23rd International Symposium on Computer Architecture
and High Performance Computing - SBAC-PAD'2011
October 26-29, 2011
Vitória, Espírito Santo, Brazil

Architecture-aware Algorithms and Software for Peta and Exascale Computing

Jack Dongarra

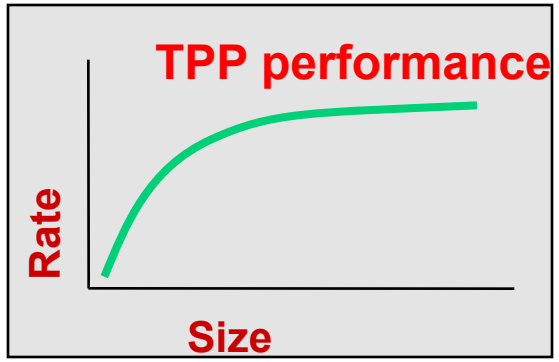
University of Tennessee
Oak Ridge National Laboratory
University of Manchester

Top500 List of Supercomputers

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

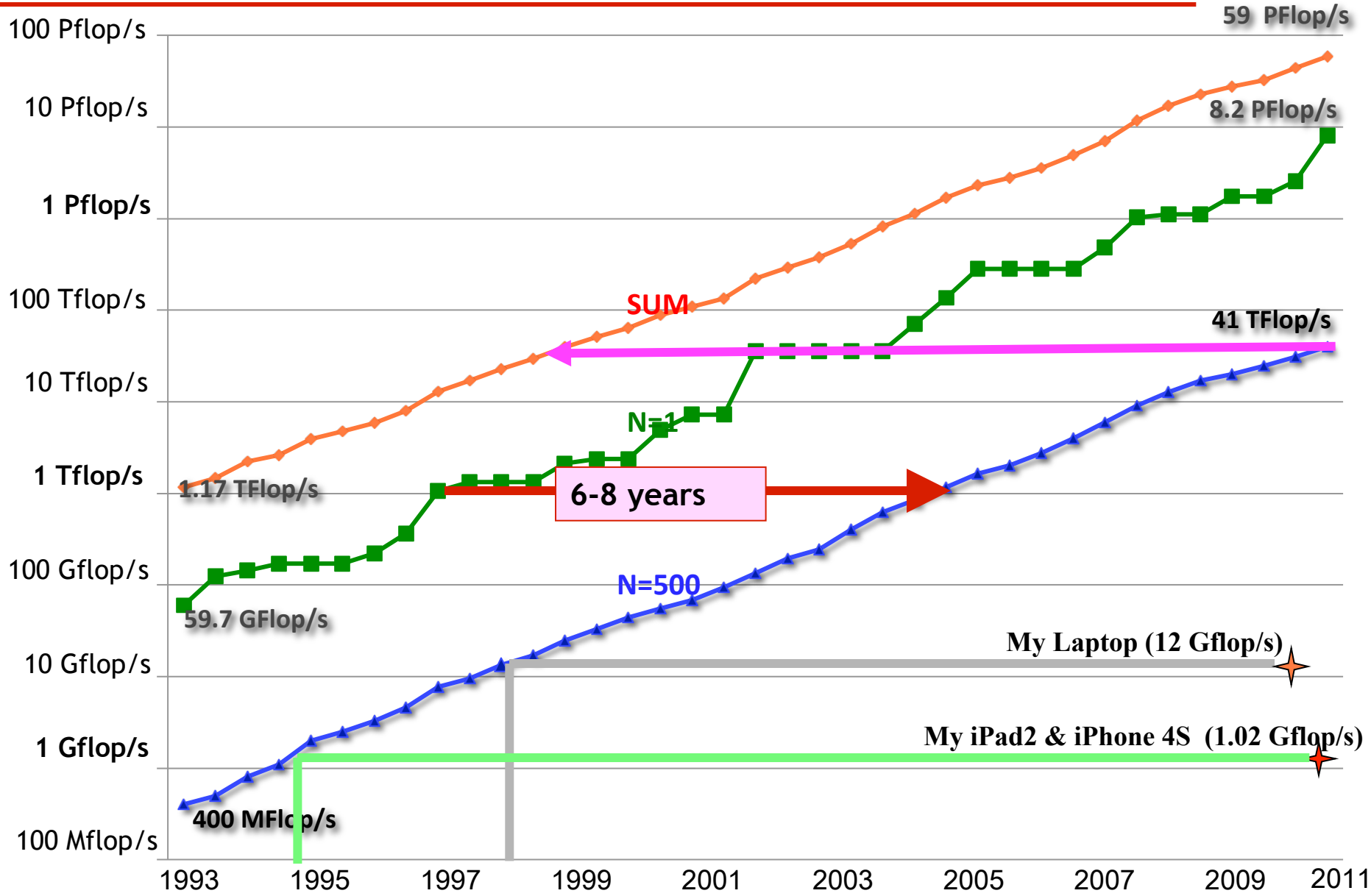
- Listing of the 500 most powerful Computers in the World
- Yardstick: Rmax from LINPACK MPP

$Ax=b$, *dense problem*

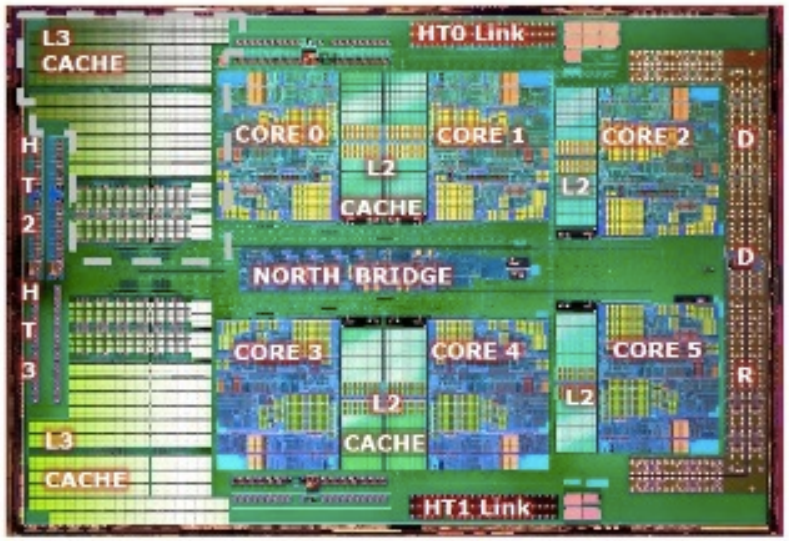


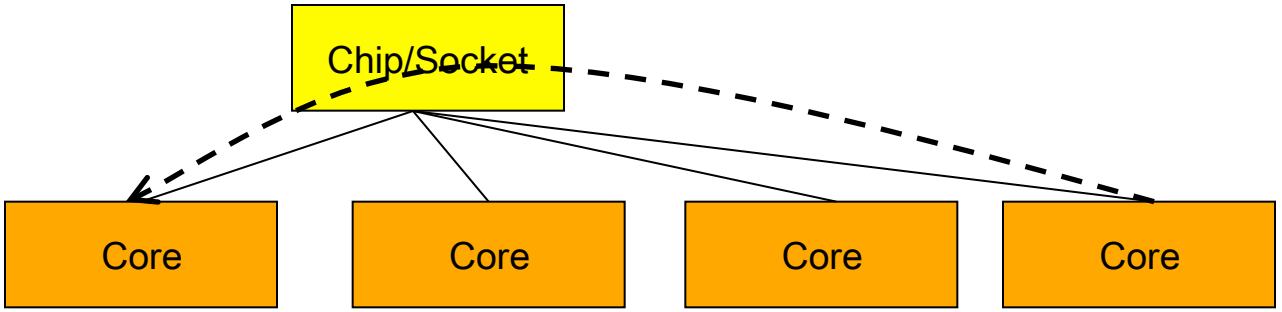
- Updated twice a year
  SC‘xy in the States in November
  Meeting in Germany in June
- All data available from **www.top500.org**

Performance Development

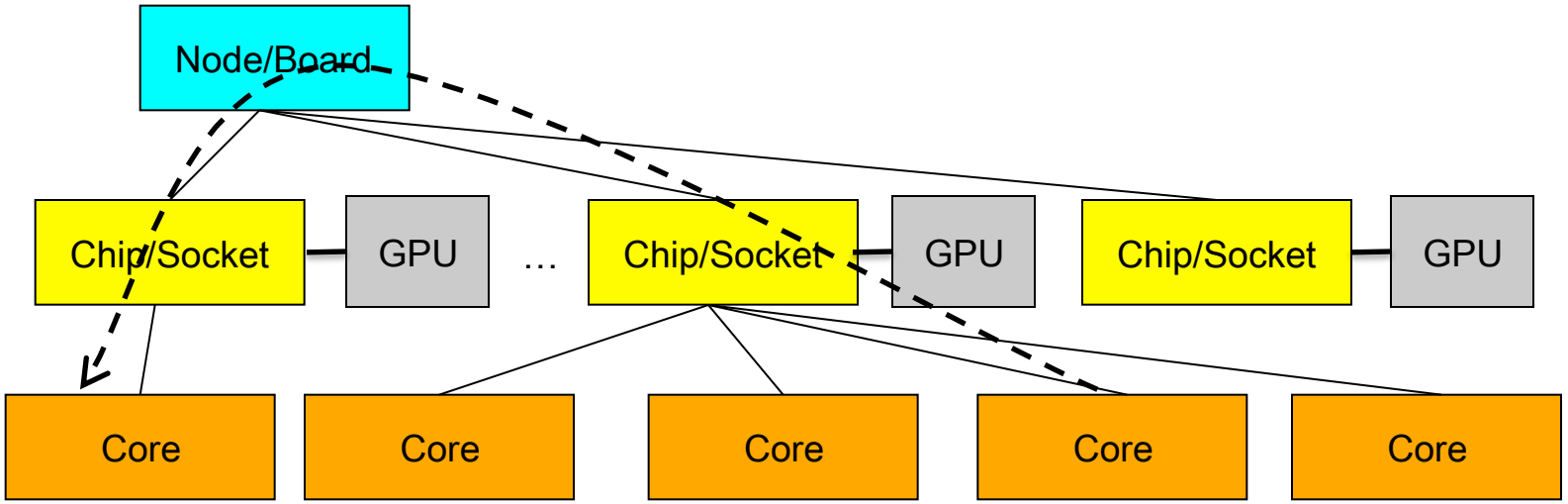


Example of typical parallel machine



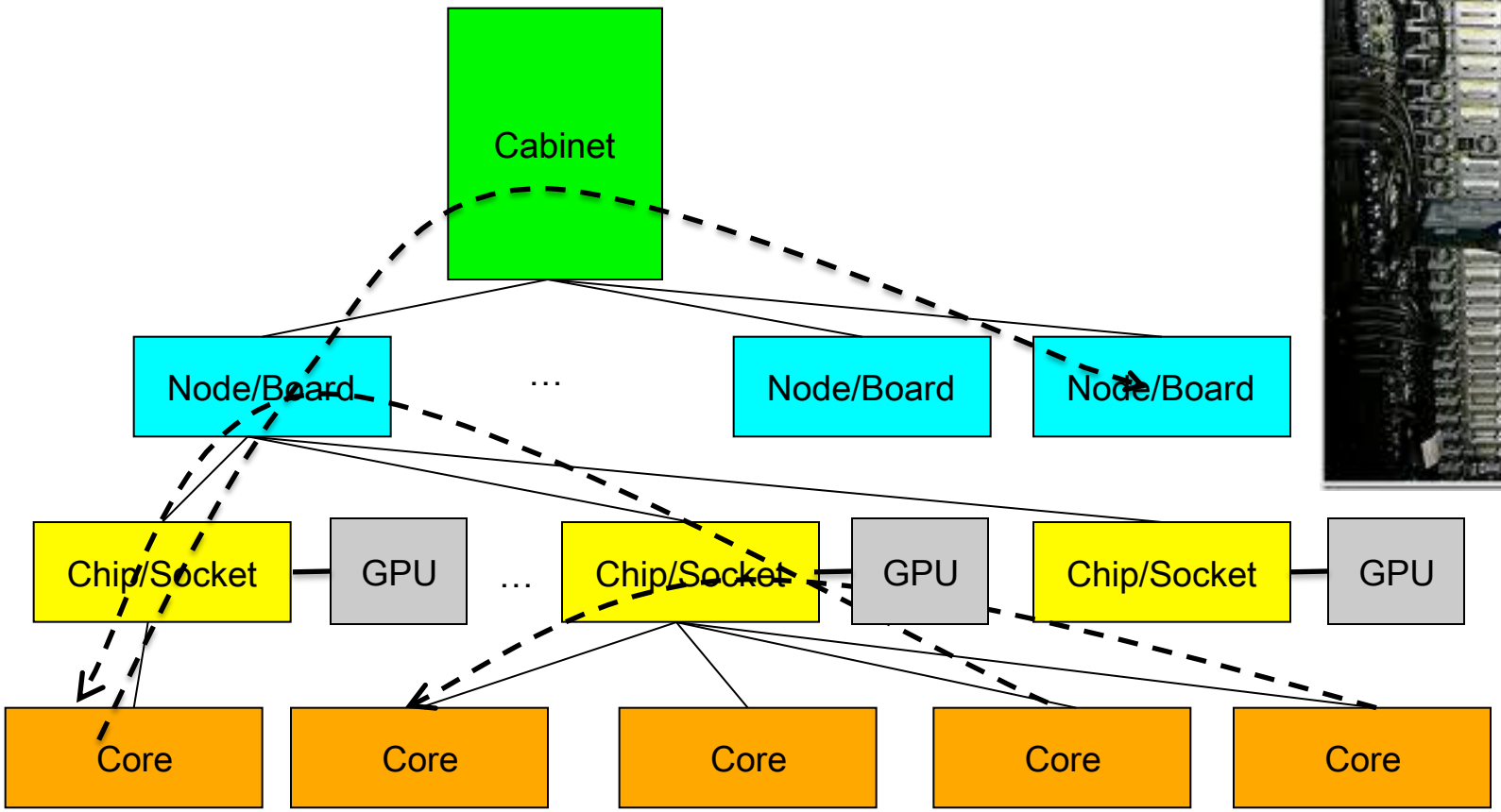


Example of typical parallel machine

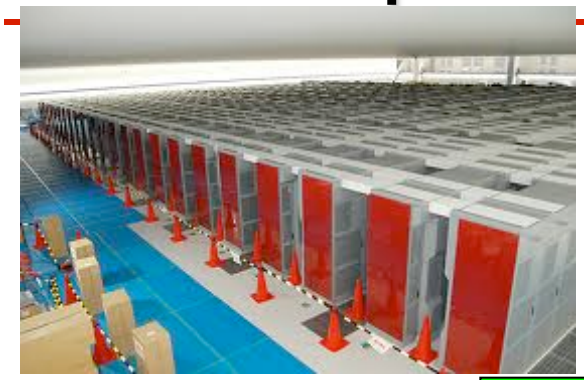


Example of typical parallel machine

Shared memory programming between processes on a board and
a combination of shared memory and distributed memory programming
between nodes and cabinets



