Future Directions in High Performance Computing

Jack Dongarra
INNOVATIVE COMPUTING LABORATORY

University of Tennessee
Oak Ridge National Laboratory
University of Manchester

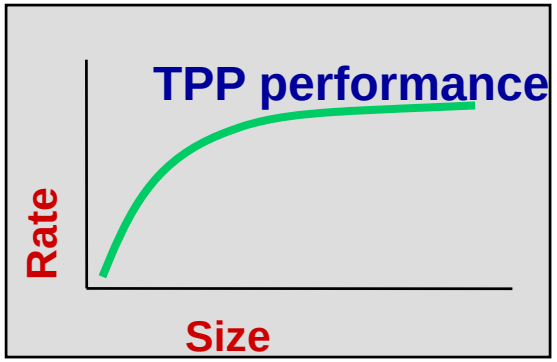
Outline

- **Top500 Results**
- **Four Important Concepts that Will Effect Math Software**
  - **Effective Use of Many-Core**
  - **Exploiting Mixed Precision in Our Numerical Computations**
  - **Self Adapting / Auto Tuning of Software**
  - **Fault Tolerant Algorithms**

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

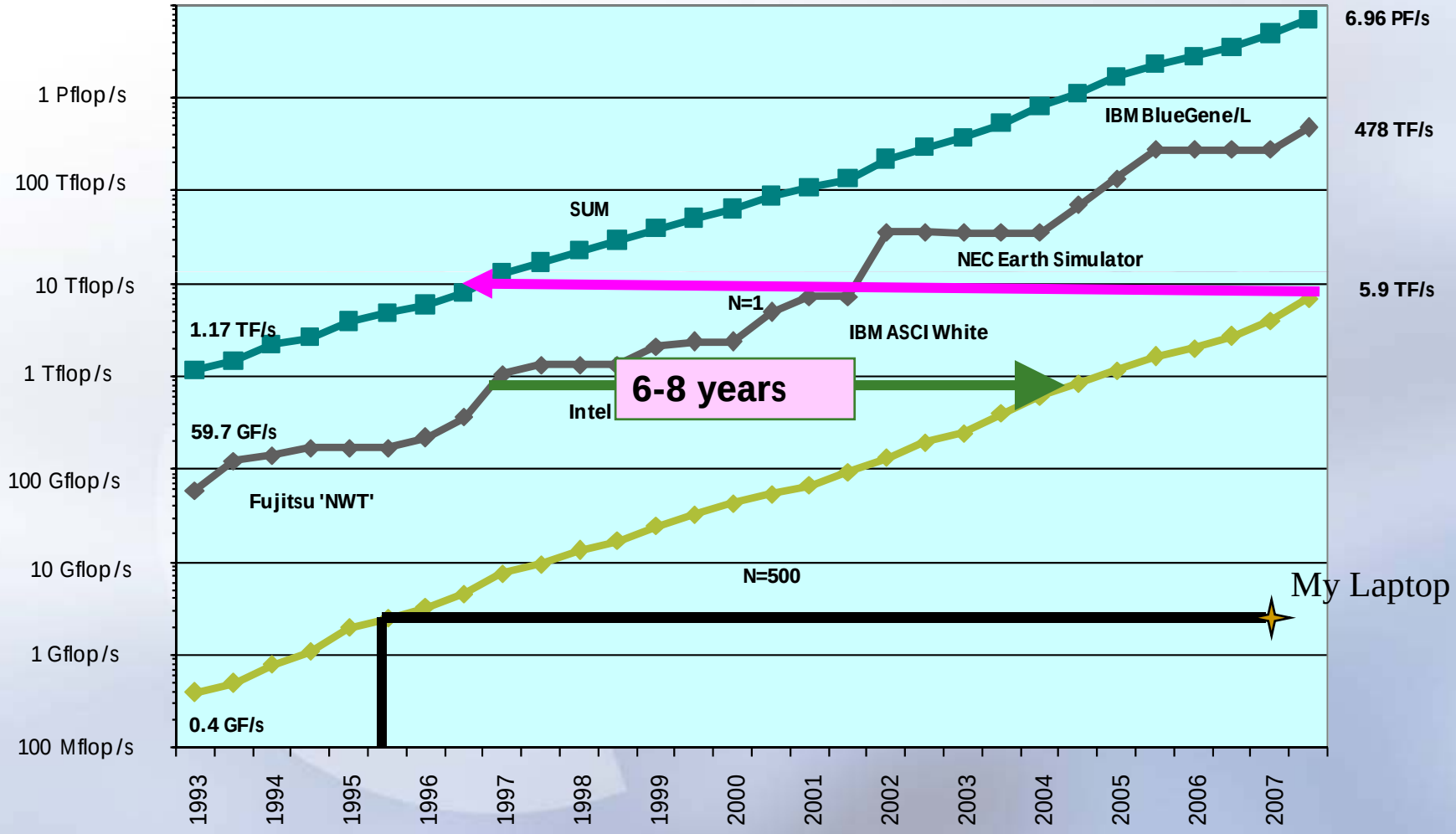
- Listing of the 500 most powerful Computers in the World
- Yardstick: Rmax from LINPACK MPP

$Ax=b$, *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**

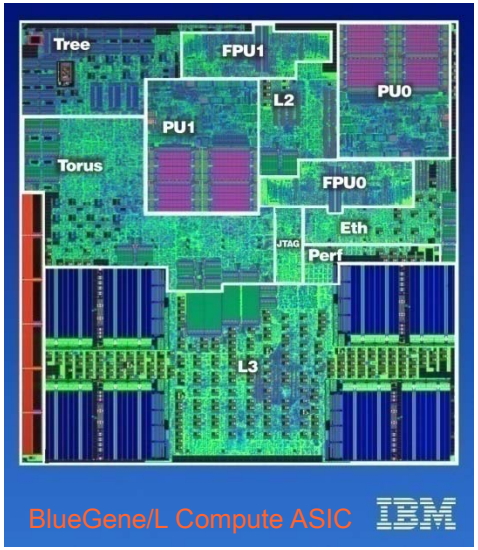
Performance Development



30th Edition: The TOP10

	Manufacturer	Computer	Rmax [TF/s]	Installation Site	Country	Year	#Cores
1	IBM	Blue Gene/L eServer Blue Gene Dual Core .7 GHz	478	DOE Lawrence Livermore Nat Lab	USA	2007 Custom	212,992
2	IBM	Blue Gene/P Quad Core .85 GHz	167	Forschungszentrum Jülich	Germany	2007 Custom	65,536
3	SGI	Altix ICE 8200 Xeon Quad Core 3 GHz	127	SGI/New Mexico Computing Applications Center	USA	2007 Hybrid	14,336
4	HP	Cluster Platform Xeon Dual Core 3 GHz	118	Computational Research Laboratories, TATA SONS	India	2007 Commod	14,240
5	HP	Cluster Platform Dual Core 2.66 GHz	102.8	Government Agency	Sweden	2007 Commod	13,728
6	Cray	Opteron Dual Core 2.4 GHz	102.2	DOE Sandia Nat Lab	USA	2007 Hybrid	26,569
7	Cray	Opteron Dual Core 2.6 GHz	101.7	DOE Oak Ridge National Lab	USA	2006 Hybrid	23,016
8	IBM	eServer Blue Gene/L Dual Core .7 GHz	91.2	IBM Thomas J. Watson Research Center	USA	2005 Custom	40,960
9	Cray	Opteron Dual Core 2.6 GHz	85.4	DOE Lawrence Berkeley Nat Lab	USA	2006 Hybrid	19,320
10	IBM	eServer Blue Gene/L Dual Core .7 GHz	82.1	Stony Brook/BNL, NY Center for Computational Sciences	USA	2006 Custom	36,864

IBM BlueGene/L #1 212,992 Cores



2.6 MWatts (2600 homes)
70,000 ops/s/person

(104 racks, 104x32x32)
212992 procs

Rack
(32 Node boards, 8x8x16)
2048 processors

Node Board
(32 chips, 4x4x2)
16 Compute Cards
64 processors

Compute Card
(2 chips, 2x1x1)
4 processors

Chip
(2 processors)

298/596 TF/s
32 TB DDR

2.9/5.7 TF/s
0.5 TB DDR

90/180 GF/s
16 GB DDR

5.6/11.2 GF/s
1 GB DDR

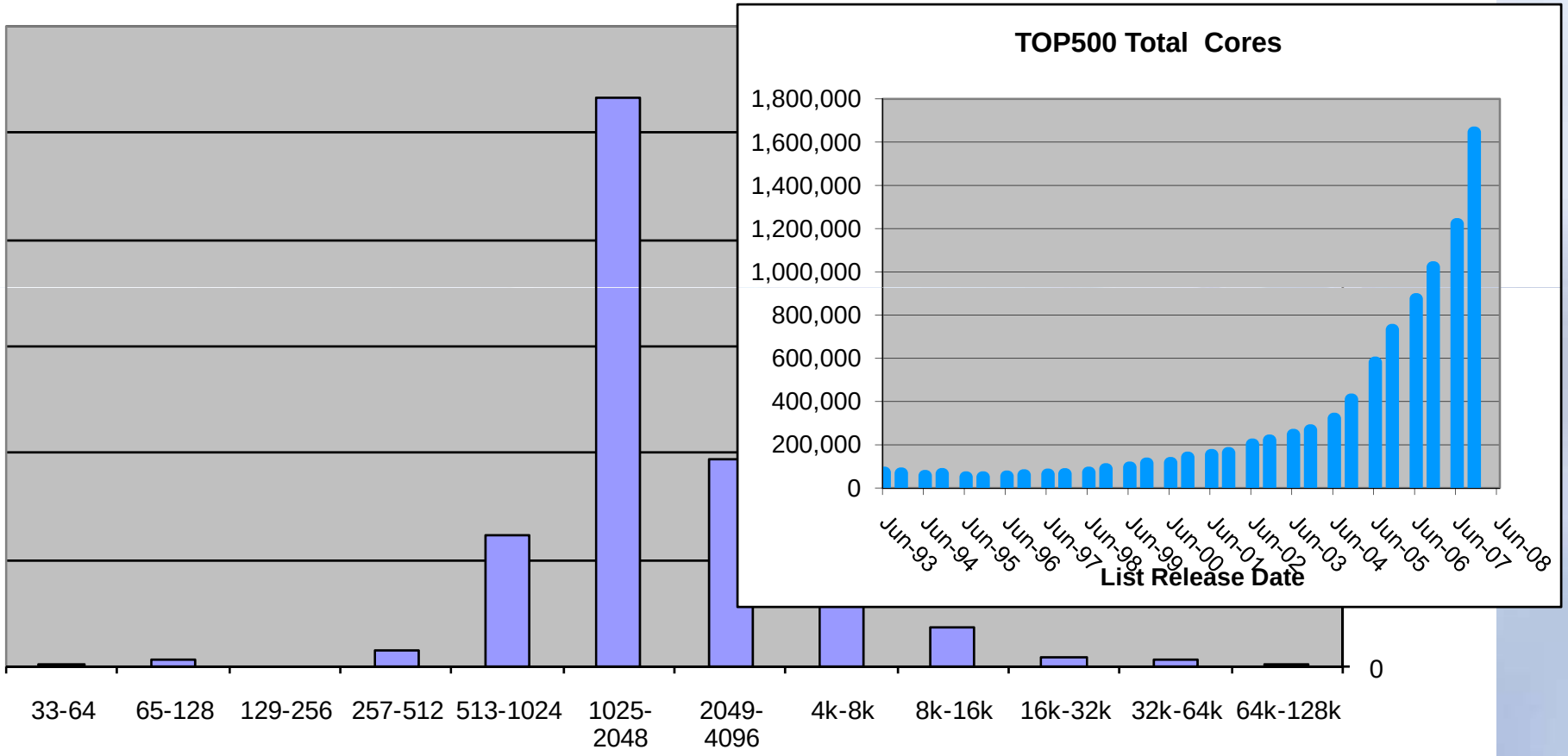
2.8/5.6 GF/s
4 MB (cache)

