Nuclear Science and Technology Division Seminar

OAK RIDGE NATIONAL LABORATORY

Managed by UT-Battelle for the Department of Energy

An Overview of High-Performance Computing and Challenges for the Future

Jack Dongarra

University of Tennessee
Oak Ridge National Laboratory
University of Manchester

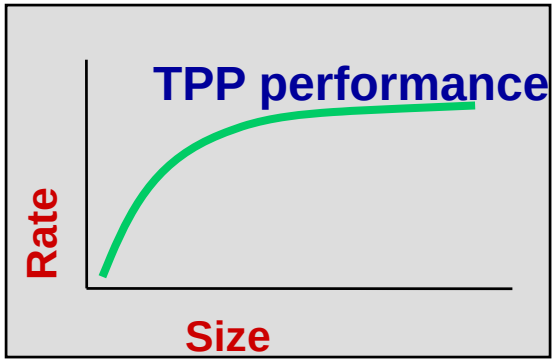
Overview

- **Quick look at High Performance Computing**
  - **Top500**
- **Six important concepts to keep in mind for multicore**

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*



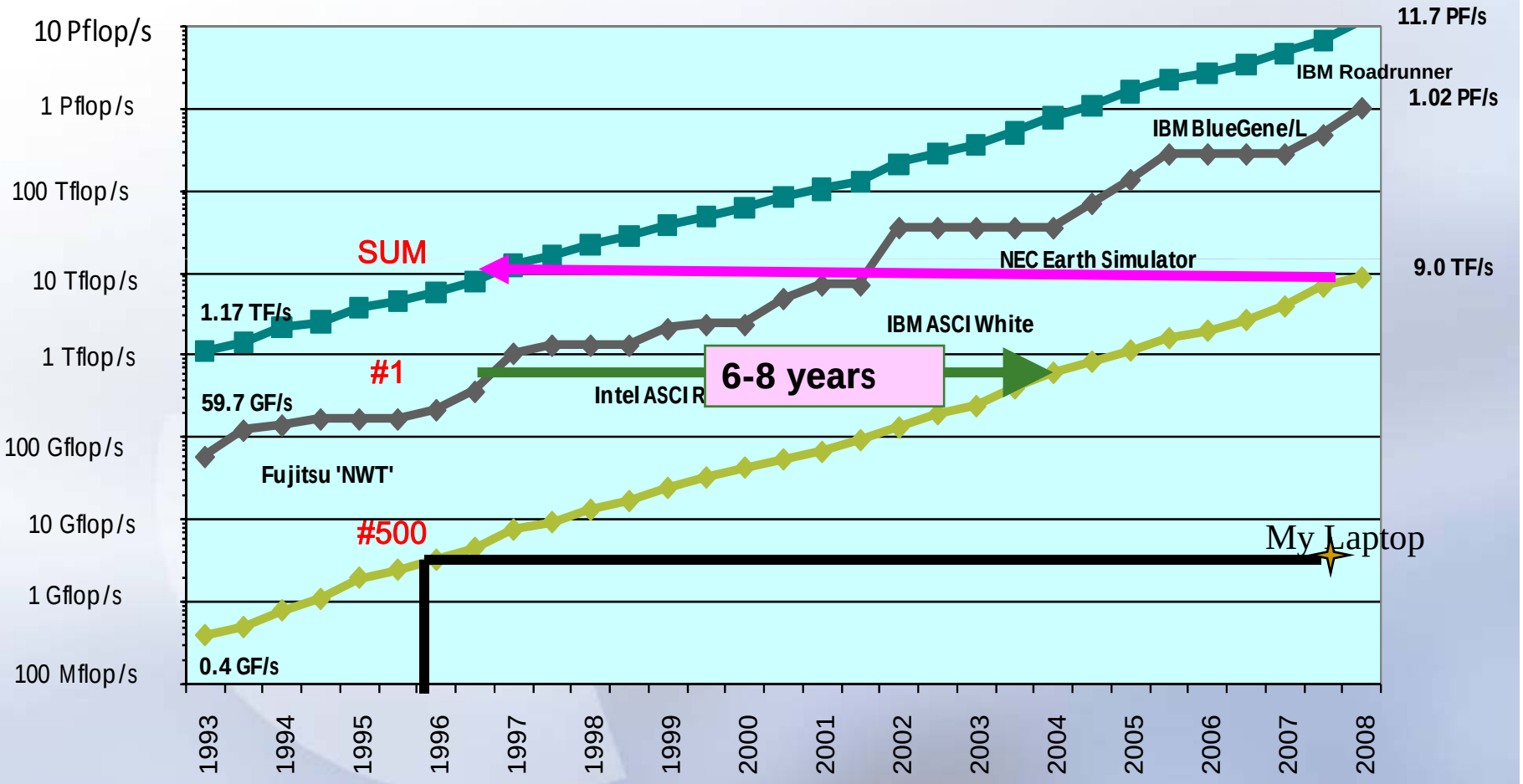
- Updated twice a year

  SC‘xy in the States in November

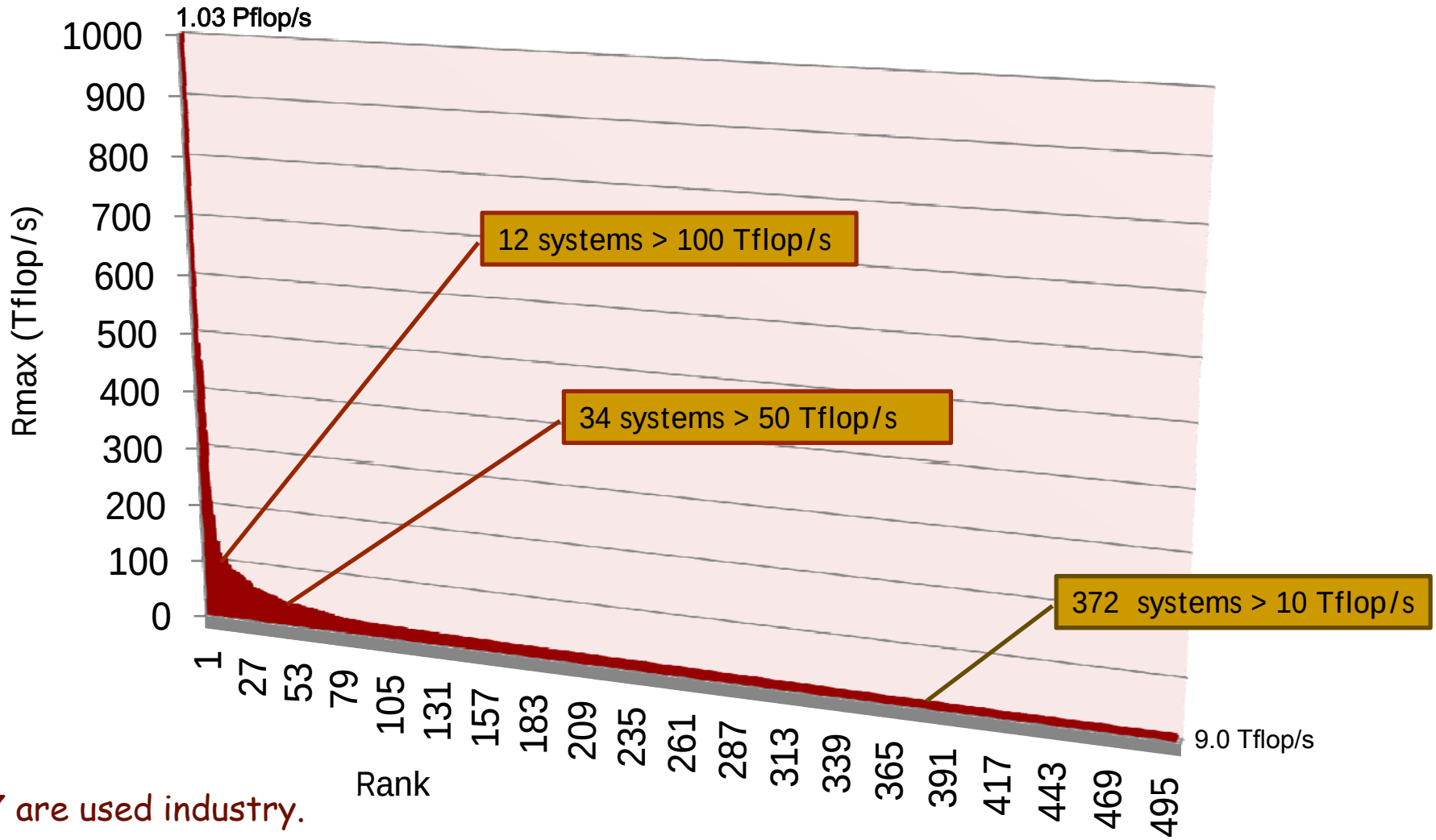
  Meeting in Germany in June

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

Performance Development



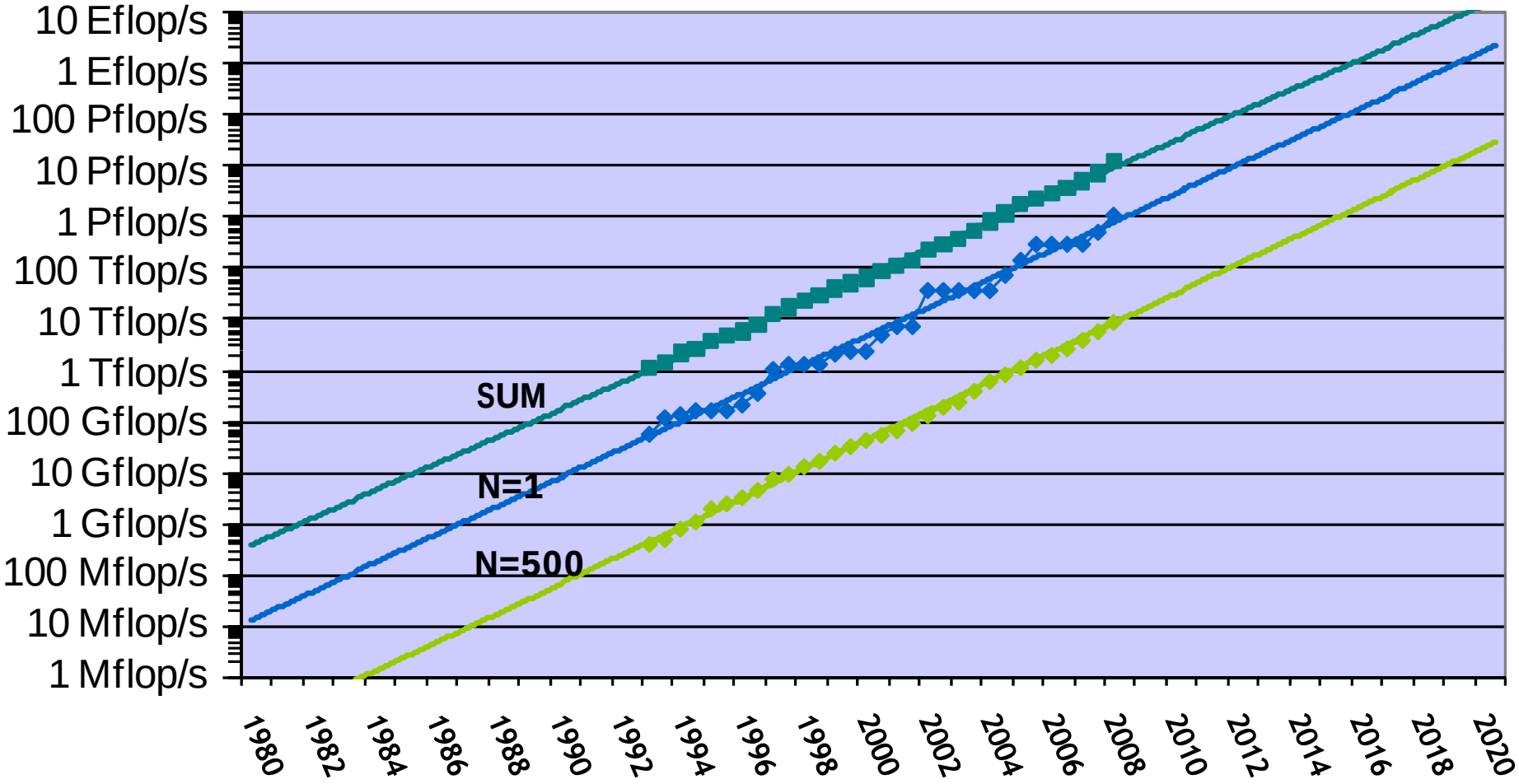
Distribution of the Top500 (6/08)