Example of typical parallel machine



Combination of shared memory and distributed memory programming

June 2011: The TOP10

Rank	Site	Computer	Country	Cores	Rmax [Pflops]	% of Peak
1	RIKEN Advanced Inst for Comp Sci	K Computer Fujitsu SPARC64 VIIIfx + custom	Japan	548,352	8.16	93
2	Nat. SuperComputer Center in Tianjin	Tianhe-1A, NUDT Intel + Nvidia GPU + custom	China	186,368	2.57	55
3	DOE / OS Oak Ridge Nat Lab	Jaguar, Cray AMD + custom	USA	224,162	1.76	75
4	Nat. Supercomputer Center in Shenzhen	Nebulea, Dawning Intel + Nvidia GPU + IB	China	120,640	1.27	43
5	GSIC Center, Tokyo Institute of Technology	Tusbame 2.0, HP Intel + Nvidia GPU + IB	Japan	73,278	1.19	52
6	DOE / NNSA LANL & SNL	Cielo, Cray AMD + custom	USA	142,272	1.11	81
7	NASA Ames Research Center/NAS	Plelades SGI Altix ICE 8200EX/8400EX + IB	USA	111,104	1.09	83
8	DOE / OS Lawrence Berkeley Nat Lab	Hopper, Cray AMD + custom	USA	153,408	1.054	82
9	Commissariat a l'Energie Atomique (CEA)	Tera-10, Bull Intel + IB	France	138,368	1.050	84
10	DOE / NNSA Los Alamos Nat Lab	Roadrunner, IBM AMD + Cell GPU + IB	USA	122,400	1.04	76

June 2011: The TOP10

Rank	Site	Computer	Country	Cores	Rmax [Pflops]	% of Peak	Power [MW]	GFlops/ Watt
1	RIKEN Advanced Inst for Comp Sci	K Computer Fujitsu SPARC64 VIIIfx + custom	Japan	548,352	8.16	93	9.9	824
2	Nat. SuperComputer Center in Tianjin	Tianhe-1A, NUDT Intel + Nvidia GPU + custom	China	186,368	2.57	55	4.04	636
3	DOE / OS Oak Ridge Nat Lab	Jaguar, Cray AMD + custom	USA	224,162	1.76	75	7.0	251
4	Nat. Supercomputer Center in Shenzhen	Nebulae, Dawning Intel + Nvidia GPU + IB	China	120,640	1.27	43	2.58	493
5	GSIC Center, Tokyo Institute of Technology	Tusbame 2.0, HP Intel + Nvidia GPU + IB	Japan	73,278	1.19	52	1.40	850
6	DOE / NNSA LANL & SNL	Cielo, Cray AMD + custom	USA	142,272	1.11	81	3.98	279
7	NASA Ames Research Center/NAS	Plelades SGI Altix ICE 8200EX/8400EX + IB	USA	111,104	1.09	83	4.10	265
8	DOE / OS Lawrence Berkeley Nat Lab	Hopper, Cray AMD + custom	USA	153,408	1.054	82	2.91	362
9	Commissariat a l'Energie Atomique (CEA)	Tera-10, Bull Intel + IB	France	138,368	1.050	84	4.59	229
10	DOE / NNSA Los Alamos Nat Lab	Roadrunner, IBM AMD + Cell GPU + IB	USA	122,400	1.04	76	2.35	446
500	Energy Comp	IBM Cluster, Intel + GigE	China	7,104	.041	53		

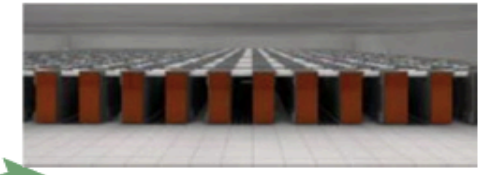
Quiz: How Many of the Top500 systems use GPUs?

Japanese K Computer

K computer Specifications

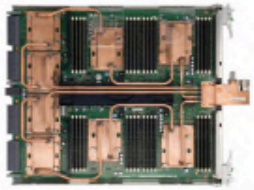
CPU (SPARC64 VIIIfx)	Cores/Node	8 cores (@2GHz)
	Performance	128GFlops
	Architecture	SPARC V9 + HPC extension
	Cache	L1(I/D) Cache : 32KB/32KB L2 Cache : 6MB
	Power	58W (typ. 30 C)
	Mem. bandwidth	64GB/s.
Node	Configuration	1 CPU / Node
	Memory capacity	16GB (2GB/core)
System board(SB)	No. of nodes	4 nodes /SB
Rack	No. of SB	24 SBs/rack
System	Nodes/system	> 80,000

Inter-connect	Topology	6D Mesh/Torus
	Performance	5GB/s. for each link
	No. of link	10 links/ node
	Additional feature	H/W barrier, reduction
	Architecture	Routing chip structure (no outside switch box)
Cooling	CPU, ICC*	Direct water cooling
	Other parts	Air cooling