Full system total of
212,992 cores

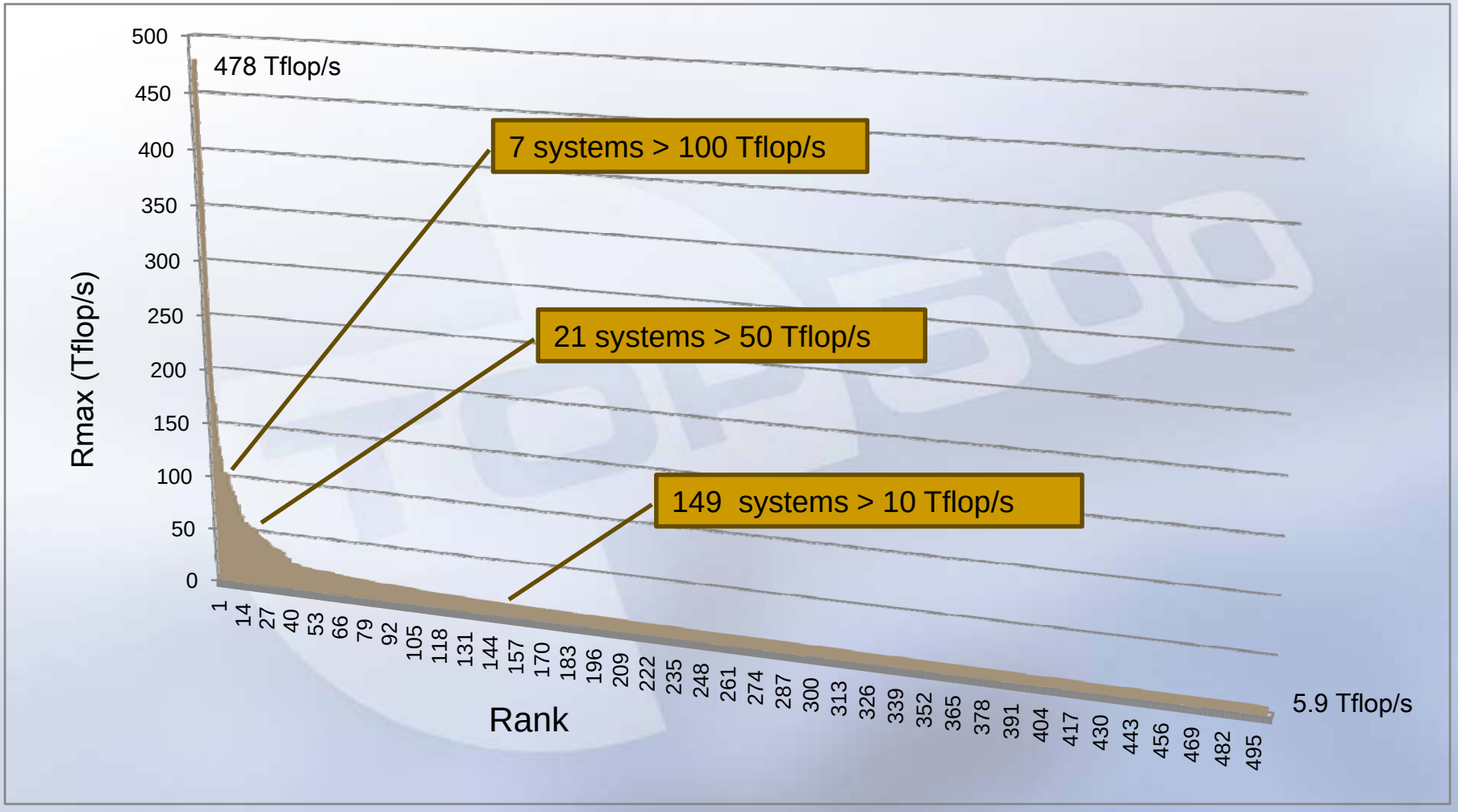
The compute node ASICs include all networking and processor functionality. Each compute ASIC includes two 32-bit superscalar PowerPC 440 embedded cores (note that L1 cache coherence is not maintained between these cores). (20.7K sec about 5.7hours; n=2.5M)

"Fastest Computer"
BG/L 700 MHz 213K proc
104 racks
Peak: 596 Tflop/s
Linpack: 498 Tflop/s
84% of peak

