

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

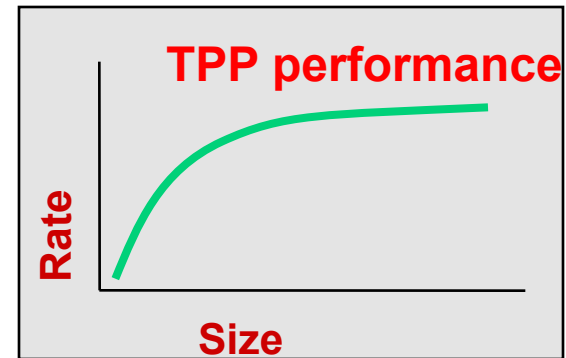
University of Tennessee
Oak Ridge National Laboratory
University of Manchester

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

$$Ax=b, \text{ dense problem}$$

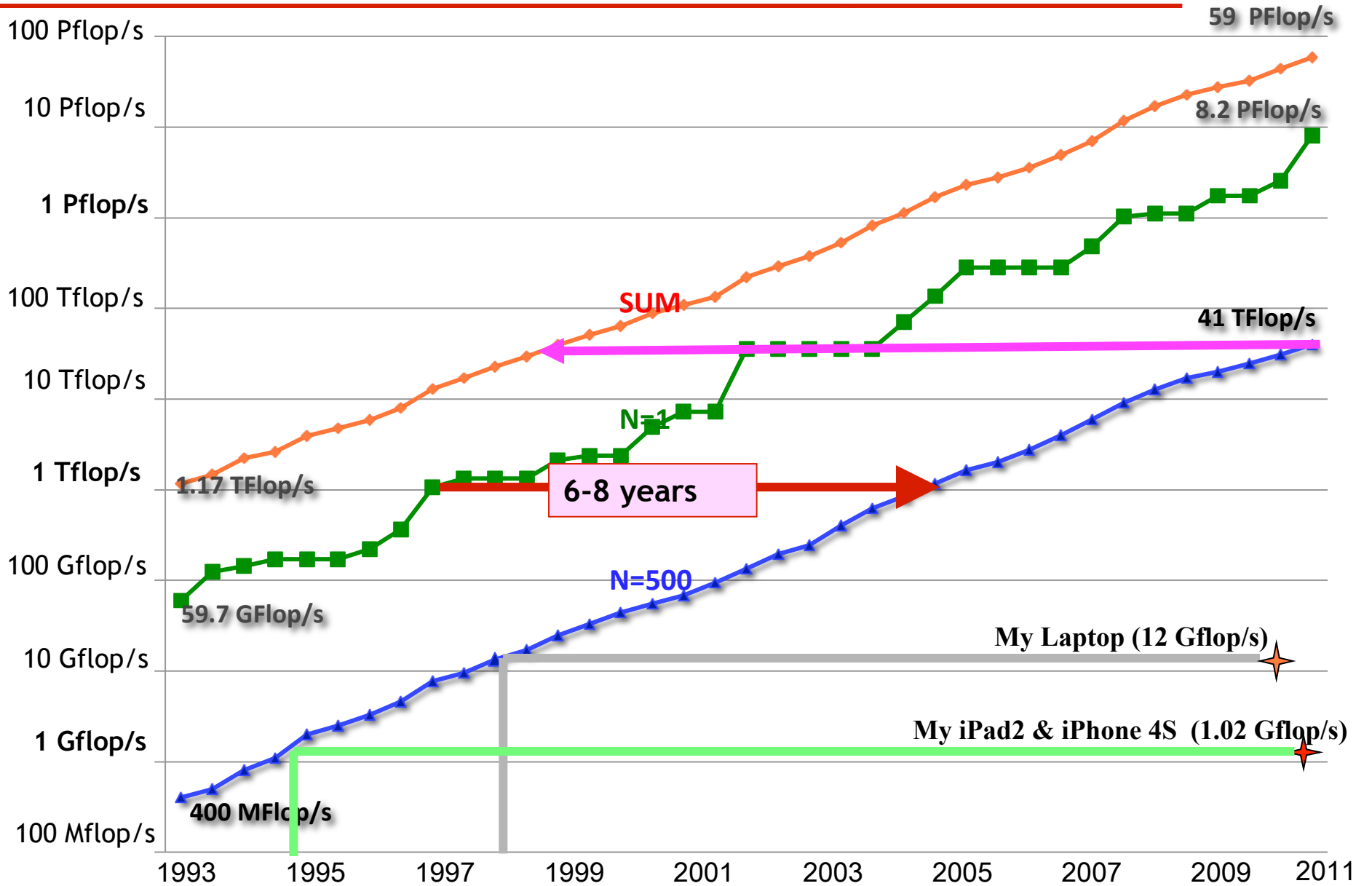


- Updated twice a year
SC'xy in the States in November
Meeting in Germany in June

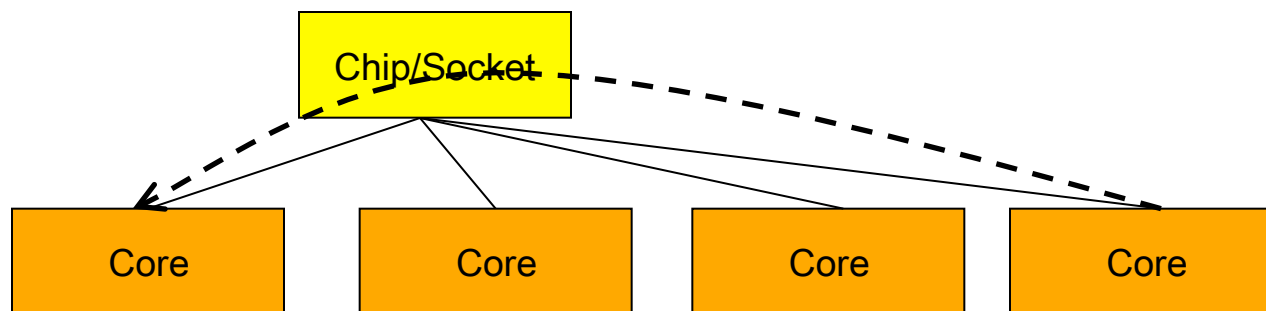
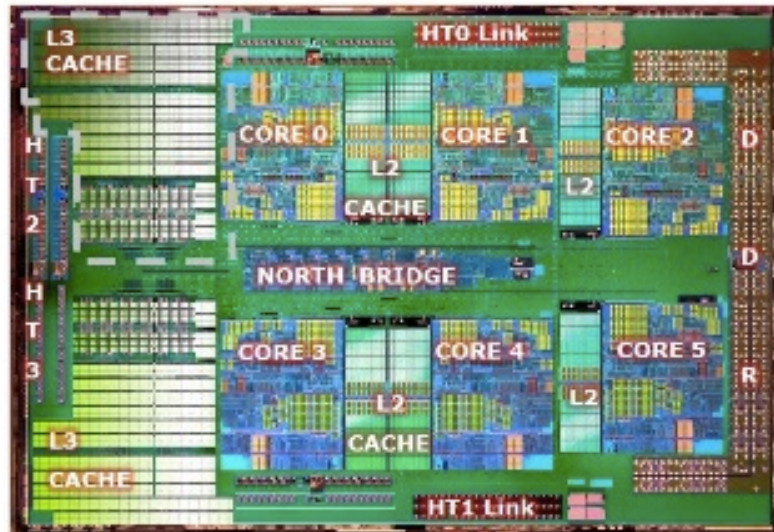
- 2 - All data available from www.top500.org