System
LINPACK 10 PFlops
over 1PB mem.
800 racks
80,000 CPUs
640,000 cores

Rack
12.3 TFlops
15TB memory



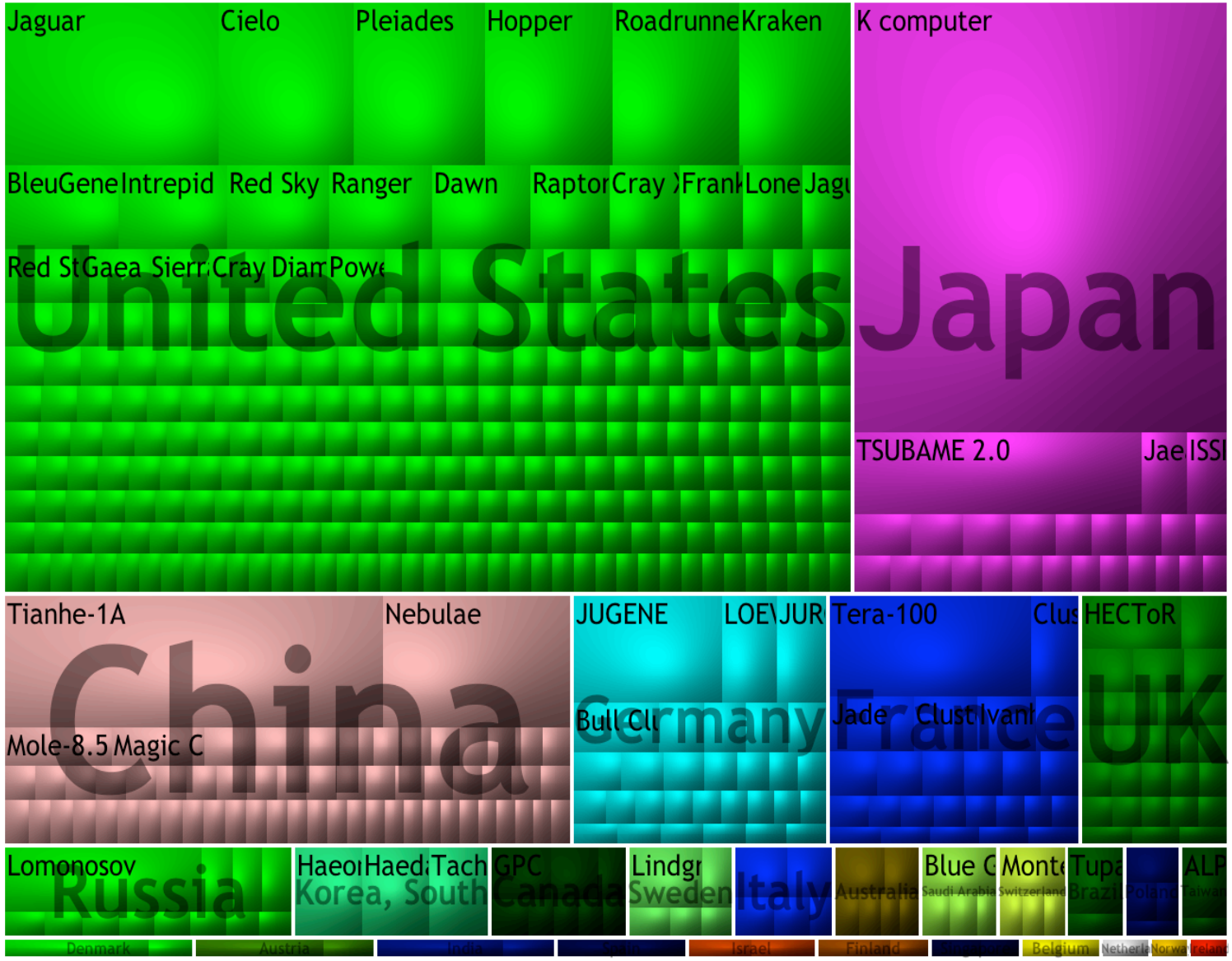
System Board
512 GFlops
64 GB memory

Node
128 GFlops
16GB Memory
64GB/s Memory band width

CPU
128GFlops
SPARC64™ VIIIfx
8 Cores@2.0GHz

* ICC : Interconnect Chip

Countries Share



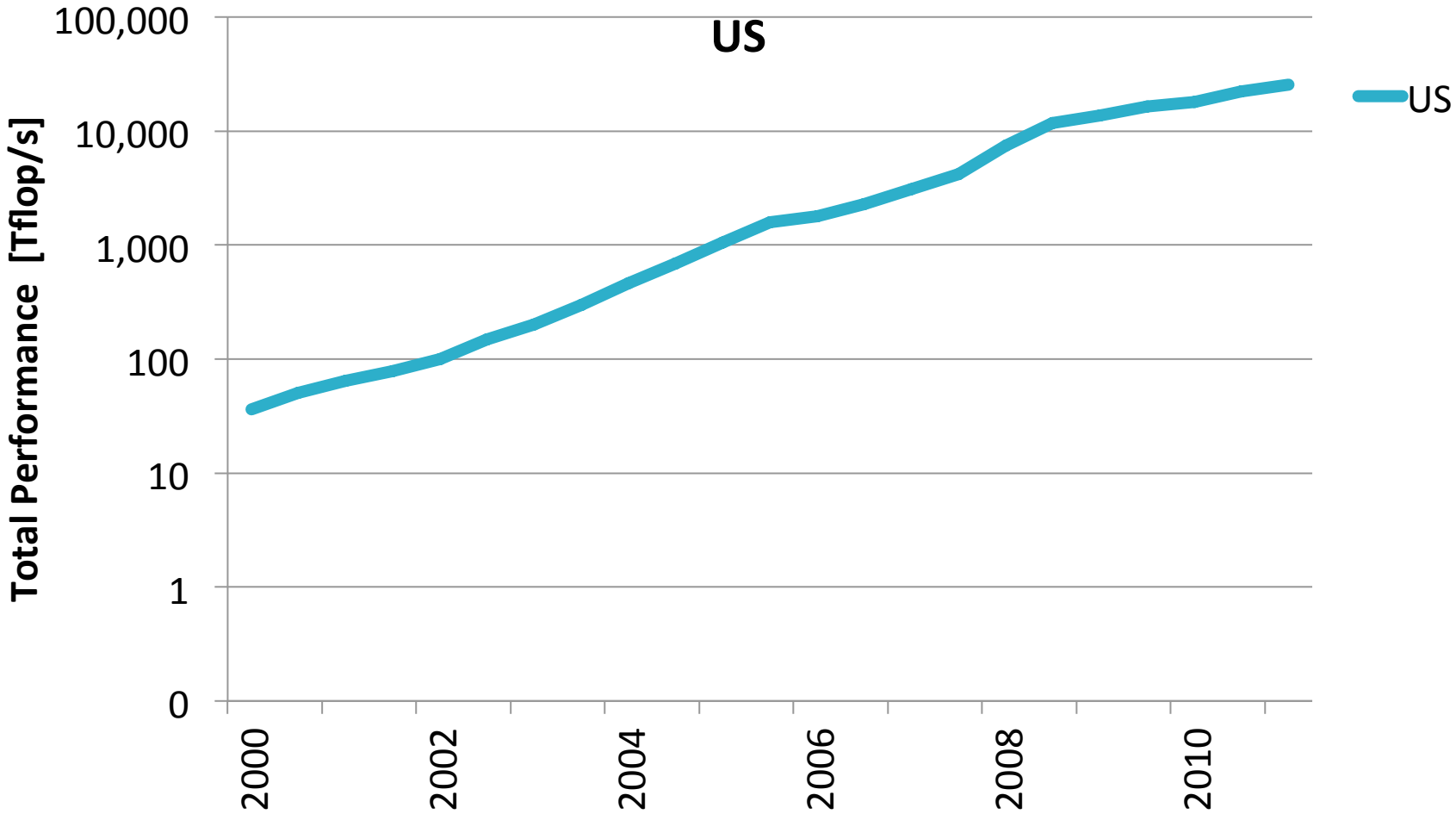
Absolute Counts

US:	251
China:	64
Germany:	31
UK:	28
Japan:	26
France:	25

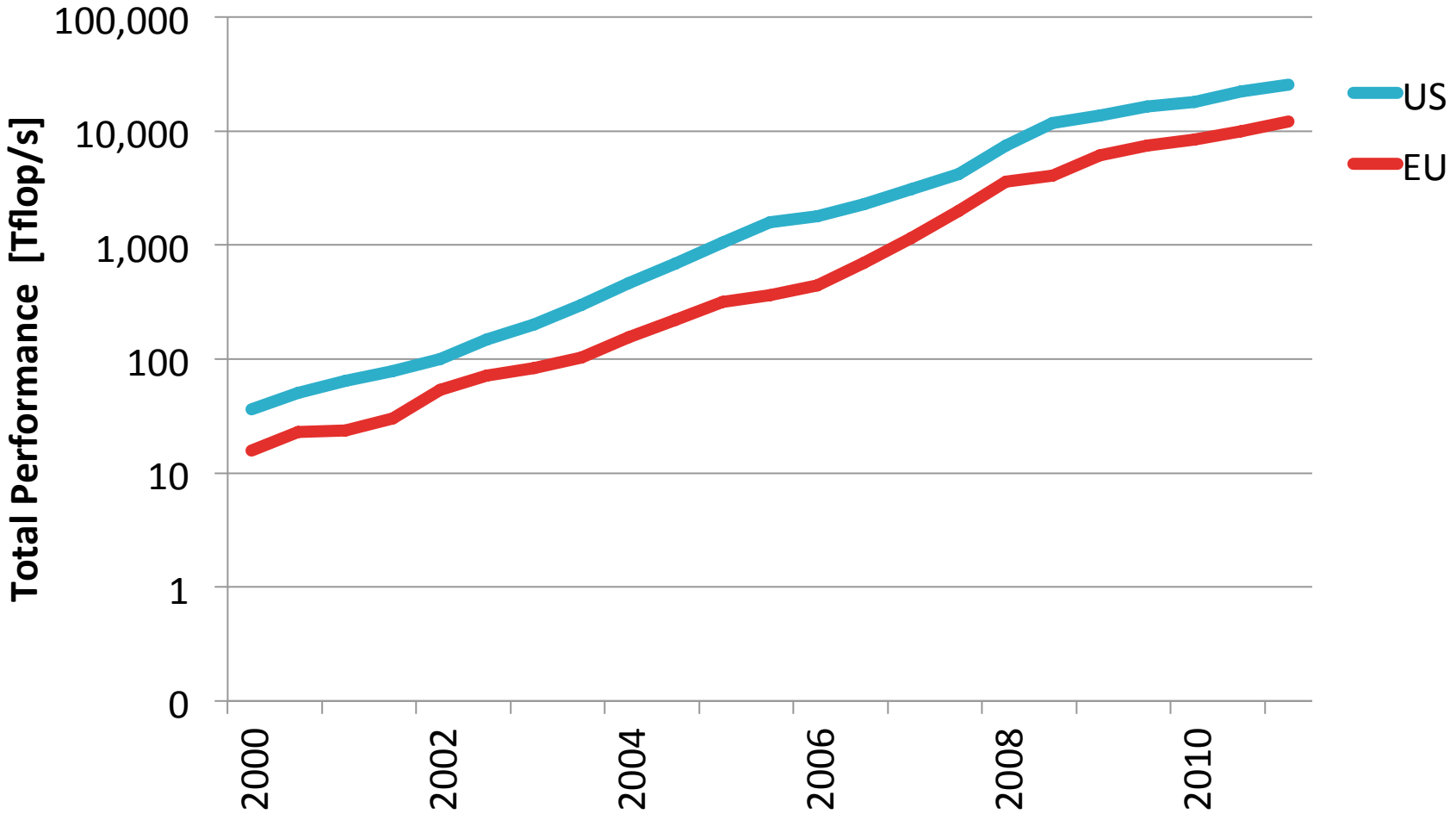
Top500 Computers in Brazil

Rank	Site	Manufacturer	Computer	Cores	RMax
34	INPE (National Institute for Space Research)	Cray Inc.	Cray XT6 12-core 2.1 GHz	30720	205100
167	NACAD/COPPE/UFRJ	Oracle	Sun Blade x6048, Xeon X5560 2.8 Ghz, Infiniband QDR	6464	64630

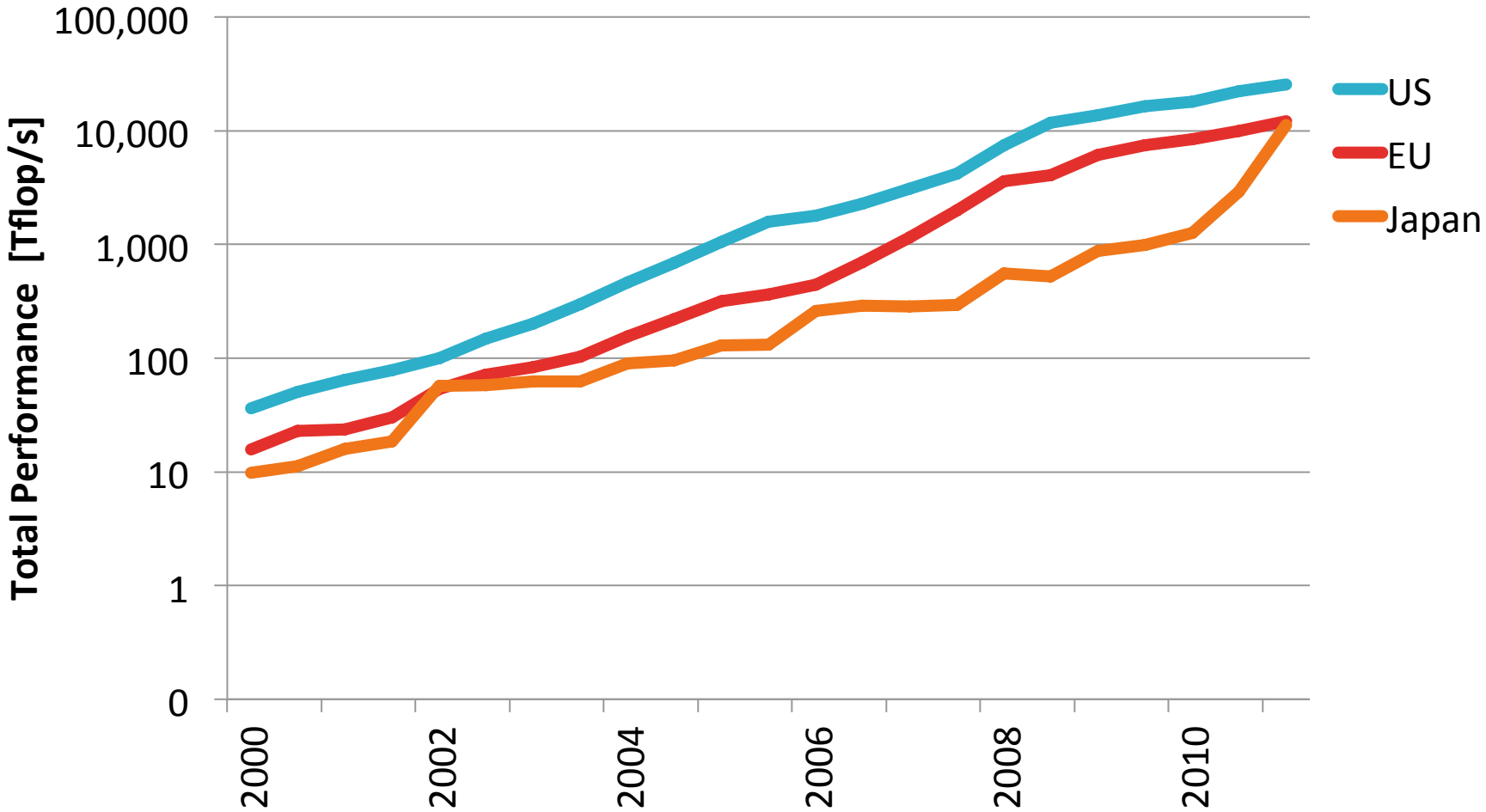
Performance by Regions



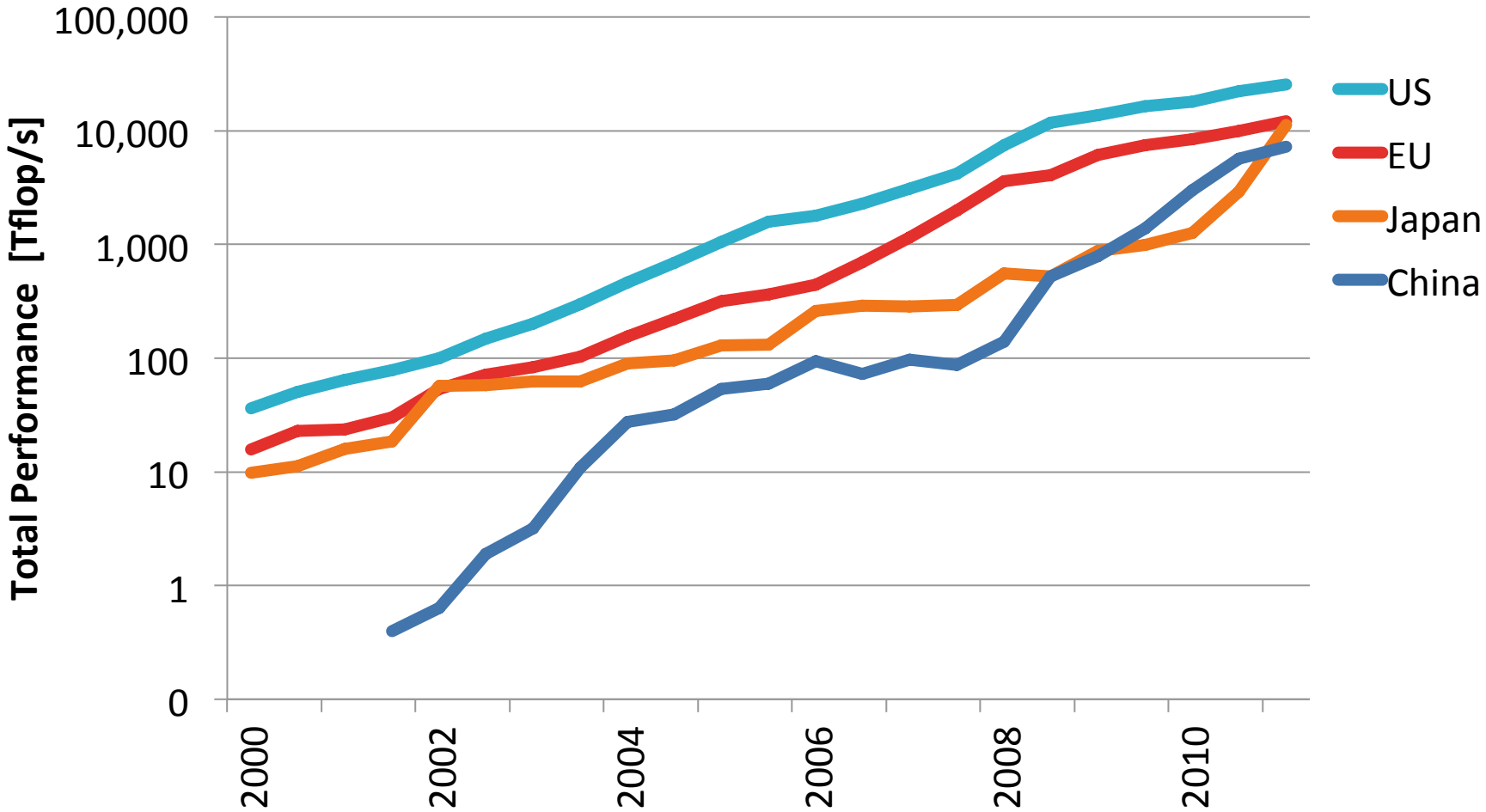
Performance by Regions



Performance by Regions

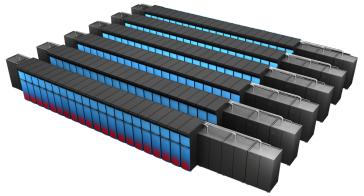


Performance by Regions



10+ Pflop/s Systems Planned in the States

- DOE SC, Titan at Oak Ridge Nat Lab,
  - Based on Cray design with Nvidia accelerators, 20 Pflop/s
- DOE NNSA, Sequoia at Lawrence Livermore Nat. Lab,
  - Based on IBM's BG/Q, 20 Pflop/s
- DOE SC, BG/Q at Argonne National Lab,
  - Based on IBM's BG/Q, 10 Pflop/s
- NSF, Blue Waters at University of Illinois, UC
  - Based on ??, 10 Pflop/s
- NSF, University of Texas, Austin (Stampede)
  - Based on Dell/Intel (Sandy Bridge + MIC), 10 Pflop/s

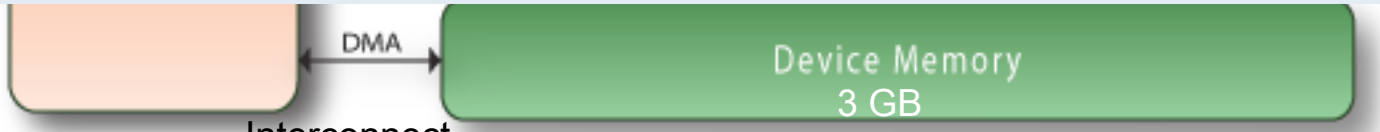


Commodity plus Accelerator

Commodity	Accelerator (GPU)
Intel Xeon	Nvidia C2050 "Fermi"
8 cores	448 CUDA cores
3 GHz	1.15 GHz
8*4 ops/cycle	448 ops/cycle
96 Gflop/s (DP)	515 Gflop/s (DP)

Quiz: How Many of the Top500 systems use GPUs?

