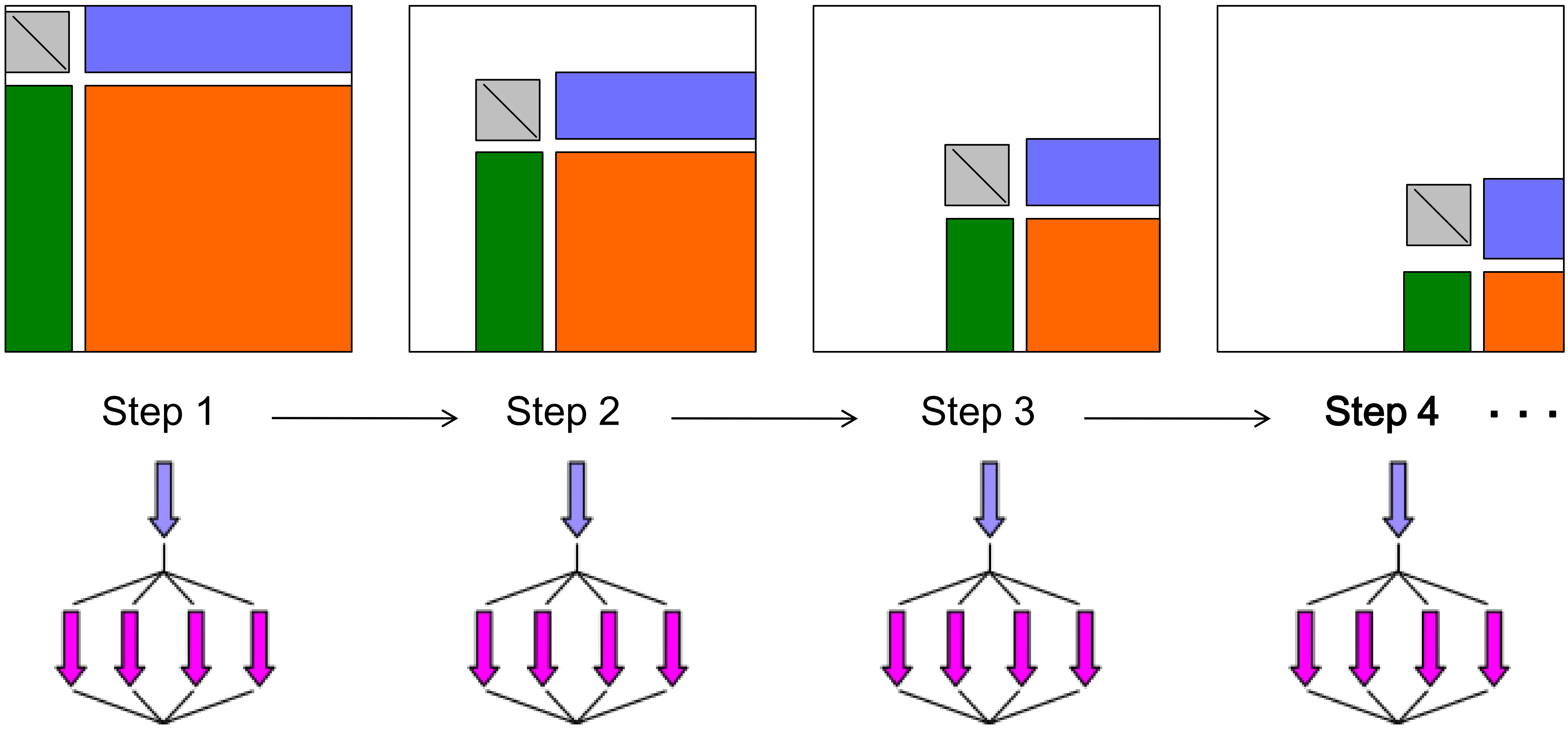
Major Changes to Software

- Must rethink the design of our software
  - Another disruptive technology
    - Similar to what happened with cluster computing and message passing
  - Rethink and rewrite the applications, algorithms, and software
- Numerical libraries for example will change
  - For example, both LAPACK and ScaLAPACK will undergo major changes to accommodate this