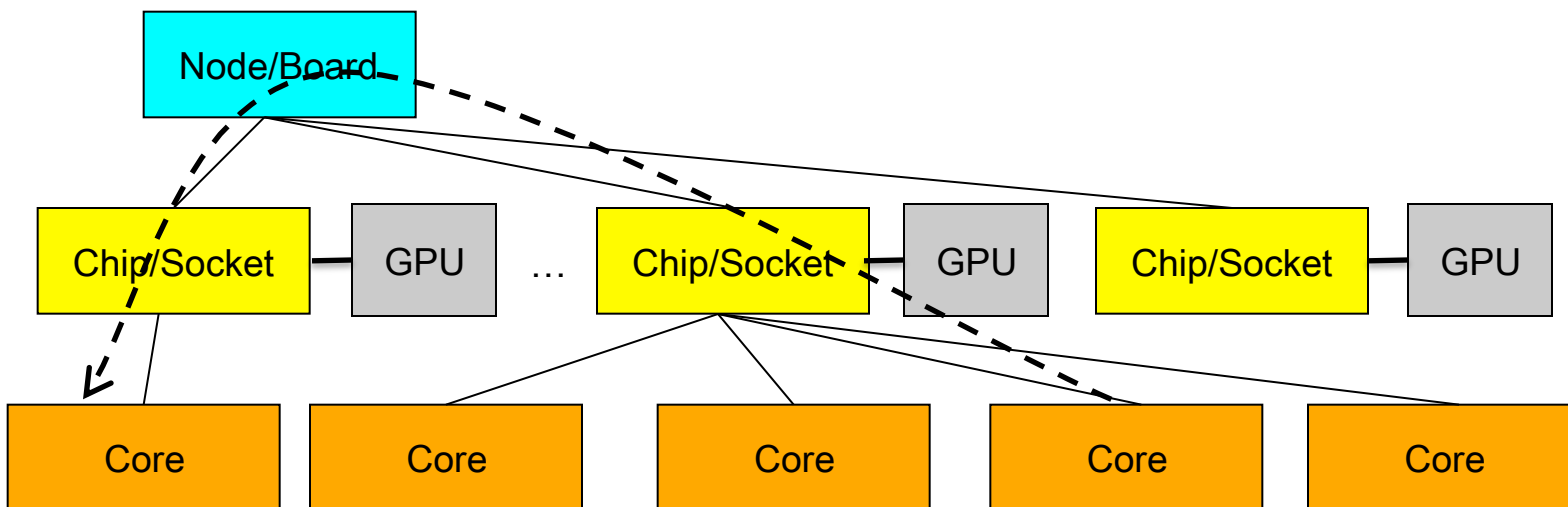
Performance Development



Example of typical parallel machine

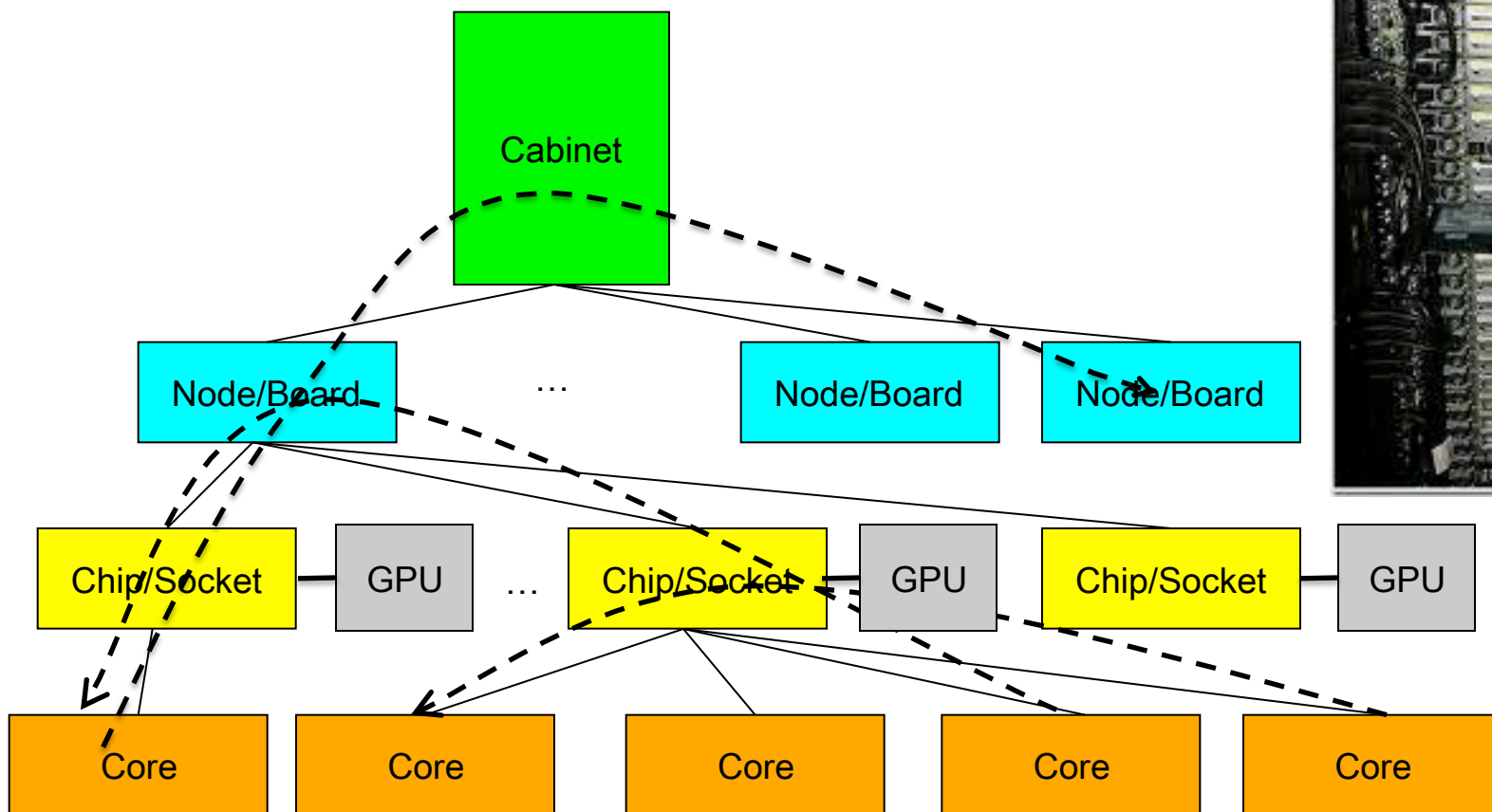


Example of typical parallel machine



Example of typical parallel machine

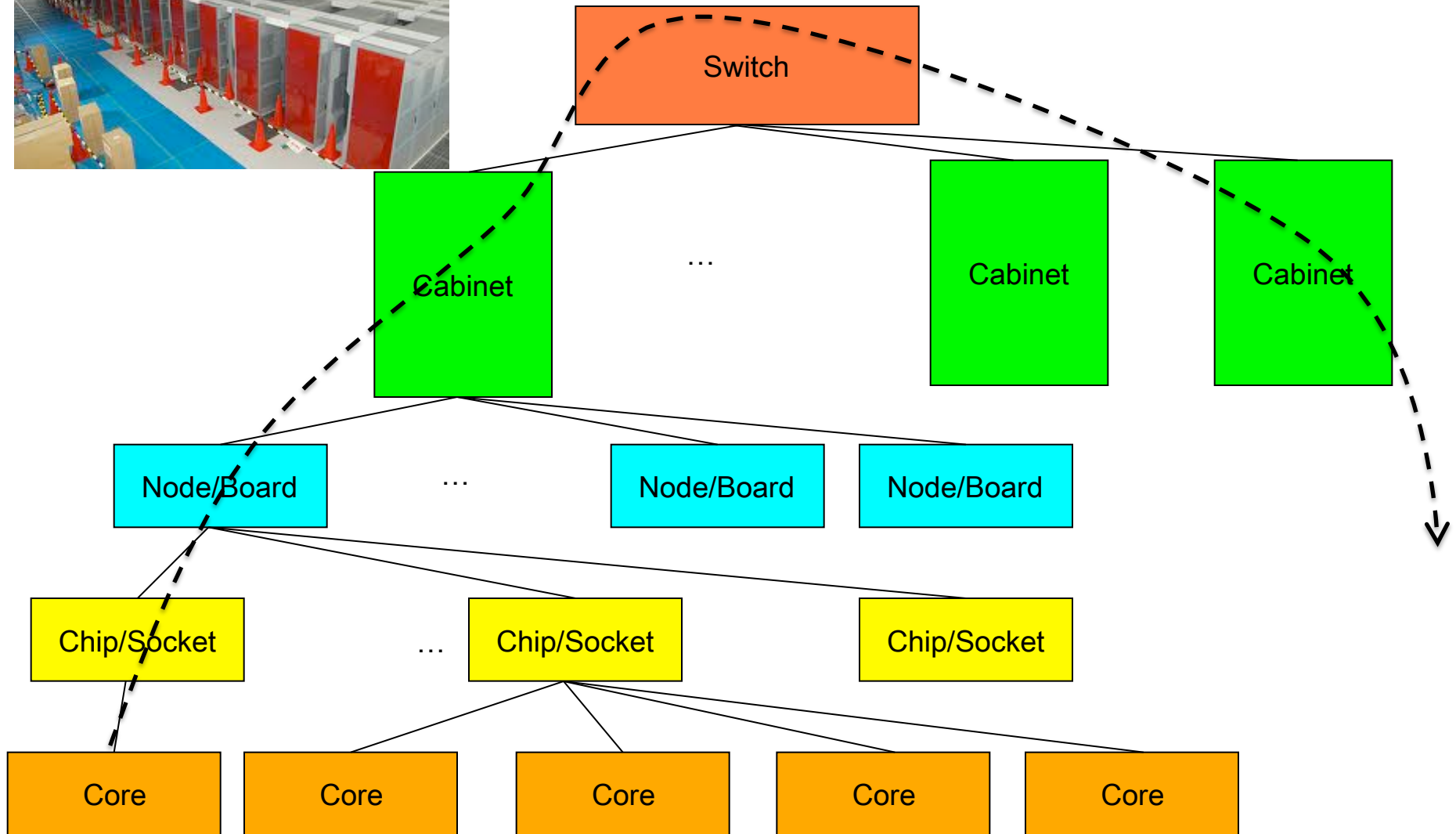
Shared memory programming between processes on a board and
a combination of shared memory and distributed memory programming
between nodes and cabinets



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 Suzhou	Tianhe-1A, NUDT Intel + Nvidia GPU + IB	China	120,640	1.27	53	2.58	493
5	GSTC Center, Tsinghua Univ of Technology	Tusname 2.0, HP Intel + Nvidia GPU + IB	China	73,728	0.19	52	1.45	85
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 Top 500 systems use GPUs?

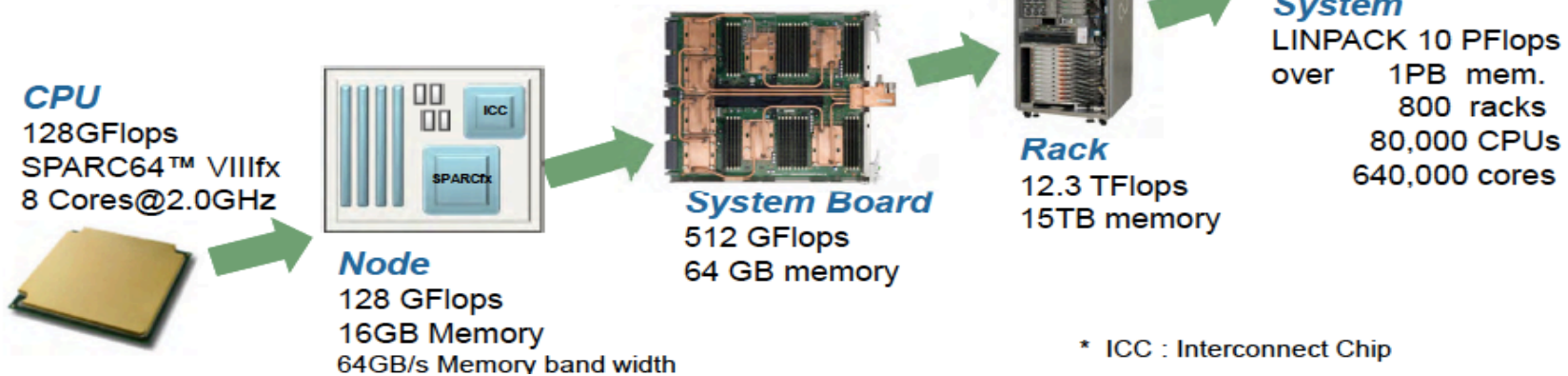
Japanese K Computer

K computer Specifications



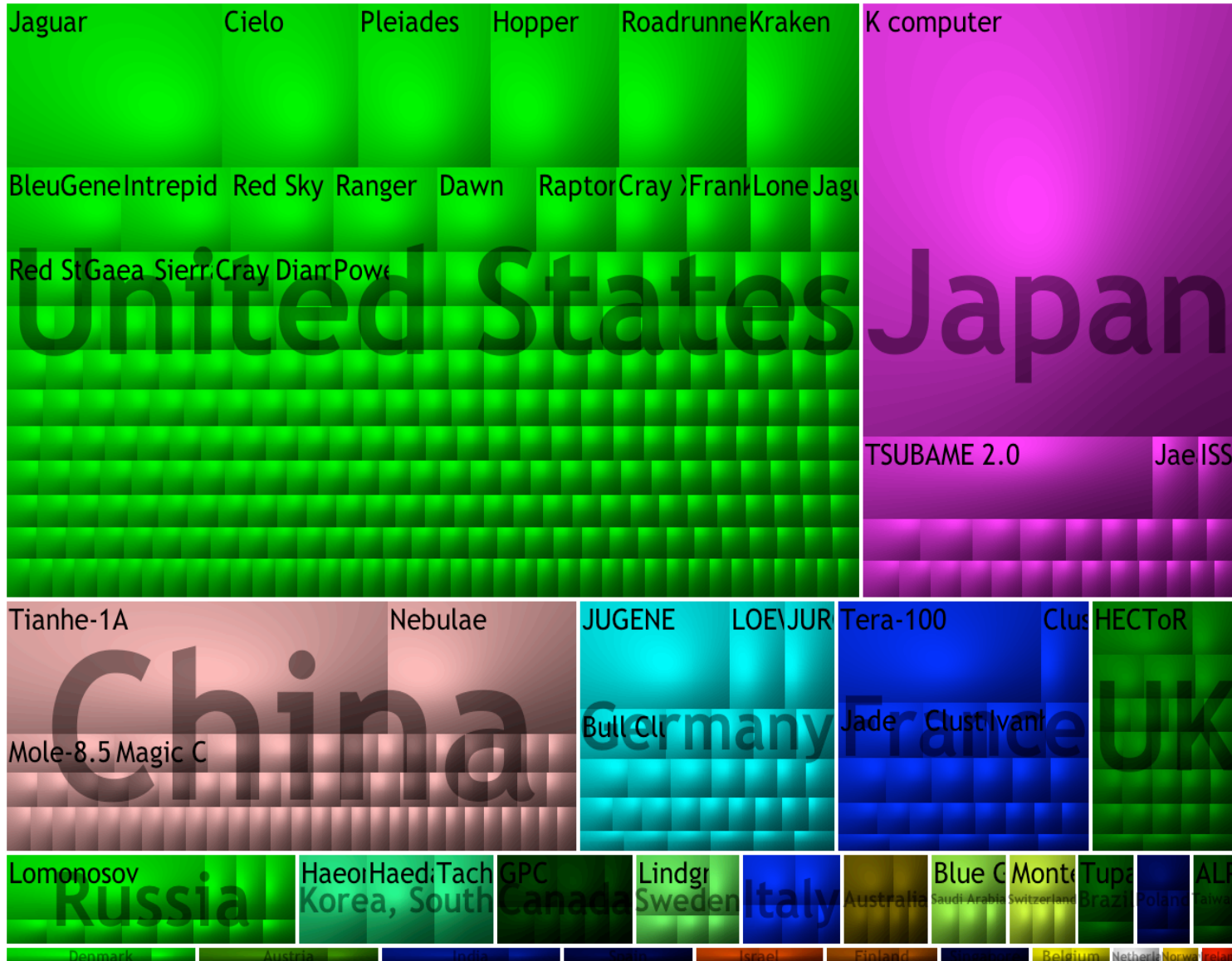
CPU (SPARC64 VIIIfx)	Cores/Node	8 cores (@2GHz)
	Performance	128GFlops
	Architecture	SPARC V9 + HPC extension
	Cache	L1(I/D) Cache : 32KB/32KB L2 Cache : 6MB
	Power	58W (typ. 30 C)
	Mem. bandwidth	64GB/s.
Node	Configuration	1 CPU / Node
	Memory capacity	16GB (2GB/core)
System board(SB)	No. of nodes	4 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
Cooling	Architecture	Routing chip structure (no outside switch box)
	CPU, ICC*	Direct water cooling
	Other parts	Air cooling