Answer:
Today only 19 systems on the TOP500 use GPUs

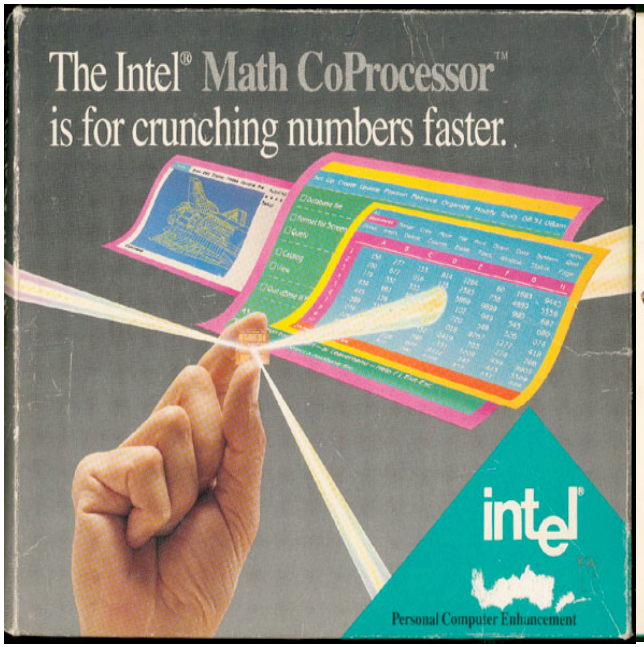


Interconnect
PCI-X 16 lane
64 Gb/s
1 GW/s

We Have Seen This Before

- Floating Point Systems FPS-164/ MAX Supercomputer (1976)
- Intel Math Co-processor (1980)
- Weitek Math Co-processor (1981)

1976



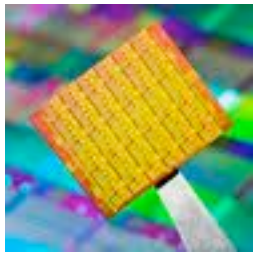
1980

Balance Between Data Movement and Floating point

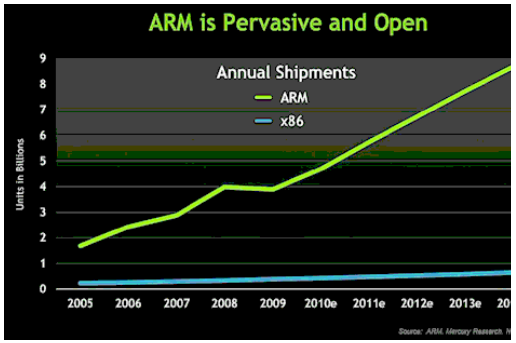
- FPS-164 and VAX (1976)
  - 11 Mflop/s; transfer rate 44 MB/s
  - Ratio of flops to bytes of data movement: 1 flop per 4 bytes transferred
- Nvidia Fermi and PCI-X to host
  - 500 Gflop/s; transfer rate 8 GB/s
  - Ratio of flops to bytes of data movement: 62 flops per 1 byte transferred
- Flop/s are cheap, so are provisioned in excess

Future Computer Systems

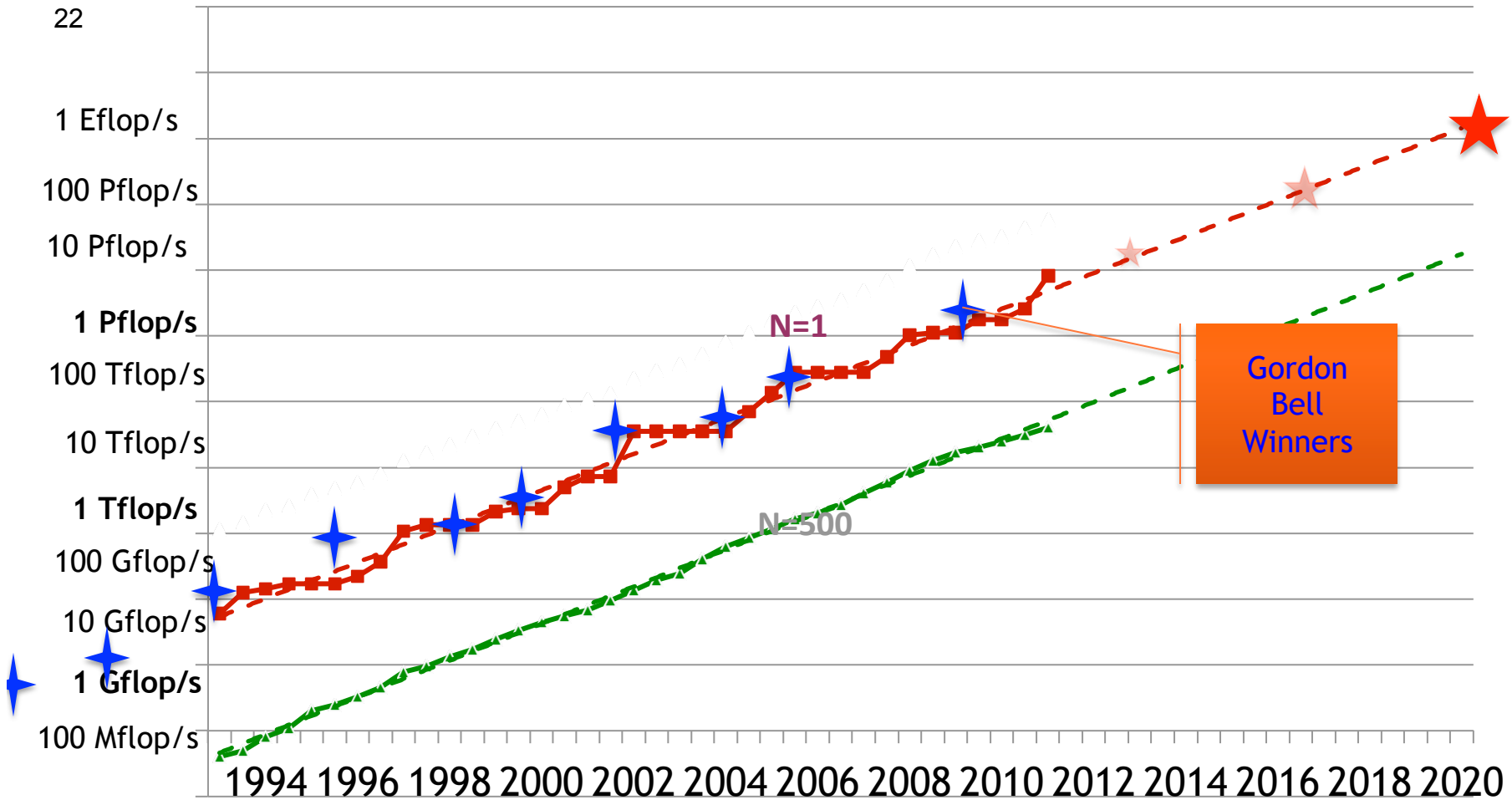
- Most likely be a hybrid design
  - Think standard multicore chips and accelerator (GPUs)
- Today accelerators are attached
- Next generation more integrated
- Intel's MIC architecture "Knights Ferry" and "Knights Corner" to come.
  - 48 x86 cores
- AMD's Fusion in 2012 - 2013
  - Multicore with embedded graphics ATI
- Nvidia's Project Denver plans to develop an integrated chip using ARM architecture in 2013.



AMD
The future is fusion



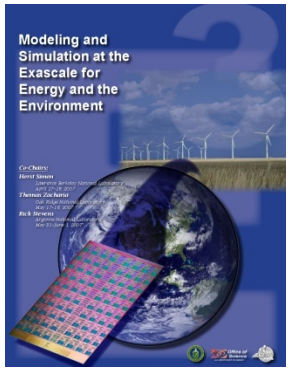
Performance Development in Top500

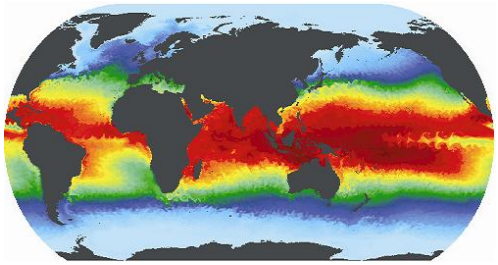


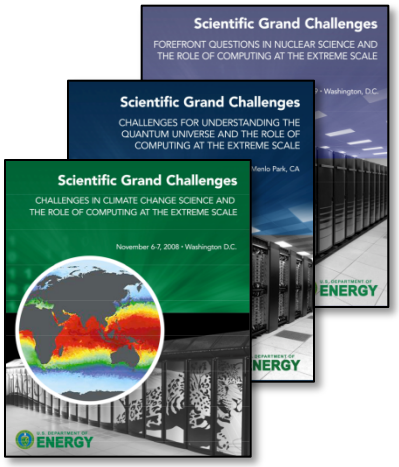
Broad Community Support and Development of the Exascale Initiative Since 2007

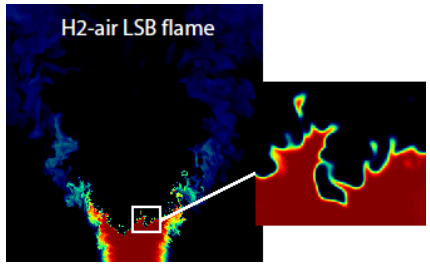
http://science.energy.gov/ascr/news-and-resources/program-documents/

- **Town Hall Meetings April-June 2007**
- **Scientific Grand Challenges Workshops Nov, 2008 – Oct, 2009**
  - **Climate Science (11/08)**
  - **High Energy Physics (12/08)**
  - **Nuclear Physics (1/09)**
  - **Fusion Energy (3/09)**
  - **Nuclear Energy (5/09)**
  - **Biology (8/09)**
  - **Material Science and Chemistry (8/09)**
  - **National Security (10/09)**
  - **Cross-cutting technologies (2/10)**
- **Exascale Steering Committee**
  - **"Denver" vendor NDA visits (8/09)**
  - **SC09 vendor feedback meetings**
  - **Extreme Architecture and Technology Workshop (12/09)**
- **International Exascale Software Project**
  - **Santa Fe, NM (4/09); Paris, France (6/09); Tsukuba, Japan (10/09); Oxford (4/10); Maui (10/10); San Francisco (4/11); Cologne (10/11)**

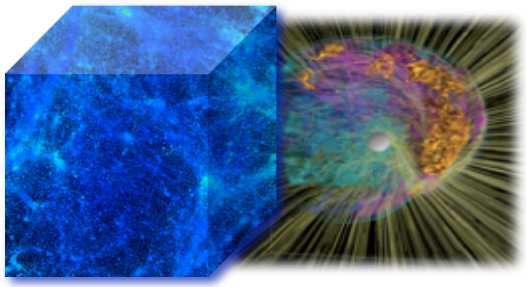








Mission Imperatives



Fundamental Science

Potential System Architecture

Systems	2011 K Computer
System peak	**8.7 Pflop/s**
Power	**10 MW**
System memory	1.6 PB
Node performance	128 GF
Node memory BW	64 GB/s
Node concurrency	8
Total Node Interconnect BW	20 GB/s
System size (nodes)	68,544
Total concurrency	548,352
MTTI	days

Potential System Architecture with a cap of $200M and 20MW

Systems	2011 K Computer	2019	Difference Today & 2019
System peak	**8.7 Pflop/s**	**1 Eflop/s**	**O(100)**
Power	**10 MW**	**~20 MW**	
System memory	1.6 PB	32 - 64 PB	O(10)
Node performance	128 GF	1,2 or 15TF	O(10) – O(100)
Node memory BW	64 GB/s	2 - 4TB/s	O(100)
Node concurrency	8	O(1k) or 10k	O(100) – O(1000)
Total Node Interconnect BW	20 GB/s	200-400GB/s	O(10)
System size (nodes)	68,544	O(100,000) or O(1M)	O(10) – O(100)
Total concurrency	548,352	O(billion)	O(1,000)
MTTI	days	O(1 day)	- O(10)

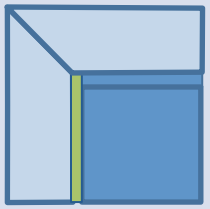
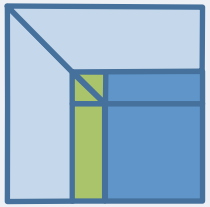
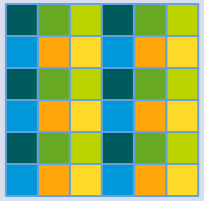
Major Changes to Software & Algorithms

- **Must rethink the design of our algorithms and software**
  - **Another disruptive technology**
    - Similar to what happened with cluster computing and message passing
  - **Rethink and rewrite the applications, algorithms, and software**
  - **Data movement is expense**
  - **Flop/s are cheap, so are provisioned in excess**

Critical Issues at Peta & Exascale for Algorithm and Software Design

- Synchronization-reducing algorithms
  - Break Fork-Join model
- Communication-reducing algorithms
  - Use methods which have lower bound on communication
- Mixed precision methods
  - 2x speed of ops and 2x speed for data movement
- Autotuning
  - Today's machines are too complicated, build "smarts" into software to adapt to the hardware
- Fault resilient algorithms
  - Implement algorithms that can recover from failures/bit flips
- Reproducibility of results
  - Today we can't guarantee this. We understand the issues, but some of our "colleagues" have a hard time with this.

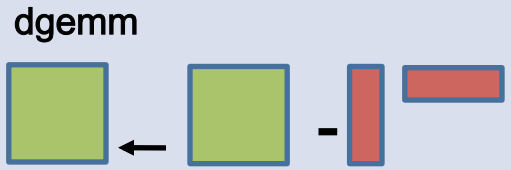
Do you remember the 80's and 90's?