TOP500® Cores per System – November 2007

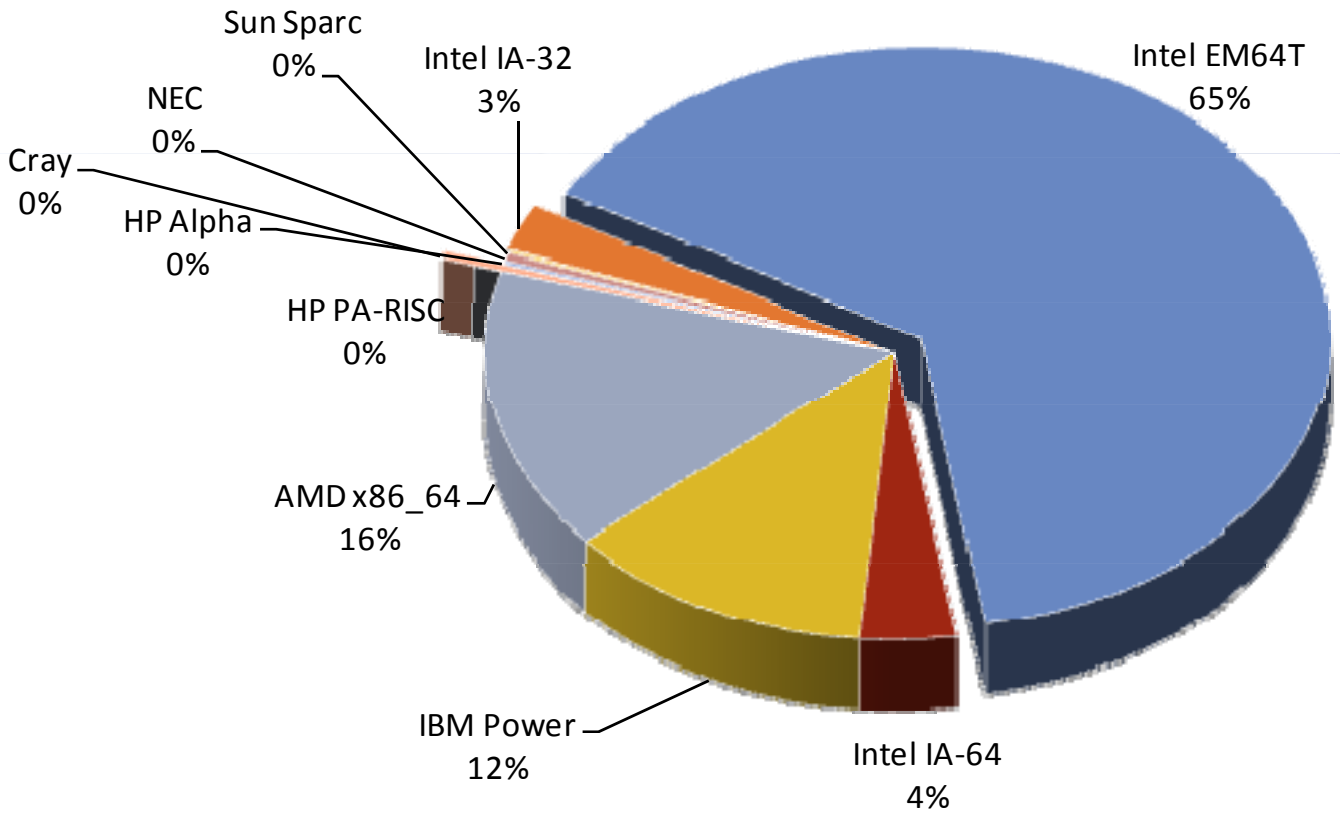


Top500 Systems November 2007



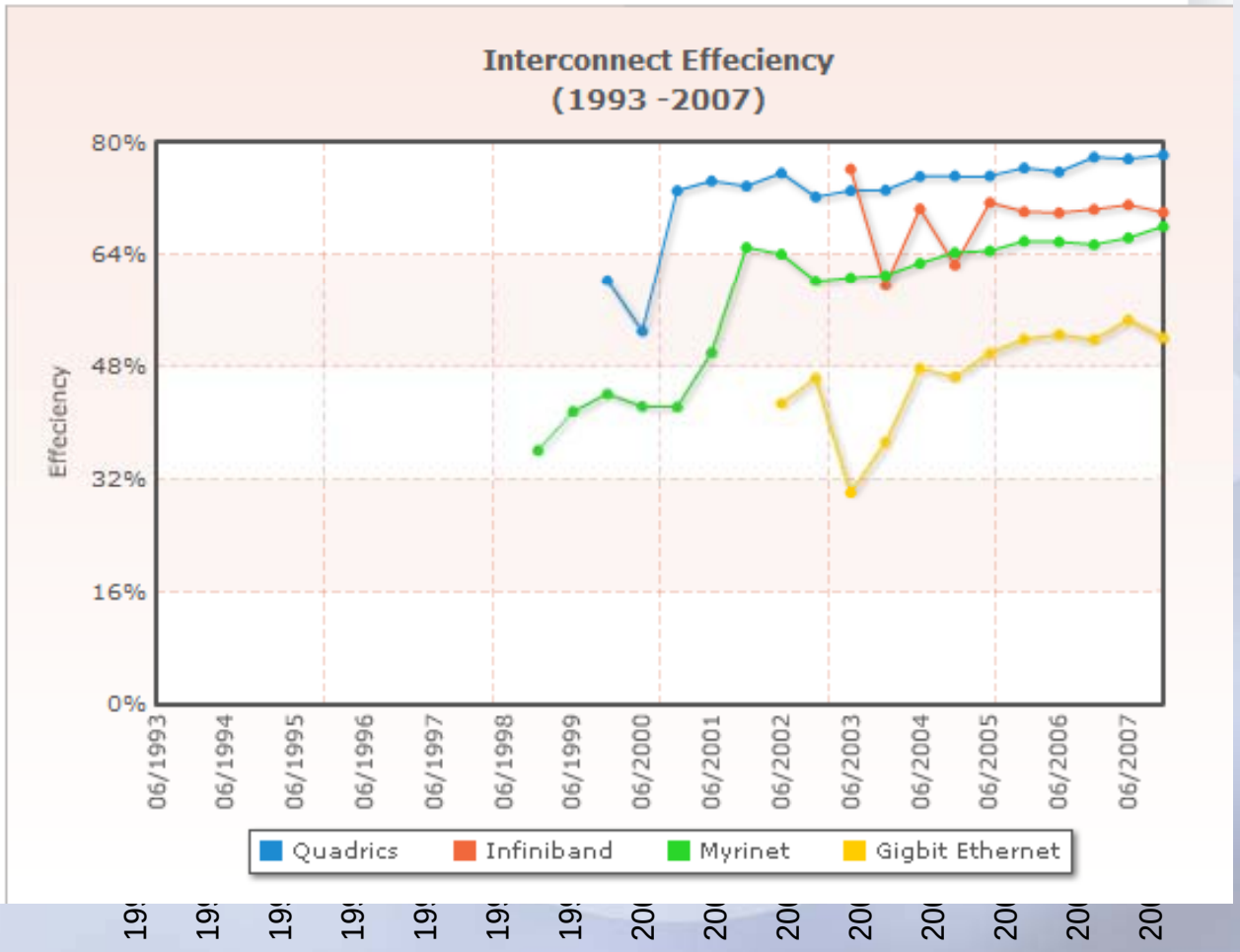
Chips Used in Each of the 500 Systems

72% Intel
12% IBM
16% AMD



Interconnects / Systems

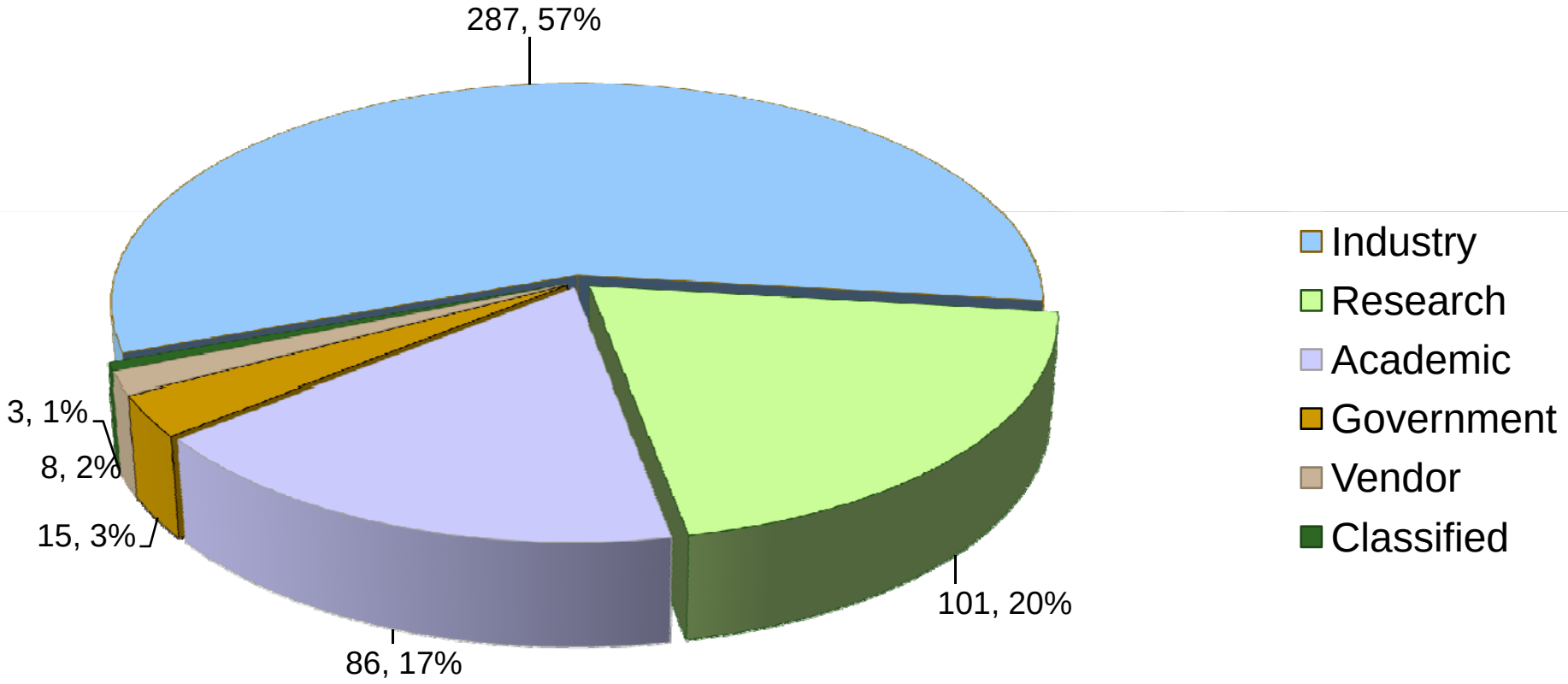
Interconnect Analysis



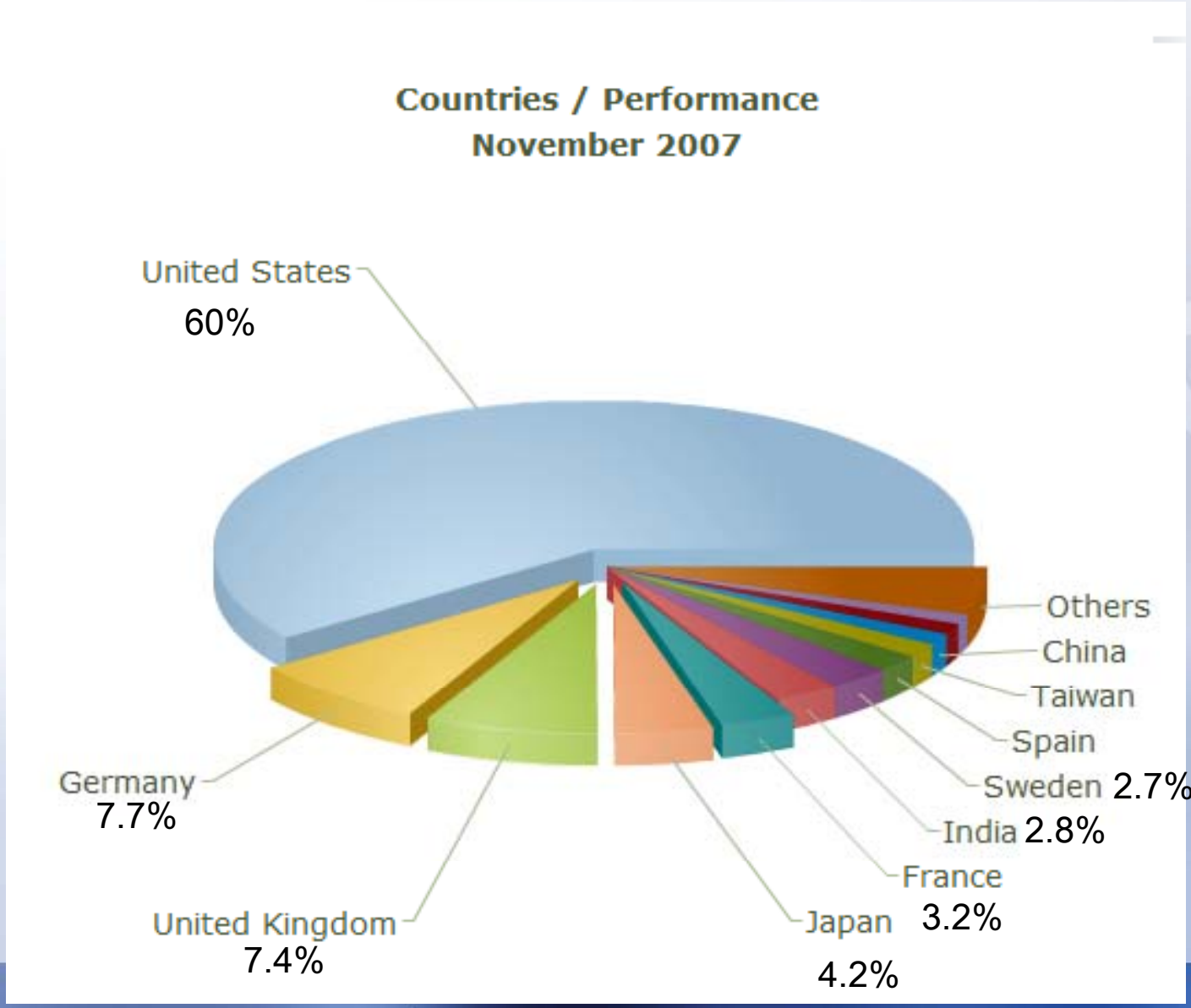
- Others
- Cray Interconnect
- SP Switch
- Crossbar
- Quadrics
- Infiniband (121)
- Myrinet (18)
- Gigabit Ethernet (270)
- N/A