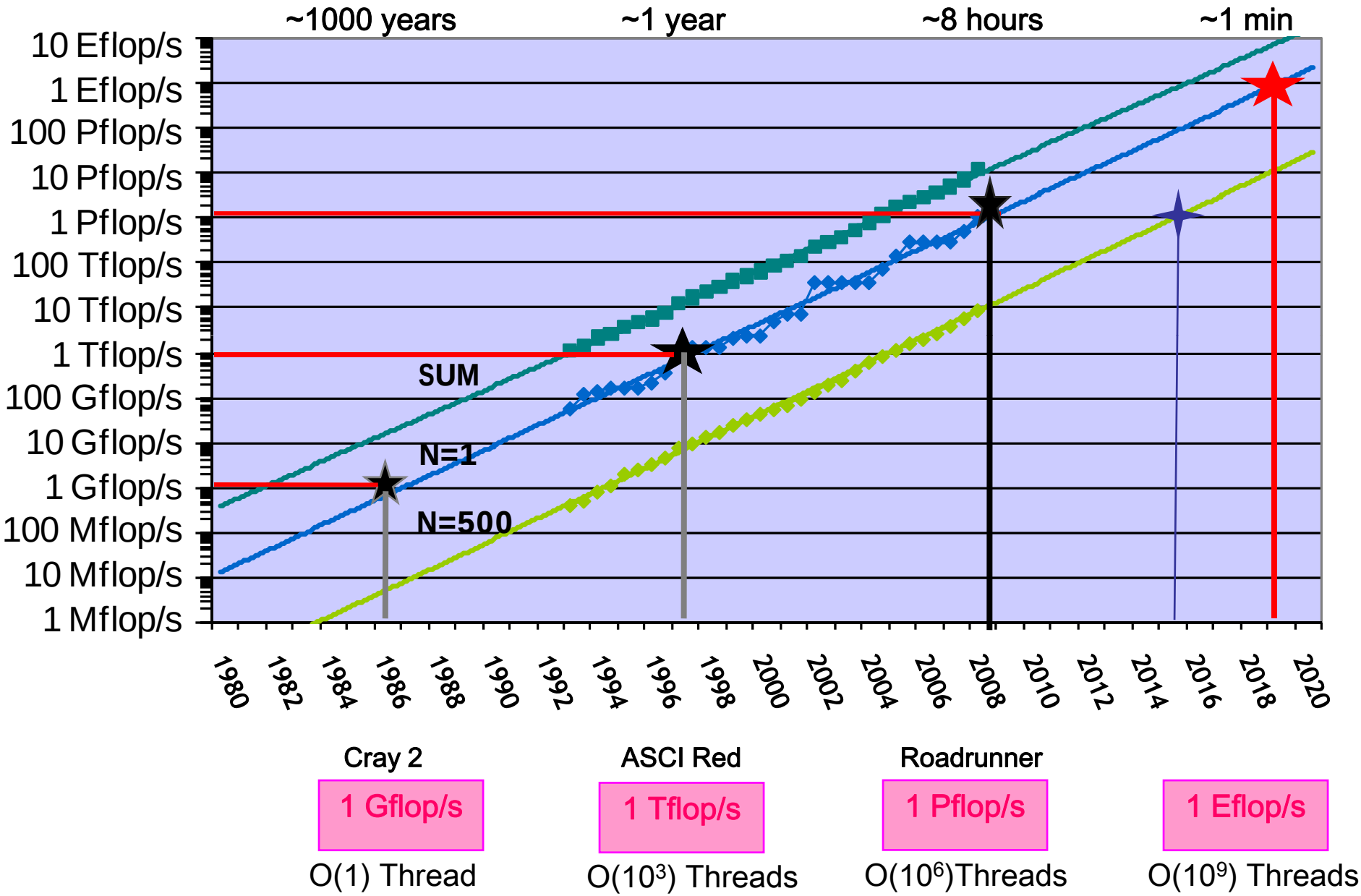
287 are used industry.

301 systems were replaced in the June 2008 list.

Performance Development & Projections

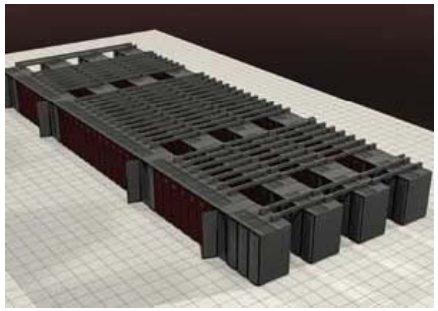


Performance Development & Projections



ASCI Red Compared to ASC Roadrunner

- ASCI Red Computer
  - 1997
  - First TFlop/s Computer
  - ~10,000 Pentium Pro
  - ~1 MWatt
  - 200 MHz
  - 200 MFlop/s each proc
    - 104 cabinets
  - ~2500 sq ft (230 m²).



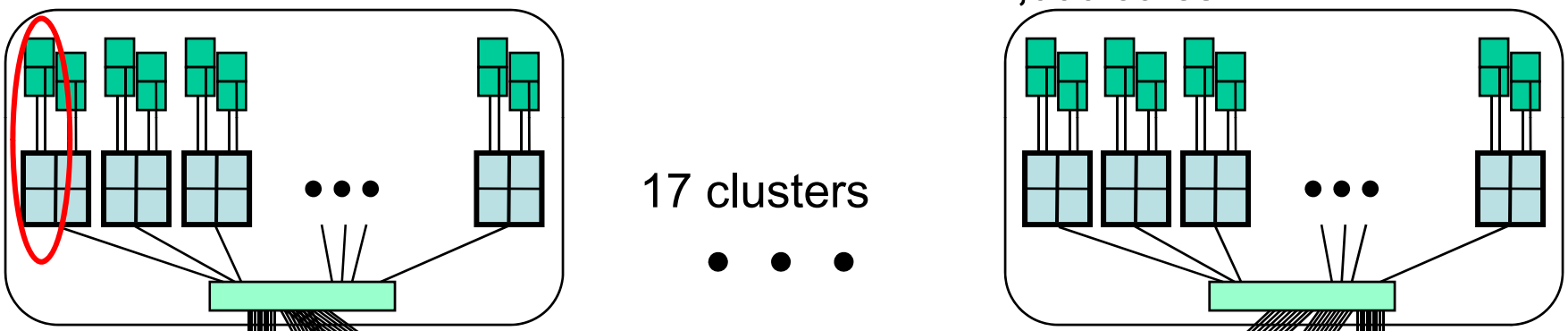
- ASC Roadrunner
  - 2008
  - First PFlop/s Computer
  - Hybrid: ~130,000 cores
    - 6,948 dual-core Opterons + 12,960 Cell processors
  - ~3 MWatts
  - 1.8 GHz Opteron + 3.2 GHz Cell
  - ~400 GFlop/s Node
    - Node: 2 socket (each dual core Opteron; each Opteron core has a Cell Chip)
    - 180 nodes/connect unit
    - 18 connect units in the system
  - ~5500 sq ft

LANL Roadrunner
A Petascale System in 2008

"Connected Unit" cluster
192 Opteron nodes
(180 w/ 2 dual-Cell blades
connected w/ 4 PCIe x8 links)

≈ 13,000 Cell HPC chips
- ≈ **1.33 PetaFlop/s** (from Cell)

≈ 7,000 dual-core Opterons
≈ **122,000 cores**

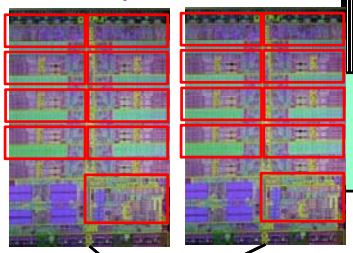


17 clusters

• • •

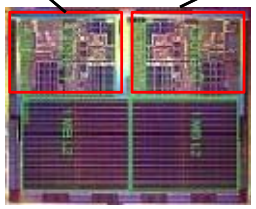
2nd stage InfiniBand 4x DDR interconnect
(18 sets of 12 links to 8 switches)

Cell chip for each core



2nd stage InfiniBand interconnect (8 switches)

Based on the 100 Gflop/s (DP) Cell chip



Dual Core Opteron Chip

Hybrid Design (2 kinds of chips & 3 kinds of cores)
Programming required at 3 levels.

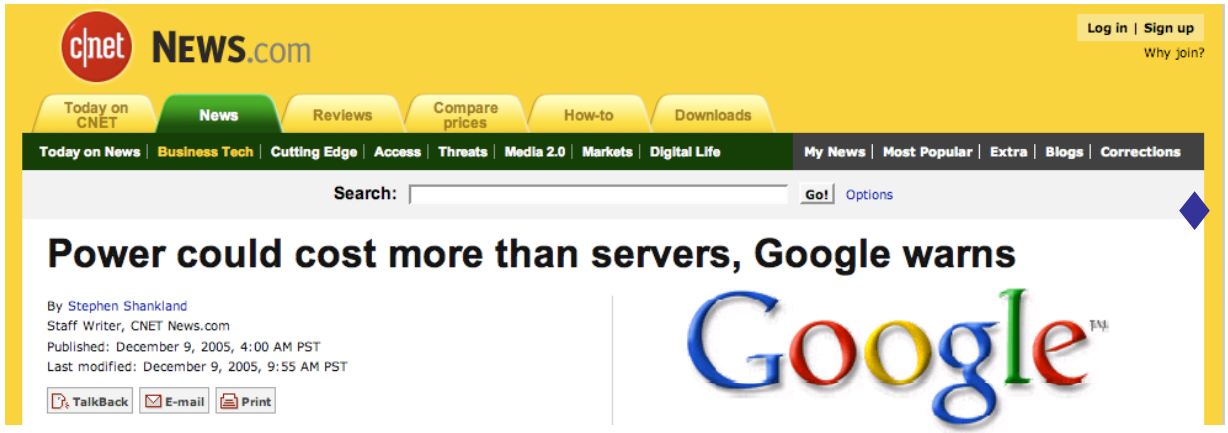
Top10 of the June 2008 List

	Computer	Rmax [TF/s]	Rmax / Rpeak	Installation Site	Country	#Cores
1	IBM / Roadrunner BladeCenter QS22/LS21	1,026	75%	DOE/NNSA/LANL	USA	122,400
2	IBM / BlueGene/L eServer Blue Gene Solution	478	80%	DOE/NNSA/LLNL	USA	212,992
3	IBM / Intrepid Blue Gene/P Solution	450	81%	DOE/OS/ANL	USA	163,840
4	SUN / Ranger SunBlade x6420	326	65%	NSF/TACC	USA	62,976
5	CRAY / Jaguar Cray XT4 QuadCore	205	79%	DOE/OS/ORNL	USA	30,976
6	IBM / JUGENE Blue Gene/P Solution	180	81%	Forschungszentrum Juelich (FZJ)	Germany	65,536
7	SGI / Encanto SGI Altix ICE 8200	133.2	77%	New Mexico Computing Applications Center	USA	14,336
8	HP / EKA Cluster Platform 3000 BL460c	132.8	77%	Computational Research Lab, TATA SONS	India	14,384
9	IBM / Blue Gene/P Solution	112	81%	IDRIS	France	40,960
10	SGI / Altix ICE 8200EX	106	86%	Total Exploration Production	France	10,240