Countries Share



Absolute Counts

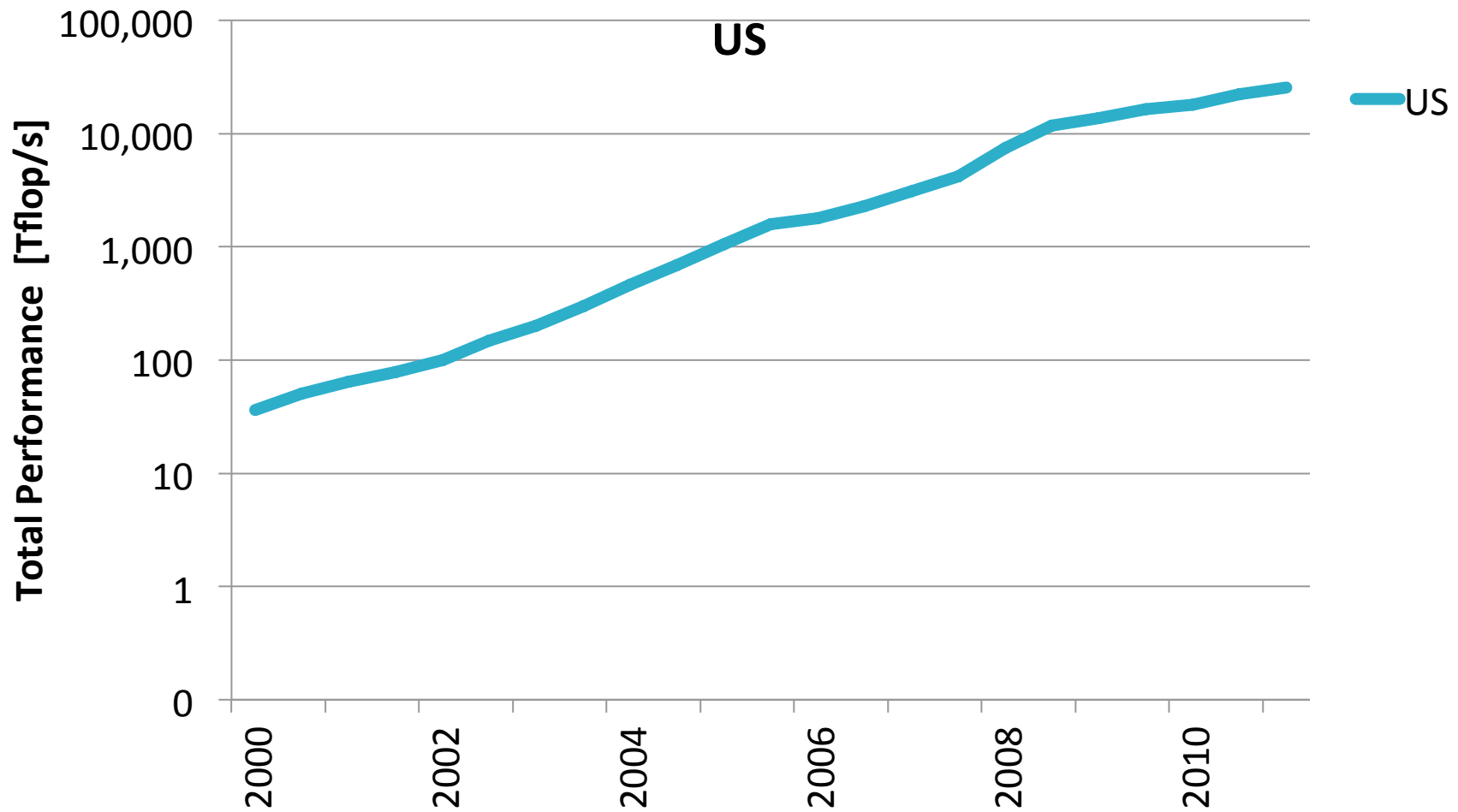
US:	251
China:	64
Germany:	31
UK:	28
Japan:	26
France:	25



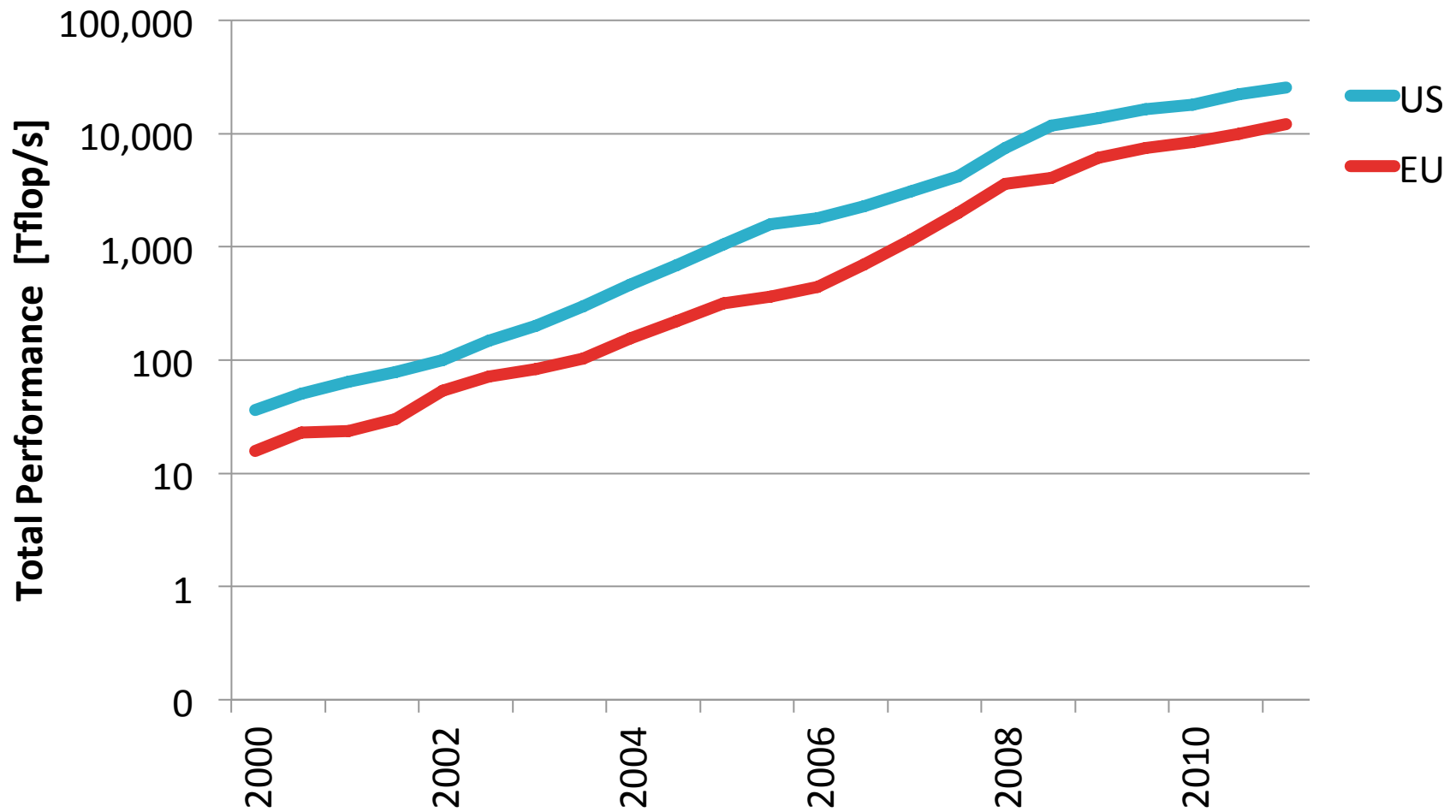
Top500 Computers in Brazil

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

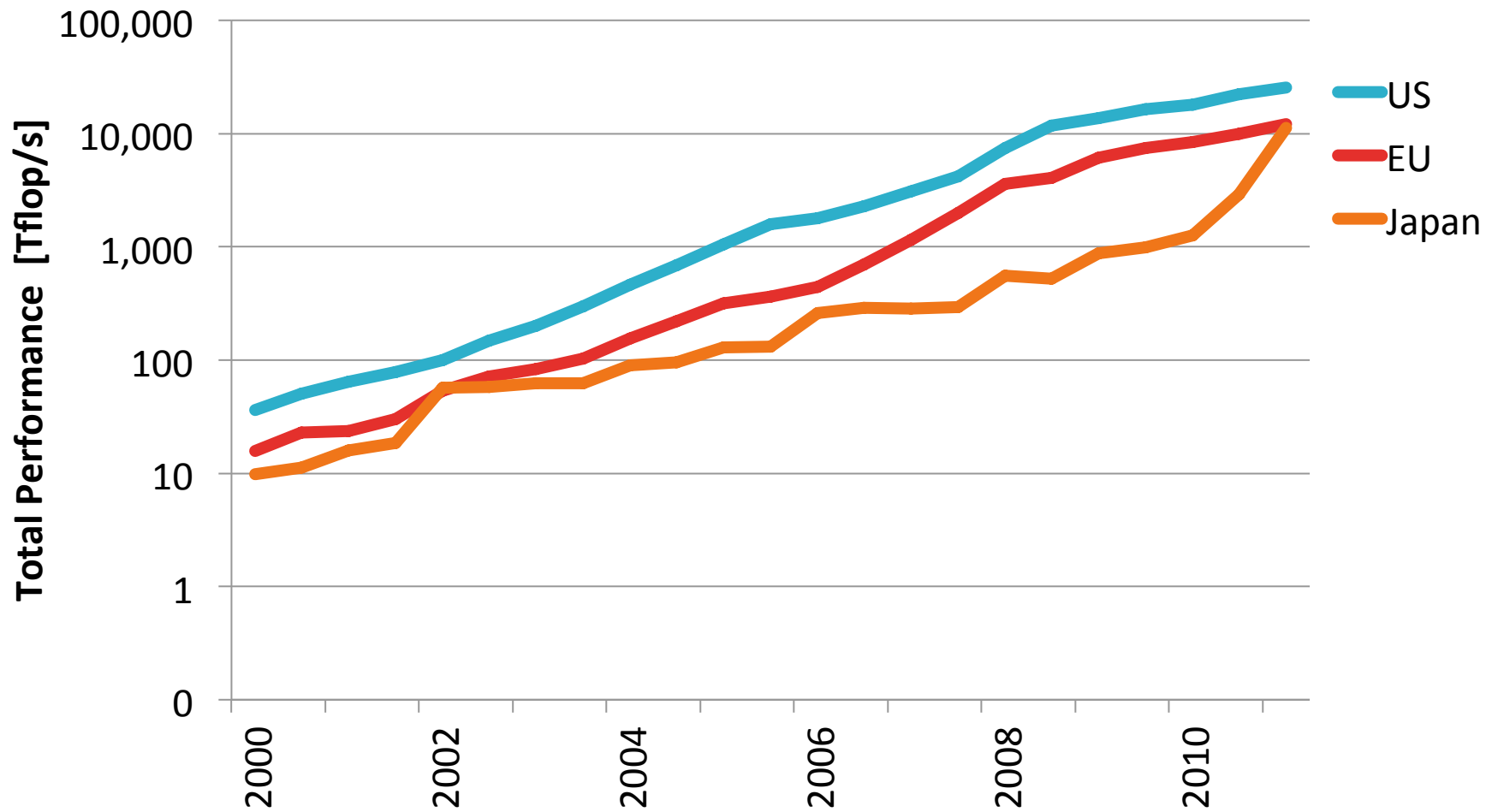
Performance by Regions



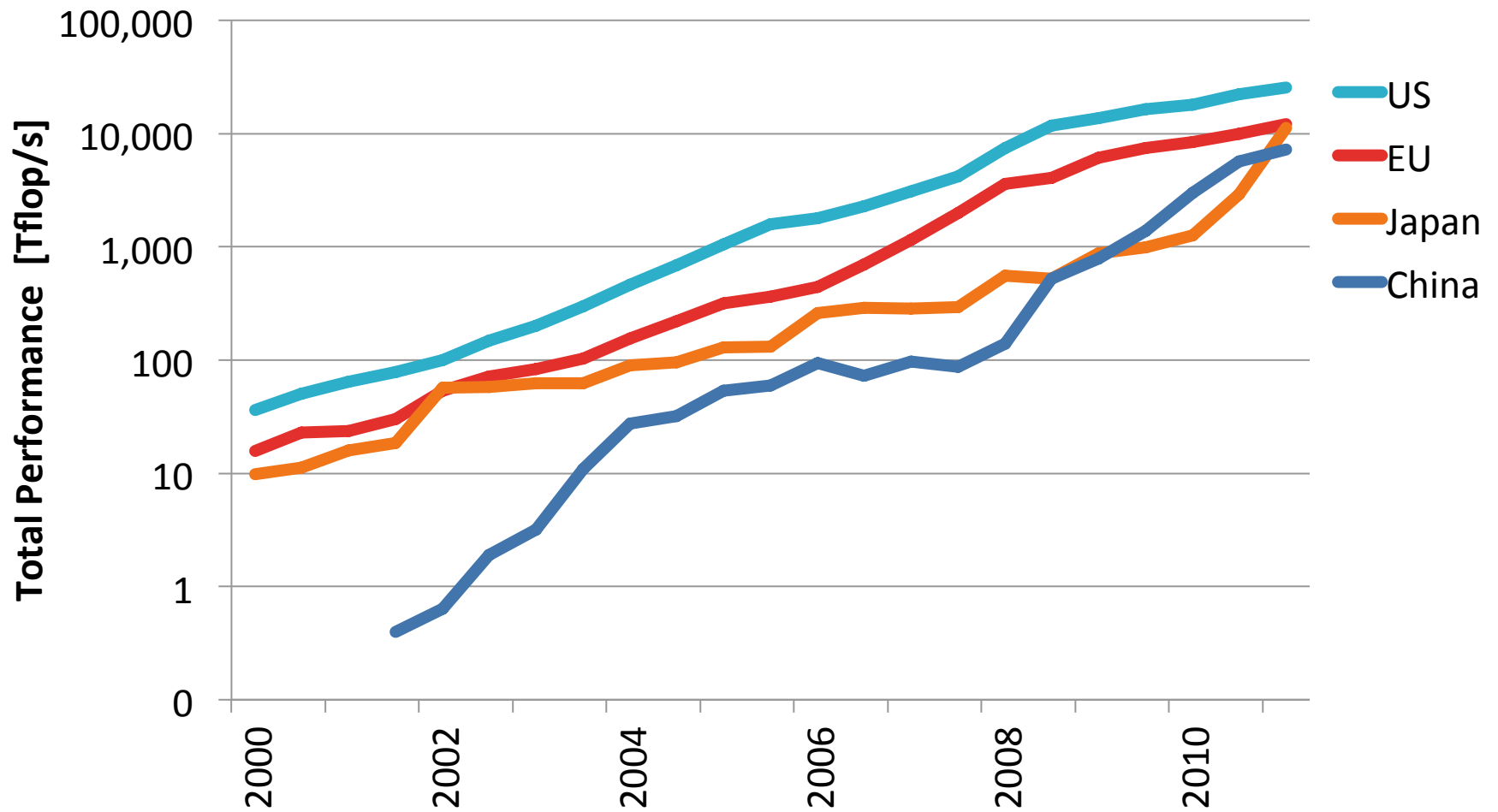
Performance by Regions



Performance by Regions



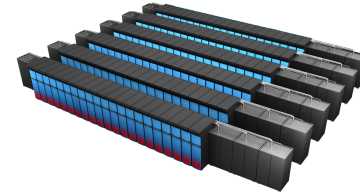
Performance by Regions





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)

Quiz: How Many of the

Top 500 systems use GPUs?