Five Important Software Features to Consider When Computing at Scale

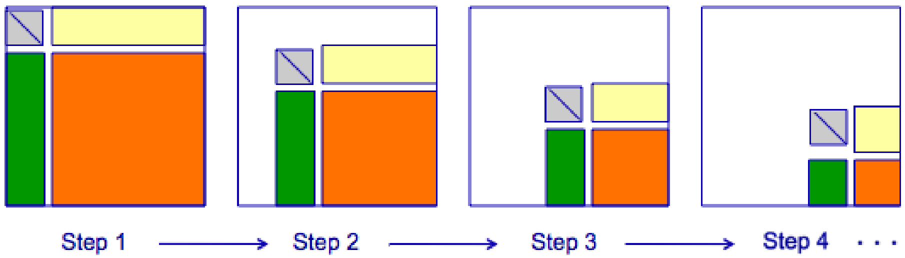
1. Effective Use of Many-Core and Hybrid architectures
   - Break fork-join parallelism
   - Dynamic Data Driven Execution
   - Block Data Layout
2. Exploiting Mixed Precision in the Algorithms
   - Single Precision is 2X faster than Double Precision
   - With GP-GPUs 10x
   - Power saving issues
3. Self Adapting / Auto Tuning of Software
   - Too hard to do by hand
4. Fault Tolerant Algorithms
   - With 1,000,000's of cores things will fail
5. Communication Reducing Algorithms
   - For dense computations from O(n *log* p) to O(*log* p) communications
   - Asynchronous iterations
   - GMRES k-step compute (x, Ax, A^2x, … A^kx)