Top10 of the June 2008 List

	Computer	Rmax [TF/s]	Rmax / Rpeak	Installation Site	Country	#Cores	Power [MW]	**MFlops/ Watt**
1	IBM / Roadrunner BladeCenter QS22/LS21	1,026	75%	DOE/NNSA/LANL	USA	122,400	2.35	437
2	IBM / BlueGene/L eServer Blue Gene Solution	478	80%	DOE/NNSA/LLNL	USA	212,992	2.33	205
3	IBM / Intrepid Blue Gene/P Solution	450	81%	DOE/OS/ANL	USA	163,840	1.26	357
4	SUN / Ranger SunBlade x6420	326	65%	NSF/TACC	USA	62,976	2.00	163
5	CRAY / Jaguar Cray XT4 QuadCore	205	79%	DOE/OS/ORNL	USA	30,976	1.58	130
6	IBM / JUGENE Blue Gene/P Solution	180	81%	Forschungszentrum Juelich (FZJ)	Germany	65,536	0.50	357
7	SGI / Encanto SGI Altix ICE 8200	133.2	77%	New Mexico Computing Applications Center	USA	14,336	0.86	155
8	HP / EKA Cluster Platform 3000 BL460c	132.8	77%	Computational Research Lab, TATA SONS	India	14,384	0.79	169
9	IBM / Blue Gene/P Solution	112	81%	IDRIS	France	40,960	0.32	357
10	SGI / Altix ICE 8200EX	106	86%	Total Exploration Production	France	10,240	0.44	240

Power is an Industry Wide Problem



The New York Times **"Hiding in Plain Sight, Google Seeks More Power", by John Markoff, June 14, 2006**

Google Plant in The Dalles, Oregon, from NYT, June 14, 2006

Google facilities

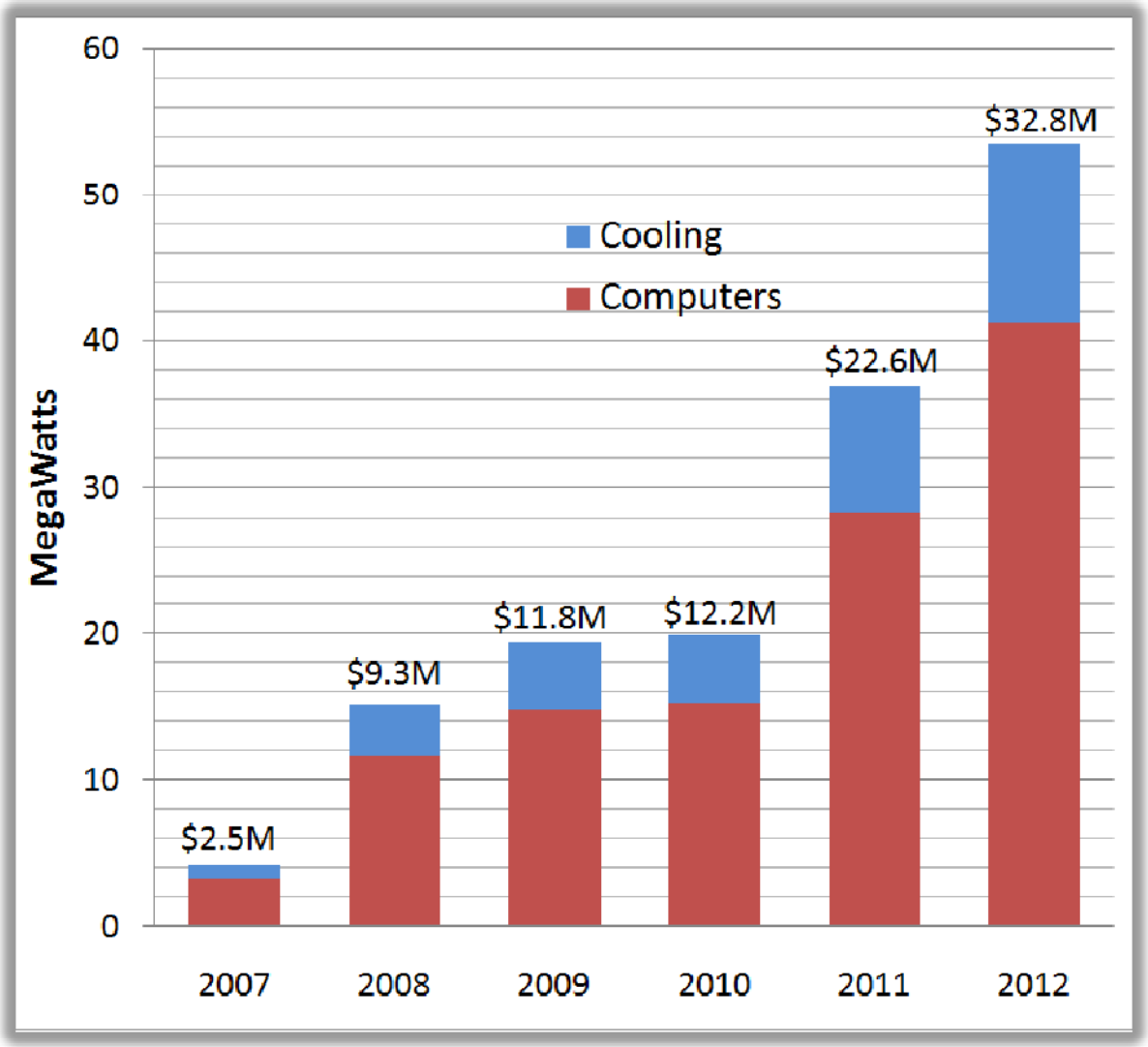
- leveraging hydroelectric power
  - old aluminum plants

Microsoft and Yahoo are building big data centers upstream in Wenatchee and Quincy, Wash.
– To keep up with Google, which means they need cheap electricity and readily accessible data networking

Microsoft Quincy, Wash.
470,000 Sq Ft, 47MW!

ORNL/UTK Computer Power Cost Projections 2007-2012

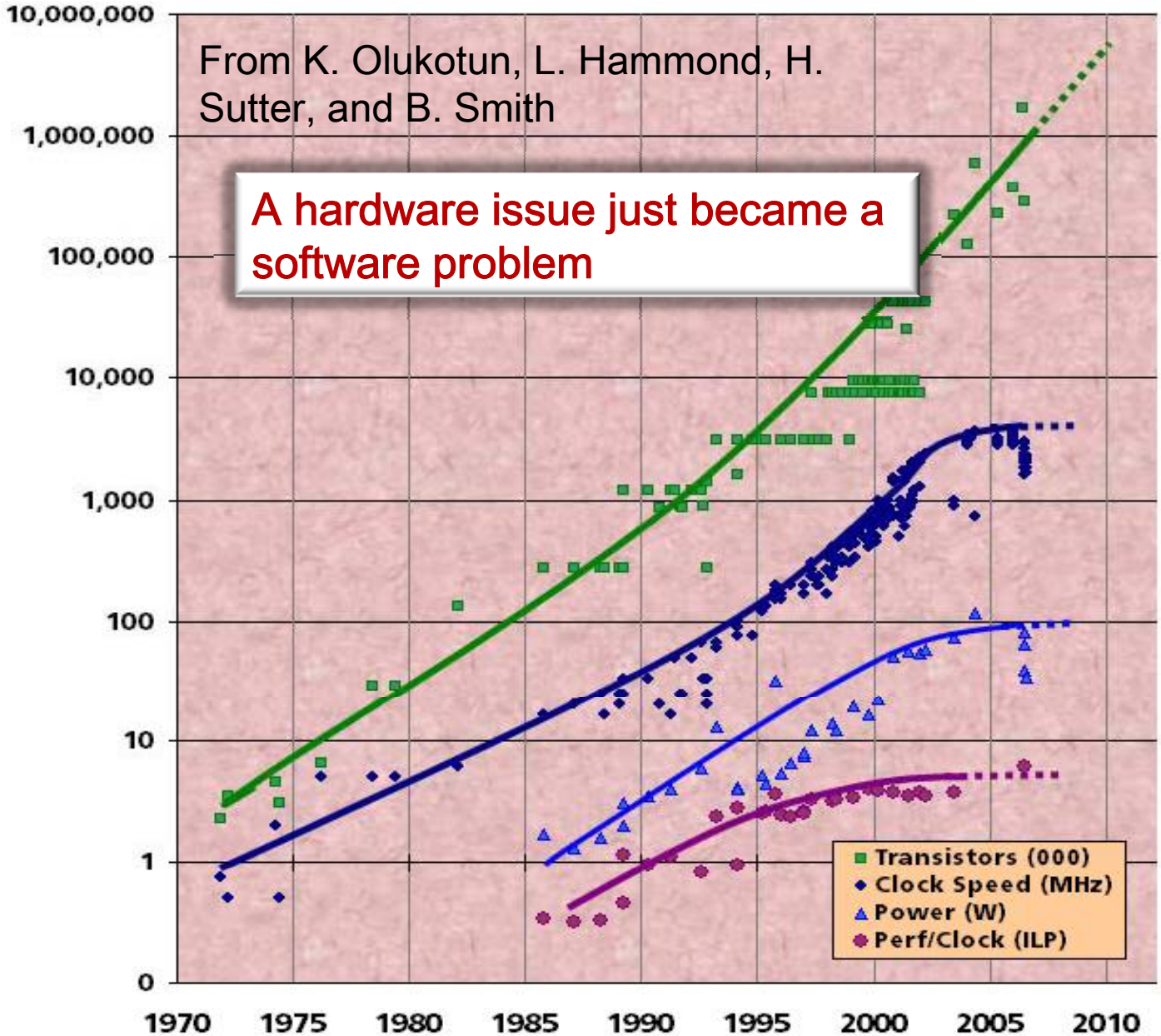
- Over the next 5 years ORNL/UTK will deploy 2 large Petascale systems
- Using 4 MW today, going to 15MW before year end
- By 2012 could be using more than 50MW!!
- Cost estimates based on $0.07 per KwH



Power becomes the architectural driver for future large systems

Cost Per Year
Includes both DOE and NSF systems.

Something's Happening Here…



- **In the "old days" it was: each year processors would become faster**
- **Today the clock speed is fixed or getting slower**
- **Things are still doubling every 18 -24 months**
- **Moore's Law reinterpretated.**
  - **Number of cores double every 18-24 months**

Power Cost of Frequency

- Power ∝ Voltage^2 x Frequency (V^{2}F)
- **Frequency** ∝ Voltage
- Power ∝ Frequency^3

	Cores	V	Freq	Perf	Power	PE (Bops/watt)
Superscalar	1	1	1	1	1	1
"New" Superscalar	1X	1.5X	1.5X	1.5X	3.3X	0.45X

Power Cost of Frequency

- Power ∝ Voltage^2 x Frequency (V^{2}F)
- **Frequency** ∝ Voltage
- Power ∝ Frequency^3