Intel Xeon
2 cores
3 GHz

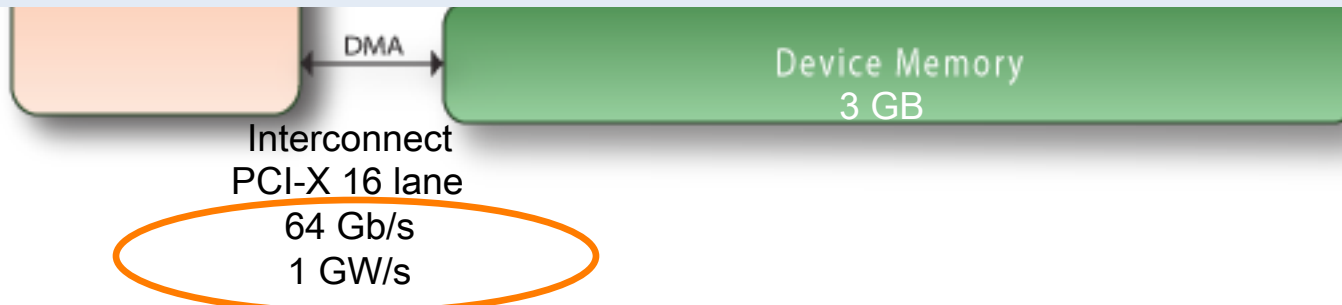
Nvidia C2050 "Fermi"
448 CUDA cores
1.15 GHz

8*4 ops/cycle
96 Gflops/DP

448 ops/cycle
115 Gflops/DP

Answer:

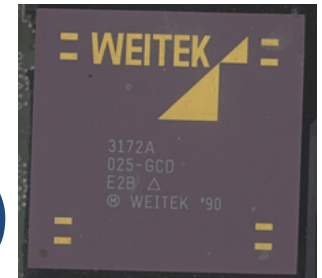
Today only 19 systems on the TOP500 use GPUs





We Have Seen This Before

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



1976

THREE HUNDRED FORTY ONE MILLION FLOATING POINT OPERATIONS PER SECOND. THE FPS-164/MAX.

Rapid scientific and engineering problems increasingly call for super-computing machines and higher technical skills to solve them in very large numbers. The small size, cost of a super-computer with the speed and accuracy of a mainframe machine has become a reality for many.

Now, there's the FPS-164/MAX — a special purpose, modular supercomputer that makes the most of CPU, CYBER and other microprocessor technologies available in a fraction of the cost.

The FPS-164/MAX is fast. 3000 point operations a second. From 10 to 100 million floating point operations per second, depending on configuration, scaling to 700 million if 48 bit accuracy is available. In the cost that other FPS-164/MAX gives you at the speed and accuracy you need to solve those really computationally intensive problems.

The FPS-164/MAX configuration is able to incorporate up to 24 vector channels at one time, allowing a fully configured FPS-164/MAX to factor a 100 by 1000 matrix in about 2 seconds, complete two 1000 by 10,000 matrices in two days.

The FPS-164/MAX is powerful. A parallel pipelined processor designed to run 100MHz or higher speed, the FPS-164/MAX has all the vector capability of our original FPS-164. We've just added a lot more power with super-special processing units which amplify the vector processing capability of the original FPS-164 by up to 10 times.

The FPS-164/MAX is cost-effective. Its modular architecture, optional channels, and channels, built from open, microprocessor modules, is any application requiring the handling of large matrices, the FPS-164/MAX offers unparalleled cost efficiency. In fact, it's the only supercomputer that can be built or broken down supercomputer cost saving up to 50%.

Whether you're looking to upgrade your existing FPS-164 — or searching for a completely new system — you need the supercomputer performance for one million dollars or less.

Many more. The FPS-164/MAX is designed for the demanding requirements of Floating Point Systems, 48-bit IEEE service offers world-wide, full vector diagnostic capabilities, and a wealth of product quality and reliability options to make you can use the FPS-164/MAX with up, modified, and ready to solve your problem solving needs.

For complete information and applications, call toll free 1-800-567-8143.

FPS-164/MAX Specifications

Point Operations (100MHz)	3000
Number of Channels	24
Number of Vector Registers	60
Number of Floating Point Processors	24
Vector Register Capacity	20 x 32 bit
Maximum Capacity	1.0 Gbytes
Word Size	32 bit
Number of Channels	24
Number of Vector Registers	60
Number of Floating Point Processors	24
Vector Register Capacity	20 x 32 bit
Maximum Capacity	1.0 Gbytes
Word Size	32 bit
Number of Channels	24
Number of Vector Registers	60
Number of Floating Point Processors	24
Vector Register Capacity	20 x 32 bit
Maximum Capacity	1.0 Gbytes
Word Size	32 bit
Number of Channels	24
Number of Vector Registers	60
Number of Floating Point Processors	24
Vector Register Capacity	20 x 32 bit
Maximum Capacity	1.0 Gbytes
Word Size	32 bit

floating point systems, inc.
P.O. Box 20480
Palo Alto, CA 94303
(415) 344-3573
U.S. MAILING LIST AVAILABLE

Circle Number 318 on Reader Service Card

The Intel® Math CoProcessor™ is for crunching numbers faster.

intel
Personal Computer Enhancement

There's one for every machine.

80387™ Family, for 80386™ based machines.

80287™ Family, for 80286™ based machines.

80187™ Family, for 8086™ and 8088™ based machines.

It's FAST!
The Intel Math CoProcessor dramatically speeds up the number crunching that's part of the work you do every day: budgeting, statistical analysis, financial analysis, CAD and other engineering analysis. In fact, the Math CoProcessor is supported by more than 100 commonly used software packages including Lotus 1-2-3, dBase IV, AutoCAD, and most language and statistical packages.

It's EASY!
Intel makes a variety of math coprocessors. Every PC has a built-in socket. Just plug it in and go.

It's SAFE!
Made by Intel, the same people who designed your PC's microprocessor, each and every Math CoProcessor is backed by an industry leading the way warranty and full free technical support. You are assured the highest degree of quality, compatibility, reliability and support for your investment.

For more information, or technical support call:
(800) 538-3173 in the U.S. and Canada
(510) 652-7154 for International

intel
Personal Computer Enhancement

1980



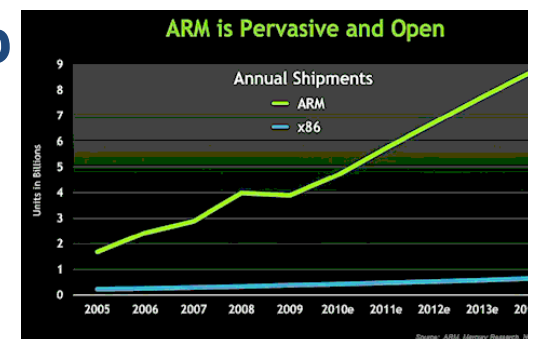
Balance Between Data Movement and Floating point

- .. **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

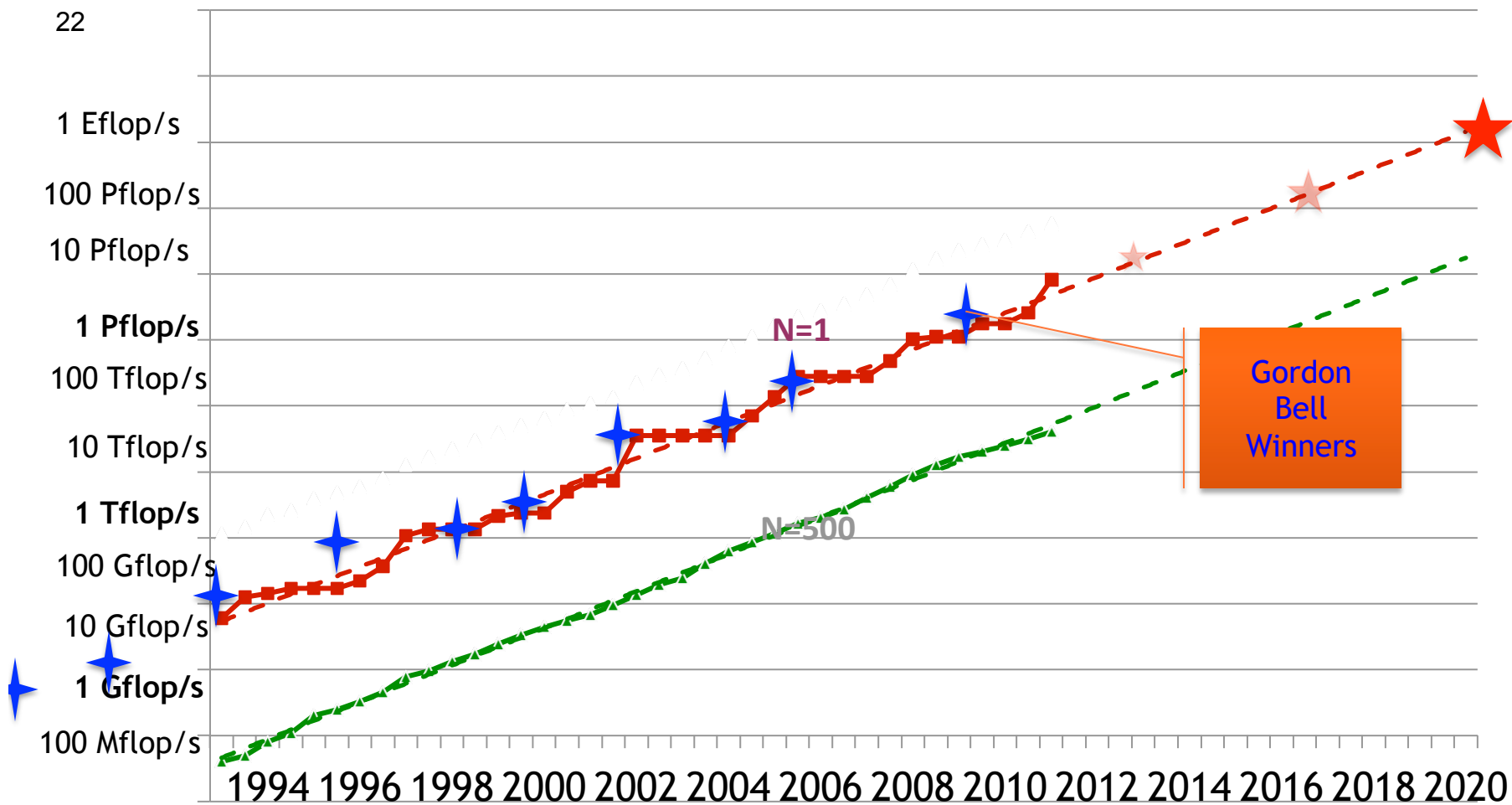


- .. Most likely be a hybrid design
 - Think standard multicore chips and accelerator (GPUs)
- .. Today accelerators are attached
- .. Next generation more integrated
- .. Intel's MIC architecture "Knights Ferry" and "Knights Corner" to come.
 - 48 x86 cores
- .. AMD's Fusion in 2012 - 2013
 - Multicore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





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

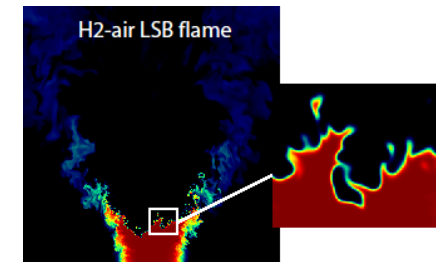
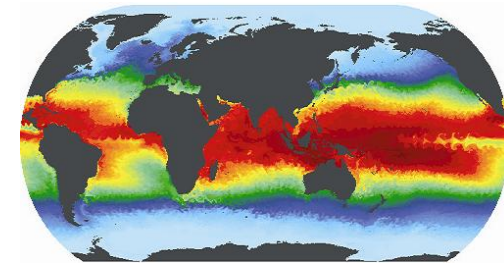
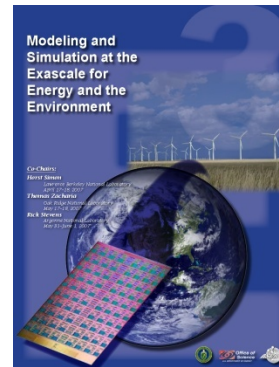
- Climate Science (11/08)
- High Energy Physics (12/08)
- Nuclear Physics (1/09)
- Fusion Energy (3/09)
- Nuclear Energy (5/09)
- Biology (8/09)
- Material Science and Chemistry (8/09)
- National Security (10/09)
- Cross-cutting technologies (2/10)

•• Exascale Steering Committee

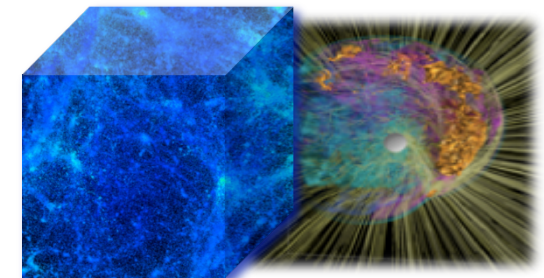
- “Denver” vendor NDA visits (8/09)
- SC09 vendor feedback meetings
- Extreme Architecture and Technology Workshop (12/09)

•• 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**

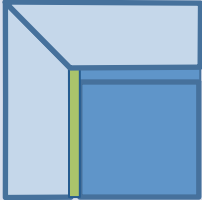
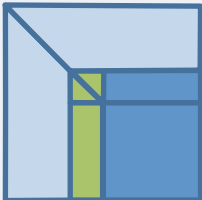
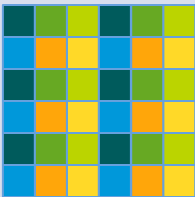


Critical Issues at Peta & Exascale for Algorithm and Software Design

- **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?