LAPACK LU/LL^T/QR

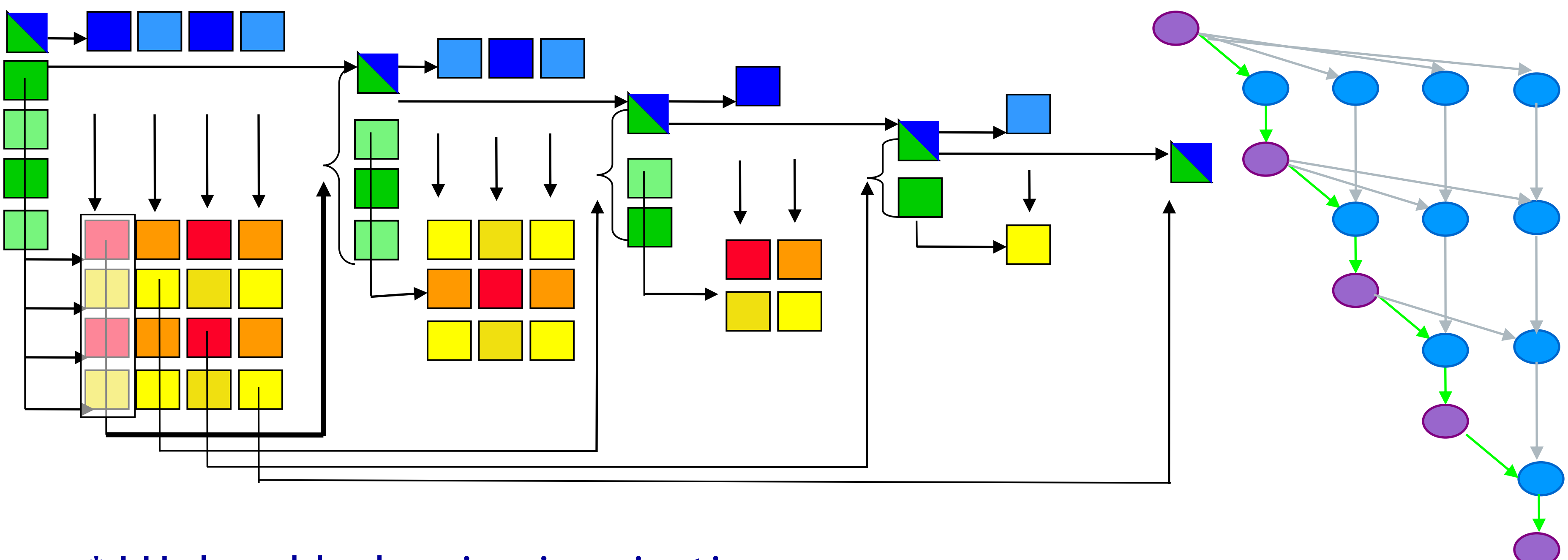


- **Fork-join, bulk synchronous processing**

Parallel Tasks in LU/LL^T/QR



- Break into smaller tasks and remove dependencies

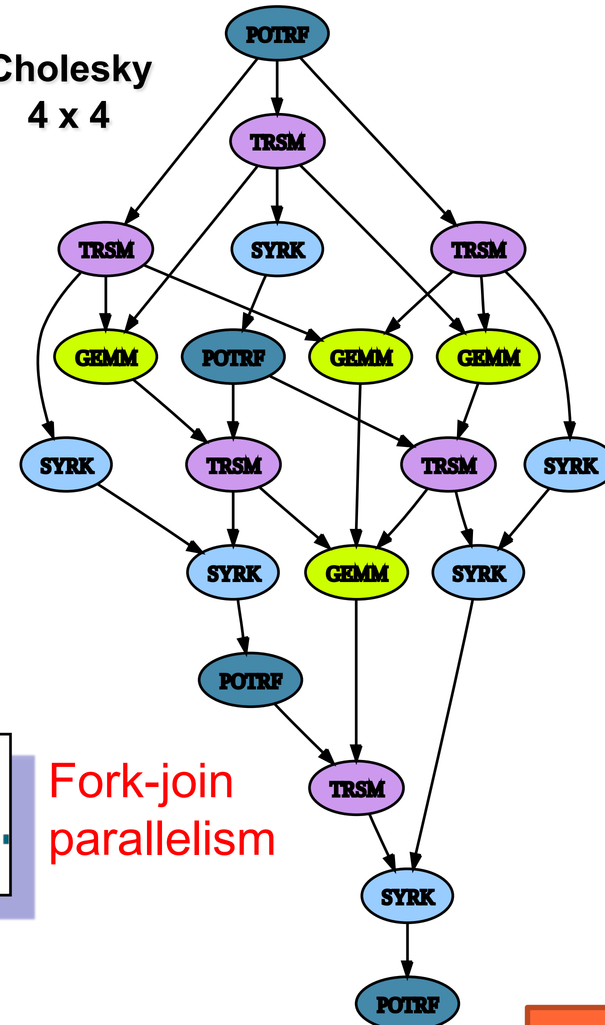


* LU does block pair wise pivoting

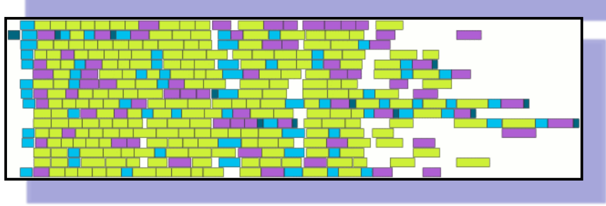
PLASMA: Parallel Linear Algebra s/w for Multicore Architectures

- **Objectives**
  - High utilization of each core
  - Scaling to large number of cores
  - Shared or distributed memory
- **Methodology**
  - Dynamic DAG scheduling
  - Explicit parallelism
  - Implicit communication
  - Fine granularity / block data layout
- **Arbitrary DAG with dynamic scheduling**