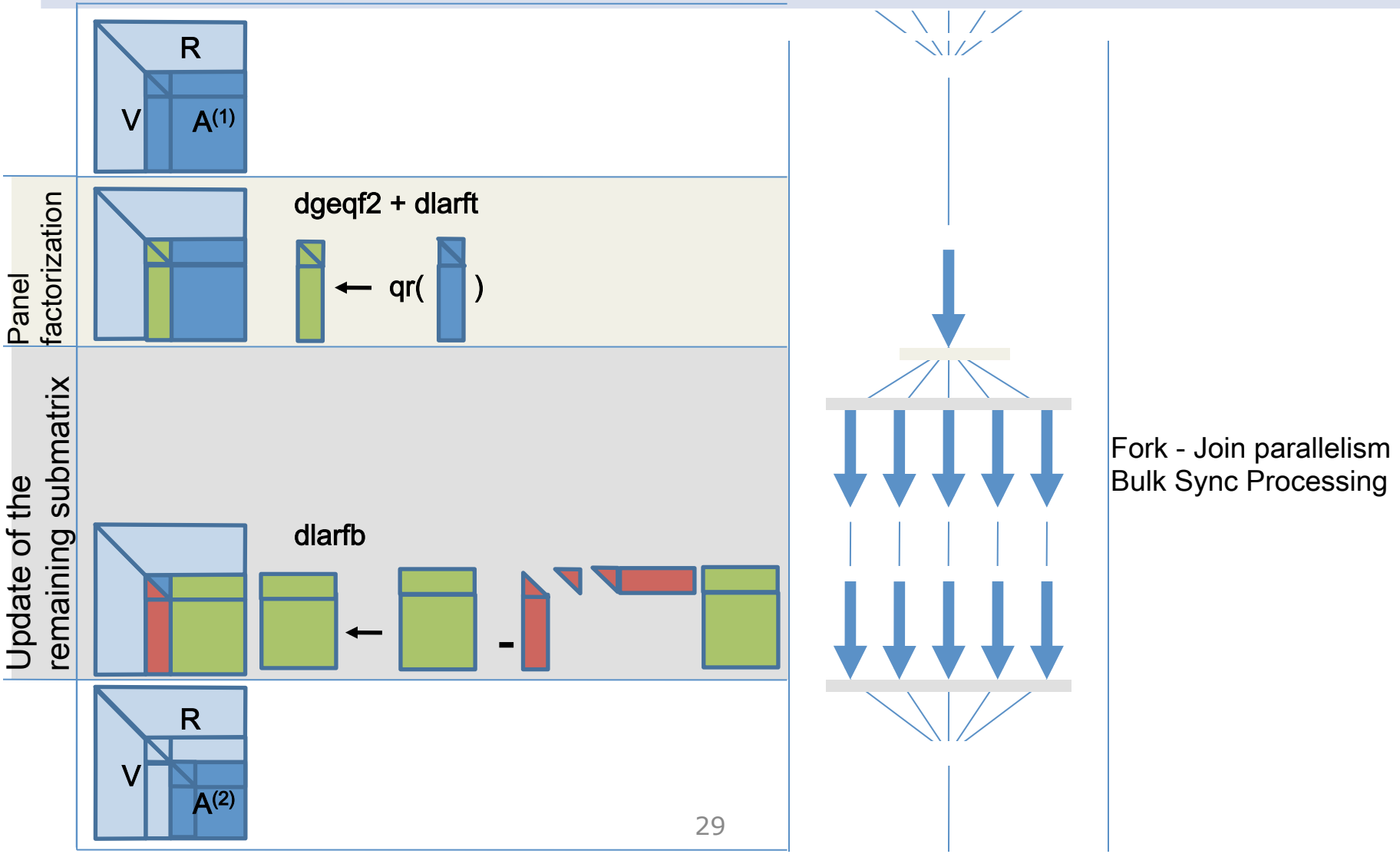
Algorithms follow hardware evolution along time.		
LINPACK (80's) (Vector operations)		Rely on - Level-1 BLAS operations
LAPACK (90's) (Blocking, cache friendly)		Rely on - Level-3 BLAS operations
ScaLAPACK (00's) (Distributed memory, Message passing)		Rely on -Level-3 BLAS operations - MPI for message passing

Parallelization of QR Factorization

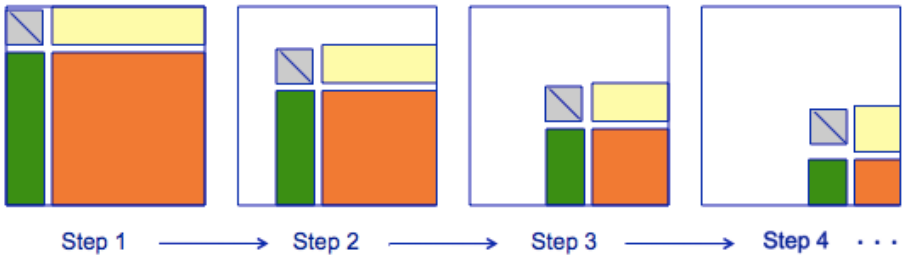
Parallelize the update:

- Easy and done in any reasonable software.
- This is the $2/3n^3$ term in the FLOPs count.
- Can be done "efficiently" with LAPACK+multithreaded BLAS

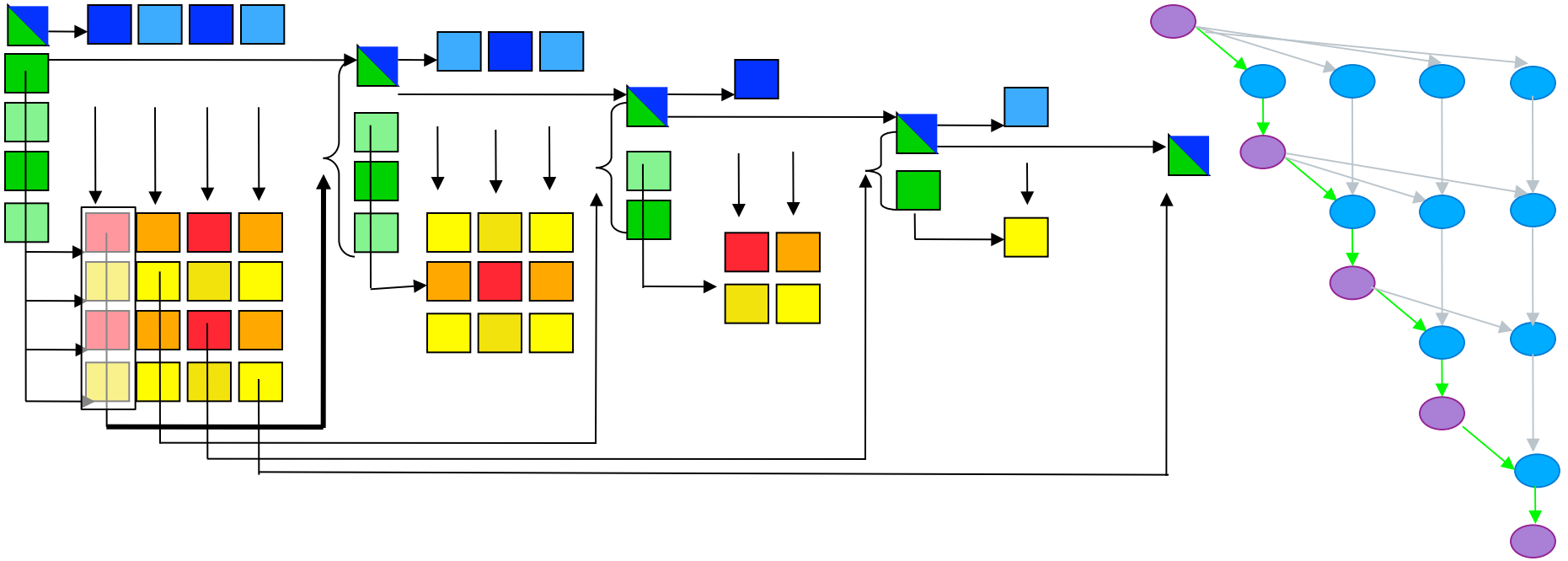




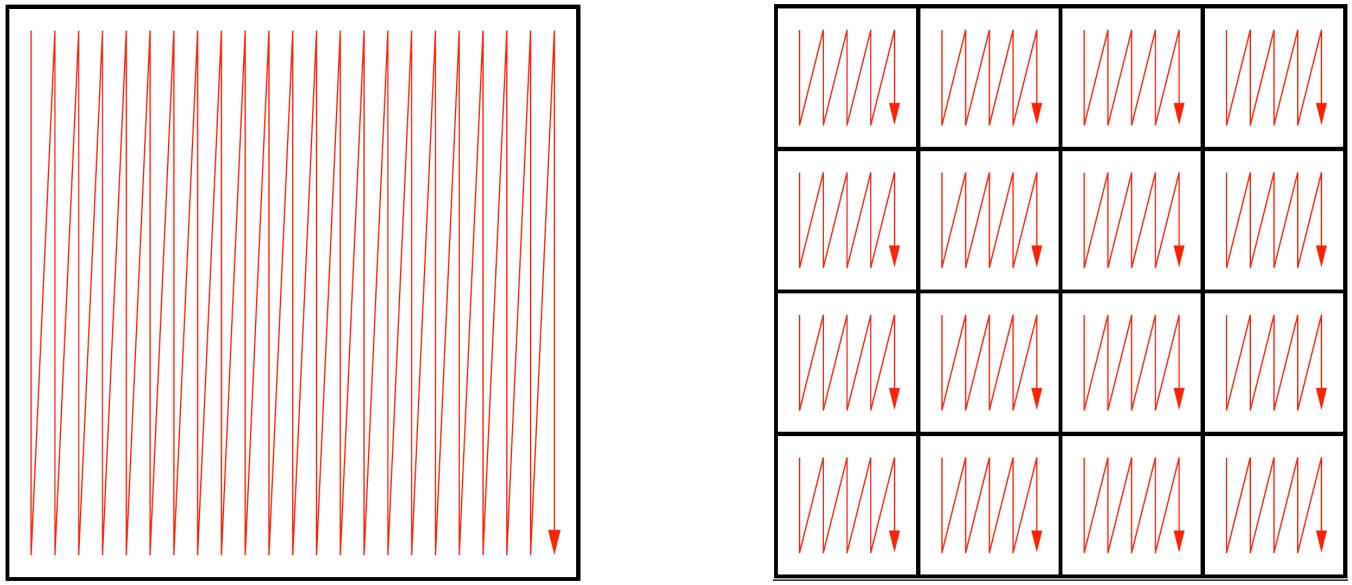
Parallel Tasks in LU/LL^T/QR



- Break into smaller tasks and remove dependencies



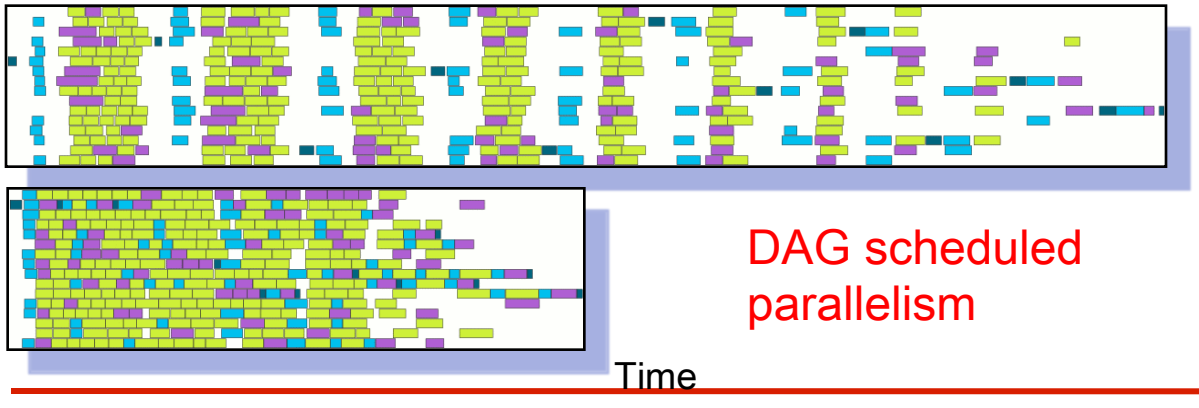
Data Layout is Critical

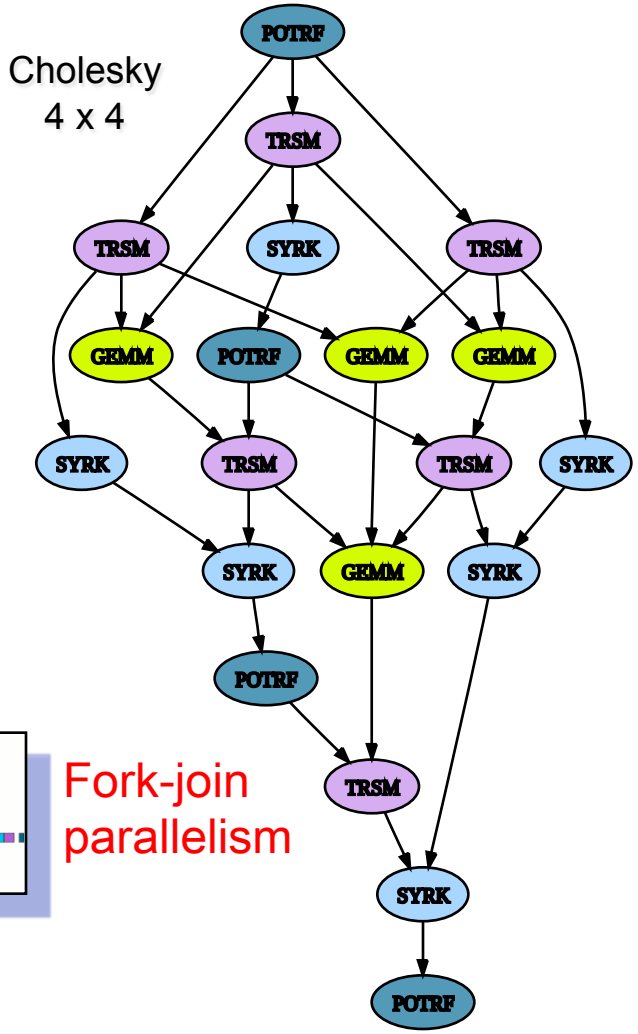


- Tile data layout where each data tile is contiguous in memory
- Decomposed into several fine-grained tasks, which better fit the memory of the small core caches

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 (QUARK)
  - Explicit parallelism
  - Implicit communication
  - Fine granularity / block data layout
- **Arbitrary DAG with dynamic scheduling**