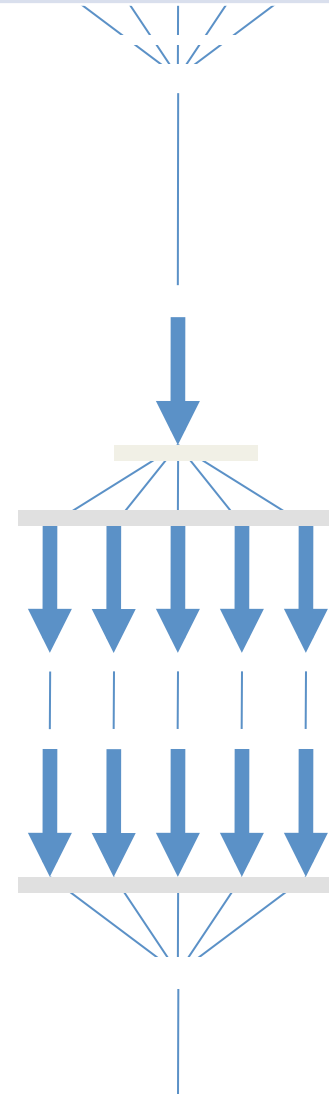
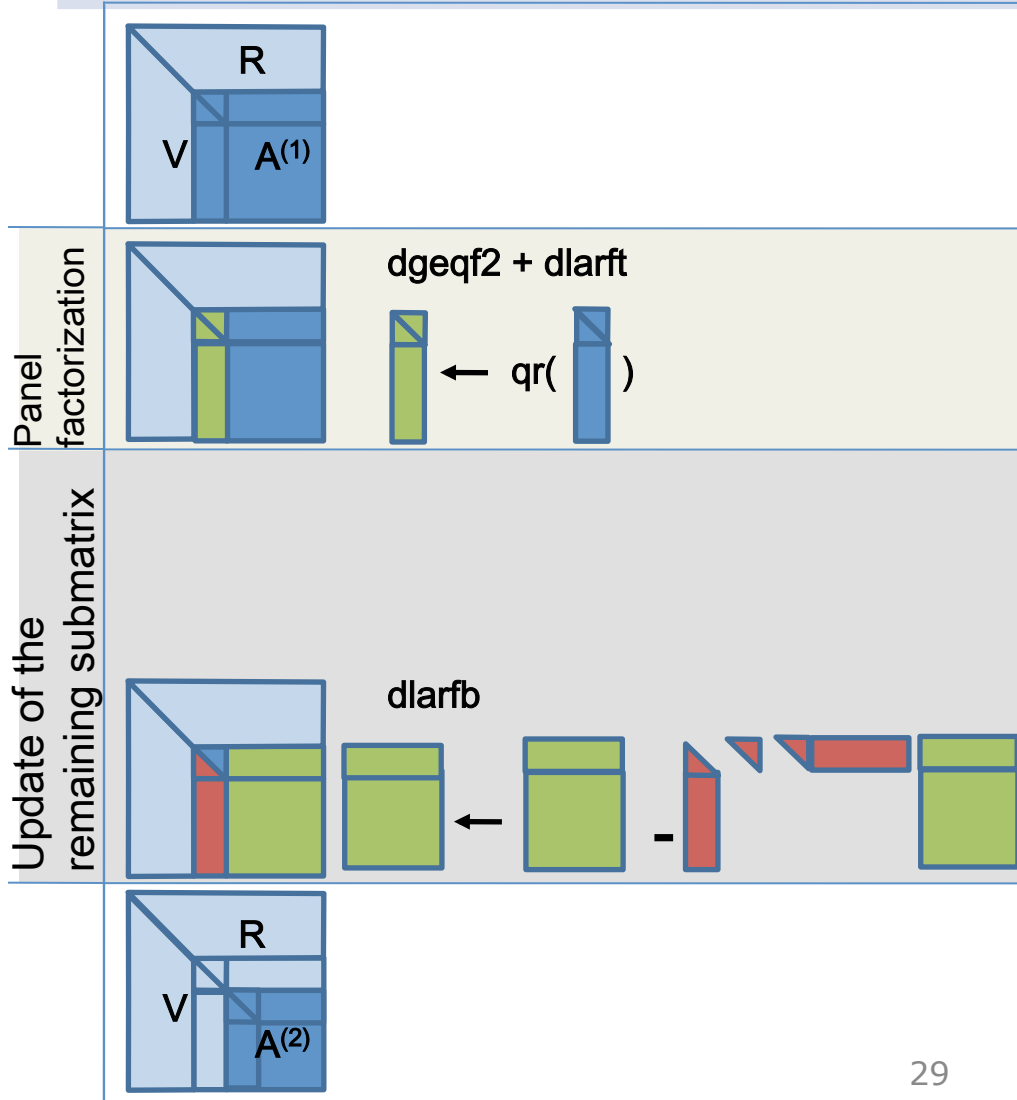
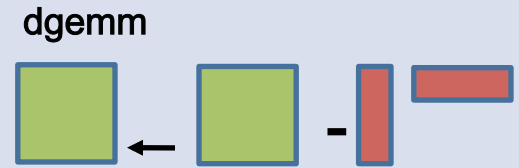
Algorithms follow hardware evolution along time.

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

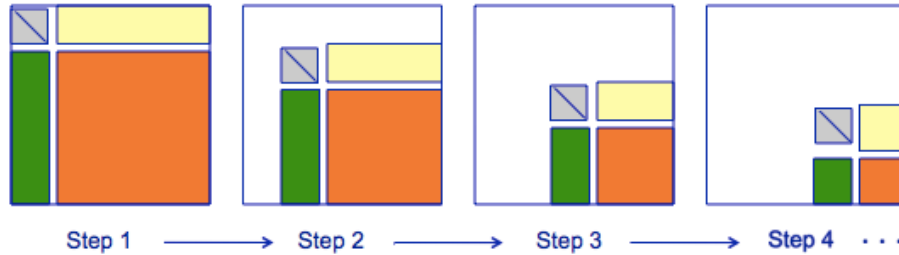
Parallelize the update:

- 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

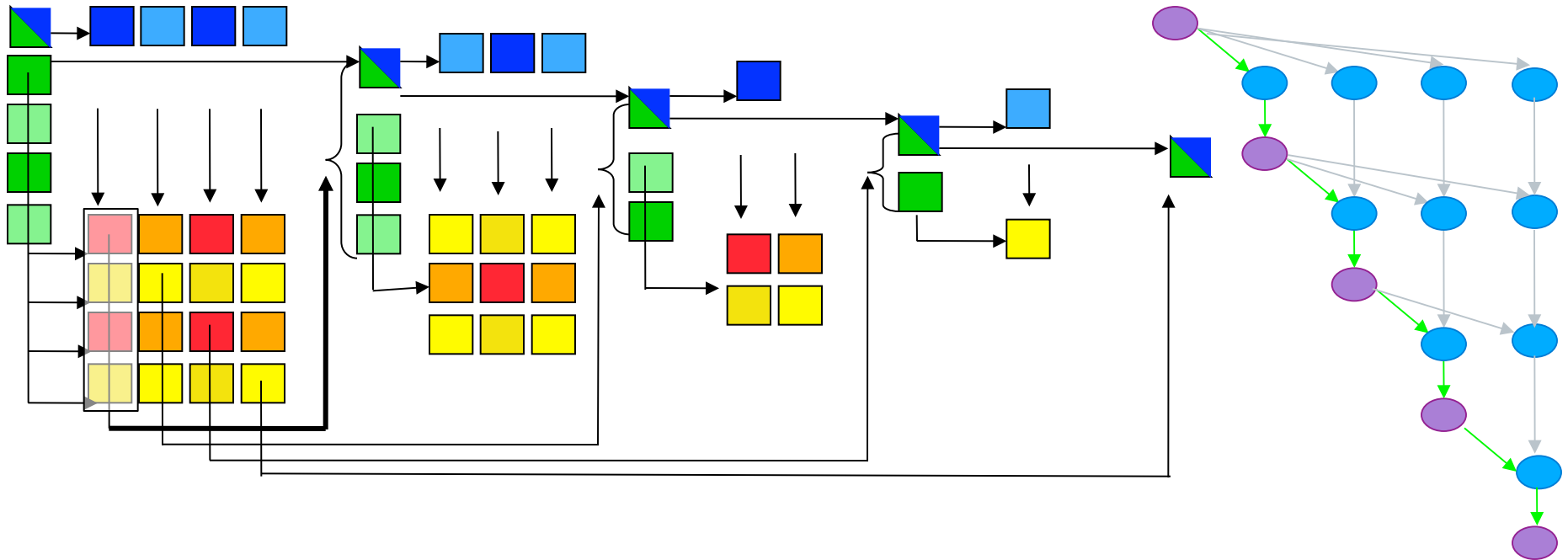


Fork - Join parallelism
Bulk Sync Processing

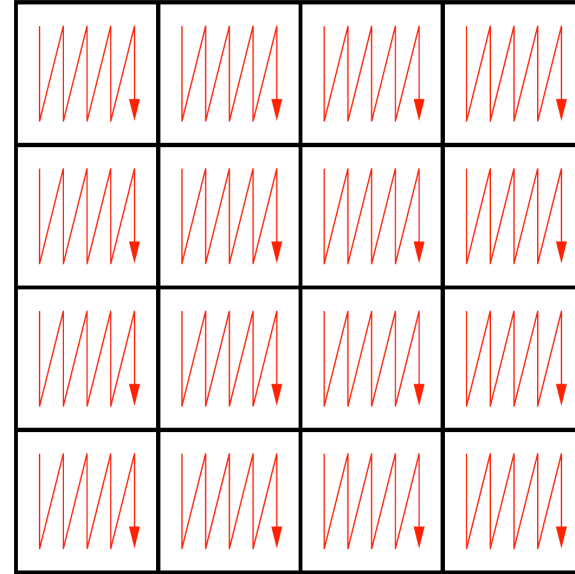
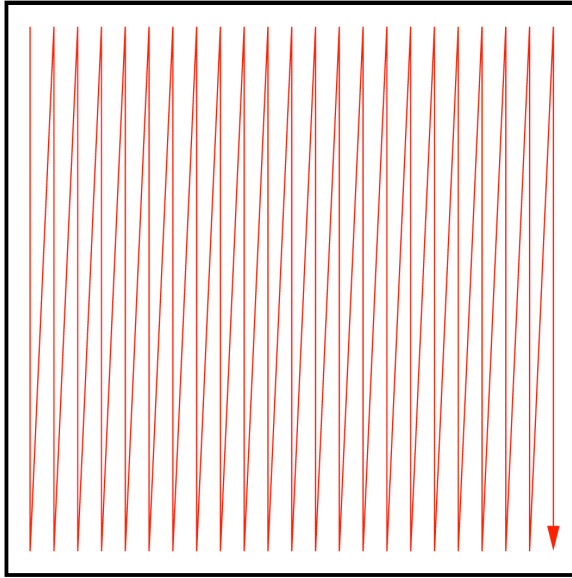
Parallel Tasks in LU/LL^T/QR