GigE + Infiniband + Myrinet = 82%

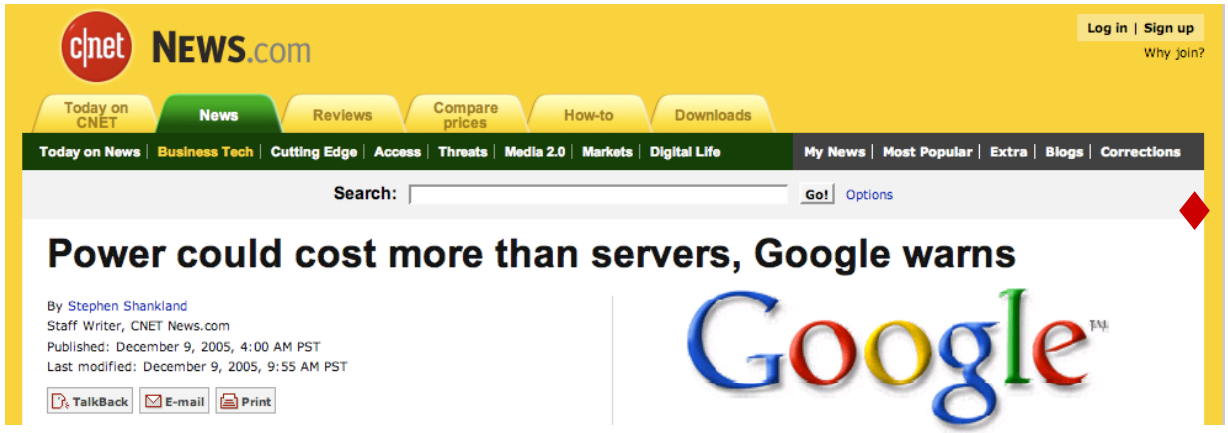
Top500 by Usage



Countries / Performance (Nov 2007)



Power is an Industry Wide Problem



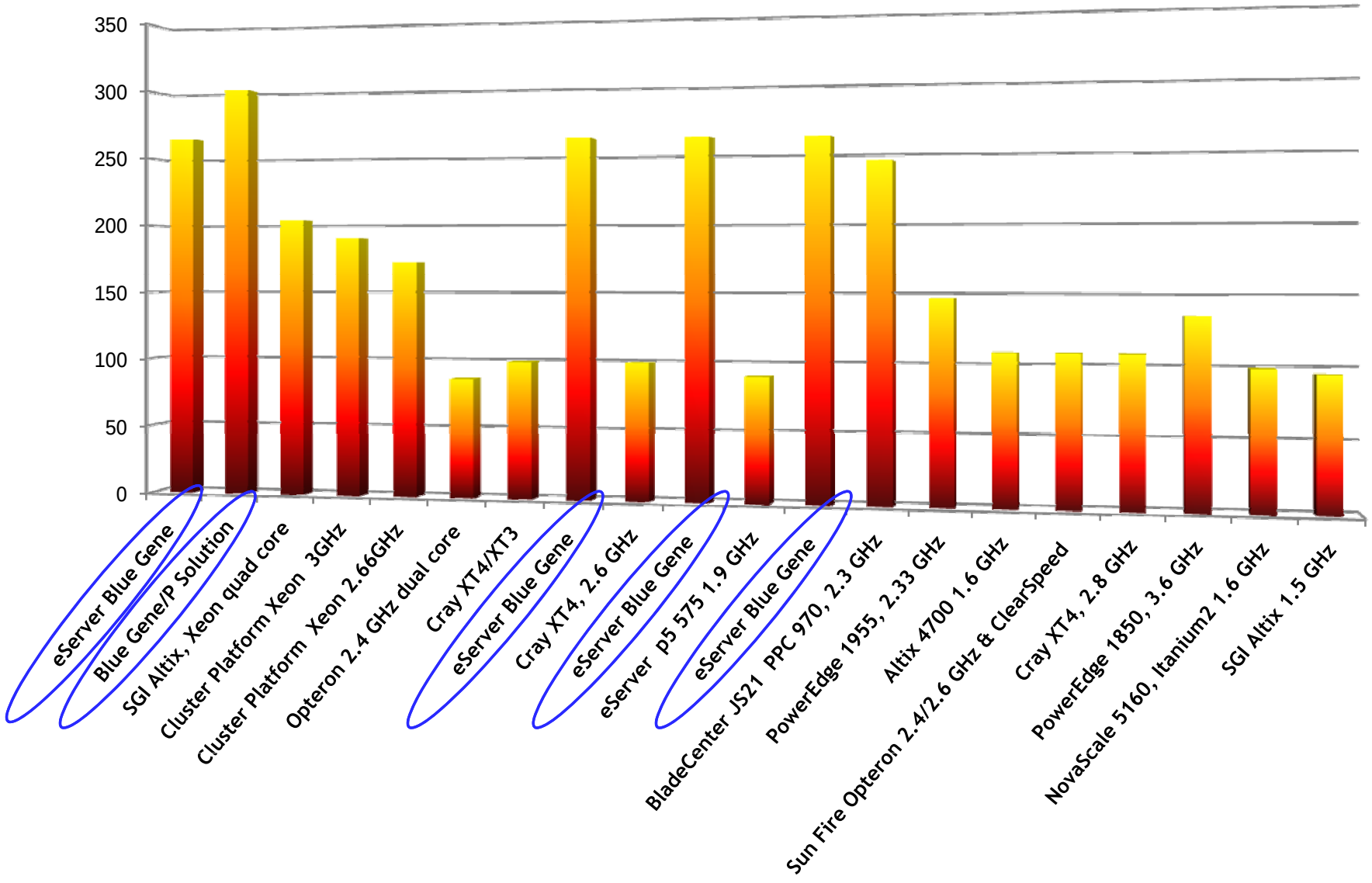
The New York Times **"Hiding in Plain Sight, Google Seeks More Power", by John Markoff, June 14, 2006**

New Google Plant in The Dulles, Oregon, from NYT, June 14, 2006

Google facilities

- leveraging hydroelectric power
  - old aluminum plants
- >500,000 servers worldwide

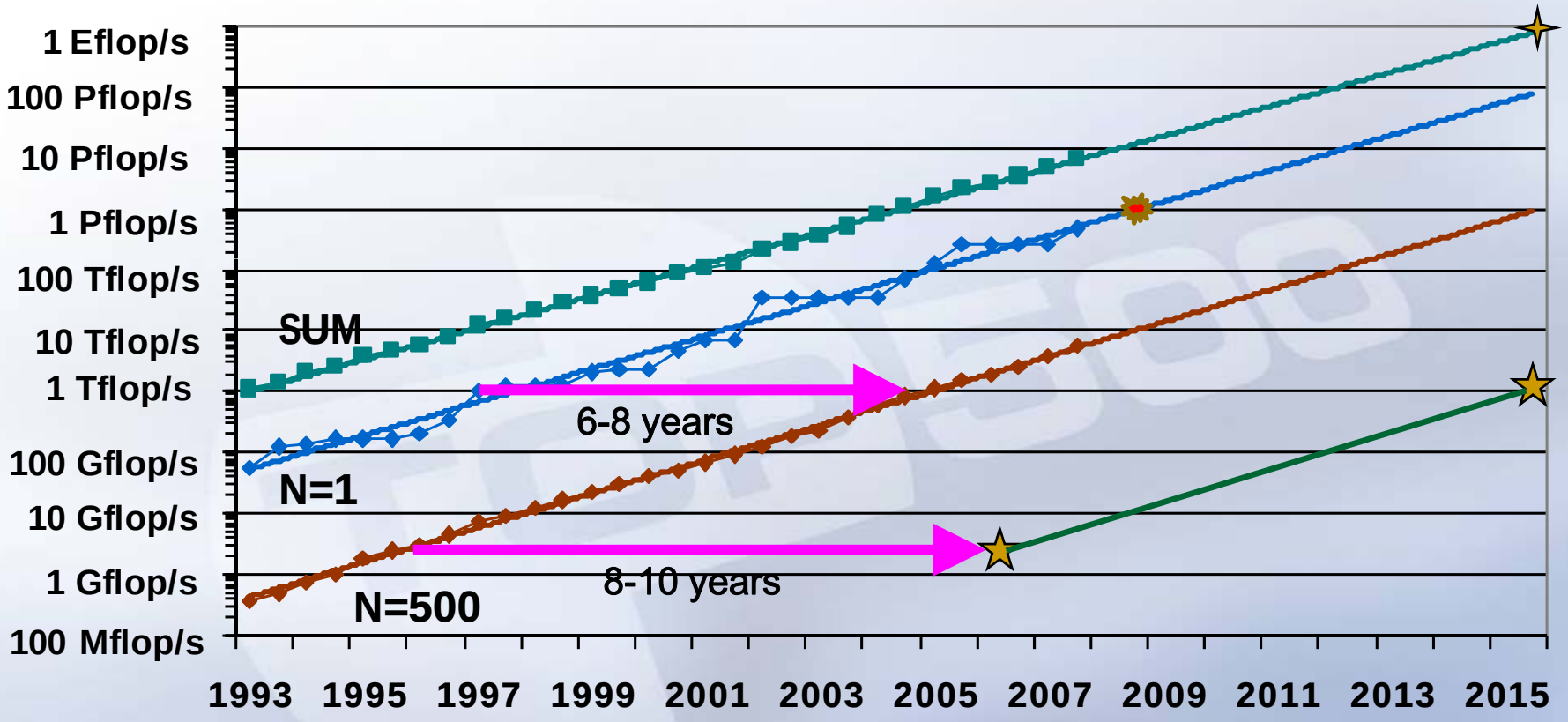
Gflop/KWatt in the Top 20



Green500

Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)	TOP500 Rank*
1	357.23	Science and Technology Facilities Council - Daresbury Laboratory	Blue Gene/P Solution	31.10	121
2	352.25	Max-Planck-Gesellschaft MPI/IPP	Blue Gene/P Solution	62.20	40
3	346.95	IBM - Rochester	Blue Gene/P Solution	124.40	24
4	336.21	Forschungszentrum Juelich (FZJ)	Blue Gene/P Solution	497.60	2
5	310.93	Oak Ridge National Laboratory	Blue Gene/P Solution	70.47	41
6	210.56	Harvard University	eServer Blue Gene Solution	44.80	170
7	210.56	High Energy Accelerator Research Organization /KEK	eServer Blue Gene Solution	44.80	171
8	210.56	IBM - Almaden Research Center	eServer Blue Gene Solution	44.80	172
9	210.56	IBM Research	eServer Blue Gene Solution	44.80	173
10	210.56	IBM Thomas J. Watson Research Center	eServer Blue Gene Solution	44.80	174