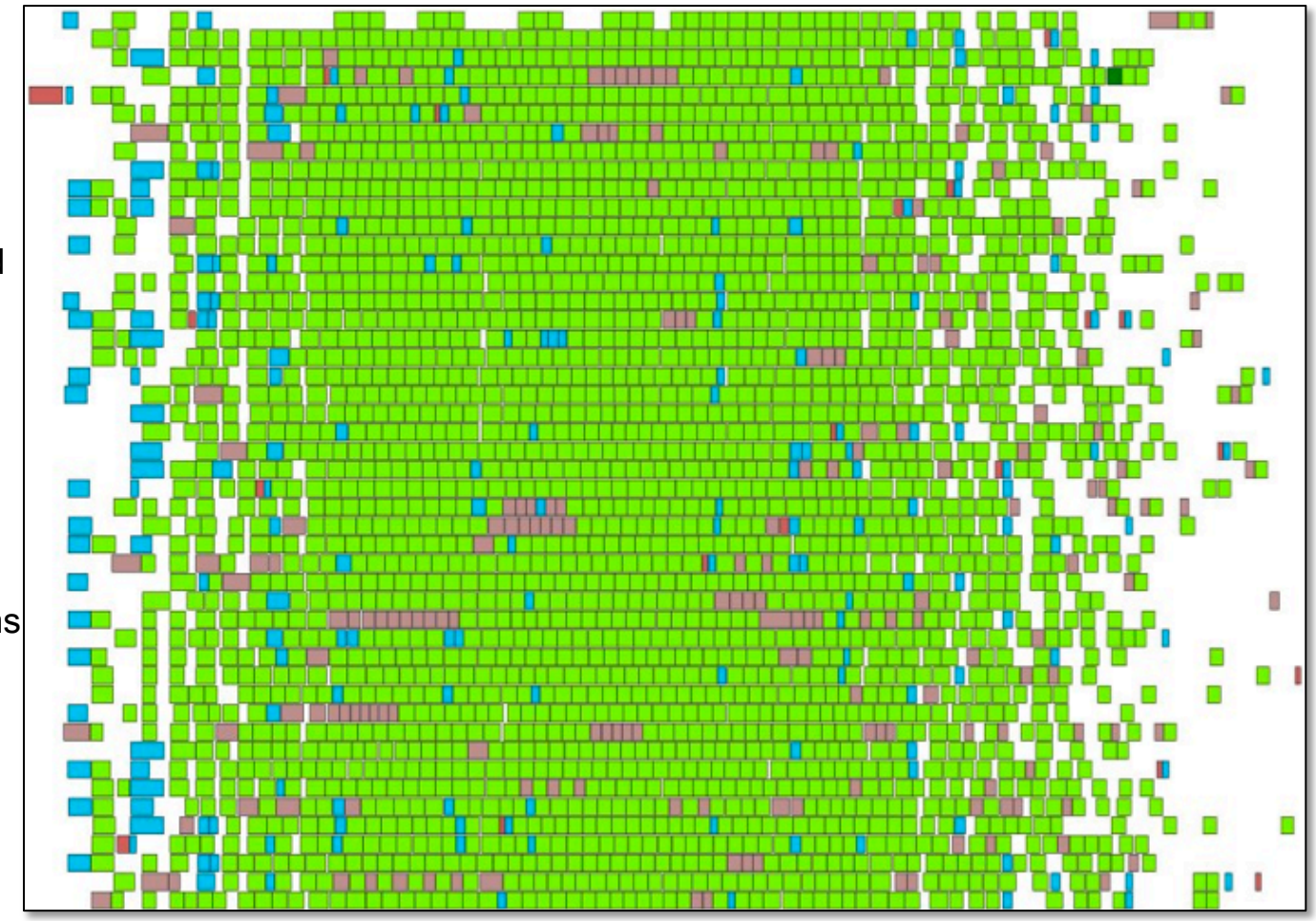




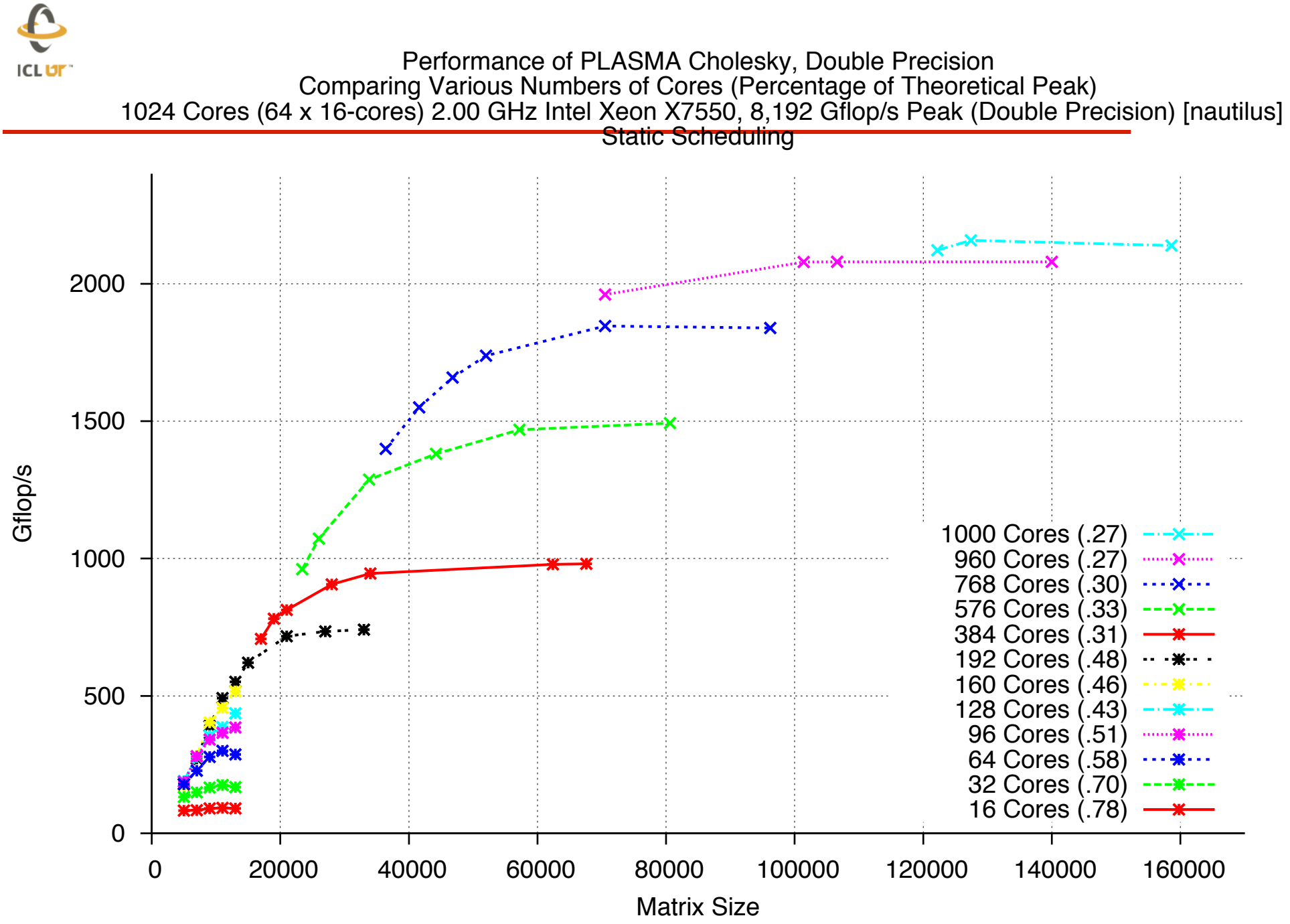
Synchronization Reducing Algorithms

- Regular trace
- Factorization steps pipelined
- Stalling only due to natural load imbalance
- Dynamic
- Out of order execution
- Fine grain tasks
- Independent block operations

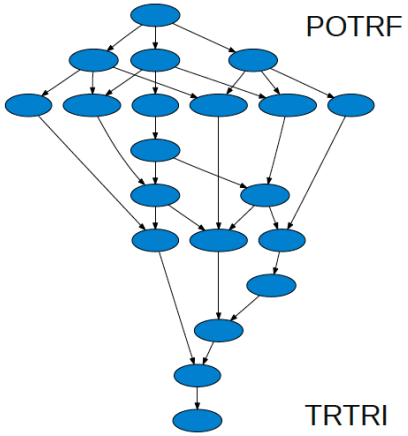
The colored area over the rectangle is the efficiency

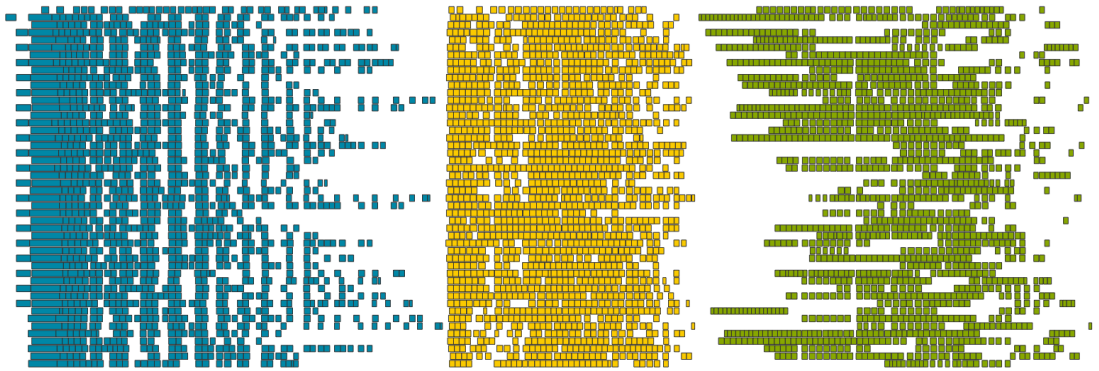


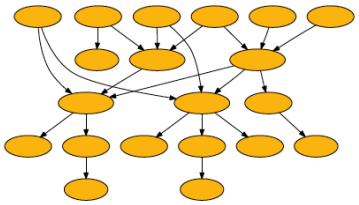
Tile QR factorization; Matrix size 4000x4000, Tile size 200
8-socket, 6-core (48 cores total) AMD Istanbul 2.8 GHz

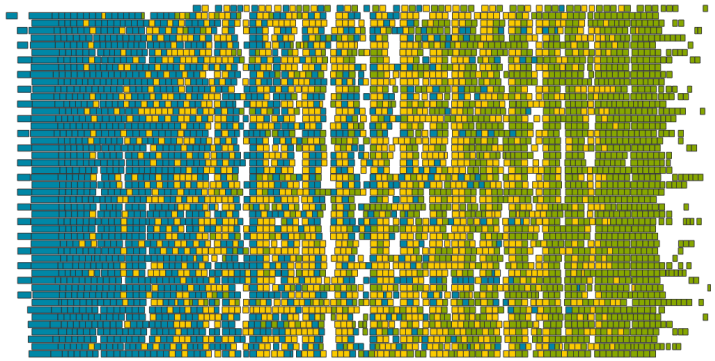

Performance of PLASMA Cholesky, Double Precision
Comparing Various Numbers of Cores (Percentage of Theoretical Peak)
1024 Cores (64 x 16-cores) 2.00 GHz Intel Xeon X7550, 8,192 Gflop/s Peak (Double Precision) [nautilus]
Static Scheduling
Gflop/s
Matrix Size
0
500
1000
1500
2000
0
20000
40000
60000
80000
100000
120000
140000
160000
1000 Cores (.27)
960 Cores (.27)
768 Cores (.30)
576 Cores (.33)
384 Cores (.31)
192 Cores (.48)
160 Cores (.46)
128 Cores (.43)
96 Cores (.51)
64 Cores (.58)
32 Cores (.70)
16 Cores (.78)

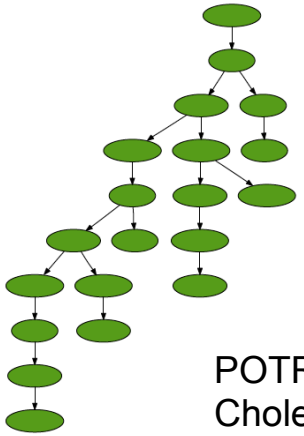
Pipelining: Cholesky Inversion
3 Steps: Factor, Invert L, Multiply L's





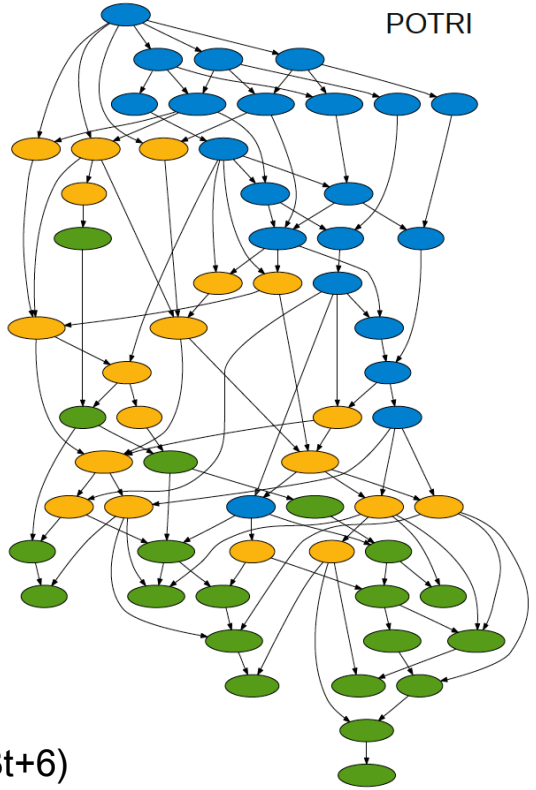




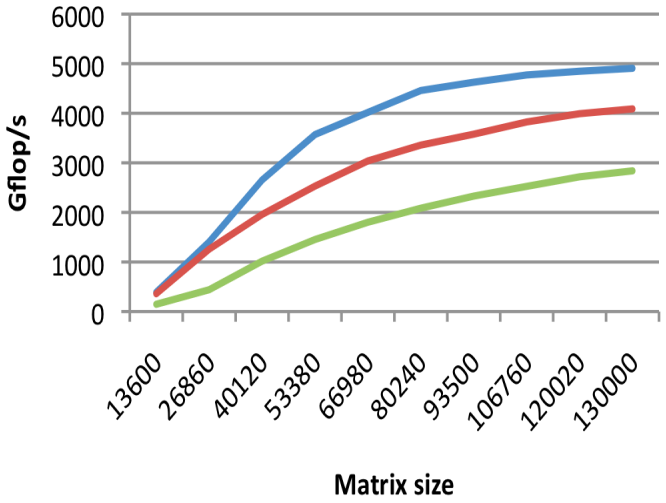


48 cores
POTRF, TRTRI and LAUUM.
The matrix is 4000 x 4000,tile size is 200 x 200,