- 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

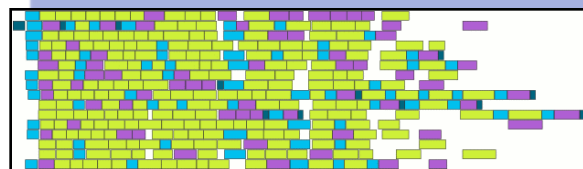
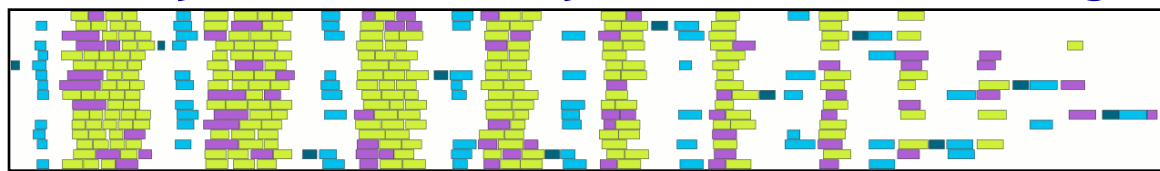
• Objectives

- High utilization of each core
- Scaling to large number of cores
- Shared or distributed memory

• Methodology

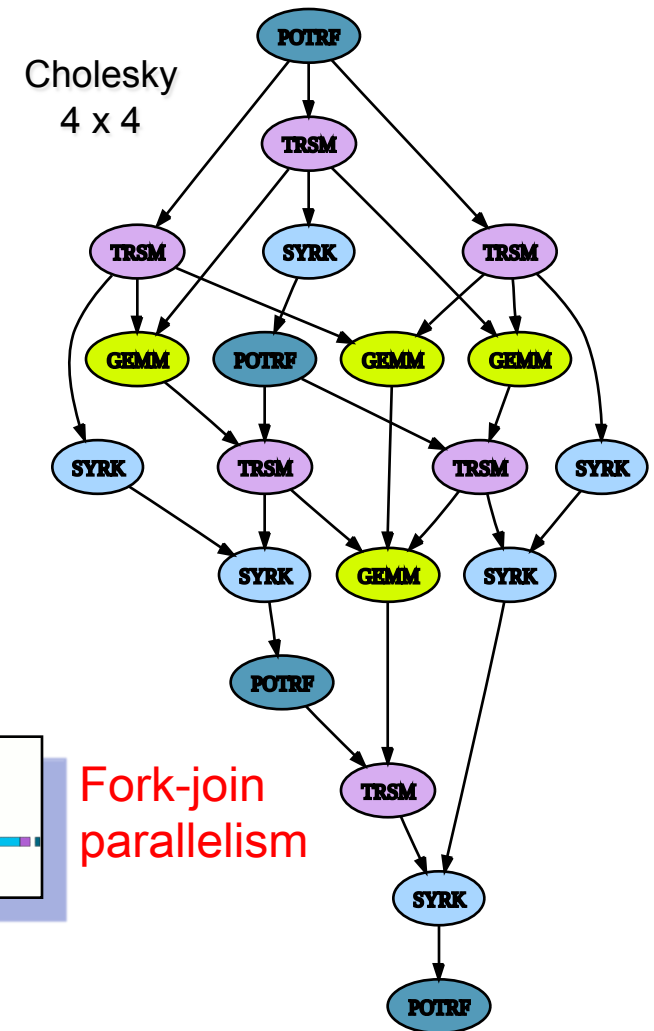
- Dynamic DAG scheduling (QUARK)
- Explicit parallelism
- Implicit communication
- Fine granularity / block data layout

• Arbitrary DAG with dynamic scheduling



DAG scheduled parallelism

Time

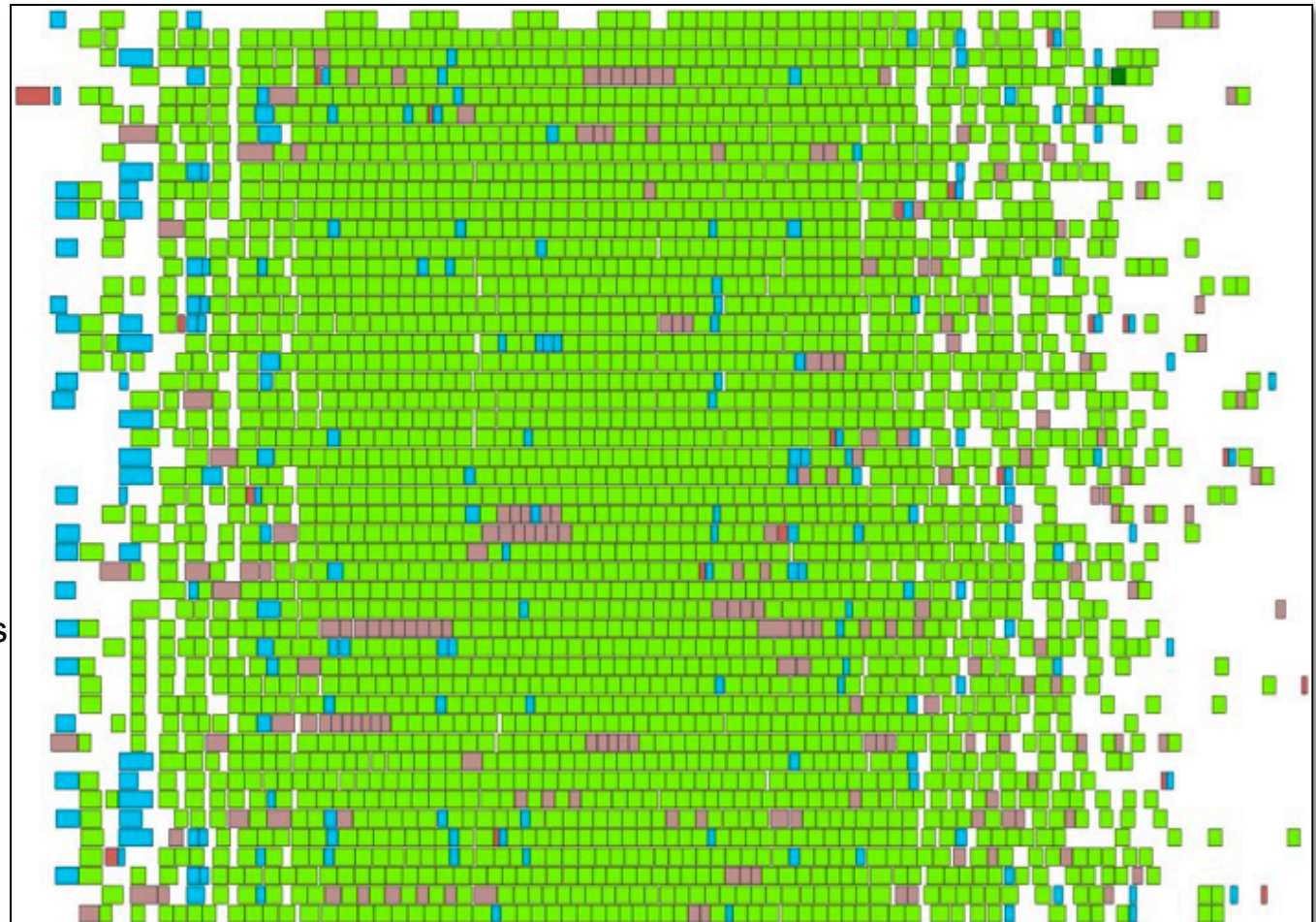


Fork-join parallelism

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



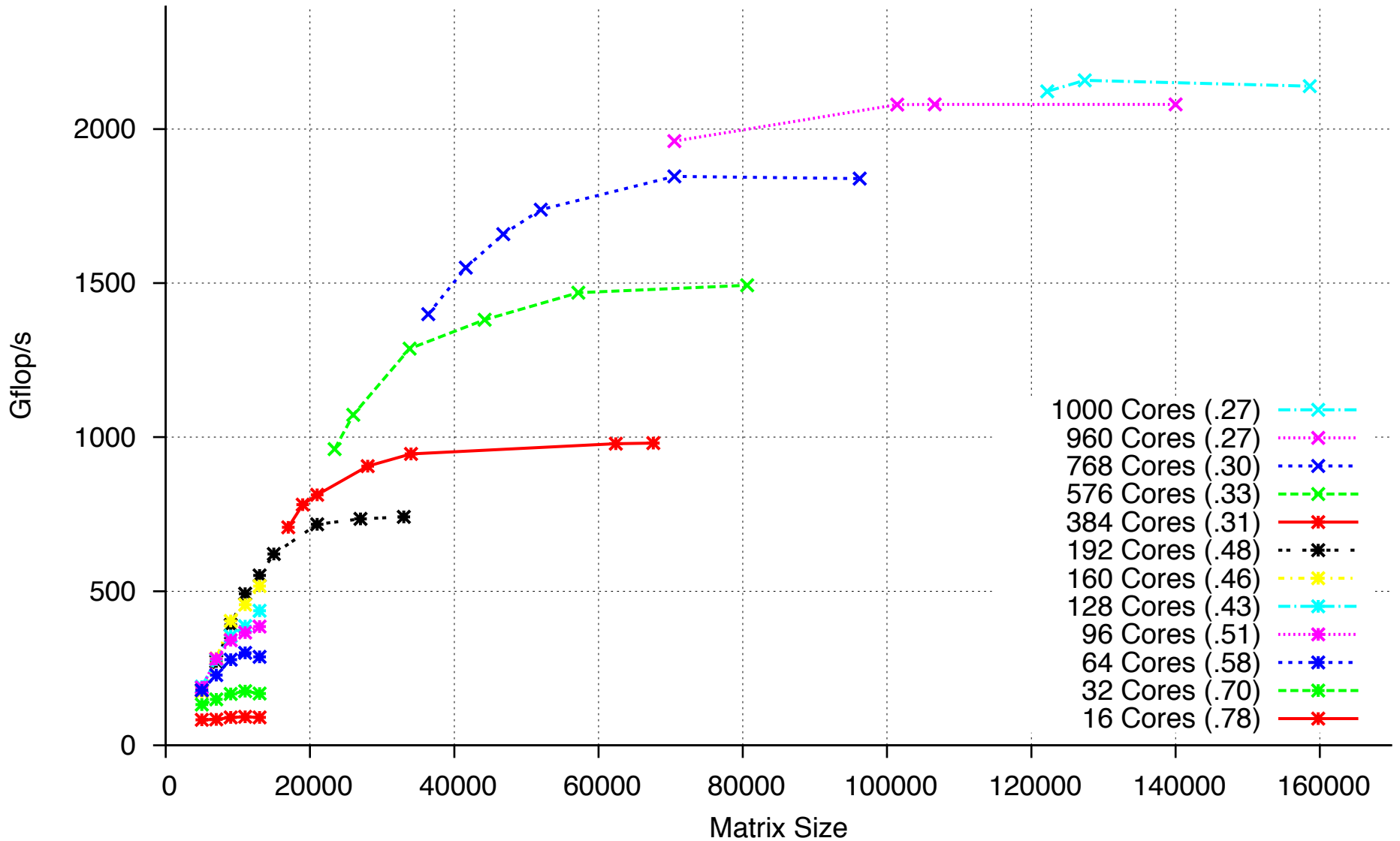
Tile QR factorization; Matrix size 4000x4000, Tile size 200
8-socket, 6-core (48 cores total) AMD Istanbul 2.8 GHz



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]

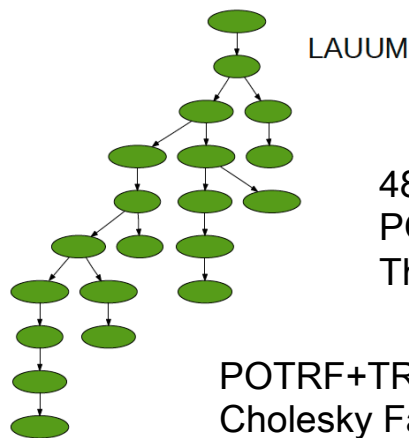
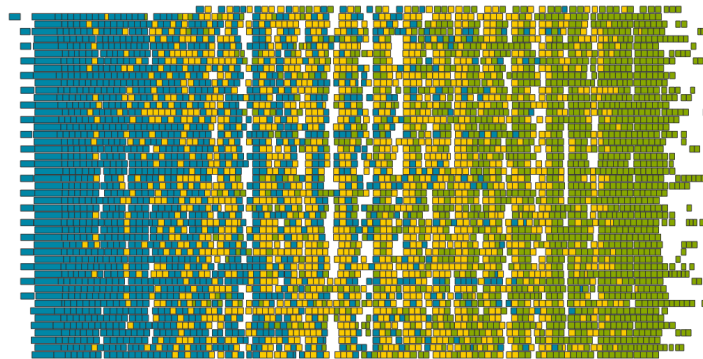
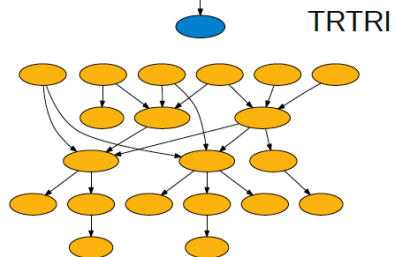
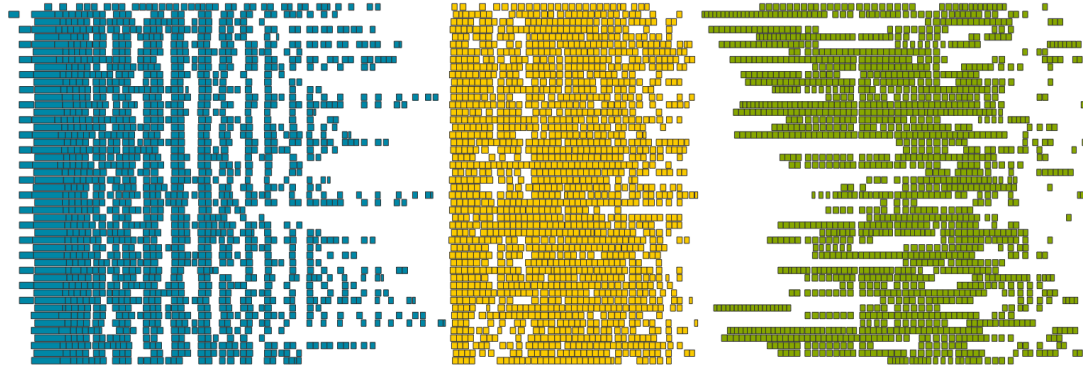
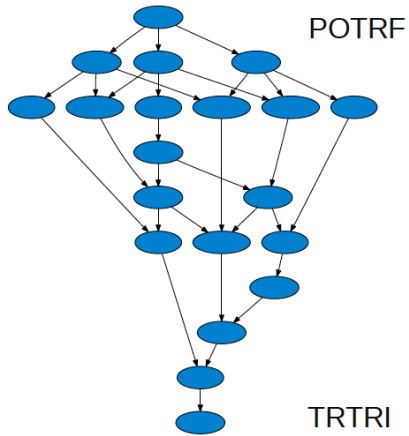
Static Scheduling