Performance Projection

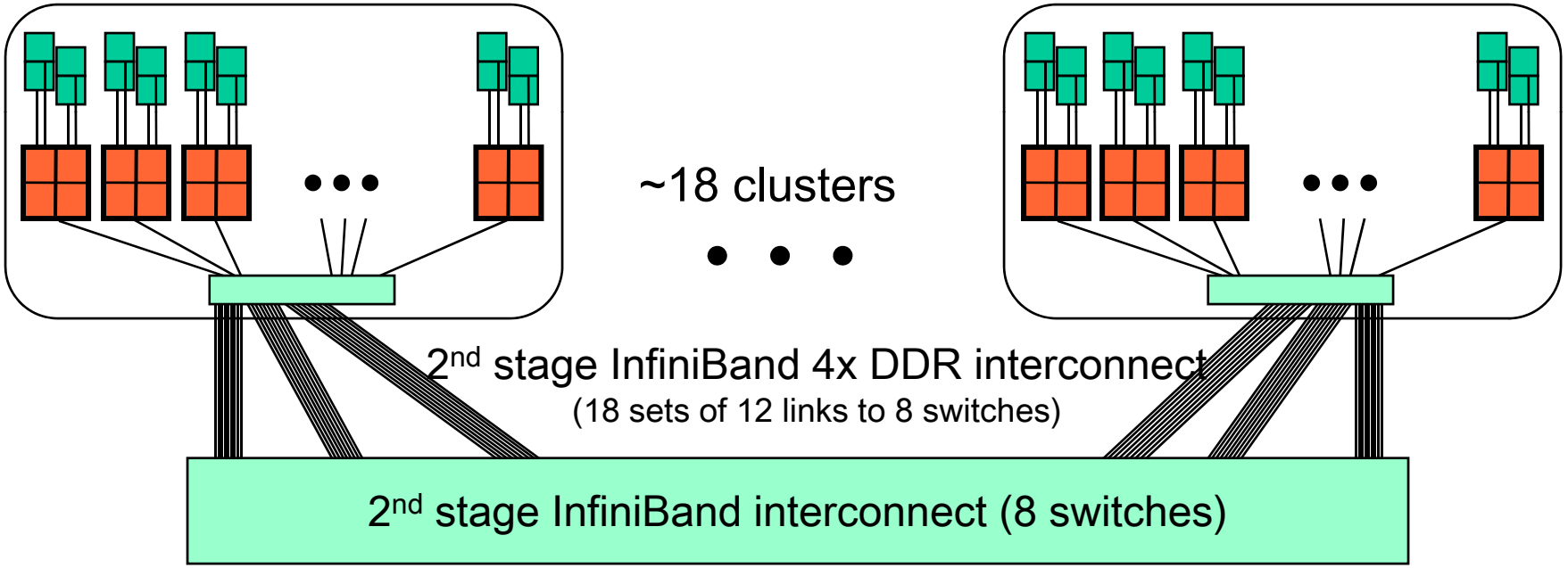


Los Alamos Roadrunner
A Petascale System in 2008

"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



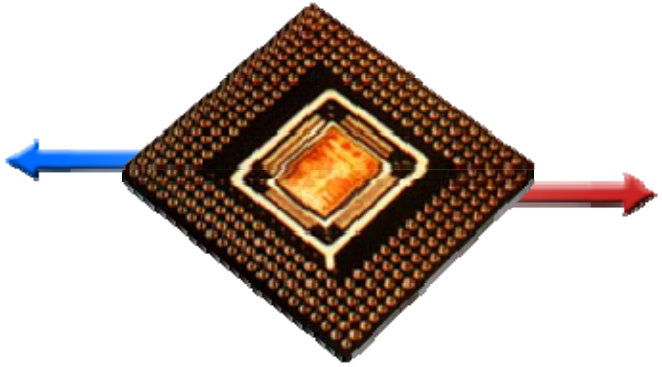
Based on the 100 Gflop/s (DP) Cell chip

Approval by DOE 12/07
First CU being built today
Expect a May Pflop/s run
Full system to LANL in December 2008

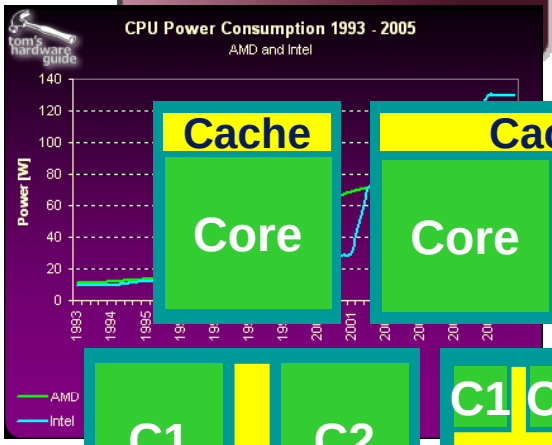
Increasing CPU Performance: A Delicate Balancing Act

Increasing the number of gates into a tight knot and decreasing the cycle time of the processor

Lower Voltage



Increase Clock Rate & Transistor Density



We have seen increasing number of gates on a chip and increasing clock speed.

Heat becoming an unmanageable problem, Intel Processors > 100 Watts

We will not see the dramatic increases in clock speeds in the future.

However, the number of gates on a chip will continue to increase.

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

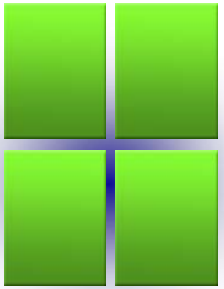
(Bigger # is better)

50% more performance with 20% less power

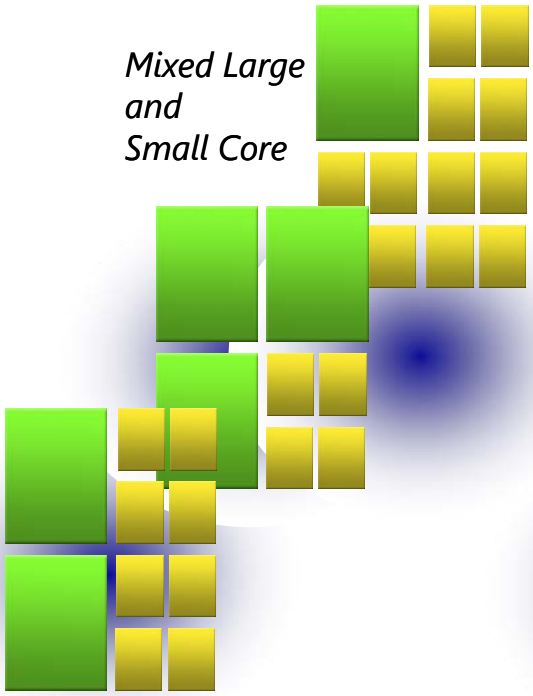
Preferable to use multiple slower devices, than one superfast device

What's Next?

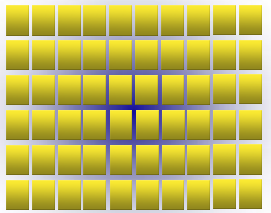
All Large Core



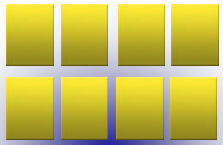
Mixed Large and Small Core



Many Small Cores

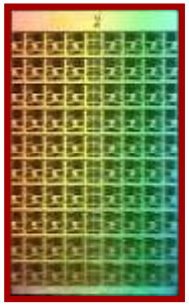


All Small Core

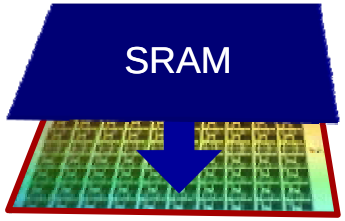


Different Classes of Chips
- Home
- Games / Graphics
- Business
- Scientific

Many Floating-Point Cores

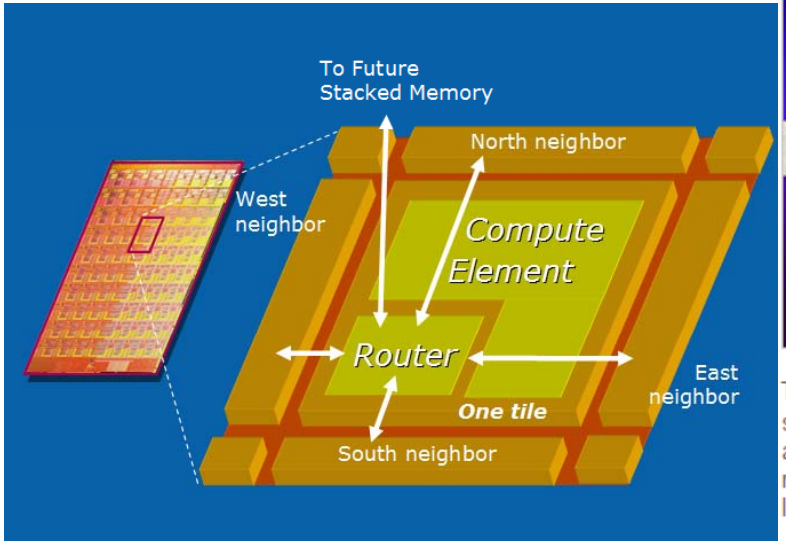


+ 3D Stacked Memory



80 Core

- **Intel's 80 Core chip**
  - **1 Tflop/s**
  - **62 Watts**
  - **1.2 TB/s internal BW**



The New York Times

Technology

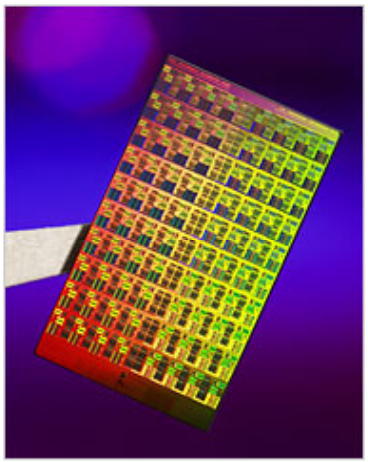
WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPINION

CAMCORDERS | CAMERAS | CELLPHONES | COMPUTERS | HANDHELDS | HOME VIDEO | MUSIC | PERIPHERALS

Intel Prototype May Herald a New Age of Processing

By JOHN MARKOFF
Published: February 12, 2007

SAN FRANCISCO, Feb. 11 — Intel will demonstrate on Monday an experimental computer chip with 80 separate processing engines, or cores, that company executives say provides a model for commercial chips that will be used widely in standard desktop, laptop and server computers within five years.



Intel

The Teraflop Chip has 80 separate processing engines and takes advantage of manufacturing technology that Intel introduced last month.

The new processor, which the company first described as a Teraflop Chip at a conference last year, will be detailed in a technical paper to be presented on the opening day of the International Solid States Circuits Conference, beginning here on Monday.