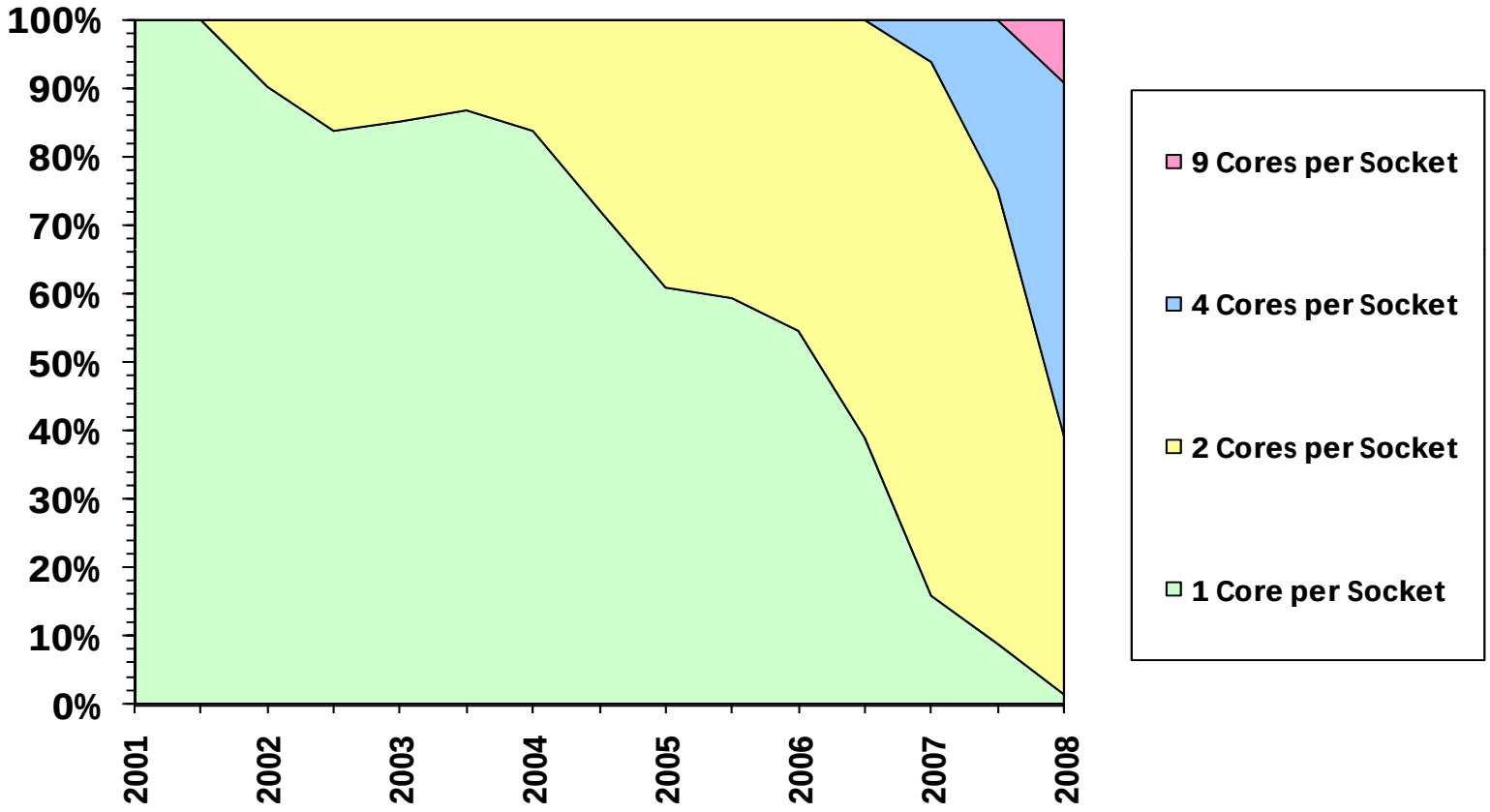
	Cores	V	Freq	Perf	Power	PE (Bops/watt)
Superscalar	1	1	1	1	1	1
"New" Superscalar	1X	1.5X	1.5X	1.5X	3.3X	0.45X
Multicore	2X	0.75X	0.75X	1.5X	0.8X	1.88X

(Bigger # is better)

50% more performance with 20% less power

Preferable to use multiple slower devices, than one superfast device

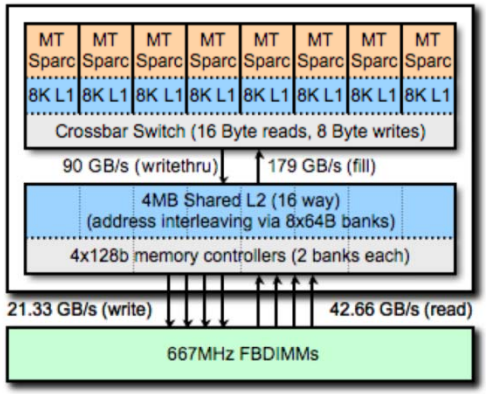
Top500 Core per Socket Over Time



Today's Multicores

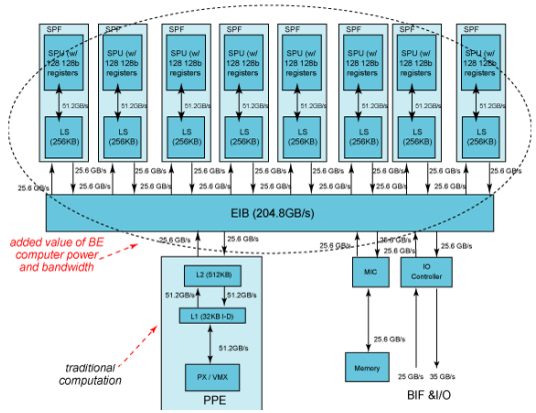
98% of Top500 Systems Are Based on Multicore

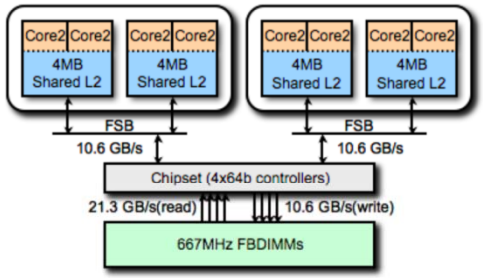
282 use Quad-Core
204 use Dual-Core
3 use Nona-core



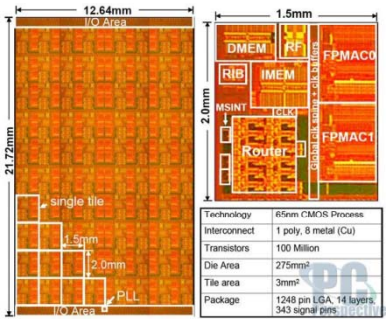
Sun Niagra2 (8 cores)

IBM Cell (9 cores)

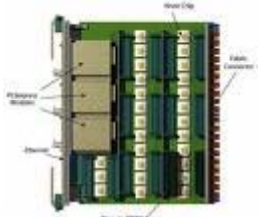




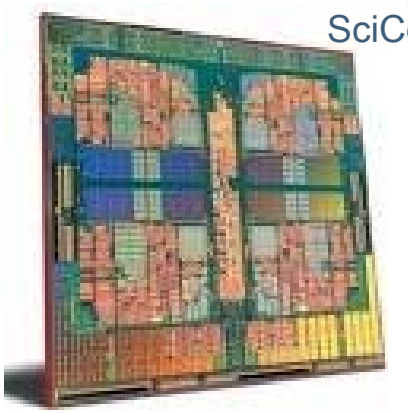
Intel Clovertown (4 cores)



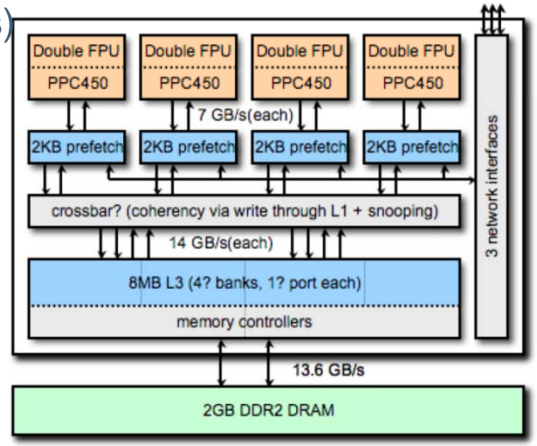
Intel Polaris (80 cores)



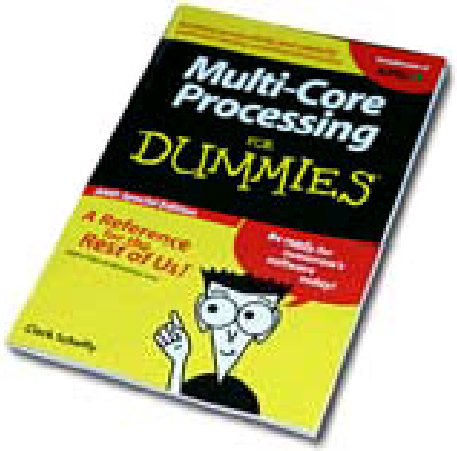
SciCortex (6 cores)



AMD Opteron (4 cores)

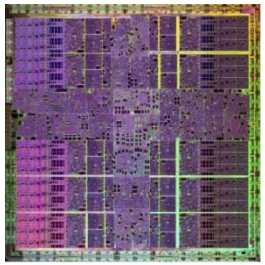


IBM BG/P (4 cores)



And then there's the GPGPU's NVIDIA's Tesla T10P

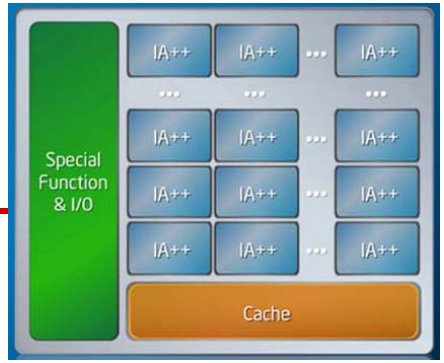
- **T10P chip**
  - **240 cores; 1.5 GHz**
  - **Tpeak 1 Tflop/s - 32 bit floating point**
  - **Tpeak 100 Gflop/s - 64 bit floating point**



- **S1070 board**
  - **4 - T10P devices;**
  - **700 Watts**

- **GTX 280**
  - **1 - T10P; 1.3 GHz**
  - **Tpeak 864 Gflop/s - 32 bit floating point**
  - **Tpeak 86.4 Gflop/s - 64 bit floating point**

Intel's Larrabee Chip



THE NEW YORK TIMES, MONDAY, AUGUST 4, 2008

Intel's Line of Graphics Chips Could Have Broader Uses

By JOHN MARKOFF

SAN FRANCISCO — **Intel** is planning to release on Monday the first technical details of a new family of chips intended to soup up computer graphics and, eventually, a broad range of computing tasks.

The new microprocessor family, code-named Larrabee, will be available in late 2009 or early 2010. Intel is releasing the details of its plans ahead of the Siggraph industry conference in Los Angeles, which starts Aug. 11.

The company said it would initially aim Larrabee at the personal-computer graphics market, where its "many-core" design, x86 instruction set, which will allow the chips to take advantage of a huge library of existing software.

Instead of speed, Intel turns to improving performance.

In 2004, after finding that it could not make its chips faster because they were overheating, Intel adopted a strategy it referred to as a "right-hand turn." It switched to improving performance by increasing the number of processing elements, or cores, on each chip. That led first to dual-core and now quad-core chips.

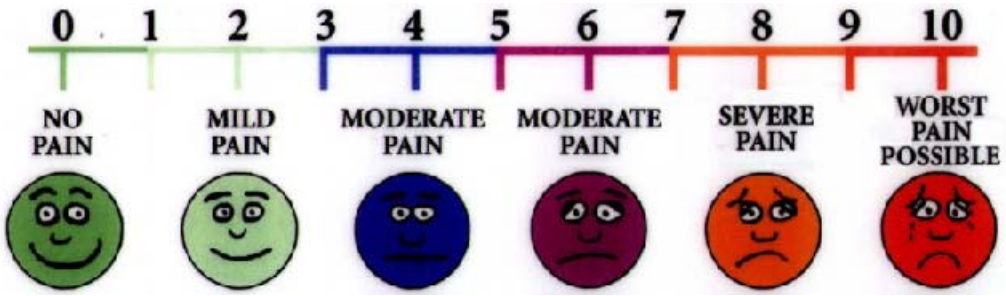
Analysts said the first generation of Larrabee may have 16 to 48 cores, depending on the performance goal.

Intel has tried several approaches to chip design, but none of them have had the impact of its x86 family, which was originally introduced three decades ago. Architectures that have been less successful include the Itanium and the 432, neither of which was adopted in mainstream computing.