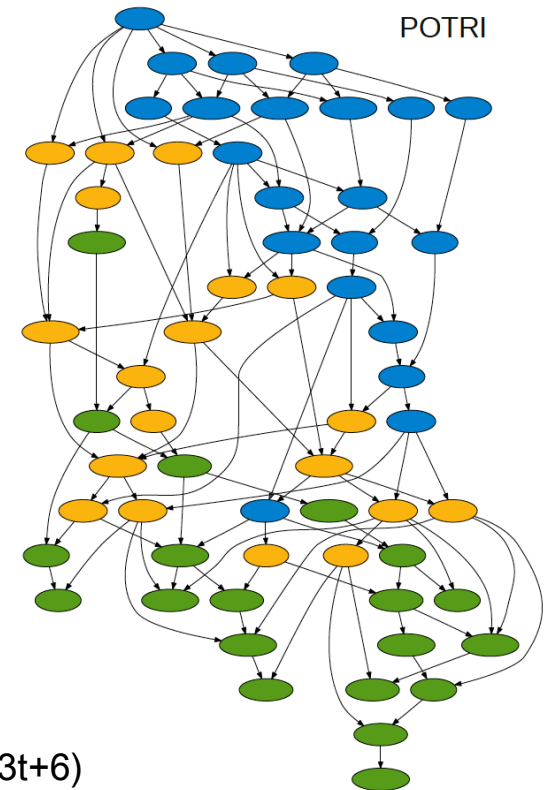
Pipelining: Cholesky Inversion

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: $3t-2$

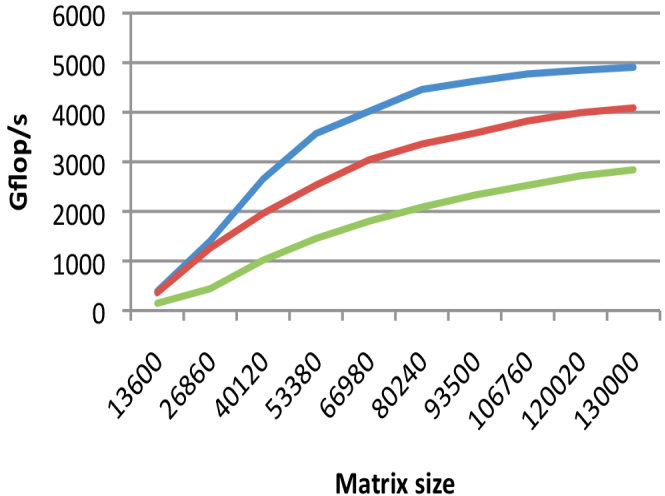


Pipelined: $18(3t+6)$

Cholesky

- DAGuE
- DSBP
- ScaLAPACK

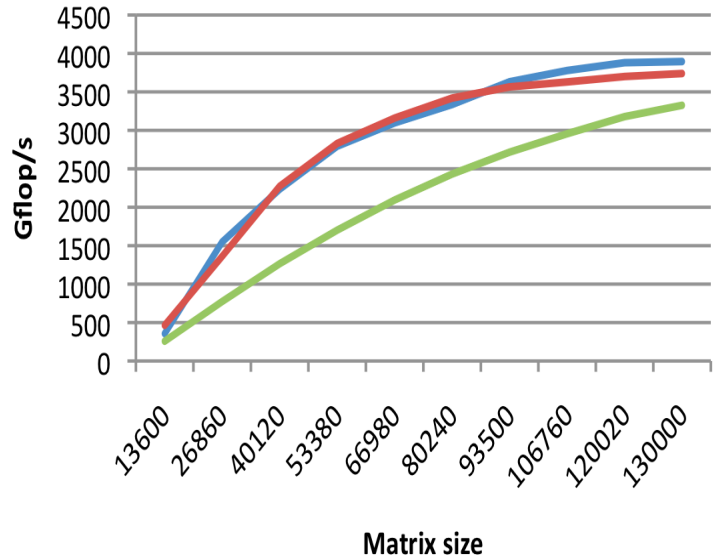
DSBP =
Distributed Square
Block Packed



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

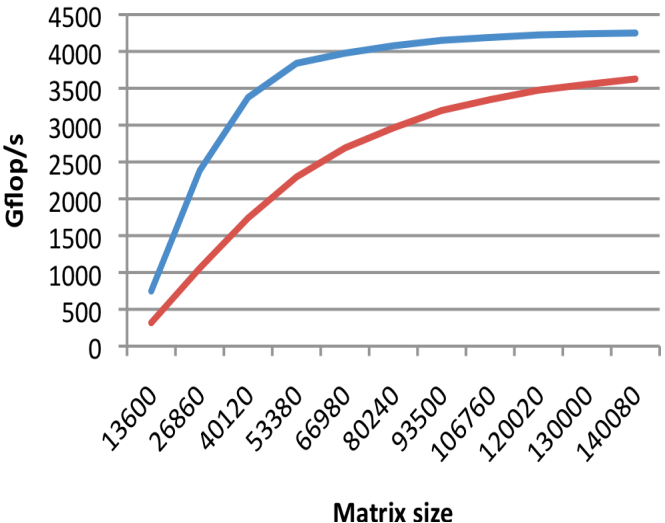
LU

- HPL
- DAGuE
- ScaLAPACK



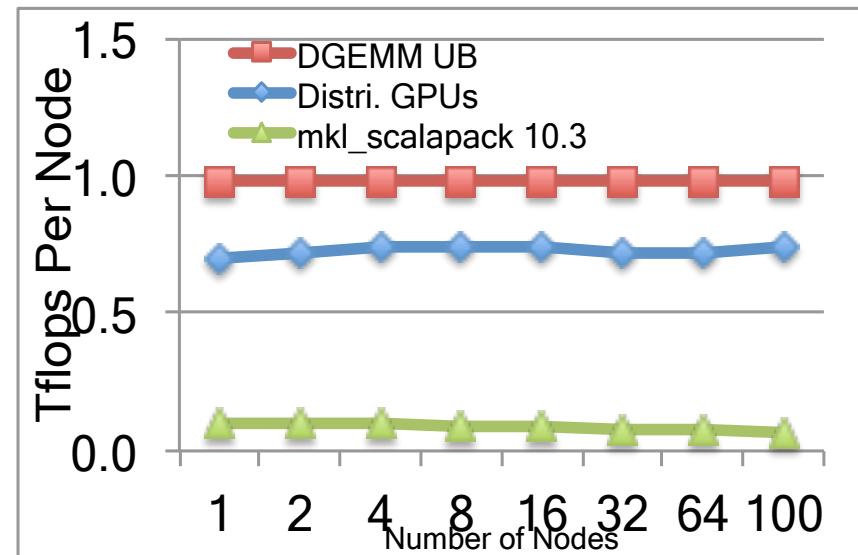
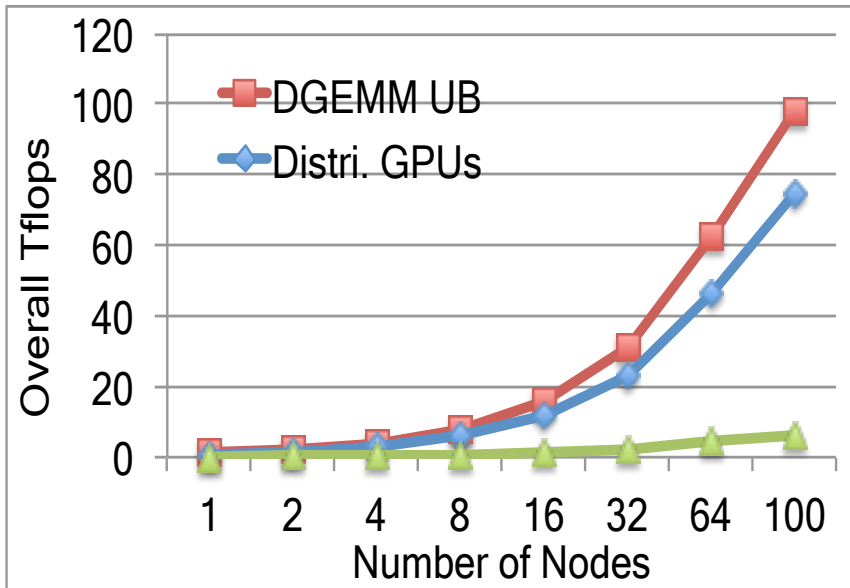
QR

- DAGuE
- ScaLAPACK



Clusters with GPUs (Cholesky)

Use 12 cores and 3 GPUs per node
Input size = $34560 \cdot \sqrt{\text{NumberNodes}}$