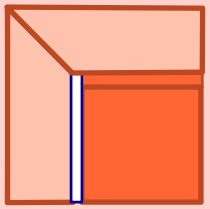
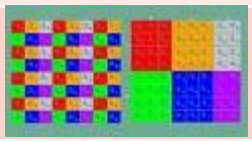
While the chip is not compatible with Intel's current chips, the company said it had already begun design work on a commercial version that would essentially have dozens or even hundreds of Intel-compatible microprocessors laid out in a tiled pattern on a single chip.

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**

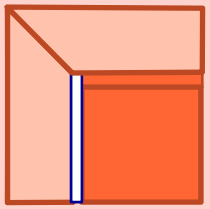
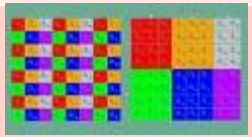
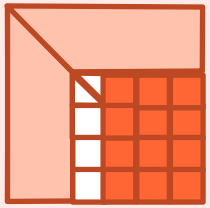
A New Generation of Software:

Parallel Linear Algebra Software for Multicore Architectures (PLASMA)

Algorithms follow hardware evolution in time		
LINPACK (70's) (Vector operations)		Rely on - Level-1 BLAS operations
LAPACK (80's) (Blocking, cache friendly)		Rely on - Level-3 BLAS operations
ScaLAPACK (90's) (Distributed Memory)		Rely on - PBLAS Mess Passing

A New Generation of Software:

Parallel Linear Algebra Software for Multicore Architectures (PLASMA)

Algorithms follow hardware evolution in time		
LINPACK (70's) (Vector operations)		Rely on - Level-1 BLAS operations
LAPACK (80's) (Blocking, cache friendly)		Rely on - Level-3 BLAS operations
ScaLAPACK (90's) (Distributed Memory)		Rely on - PBLAS Mess Passing
PLASMA (00's) New Algorithms (many-core friendly)		Rely on - a DAG/scheduler - block data layout - some extra kernels

Those new algorithms
- have a very low granularity, they scale very well (multicore, petascale computing, …)
- removes a lots of dependencies among the tasks, (multicore, distributed computing)
- avoid latency (distributed computing, out-of-core)
- rely on fast kernels

Those new algorithms need new kernels and rely on efficient scheduling algorithms.

Steps in the LAPACK LU

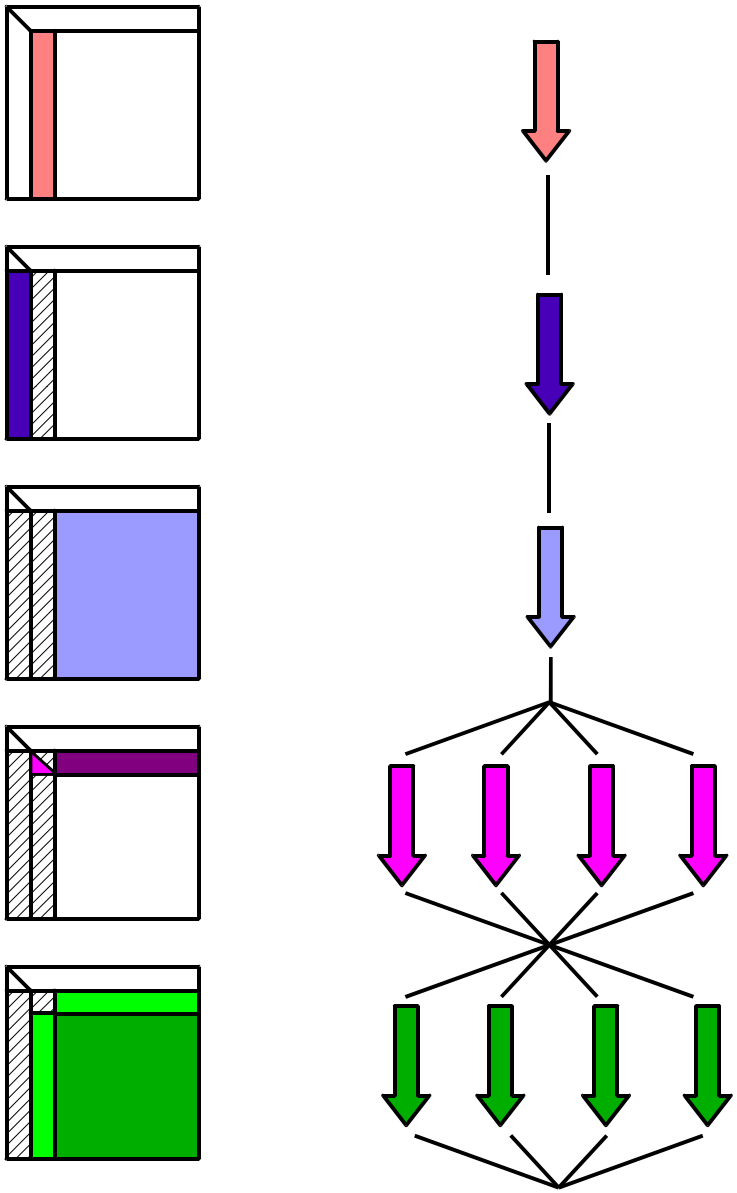
DGETF2
(Factor a panel)

DLSWP
(Backward swap)

DLSWP
(Forward swap)

DTRSM
(Triangular solve)

DGEMM
(Matrix multiply)



LAPACK

LAPACK

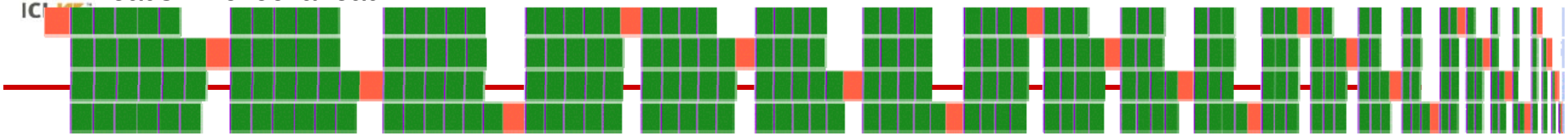
LAPACK

BLAS

BLAS

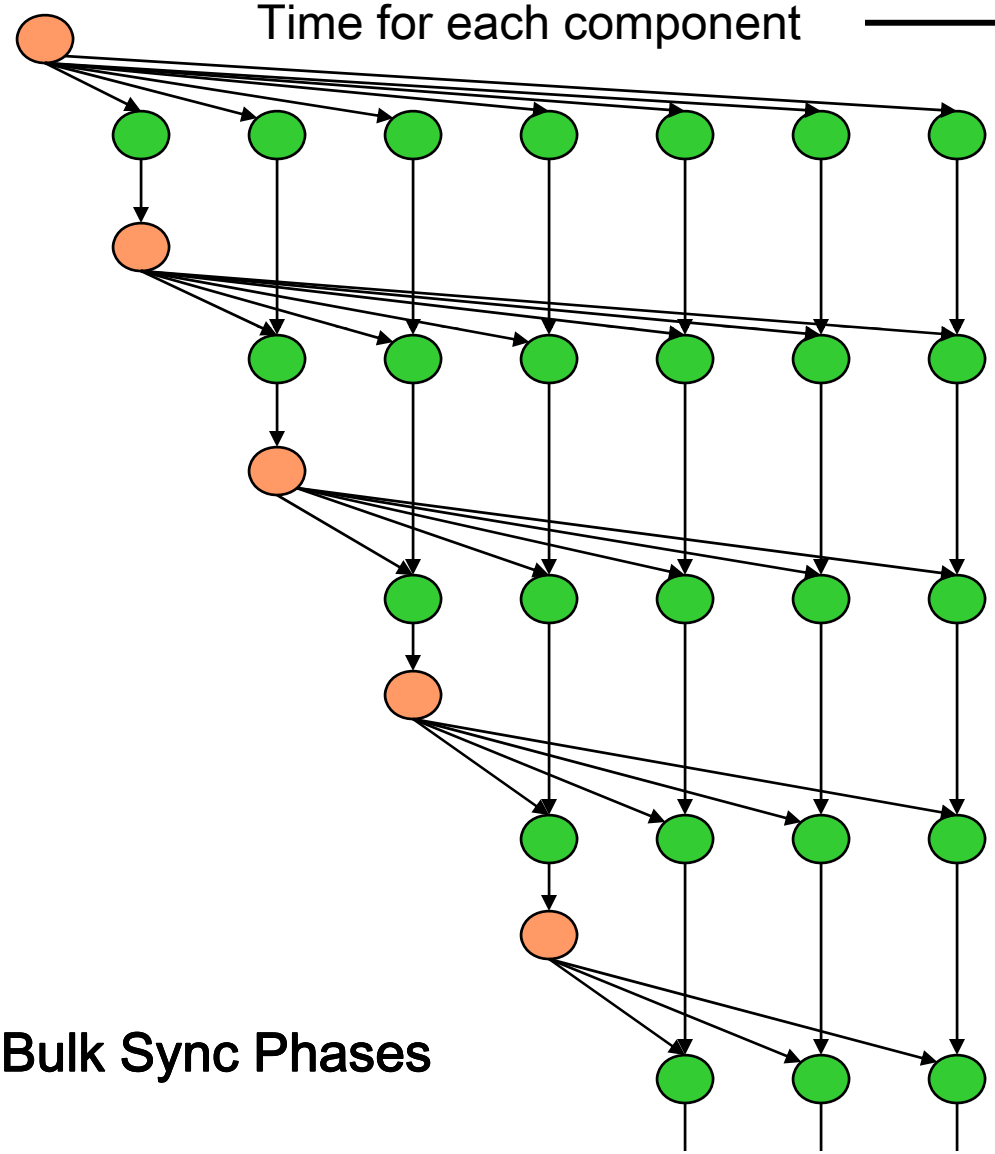
LU Timing Profile (4 processor system)