Cholesky
4 x 4



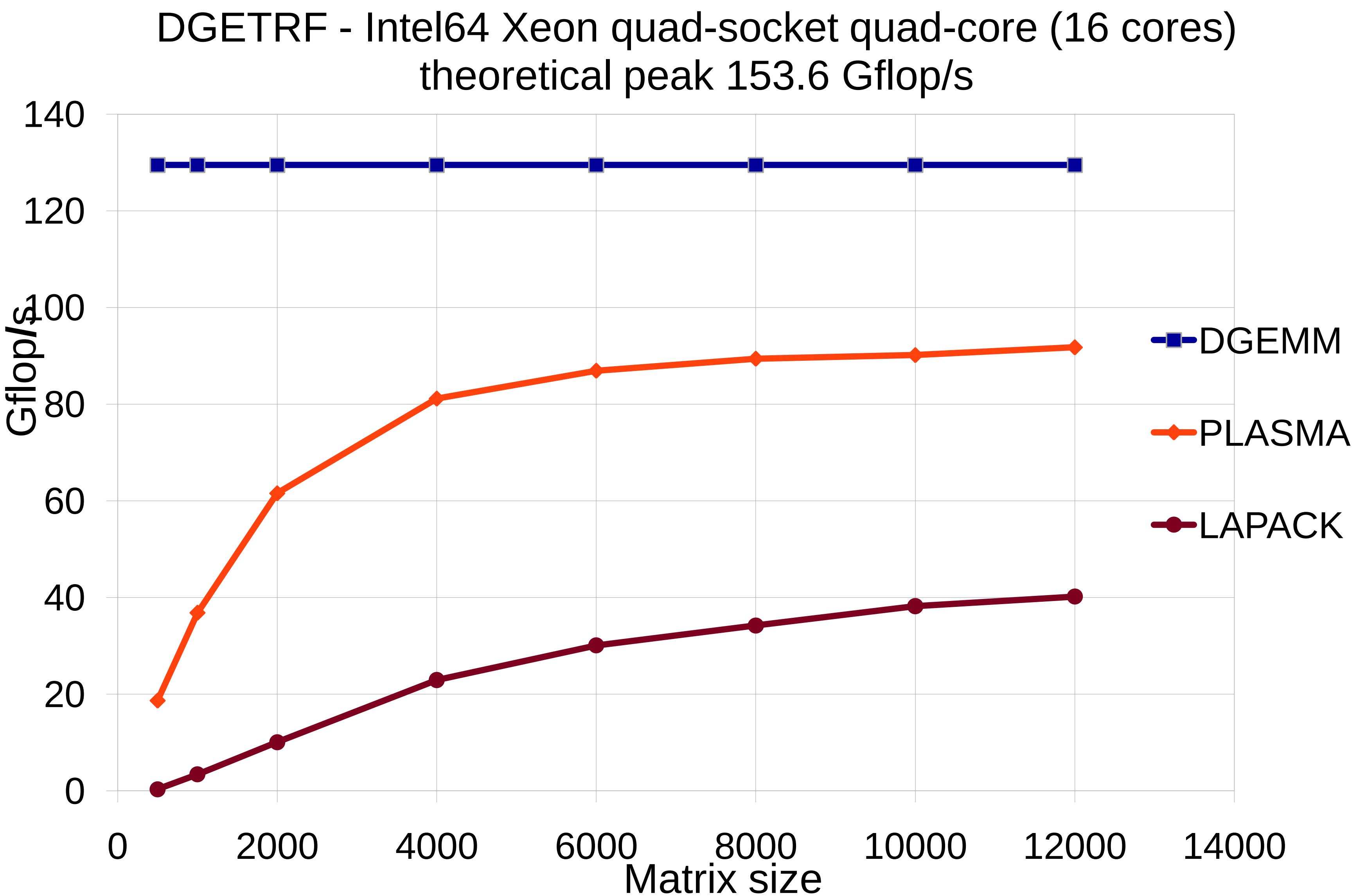
Fork-join parallelism



DAG scheduled parallelism

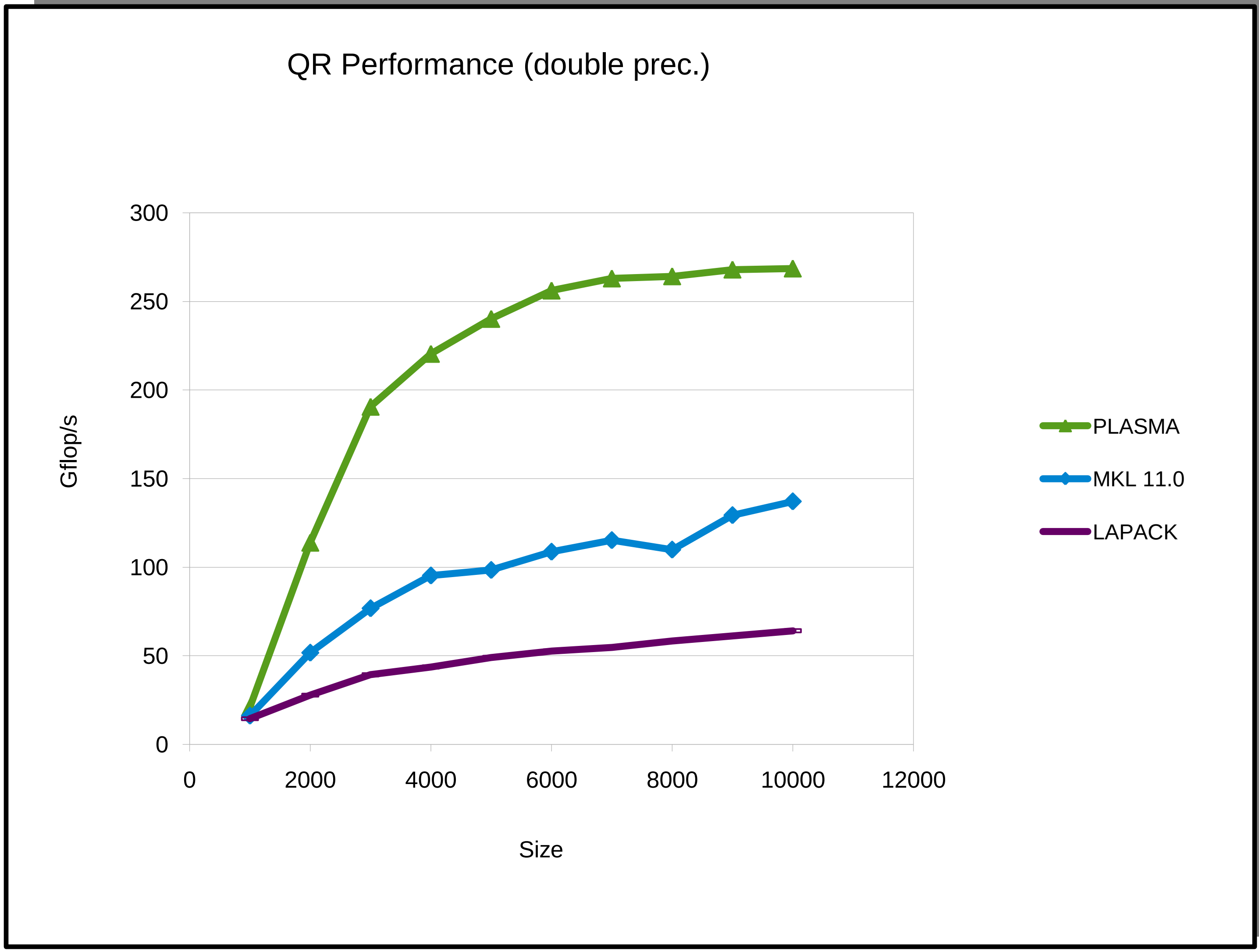
Time

LU - Intel64 - 16 cores



PLASMA Performance (QR, 48 cores)

ISTANBUL AMD 8 socket 6 core (48 cores) @2.8GHz



Challenges of using GPUs

- **High levels of parallelism**
  Many GPU cores
  [e.g. Tesla C2050 (Fermi) has 448 CUDA cores]

- **Hybrid/heterogeneous architectures**
  Match algorithmic requirements to architectural strengths
  [e.g. small, non-parallelizable tasks to run on CPU, large and parallelizable on GPU]

- **Compute *vs* communication gap**
  Exponentially growing gap; persistent challenge
  [Processor speed improves 59%, memory bandwidth 23%, latency 5.5%]
  [on all levels, e.g. a GPU Tesla C1070 (4 x C1060) has compute power of O(1,000) Gflop/s but GPUs communicate through the CPU using O(1) GB/s connection]

Matrix Algebra on GPU and Multicore Architectures

- **MAGMA**: **a new generation linear algebra (LA) libraries** to achieve the fastest possible time to an accurate solution **on hybrid/heterogeneous architectures**, starting with current multicore+MultiGPU systems
  Homepage: http://icl.cs.utk.edu/magma/

- **MAGMA & LAPACK**
  - **MAGMA -** based on LAPACK and extended for hybrid systems (multi-GPUs + multicore systems);
  - **MAGMA -** designed to be similar to LAPACK in functionality, data storage and interface, in order to allow scientists to effortlessly port any of their LAPACK-relying software components to take advantage of the new architectures
  - **MAGMA -** to leverage years of experience in developing open source LA software packages and systems like LAPACK, ScaLAPACK, BLAS, ATLAS as well as the newest LA developments (e.g. communication avoiding algorithms) and experiences on homogeneous multicores (e.g. PLASMA)

- **Support**
  - NSF, Microsoft, NVIDIA [**CUDA Center of Excellence at UTK** on the development of **Linear Algebra Libraries for CUDA-based Hybrid Architectures**]