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

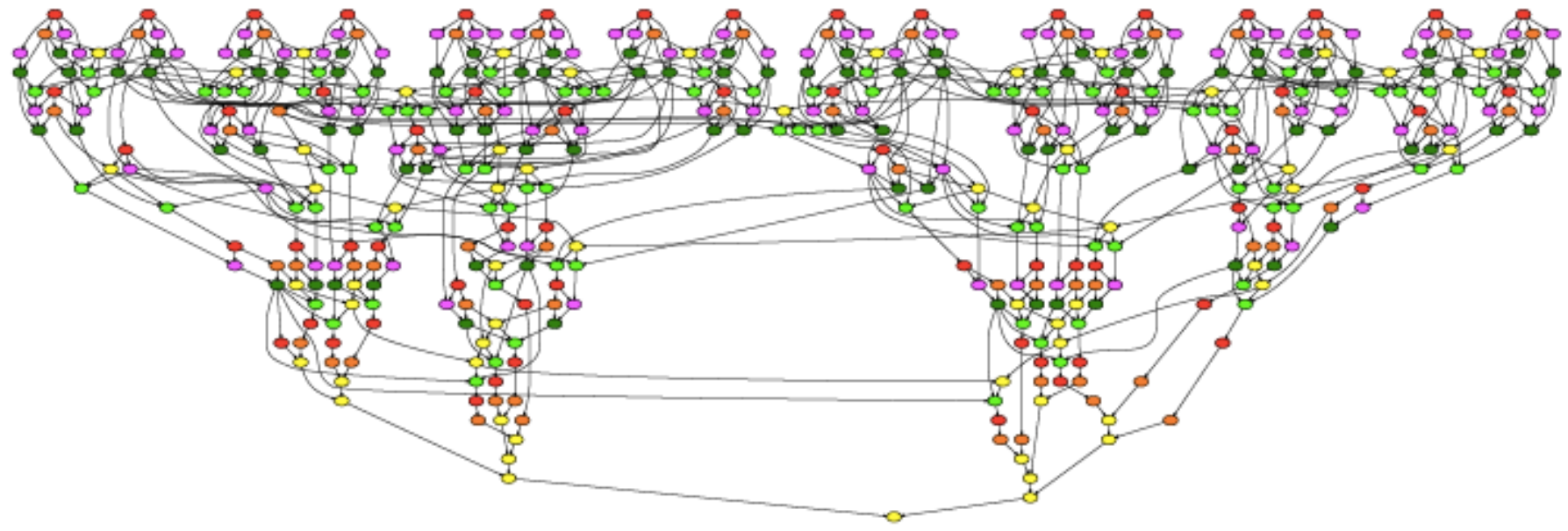
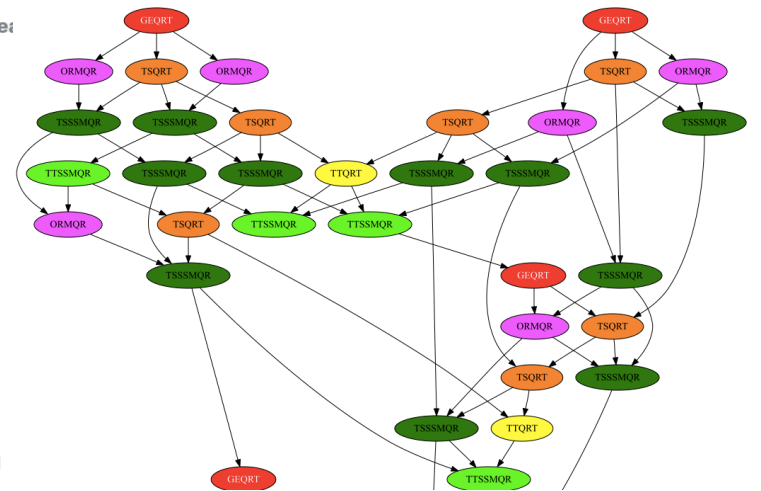
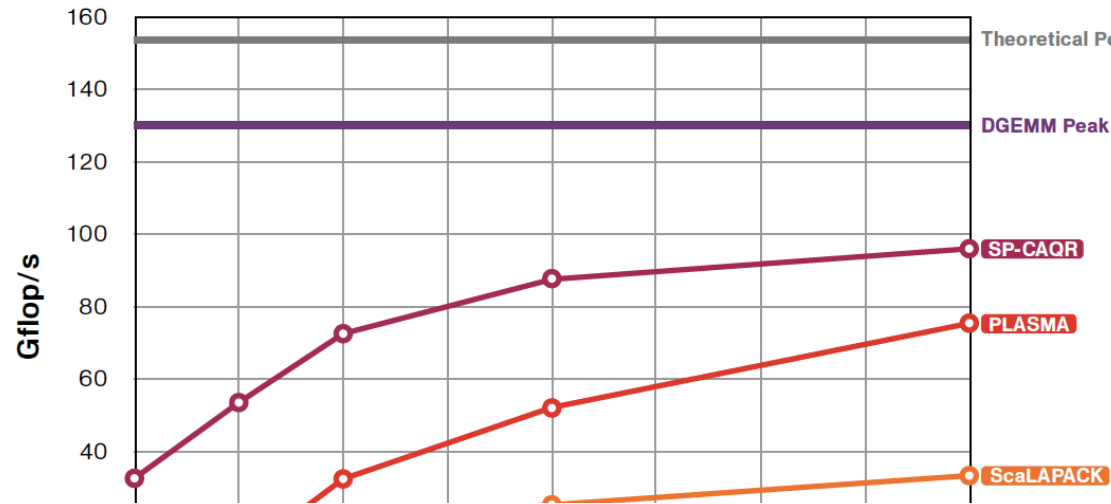


Communication Avoiding Algorithms

- 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 $Ax=b$, $Ax=\lambda x$
 - 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	$\Theta\left(\frac{mn^2}{\sqrt{W}}\right)$
# messages	$\Theta\left(\frac{mn^2}{W^{3/2}}\right)$

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 .**

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

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

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(n3)
x = L\U\b    O(n2)
r = b - Ax   O(n2)
WHILE || r || not small enough
    z = L\U\r O(n2)
    x = x + z  O(n1)
    r = b - Ax O(n2)
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.

$L U = \text{lu}(A)$	SINGLE	$O(n^3)$
$x = L \setminus (U \setminus b)$	SINGLE	$O(n^2)$
$r = b - Ax$	DOUBLE	$O(n^2)$
WHILE $\ r \ $ not small enough		
$z = L \setminus (U \setminus r)$	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.

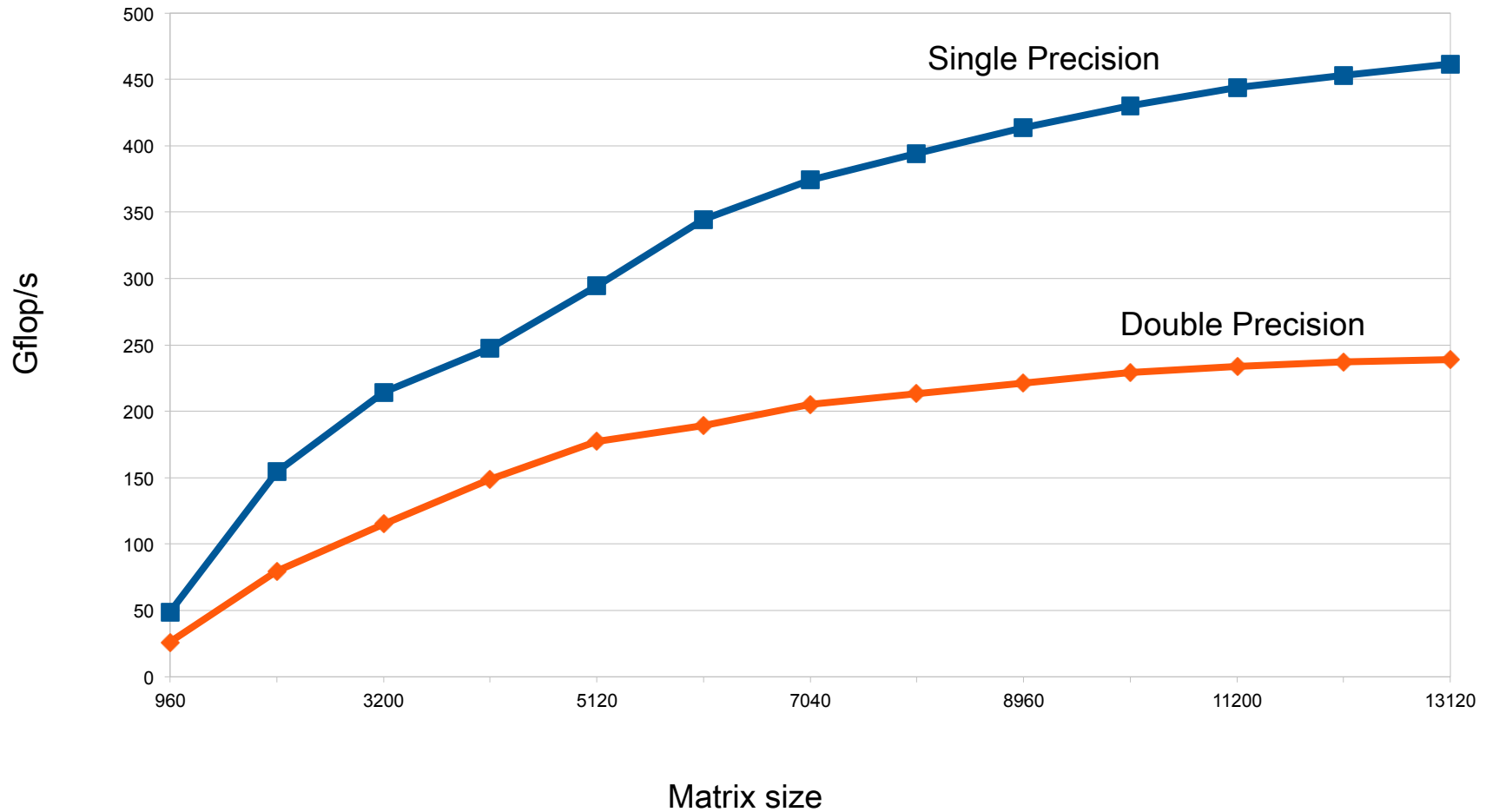
- Requires extra storage, total is 1.5 times normal;
- $O(n^3)$ work is done in lower precision
- $O(n^2)$ work is done in high precision
- Problems if the matrix is ill-conditioned in sp; $O(10^8)$



$$Ax = b$$

FERMI

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

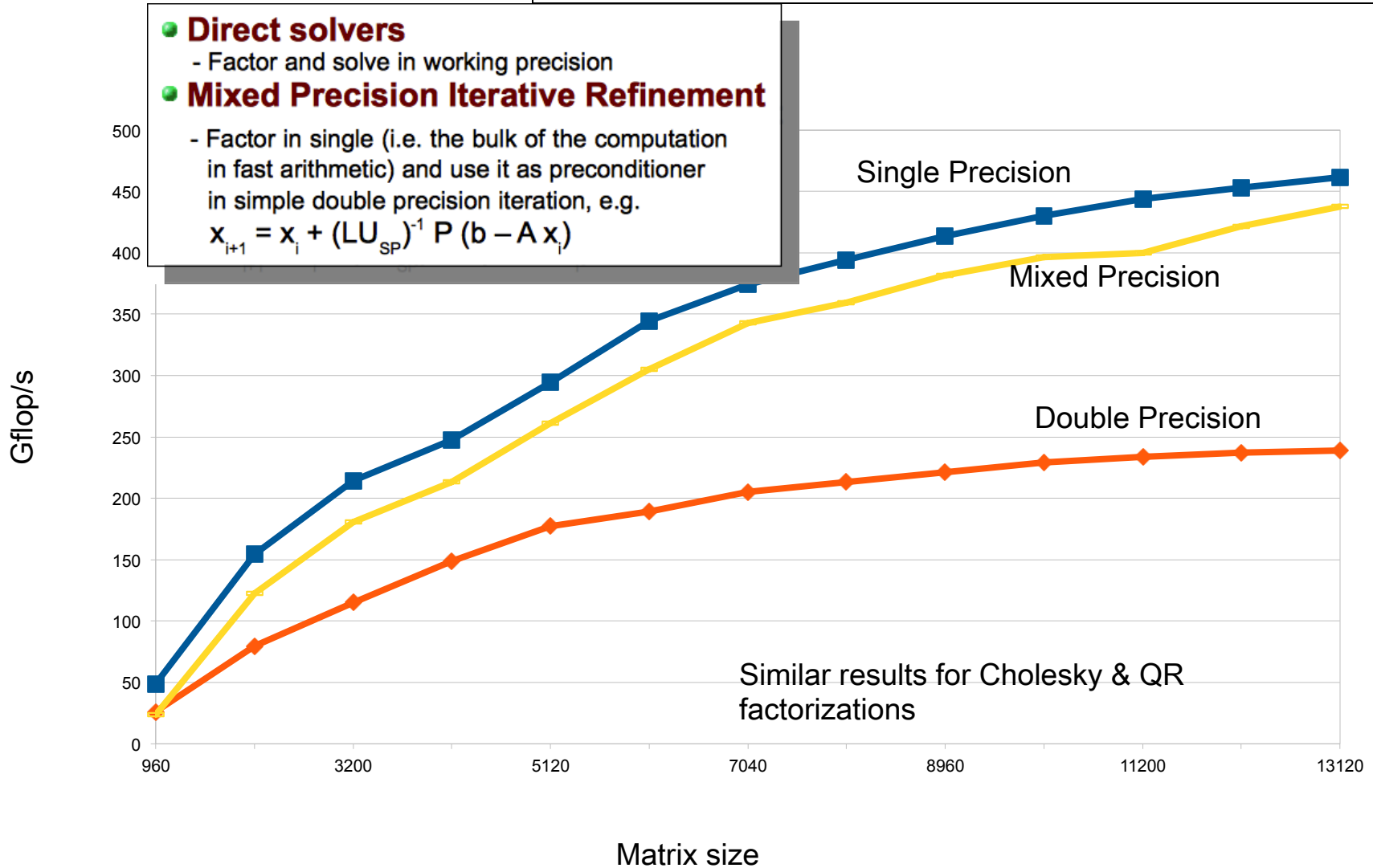




$$Ax = b$$

FERMI

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



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



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

51



Jack Dongarra	Alok Choudhary	Sanjay Kale	Matthias Mueller	Bob Sugar
Pete Beckman	Sudip Dossanjh	Richard Kenway	Wolfgang Nagel	Shinji Sumimoto
Terry Moore	Thom Dunning	David Keyes	Hiroshi Nakashima	William Tang
Patrick Aerts	Sandro Fiore	Bill Kramer	Michael E. Papka	John Taylor
Giovanni Aloisio	Al Geist	Jesus Labarta	Dan Reed	Rajeev Thakur
Jean-Claude Andre	Bill Gropp	Alain Lichnewsky	Mitsuhsa Sato	Anne Trefethen
David Barkai	Robert Harrison	Thomas Lippert	Ed Seidel	Mateo Valero
Jean-Yves Berthou	Mark Herald	Bob Lucas	John Shalf	Aad van der Steen
Taisuke Boku	Michael Heroux	Barney Maccabe	David Skinner	Jeffrey Vetter
Bertrand Braunschweig	Adolfy Hoisie	Satoshi Matsuoka	Marc Snir	Peg Williams
Franck Cappello	Koh Hotta	Paul Messina	Thomas Sterling	Robert Wisniewski
Barbara Chapman	Yutaka Ishikawa	Peter Michielse	Rick Stevens	Kathy Yelick
Xuebin Chi	Fred Johnson	Bernd Mohr	Fred Streit	

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

▪ **Alan Turing (1912 – 1954)**

SPONSORS



• www.exascale.org