Threads – no lookahead



Time for each component

1D decomposition and SGI Origir



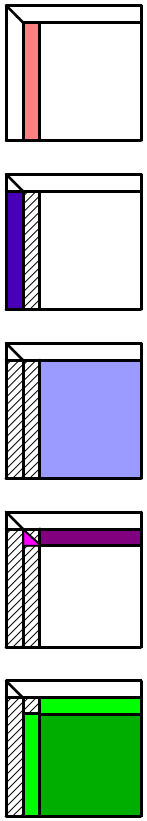
DGETF2

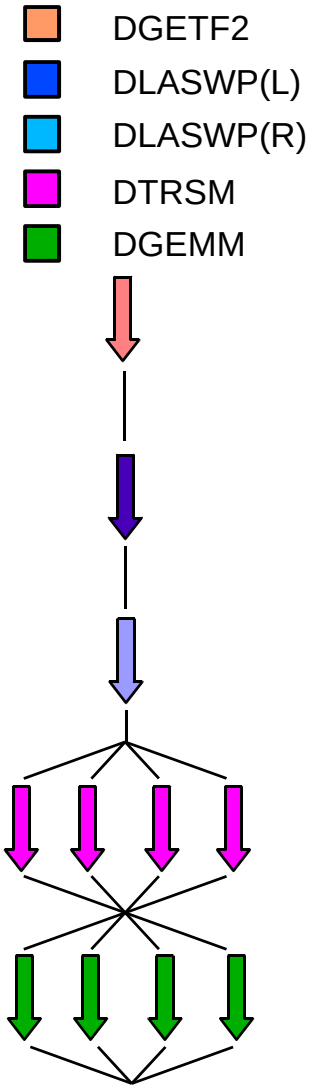
DLSWP

DLSWP

DTRSM

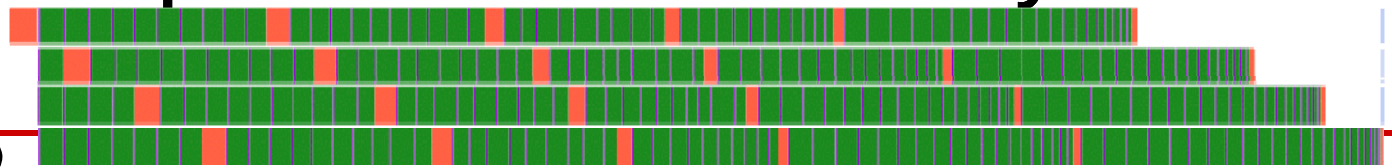
DGEMM

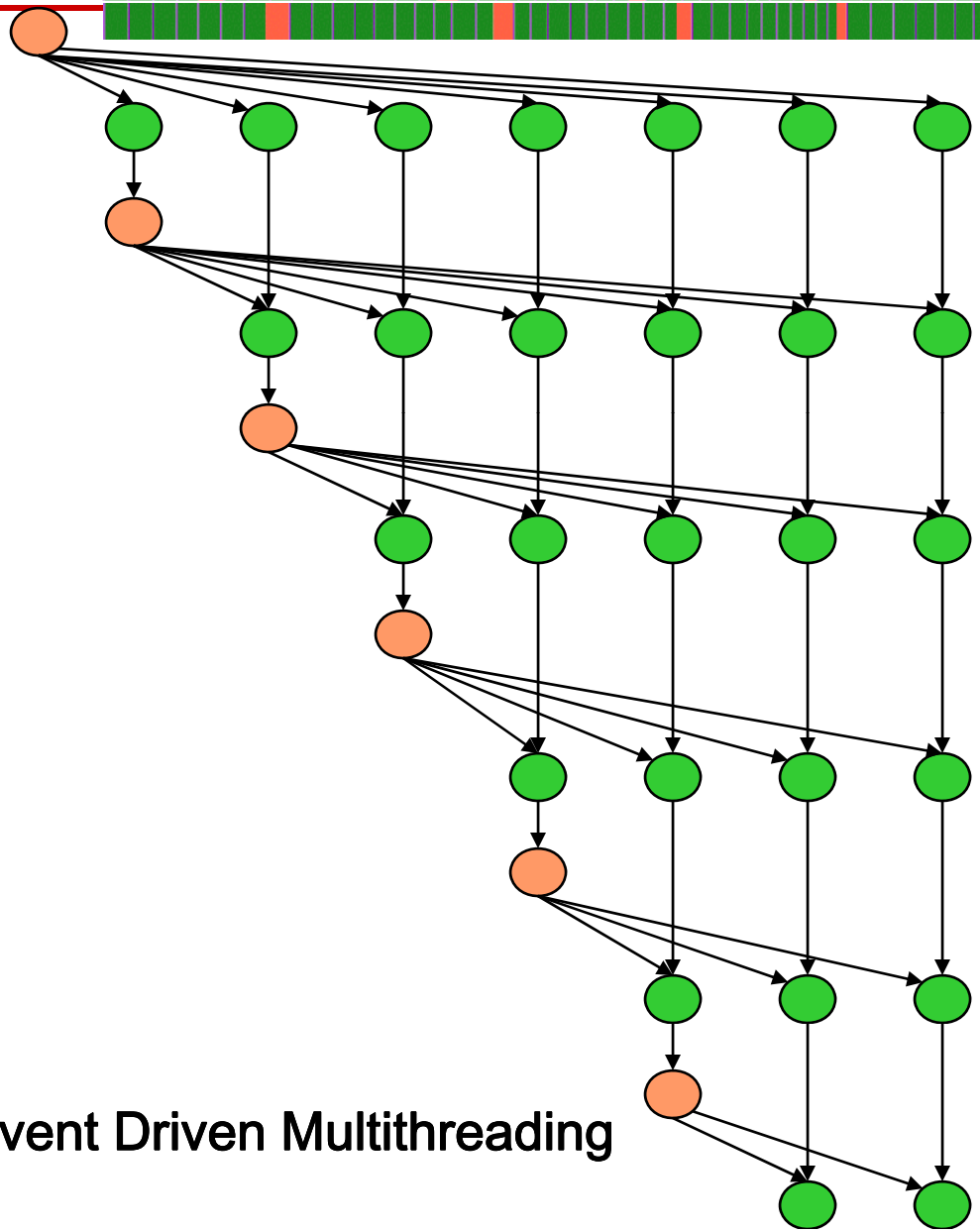




Bulk Sync Phases

Adaptive Lookahead - Dynamic



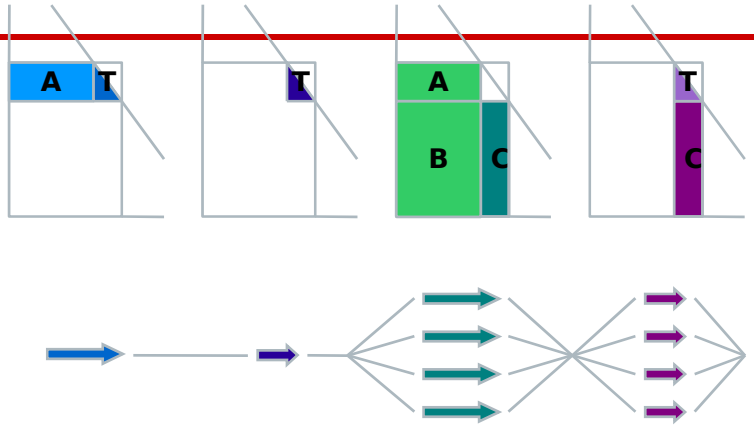


```
while(1)
    fetch_task();
    switch(task.type) {
        case PANEL:
            dgetf2();
            update_progress();
        case COLUMN:
            dlaswp();
            dtrsm();
            dgemm();
            update_progress();
        case END:
            for()
                dlaswp();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

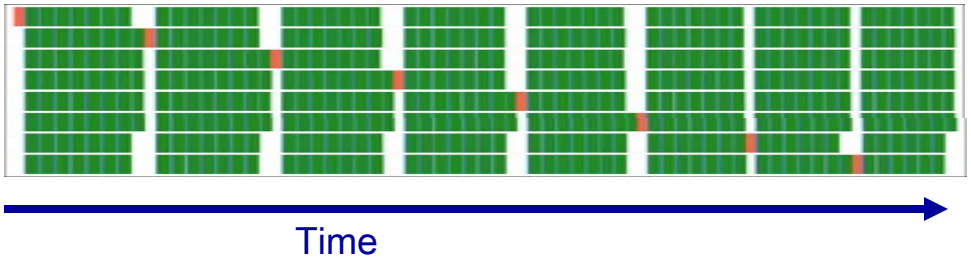
Event Driven Multithreading

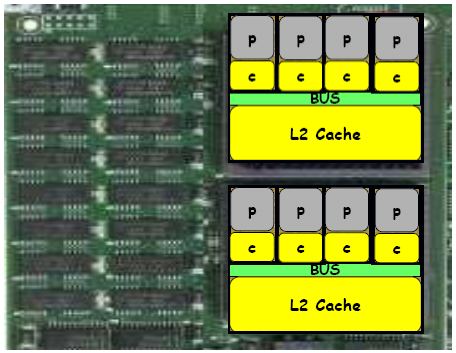
Reorganizing algorithms to use this approach

Fork-Join vs. Dynamic Execution



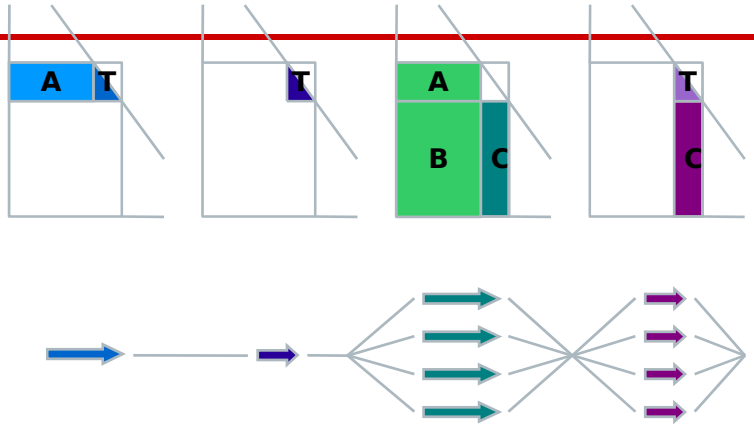
Fork-Join – parallel BLAS



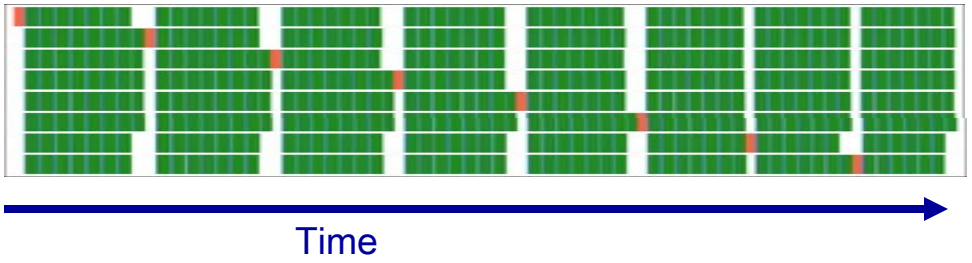


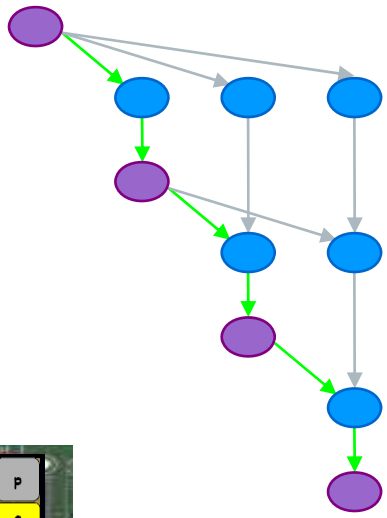
Experiments on
Intel's Quad Core Clovertown
with 2 Sockets w/ 8 Treads

Fork-Join vs. Dynamic Execution

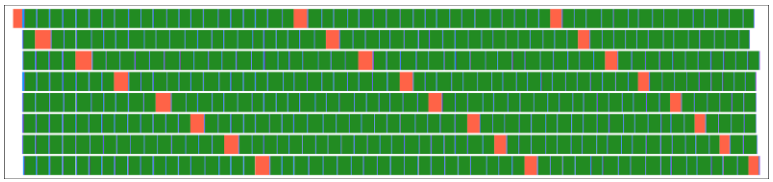


Fork-Join – parallel BLAS

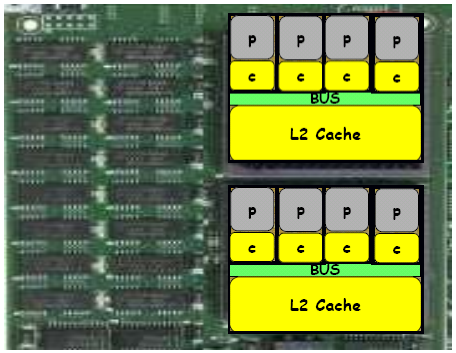




DAG-based – dynamic scheduling



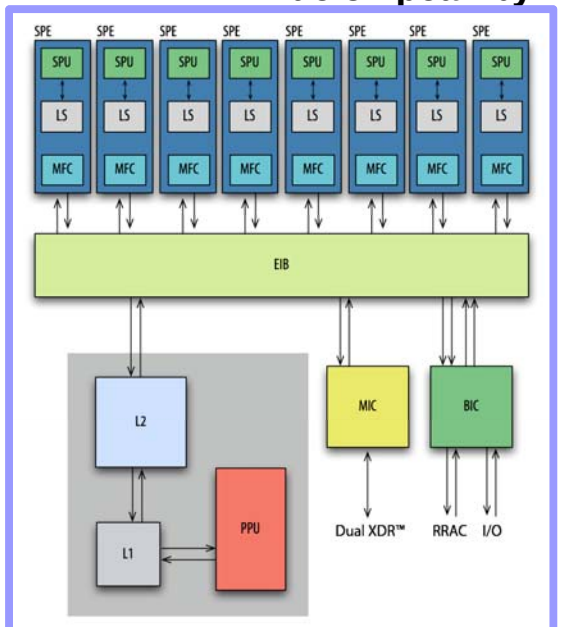
Time saved



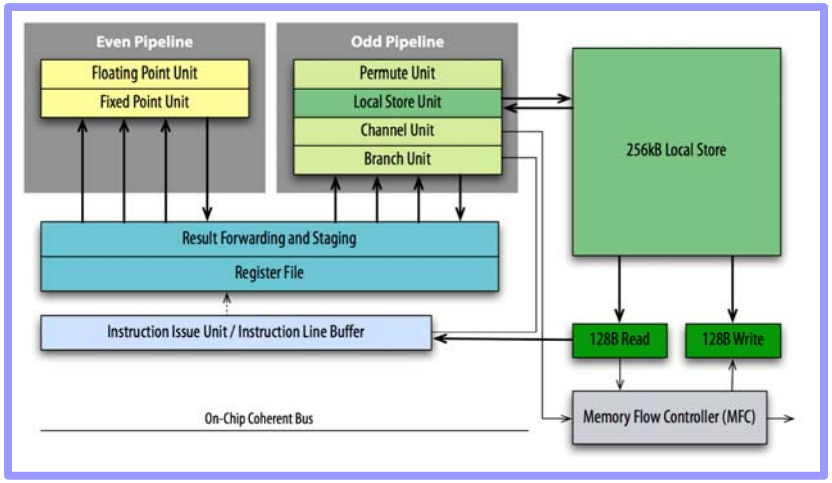
Experiments on
Intel's Quad Core Clovertown
with 2 Sockets w/ 8 Treads

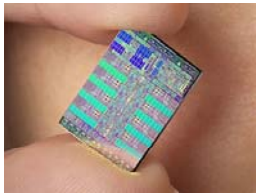