- **Many X 86 IA cores**
  - **Scalable to Tflop/s**
- **New cache architecture**
- **New vector instructions set**
  - **Vector memory operations**
  - **Conditionals**
  - **Integer and floating point arithmetic**
- **New vector processing unit / wide SIMD**

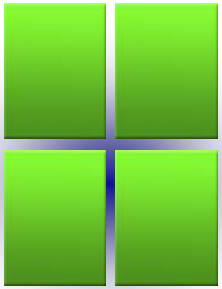
Pro's and Con's for Different Computational Platforms

	CORE	CELL	GPU	FPGA
Speed improvement				
Ease-of-programming for high performance				
Availibility of libraries				

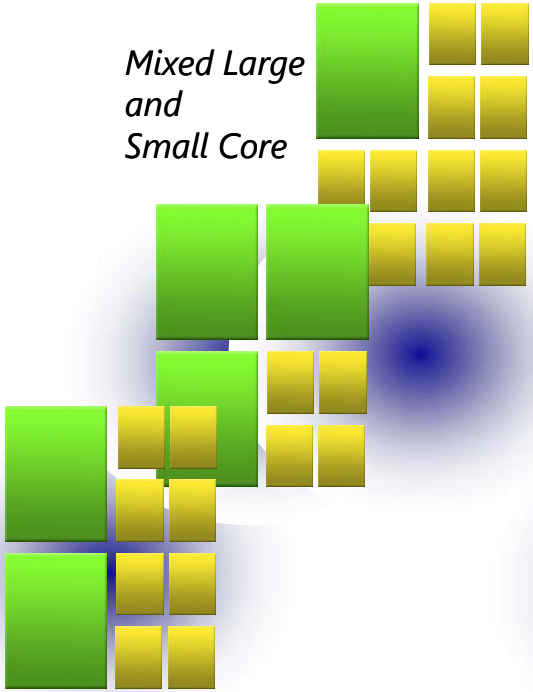


What's Next? Multicore to Manycore

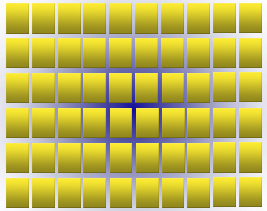
All Large Core



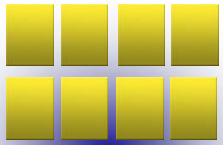
Mixed Large and Small Core



Many Small Cores

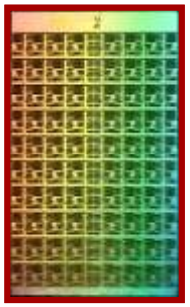


All Small Core

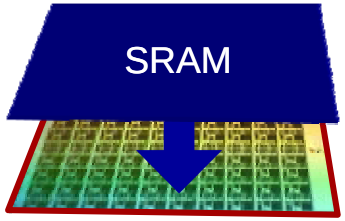


Different Classes of Chips
- Home
- Games / Graphics
- Business
- Scientific

Many Floating-Point Cores



+ 3D Stacked Memory



The question is not whether this will happen but whether we are ready

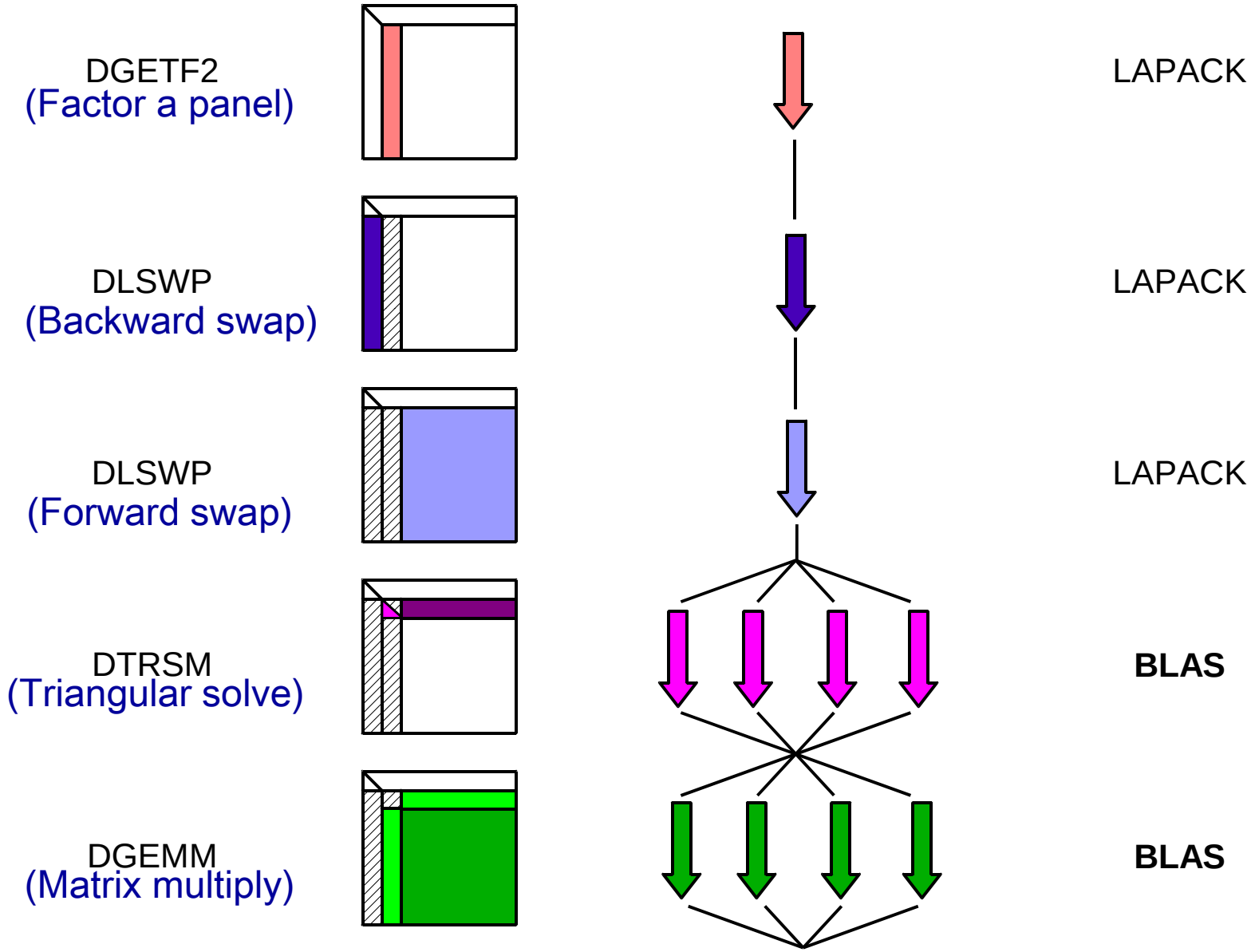
ManyCore - Parallelism for the Masses

- **We are looking at the following concepts in designing the next numerical library implementation**
  - **Dynamic Data Driven Execution**
  - **Self Adapting**
  - **Block Data Layout**
  - **Mixed Precision in the Algorithm**
  - **Exploit Hybrid Architectures**
  - **Fault Tolerant Methods**

Major Changes to Software

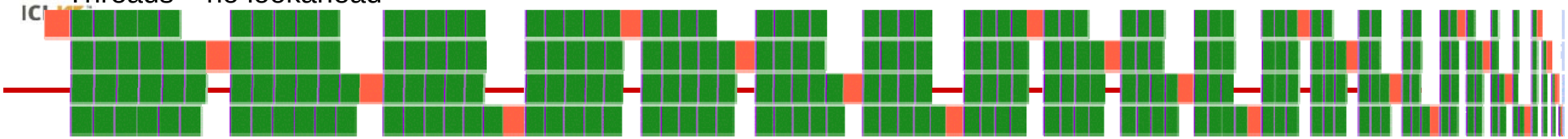
- **Must rethink the design of our software**
  - **Another disruptive technology**
    - **Similar to what happened with cluster computing and message passing**
  - **Rethink and rewrite the applications, algorithms, and software**
- **Numerical libraries for example will change**
  - **For example,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

Steps in the LAPACK LU

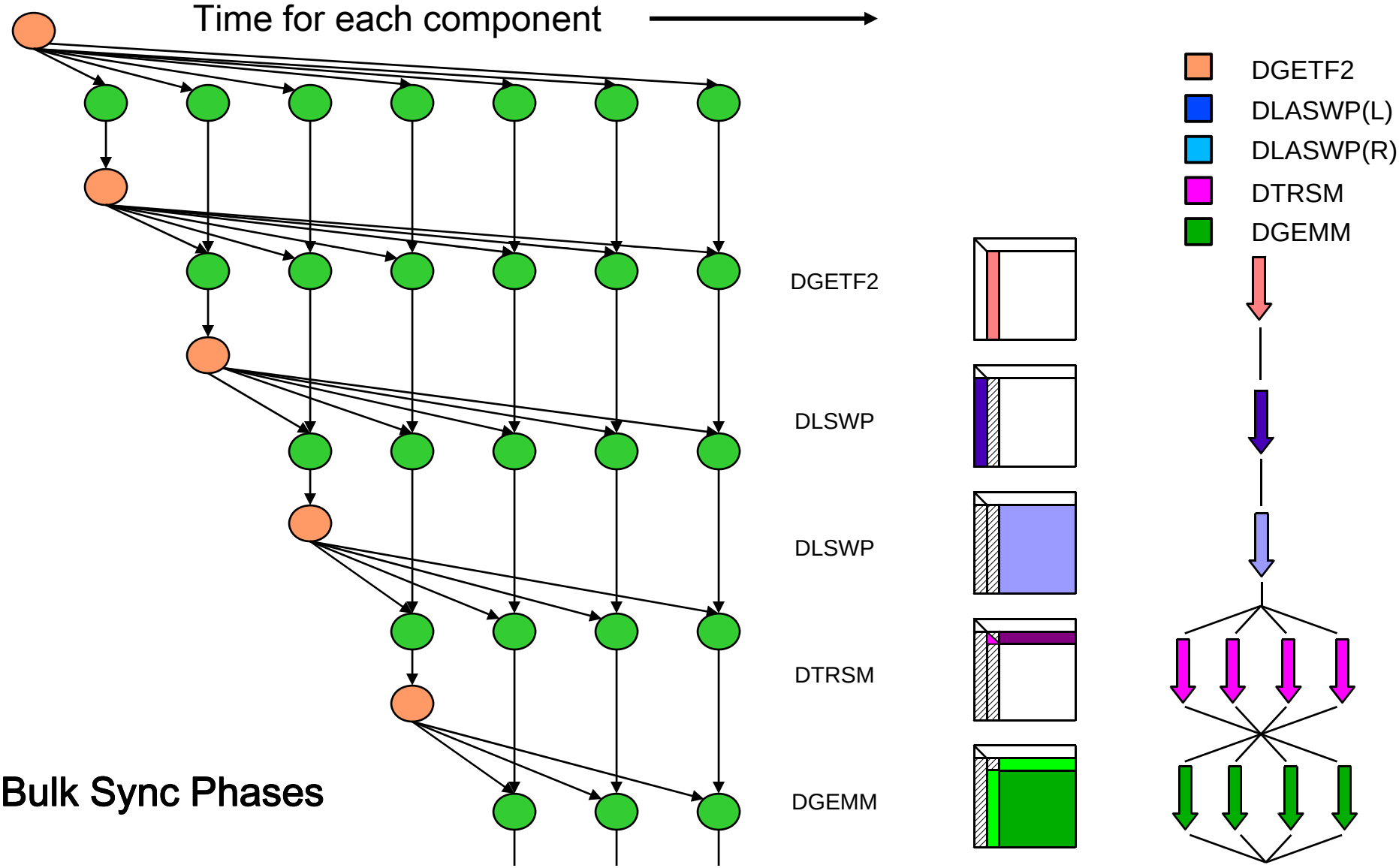


LU Timing Profile (4 processor system)