- **MAGMA developers**
  - University of Tennessee, **Knoxville**; University of California, **Berkeley**; University of Colorado, **Denver**

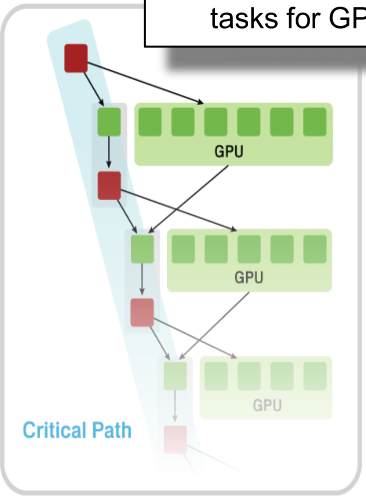
Hybridization methodology

- **MAGMA uses HYBRIDIZATION methodology based on**
  - Representing linear algebra algorithms as collections of TASKS and DATA DEPENDANCIES among them
  - Properly SCHEDULING the tasks' execution over the multicore and the GPU hardware components

- **Successfully applied to fundamental linear algebra algorithms**
  - One and two-sided factorizations and solvers
  - Iterative linear and eigen-solvers

- **Faster, cheaper, better ?**
  - High-level
  - Leveraging prior developments
  - Exceeding in performance homogeneous solutions

Hybrid CPU+GPU algorithms
(small tasks for multicores and large tasks for GPUs)



Linear solvers on Fermi

MAGMA LU-based solvers on Fermi (C2050)

FERMI Tesla C2050: 448 CUDA cores @ 1.15GHz
SP/DP peak is 1030 / 515 GFlop/s

- **Direct solvers**
  - Factor and solve in working precision
- **Mixed Precision Iterative Refinement**
  - Factor in single (i.e. the bulk of the computation in fast arithmetic) and use it as preconditioner in simple double precision iteration, e.g.
    $x_{i+1} = x_i + (LU_{SP})^{-1} P (b - A x_i)$

- Similar results for Cholesky & QR

New Release for SC2010 PLASMA 2.3

Functionality	Coverage
Linear systems and least squares	LU, Cholesky, QR & LQ
Mixed-precision linear systems	LU, Cholesky, QR
Tall and skinny factorization	QR
Generation of the Q matrix	QR, LQ, tall and skinny QR
Explicit matrix inversion	Cholesky
Level 3 BLAS	GEMM, HEMM, HER2K, HERK, SYMM, SYR2K, SYRK, TRMM, TRSM (complete set)
In-place layout translations	CM, RM, CCRB, CRRB, RCRB, RRRB (all combinations)

Features
Covering four precisions: Z, C, D, S (and mixed-precision: ZC, DS)
Static scheduling and dynamic scheduling with QUARK
Support for Linux, MS Windows, Mac OS and AIX

New Release for SC2010 MAGMA 1.0

Functionality	Coverage
Linear systems and least squares	LU, Cholesky, QR & LQ
Mixed-precision linear systems	LU, Cholesky, QR
Eigenvalue and singular value problems	Reductions to upper Hessenberg, bidiagonal, and tridiagonal forms
Generation of the Q matrix	QR, LQ, Hessenberg, bidiabonalization, and tridiagonalization
MAGMA BLAS	Subset of BLAS, critical for MAGMA performance for Tesla and Fermi

Features
Covering four precisions: Z, C, D, S (and mixed-precision: ZC, DS)
Support for multicore and one NVIDIA GPU
CPU and GPU interfaces
Support for Linux and Mac OS

Summary

- Major Challenges are ahead for extreme computing
  - Parallelism
  - Hybrid
  - Fault Tolerance
  - Power
  - … and many others not discussed here
- We will need completely new approaches and technologies to reach the Exascale level
- This opens up many new opportunities for applied mathematicians and computer scientists

Collaborators / Support

- MAGMA [Matrix Algebra on GPU and Multicore Architectures] team
  http://icl.cs.utk.edu/magma/

- PLASMA [Parallel Linear Algebra for Scalable Multicore Architectures] team
  http://icl.cs.utk.edu/plasma

- Collaborating partners

University of Tennessee, Knoxville
University of California, Berkeley
University of Colorado, Denver

University of Coimbra, Portugal
INRIA, France (StarPU team)