POTRF+TRTRI+LAUUM: 25 (7t-3)
Cholesky Factorization alone: 3t-2



Pipelined: 18 (3t+6)

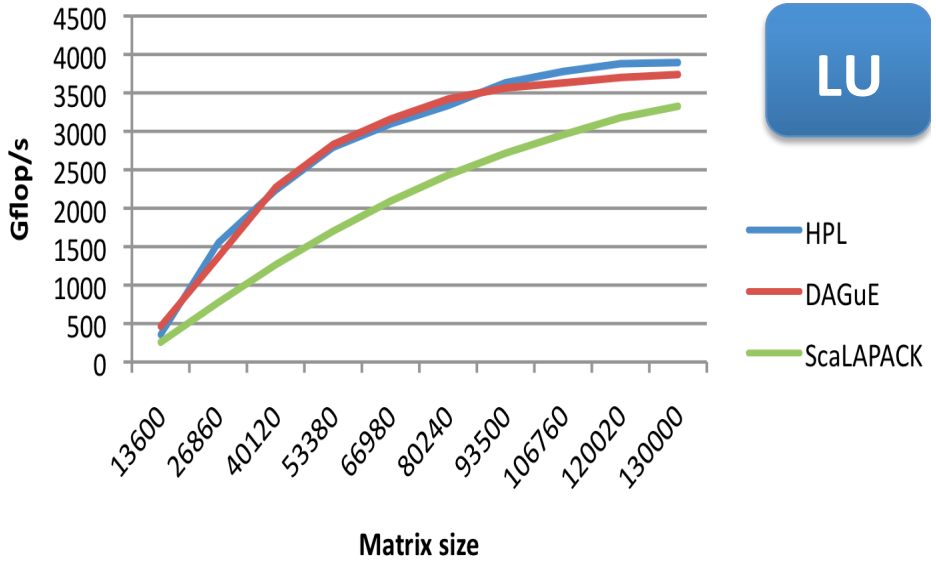


Cholesky

DAGuE
DSBP
ScaLAPACK

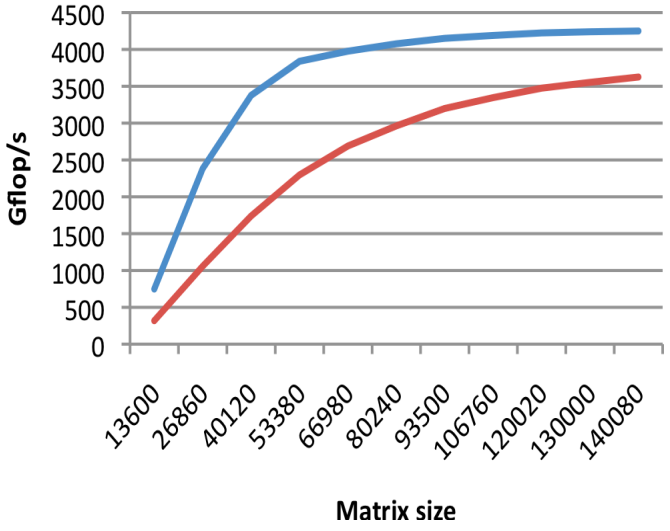
DSBP = Distributed Square Block Packed

81 nodes
Dual socket nodes
Quad core Xeon L5420
Total 648 cores at 2.5 GHz
ConnectX InfiniBand DDR 4x



LU

HPL
DAGuE
ScaLAPACK

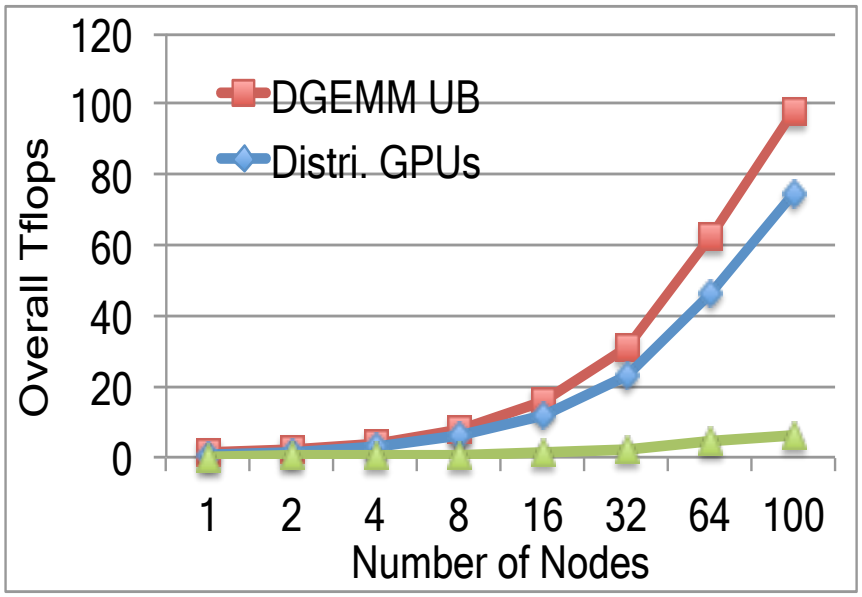


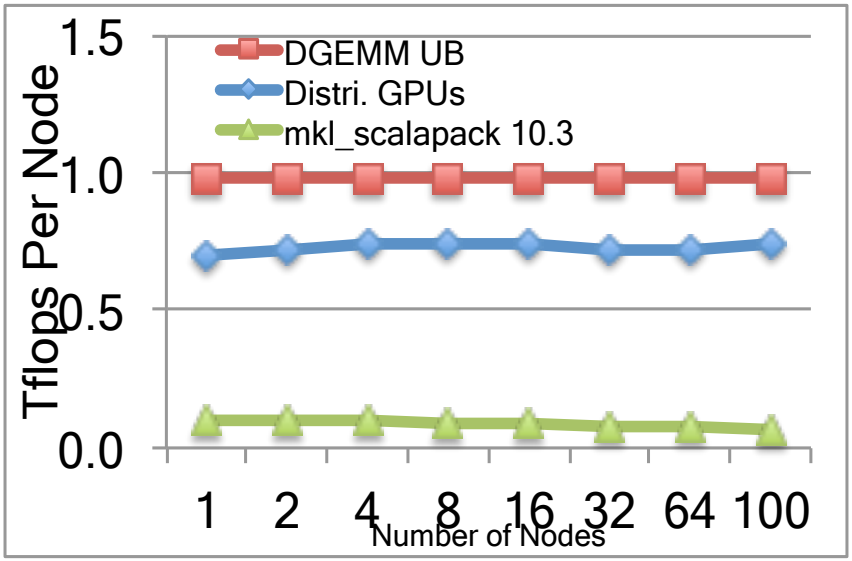
QR

DAGuE
ScaLAPACK

Clusters with GPUs (Cholesky)

Use 12 cores and 3 GPUs per node
Input size = 34560*sqrt(NumberNodes)





On the Keeneland system:

100 nodes

Each node has two 6-core Intel Westmere CPUs and three Nvidia Fermi GPUs

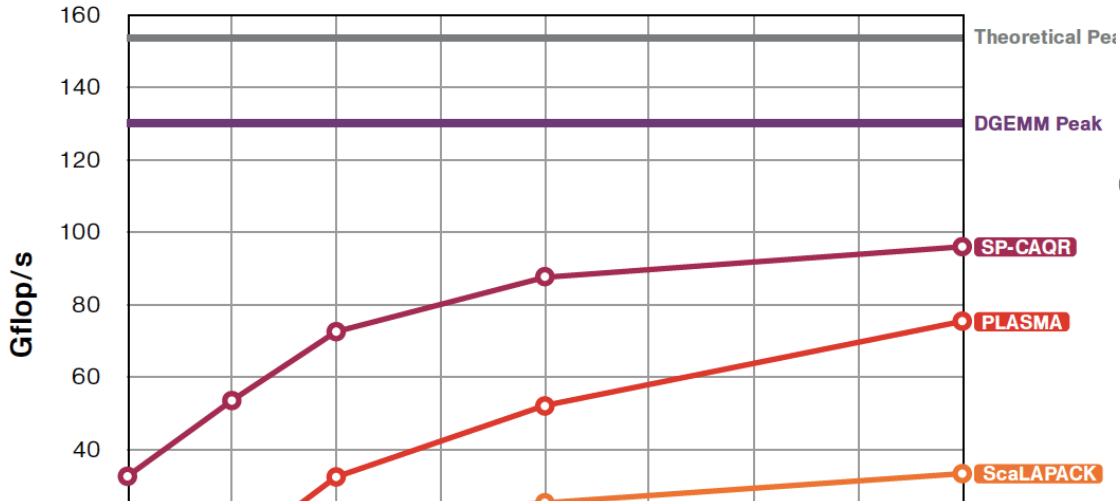
SW used: Intel MKL 10.3.5, CUDA 4.0, OpenMPI 1.5.1, PLASMA 2.4.1

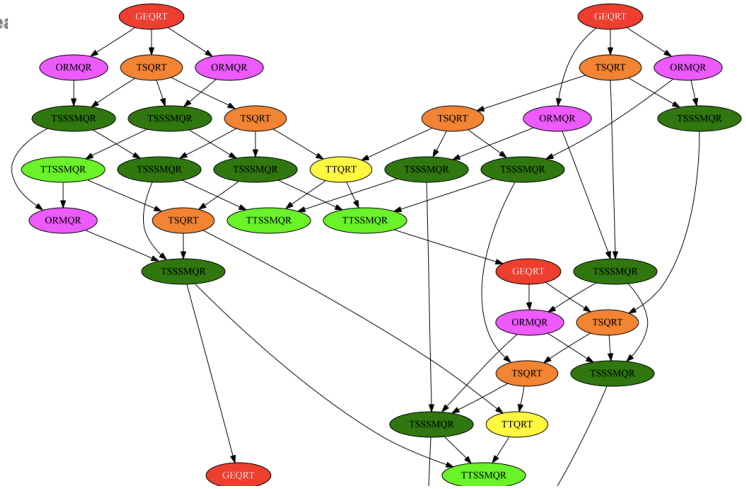
Communication Avoiding Algorithms

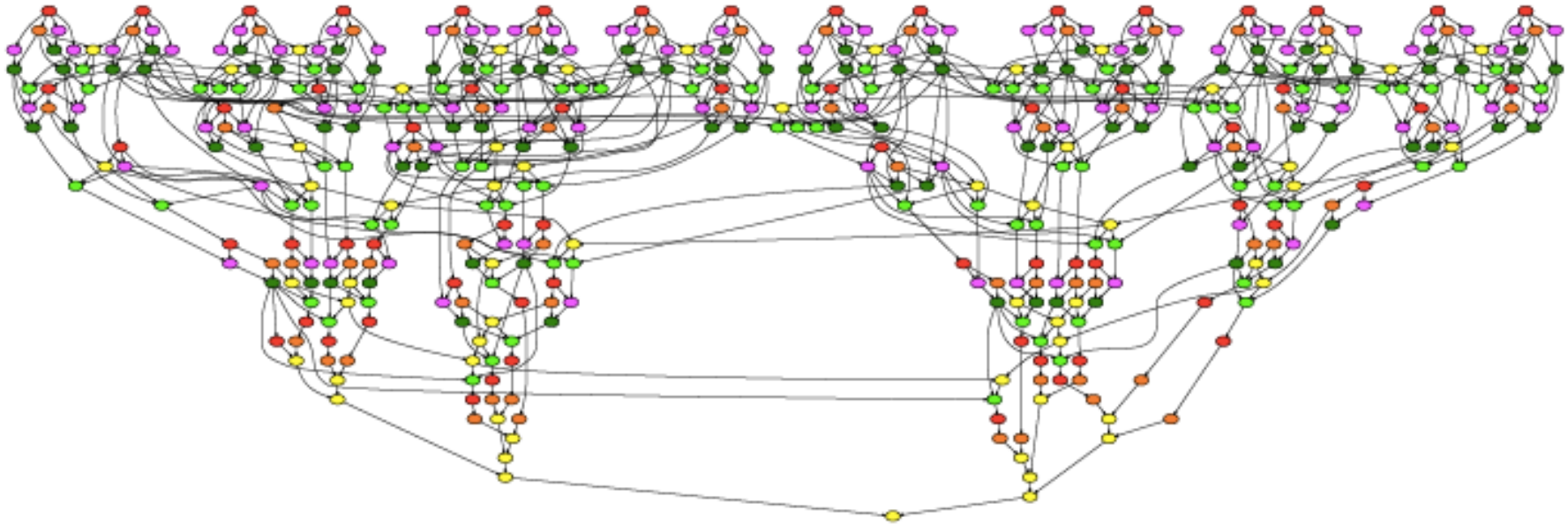
- Goal: Algorithms that communicate as little as possible
- Jim Demmel and company have been working on algorithms that obtain a provable minimum communication. (M. Anderson yesterday)
- Direct methods (BLAS, LU, QR, SVD, other decompositions)
  - Communication lower bounds for *all* these problems
  - Algorithms that attain them (*all* dense linear algebra, some sparse)
- Iterative methods - Krylov subspace methods for Ax=b, Ax=λx
  - Communication lower bounds, and algorithms that attain them (depending on sparsity structure)
- For QR Factorization they can show:

	Lower bound
# flops	$\Theta(mn^2)$
# words	$\Theta(\frac{mn^2}{\sqrt{W}})$
# messages	$\Theta(\frac{mn^2}{W^{3/2}})$

Communication Reducing QR Factorization







Mixed Precision Methods

- Mixed precision, use the lowest precision required to achieve a given accuracy outcome
  - Improves runtime, reduce power consumption, lower data movement
  - Reformulate to find correction to solution, rather than solution; Δx rather than x.