Threads – no lookahead

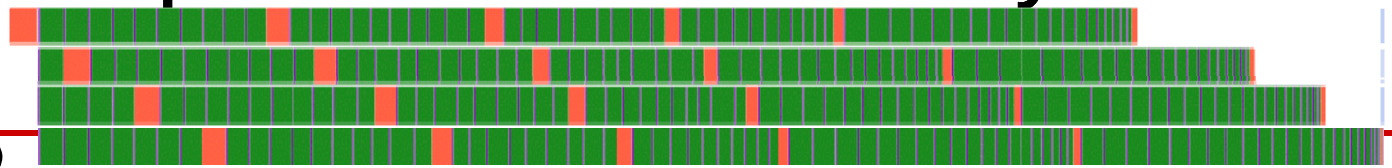


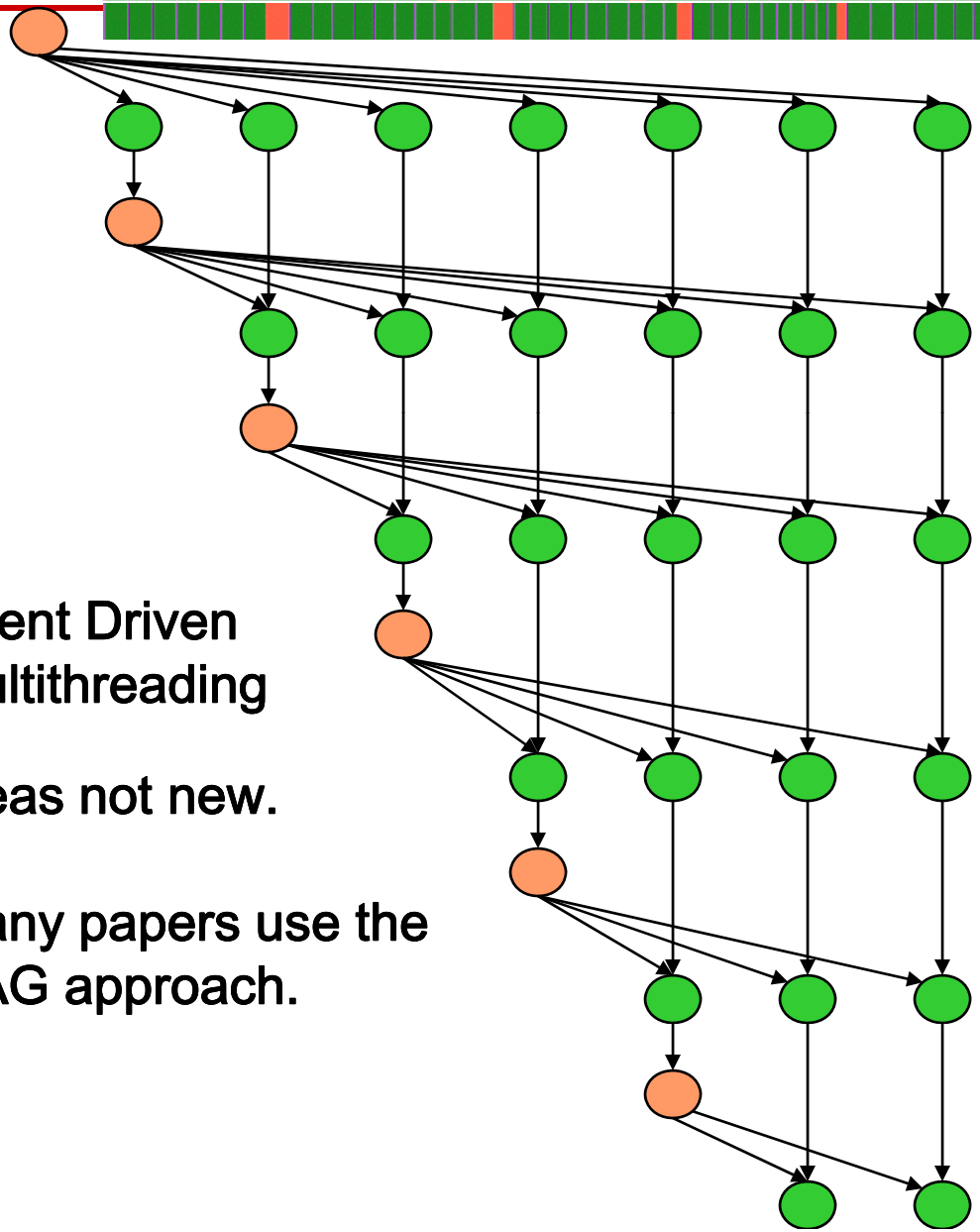
Time for each component



Bulk Sync Phases

Adaptive Lookahead - Dynamic





Event Driven Multithreading

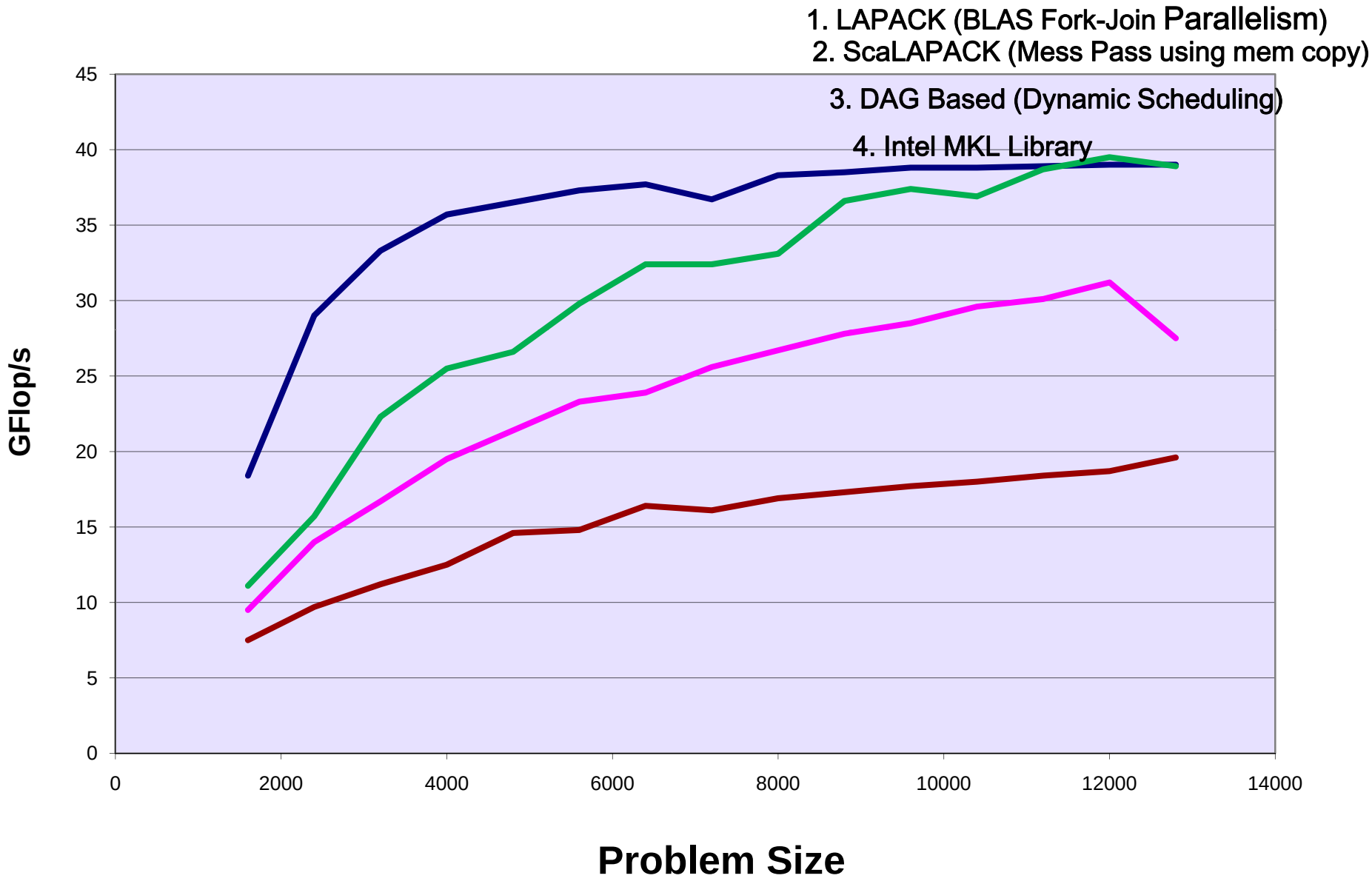
Ideas not new.

Many papers use the DAG approach.

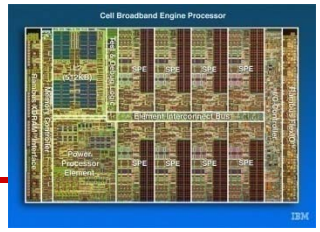
```
while(1)
    fetch_task();
    switch(task.type) {
        case PANEL:
            dgetf2();
            update_progress();
        case COLUMN:
            dlaswp();
            dtrsm();
            dgemm();
            update_progress();
        case END:
            for()
                dlaswp();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

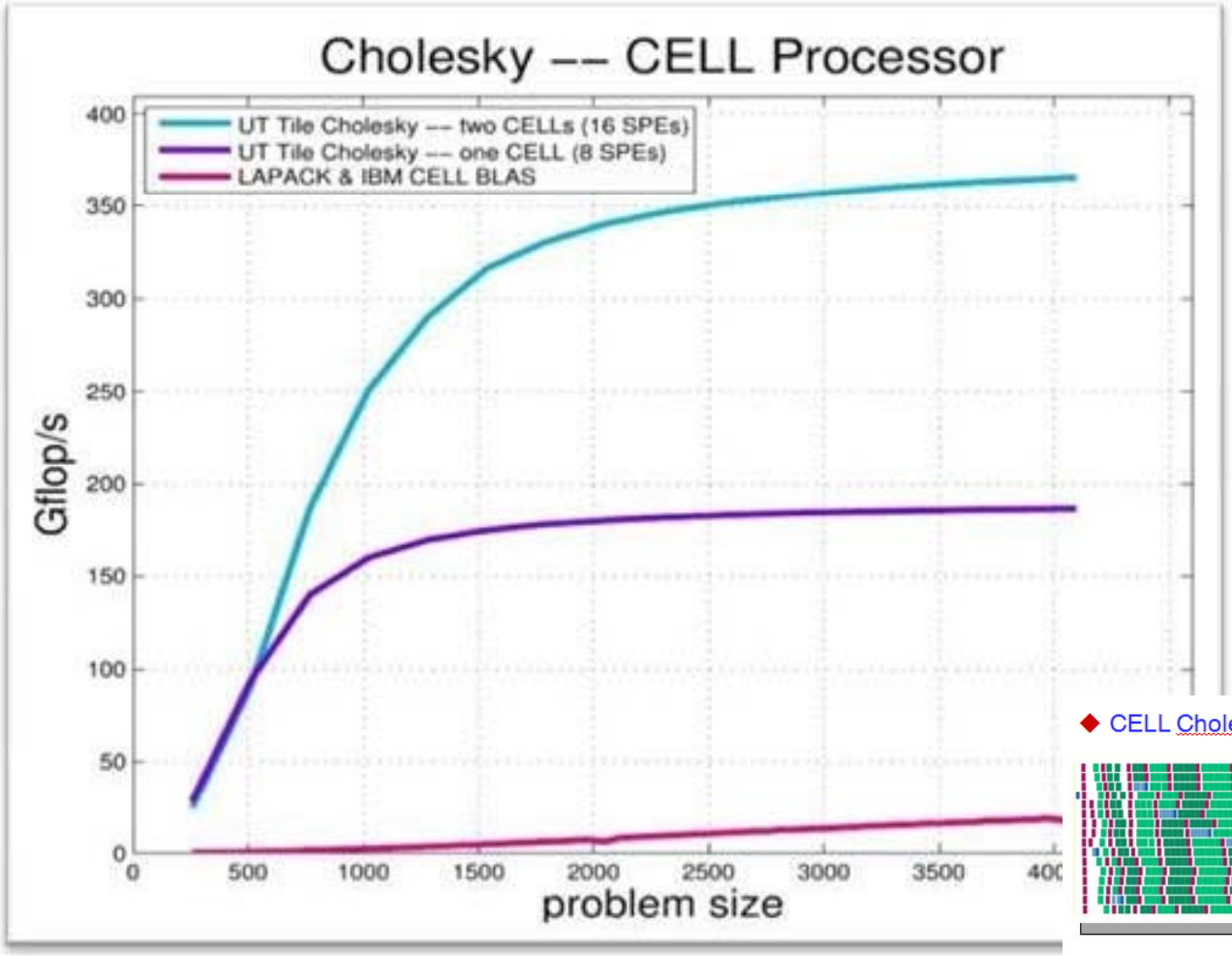
Reorganizing algorithms to use this approach

LU – 16 Core
(8 Socket - Dual Core Opteron 2.2 GHz)



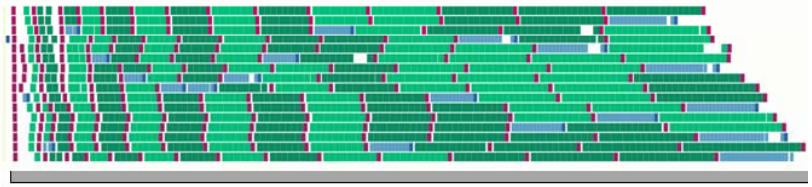
Cholesky on the CELL



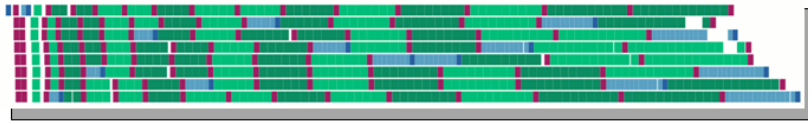


- **1 CELL (8 SPEs)**
  - **186 Gflop/s**
  - **91 % peak**
  - **97 % SGEMM peak**
- **2 CELLs (16 SPEs)**
  - **365 Gflop/s**
  - **89 % peak**
  - **95 % SGEMM peak**

◆ CELL Cholesky – 16 cores



◆ CELL Cholesky – 8 cores



Single precision results on the Cell

Performance of Single Precision on Conventional Processors

- **Realized have the similar situation on our commodity processors.**
  - **That is, SP is 2X as fast as DP on many systems**
- **The Intel Pentium and AMD Opteron have SSE2**
  - **2 flops/cycle DP**
  - **4 flops/cycle SP**
- **IBM PowerPC has AltiVec**
  - **8 flops/cycle SP**
  - **4 flops/cycle DP**
    - **No DP on AltiVec**

	Size	SGEMM/ DGEMM	Size	SGEMV/ DGEMV
AMD Opteron 246	3000	2.00	5000	1.70
UltraSparc-IIe	3000	1.64	5000	1.66
Intel PIII Coppermine	3000	2.03	5000	2.09
PowerPC 970	3000	2.04	5000	1.44
Intel Woodcrest	3000	1.81	5000	2.18
Intel XEON	3000	2.04	5000	1.82
Intel Centrino Duo	3000	2.71	5000	2.21

Single precision is faster because:
- Higher parallelism in SSE/vector units
- Reduced data motion
- Higher locality in cache

32 or 64 bit Floating Point Precision?

- **A long time ago 32 bit floating point was used**
  - **Still used in scientific apps but limited**
- **Most apps use 64 bit floating point**
  - **Accumulation of round off error**
    - **A 10 TFlop/s computer running for 4 hours performs > 1 Exaflop (10^{18}) ops.**
  - **Ill conditioned problems**
  - **IEEE SP exponent bits too few (8 bits, $10^{\pm 38}$)**
  - **Critical sections need higher precision**
    - **Sometimes need extended precision (128 bit fl pt)**
  - **However some can get by with 32 bit fl pt in some parts**
- **Mixed precision a possibility**
  - **Approximate in lower precision and then refine or improve solution to high precision.**

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.**

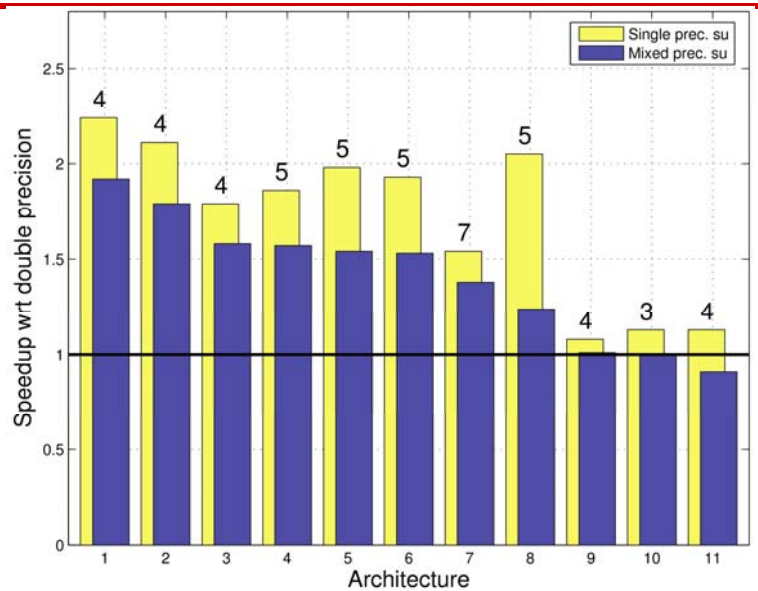
L U = lu(A)	SINGLE	$O(n^3)$
x = L\(U\b)	SINGLE	$O(n^2)$
r = b – Ax	DOUBLE	$O(n^2)$
WHILE || r || not small enough		
z = L\(U\r)	SINGLE	$O(n^2)$
x = x + z	DOUBLE	$O(n^1)$
r = b – Ax	DOUBLE	$O(n^2)$
END		

- **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; $O(10^8)$**

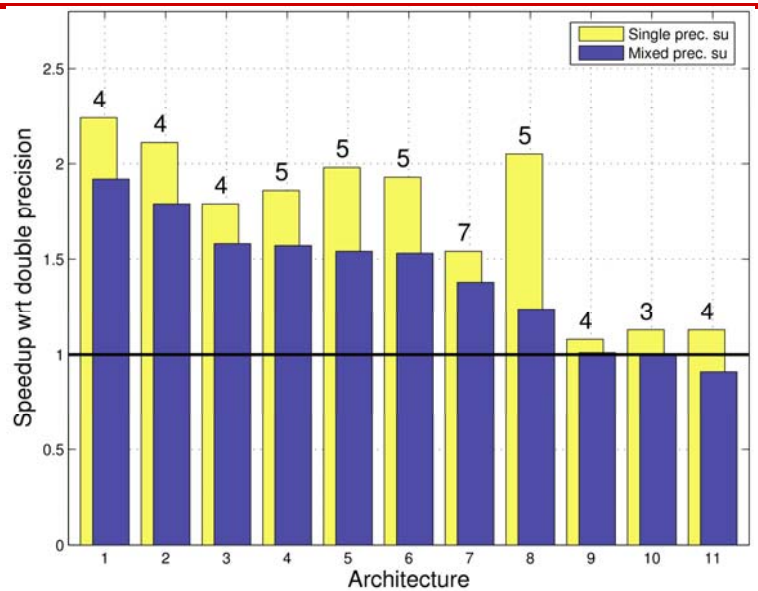
Results for Mixed Precision Iterative Refinement for Dense *Ax* = *b*



	Architecture (BLAS)
1	Intel Pentium III Coppermine (Goto)
2	Intel Pentium III Katmai (Goto)
3	Sun UltraSPARC IIe (Sunperf)
4	Intel Pentium IV Prescott (Goto)
5	Intel Pentium IV-M Northwood (Goto)
6	AMD Opteron (Goto)
7	Cray X1 (libsci)
8	IBM Power PC G5 (2.7 GHz) (VecLib)
9	Compaq Alpha EV6 (CXML)
10	IBM SP Power3 (ESSL)
11	SGI Octane (ATLAS)

- Single precision is faster than DP because:
  - **Higher parallelism within vector units**
    - **4 ops/cycle (usually) instead of 2 ops/cycle**
  - **Reduced data motion**
    - **32 bit data instead of 64 bit data**
  - **Higher locality in cache**
    - **More data items in cache**

Results for Mixed Precision Iterative Refinement for Dense *Ax* = *b*



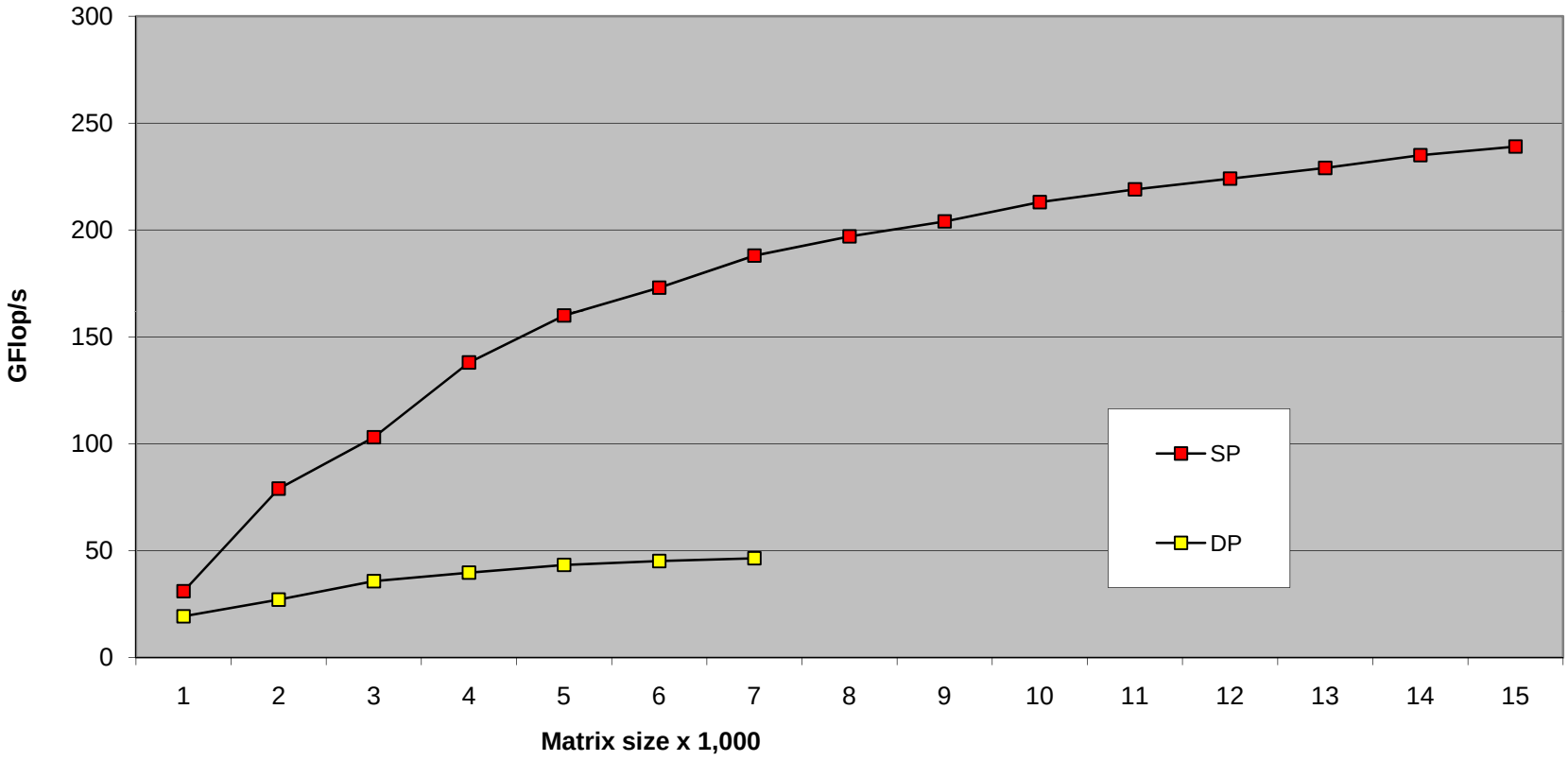
	Architecture (BLAS)
1	Intel Pentium III Coppermine (Goto)