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

$$x_{i+1} - x_i = -\frac{f(x_i)}{f'(x_i)}$$

Idea Goes Something Like This…

- Exploit 32 bit floating point as much as possible.
  - Especially for the bulk of the computation
- Correct or update the solution with selective use of 64 bit floating point to provide a refined results
- Intuitively:
  - Compute a 32 bit result,
  - Calculate a correction to 32 bit result using selected higher precision and,
  - Perform the update of the 32 bit results with the correction using high precision.

Mixed-Precision Iterative Refinement

- Iterative refinement for dense systems, $Ax = b$, can work this way.

L U = lu(A)	$O(n^3)$
x = L\(U\b)	$O(n^2)$
r = b – Ax	$O(n^2)$
WHILE || r || not small enough	
z = L\(U\r)	$O(n^2)$
x = x + z	$O(n^1)$
r = b – Ax	$O(n^2)$
END	

  - Wilkinson, Moler, Stewart, & Higham provide error bound for SP fl pt results when using DP fl pt.

Mixed-Precision Iterative Refinement

- Iterative refinement for dense systems, $Ax = b$, can work this way.

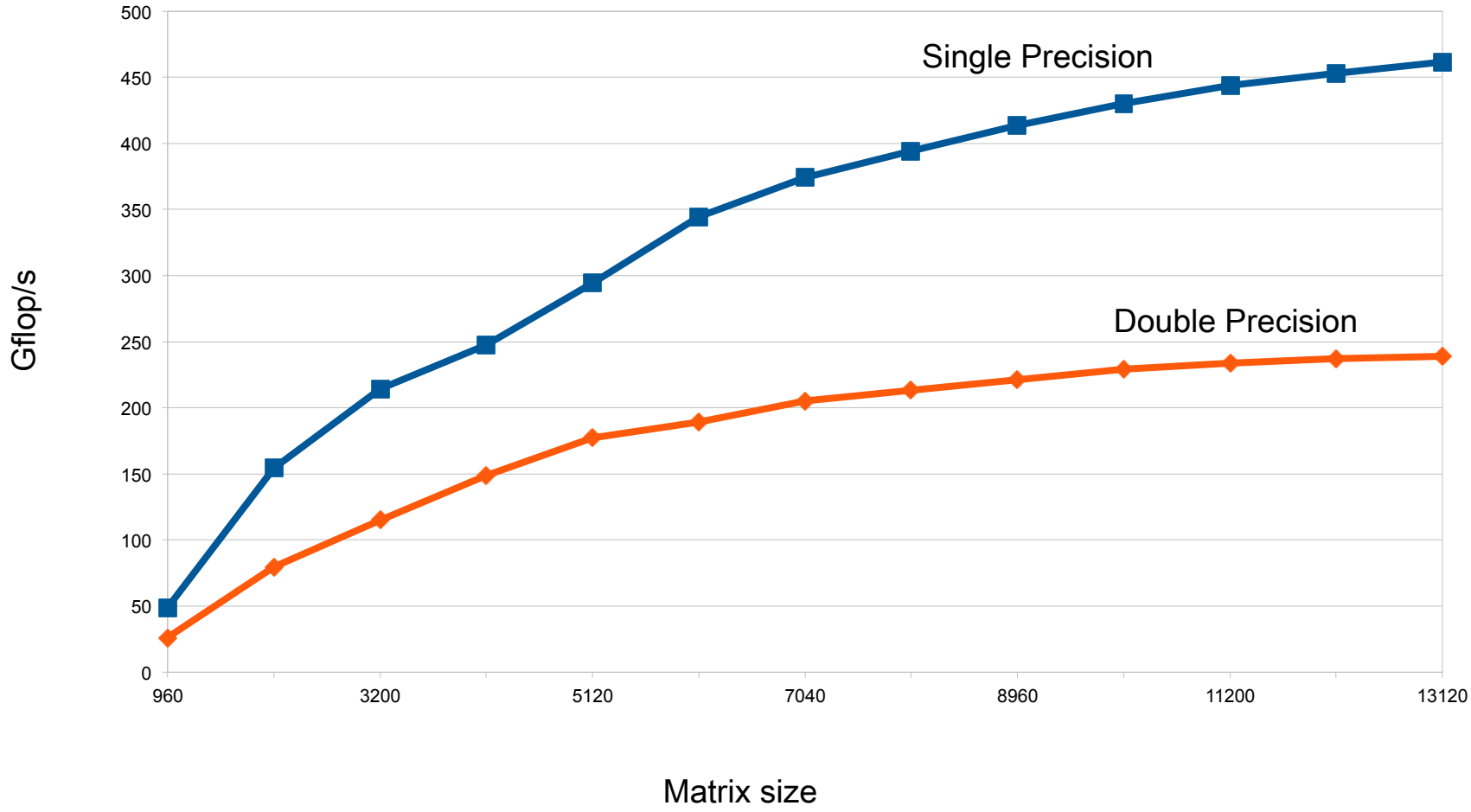
L U = lu(A)	SINGLE	$O(n^3)$
x = L\(U\b)	SINGLE	$O(n^2)$
r = b – Ax	DOUBLE	$O(n^2)$
WHILE || r || not small enough		
z = L\(U\r)	SINGLE	$O(n^2)$
x = x + z	DOUBLE	$O(n^1)$
r = b – Ax	DOUBLE	$O(n^2)$
END		

- Wilkinson, Moler, Stewart, & Higham provide error bound for SP fl pt results when using DP fl pt.
- It can be shown that using this approach we can compute the solution to 64-bit floating point precision.

- Requires extra storage, total is 1.5 times normal;
- $O(n^3)$ work is done in lower precision
- $O(n^2)$ work is done in high precision
- Problems if the matrix is ill-conditioned in sp; $O(10^8)$

Ax = b

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

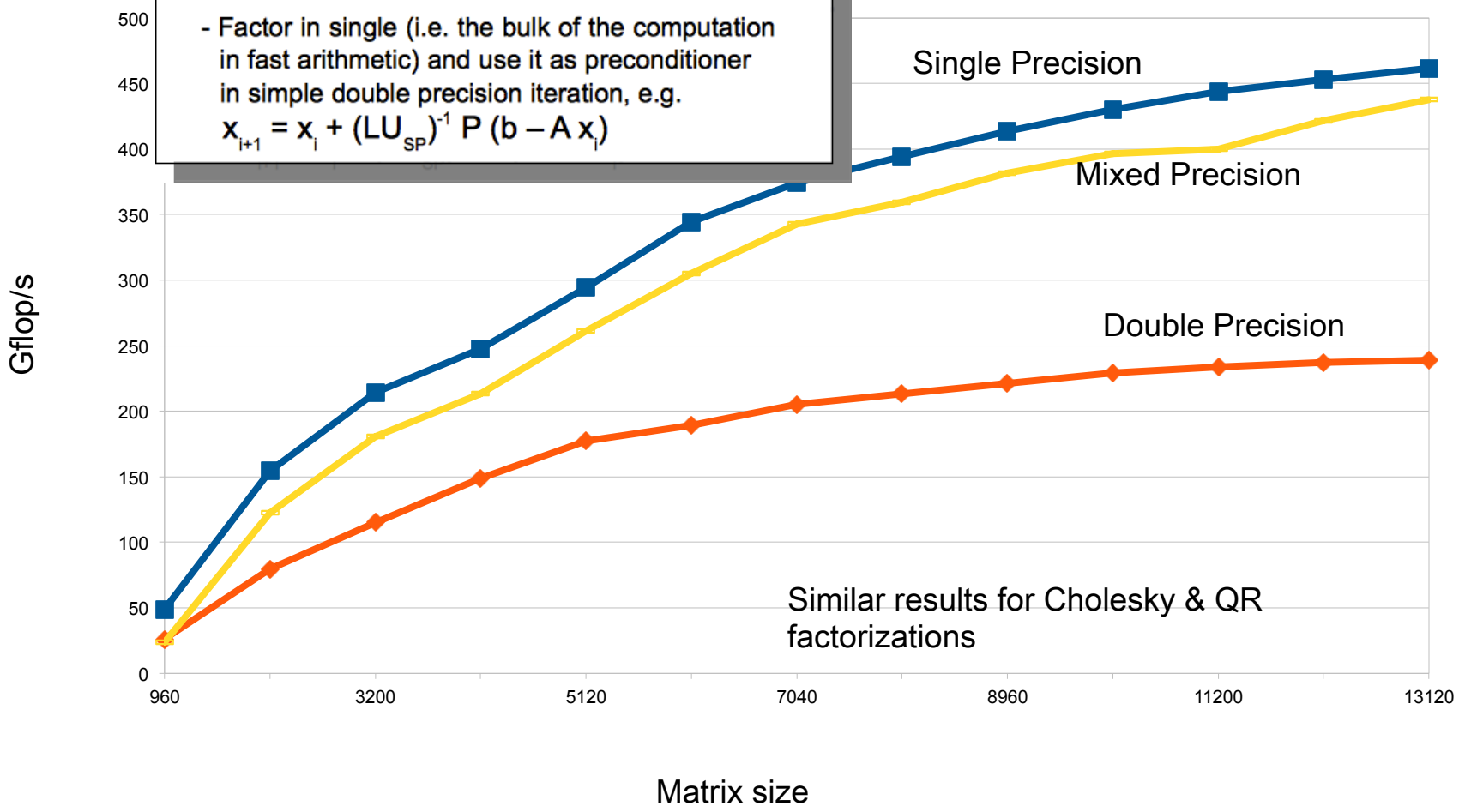


Ax = b

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.

$$x_{i+1} = x_i + (LU_{SP})^{-1} P (b - A x_i)$$



Reproducibility

- For example $\sum x_i$ when done in parallel can't guarantee the order of operations.
- Lack of reproducibility due to floating point nonassociativity and algorithmic adaptivity (including autotuning) in efficient production mode
- Bit-level reproducibility may be unnecessarily expensive most of the time
- Force routine adoption of uncertainty quantification
  - Given the many unresolvable uncertainties in program inputs, bound the error in the outputs in terms of errors in the inputs

A Call to Action: Exascale is a Global Challenge

- Hardware has changed dramatically while software ecosystem has remained stagnant
- Community codes unprepared for sea change in architectures
- No global evaluation of key missing components
- The IESP was Formed in 2008
- Goal to engage international computer science community to address common software challenges for Exascale
- Focus on open source systems software that would enable multiple platforms
- Shared risk and investment
- Leverage international talent base

International Exascale Software Program

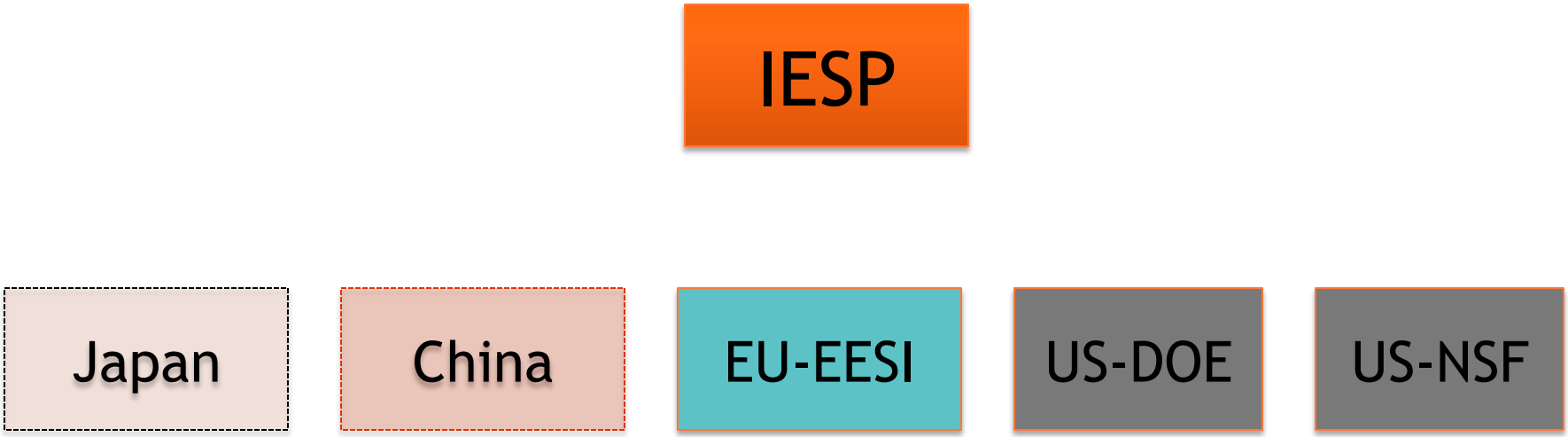
Improve the world's simulation and modeling capability by improving the coordination and development of the HPC software environment

Workshops:

Build an international plan for coordinating research for the next generation open source software for scientific high-performance computing

www.exascale.org

Example Organizational Structure: Incubation Period (today):



- **IESP provides coordination internationally, while regional groups have well managed R&D plans and milestones**

Conclusions

- For the last decade or more, the research investment strategy has been overwhelmingly biased in favor of hardware.
- This strategy needs to be rebalanced - barriers to progress are increasingly on the software side.
- High Performance Ecosystem out of balance
  - Hardware, OS, Compilers, Software, Algorithms, Applications
    - No Moore's Law for software, algorithms and applications

- Our community is needed and has a great deal to offer and contribute.

Published in the January 2011 issue of The International Journal of High Performance Computing Applications

"We can only see a short distance ahead, but we can see plenty there that needs to be done."

- *Alan Turing (1912 – 1954)*

- www.exascale.org