2	Intel Pentium III Katmai (Goto)
3	Sun UltraSPARC IIe (Sunperf)
4	Intel Pentium IV Prescott (Goto)
5	Intel Pentium IV-M Northwood (Goto)
6	AMD Opteron (Goto)
7	Cray X1 (libsci)
8	IBM Power PC G5 (2.7 GHz) (VecLib)
9	Compaq Alpha EV6 (CXML)
10	IBM SP Power3 (ESSL)
11	SGI Octane (ATLAS)

Architecture (BLAS-MPI)	# procs	*n*	DP Solve /SP Solve	DP Solve /Iter Ref	# iter
AMD Opteron (Goto – OpenMPI MX)	32	22627	1.85	1.79	6
AMD Opteron (Goto – OpenMPI MX)	64	32000	1.90	1.83	6

- Single precision is faster than DP because:
  - **Higher parallelism within vector units**
    - **4 ops/cycle (usually) instead of 2 ops/cycle**
  - **Reduced data motion**
    - **32 bit data instead of 64 bit data**
  - **Higher locality in cache**
    - **More data items in cache**

NVIDIA GTX 280 - 1.296 GHz

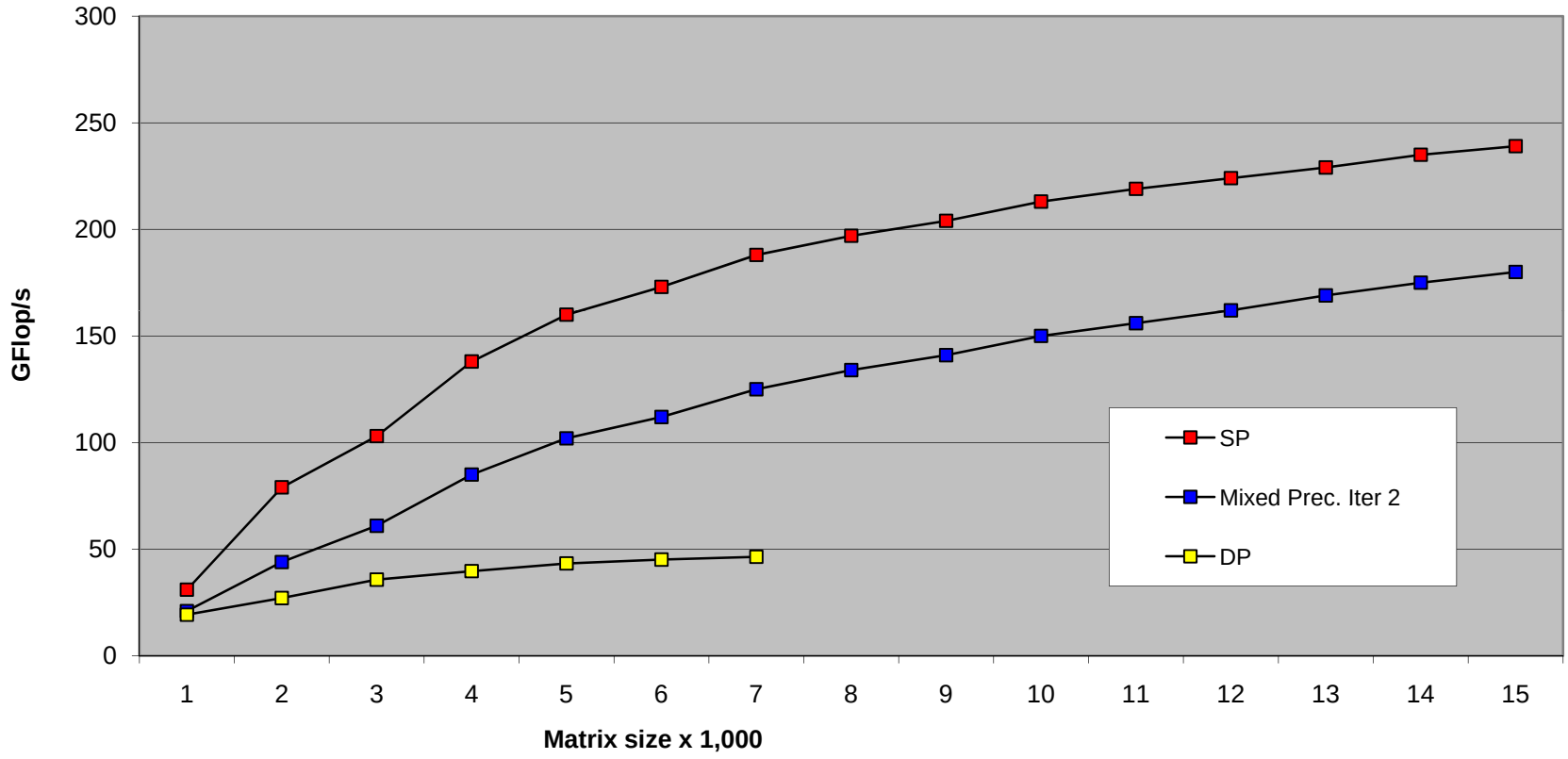
Solving Ax = b in various accuracies, A is SPD



- **Tpeak 864 Gflop/s - 32 bit floating point**
- **Tpeak 86.4 Gflop/s - 64 bit floating point**
- **Intel Xeon 2.33 GHz, host**

NVIDIA GTX 280 - 1.296 GHz

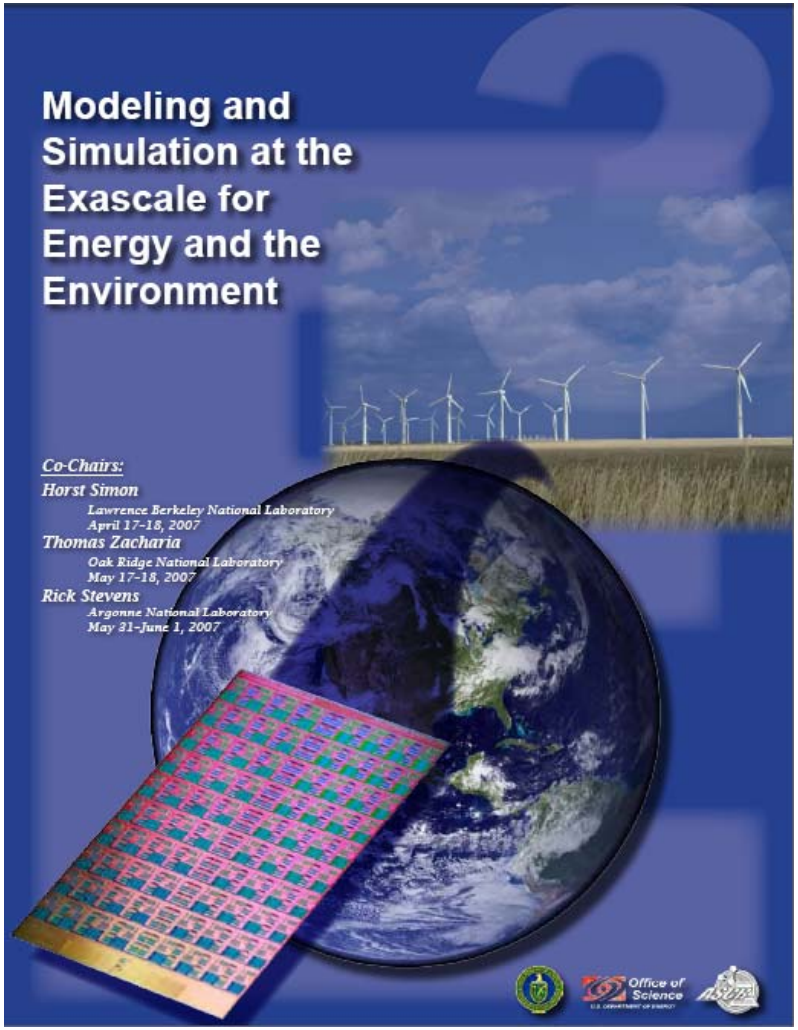
Solving Ax = b in various accuracies, A is SPD



- **Tpeak 864 Gflop/s - 32 bit floating point**
- **Tpeak 86.4 Gflop/s - 64 bit floating point**
- **Intel Xeon 2.33 GHz, host**

2007 DOE Town Hall Meeting on Exascale Computing

- //www.er.doe.gov/ASCR/ProgramDocuments/TownHall.pdf
- **Three Meetings at**
  - **Lawrence Berkeley Nat Lab**
  - **Oak Ridge Nat Lab**
  - **Argonne Nat Lab**
- **Community input on challenges for exascale computing**
- **About 450 participants**



For the Future

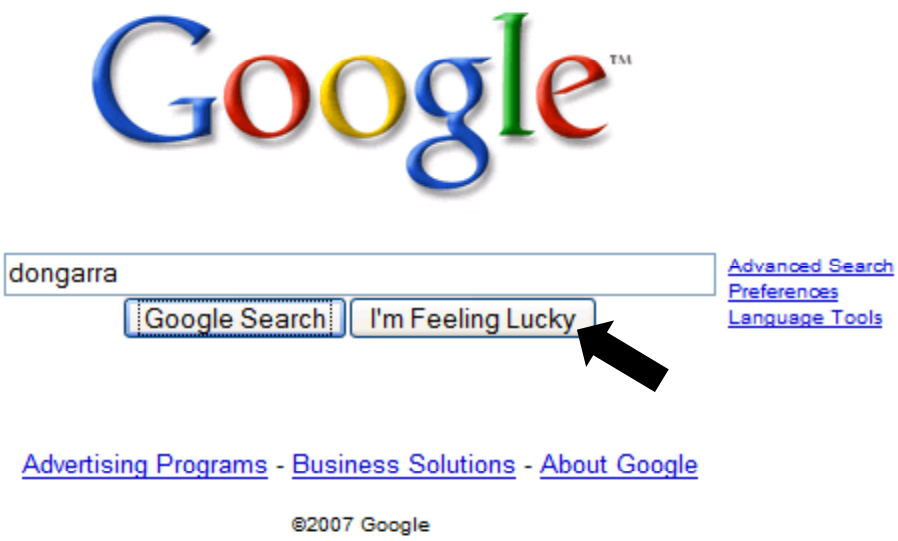
- Exascale systems are likely feasible by 2017±2
- 10-100 Million processing elements (cores or mini-cores) with chips perhaps as dense as 1,000 cores per socket, clock rates will grow more slowly
- 3D packaging likely
- Large-scale optics based interconnects
- 10-100 PB of aggregate memory
- > 10,000's of I/O channels to 10-100 Exabytes of secondary storage, disk bandwidth to storage ratios not optimal for HPC use
- Hardware and software based fault management
- Simulation and multiple point designs will be required to advance our understanding of the design space
- Achievable performance per watt will likely be the primary measure of progress

Advanced Architectures Challenges

- **Performance per watt — stretch goal 100 GF/watt of sustained performance ⇒ >> 10 - 100 MW Exascale system**
  - **Leakage current dominates power consumption**
  - **Active power switching will help manage standby power**
  - **Today Roadrunner ~ 2.35 MW; < 500 MF/watt**
- **Large-scale integration — need to package 10M-100M cores, memory and interconnect**
  - **3D packaging likely, goal of small part classes/counts**
- **Heterogeneous or Homogenous cores?**
  - **Mini cores or leverage from mass market systems**
- **Reliability — needs to increase by 10^3 in faults per PF to achieve MTBF of 1 week**
  - **Integrated HW/SW management of faults**
- **Integrated programming models (PGAS?)**
  - **Provide a usable programming model for hosting existing and future codes**

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.**
- **Moreover, the return on investment is more favorable to software.**
  - **Hardware has a half-life measured in years, while software has a half-life measured in decades.**
- **High Performance Ecosystem out of balance**
  - **Hardware, OS, Compilers, Software, Algorithms, Applications**
    - **No Moore's Law for software, algorithms and applications**

Collaborators / Support

Marc Baboulin UCoimbra
Alfredo Buttari, ENS/INRIA
Bilel Hadri, UTK
Julien Langou, UColorado
Julie Langou, UTK
Hatem Ltaief, UTK
Piotr Luszczek, MathWorks
Jakub Kurzak, UTK
Stan Tomov, UTK