With the Hype on Cell & PS3 We Became Interested

- **The PlayStation 3's CPU based on a "Cell" processor**
- **Each Cell contains a Power PC processor and 8 SPEs. (SPE is processing unit, SPE: SPU + DMA engine)**
  - **An SPE is a self contained vector processor which acts independently from the others.**
    - **4 way SIMD floating point units capable of a total of 25.6 Gflop/s @ 3.2 GHZ**
  - **204.8 Gflop/s peak!**
  - **The catch is that this is for 32 bit floating point; (Single Precision SP)**
  - **And 64 bit floating point runs at 14.6 Gflop/s total for all 8 SPEs!!**
    - **Divide SP peak by 14; factor of 2 because of DP and 7 because of latency issues**



SPE ~ 25 Gflop/s peak





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

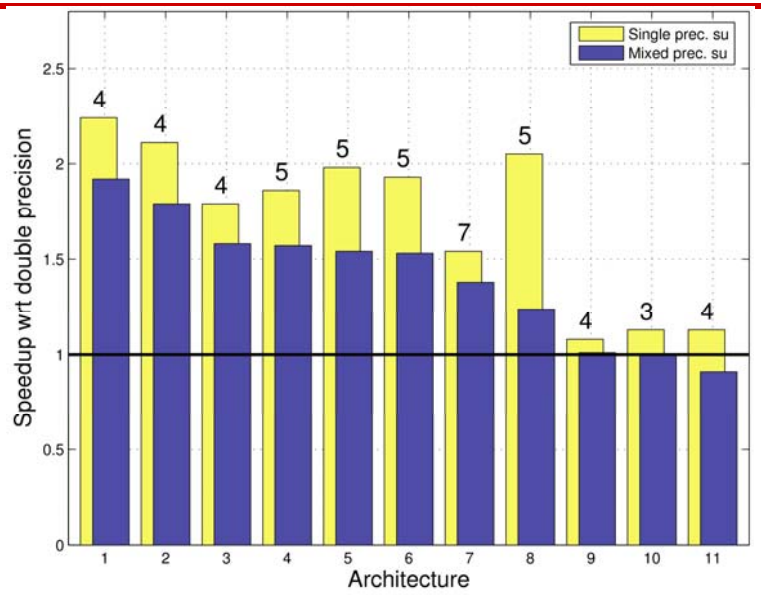
- **Iterative refinement for dense systems, $Ax = b$, can work this way.**

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; $O(10^8)$**

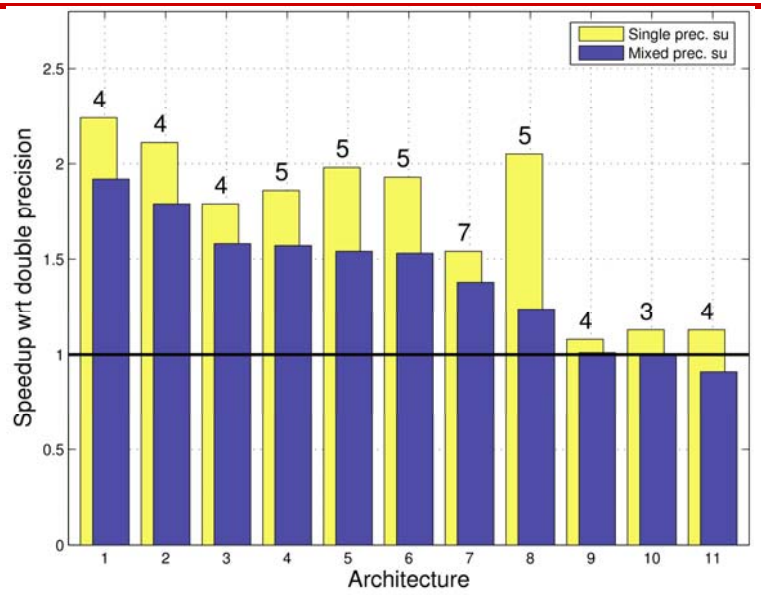
Results for Mixed Precision Iterative Refinement for Dense $Ax = b$



	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 $Ax = b$

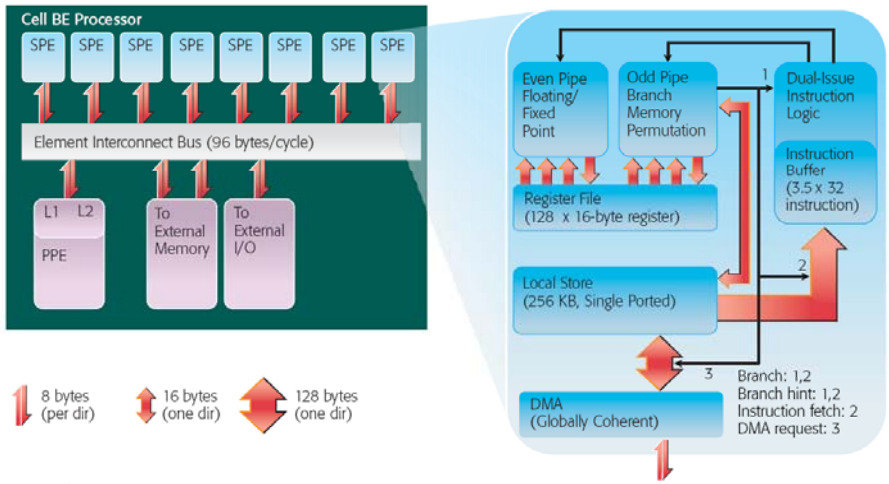


	Architecture (BLAS)
1	Intel Pentium III Coppermine (Goto)
2	Intel Pentium III Katmai (Goto)
3	Sun UltraSPARC IIe (Sunperf)
4	Intel Pentium IV Prescott (Goto)
5	Intel Pentium IV-M Northwood (Goto)
6	AMD Opteron (Goto)
7	Cray X1 (libsci)
8	IBM Power PC G5 (2.7 GHz) (VecLib)
9	Compaq Alpha EV6 (CXML)
10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

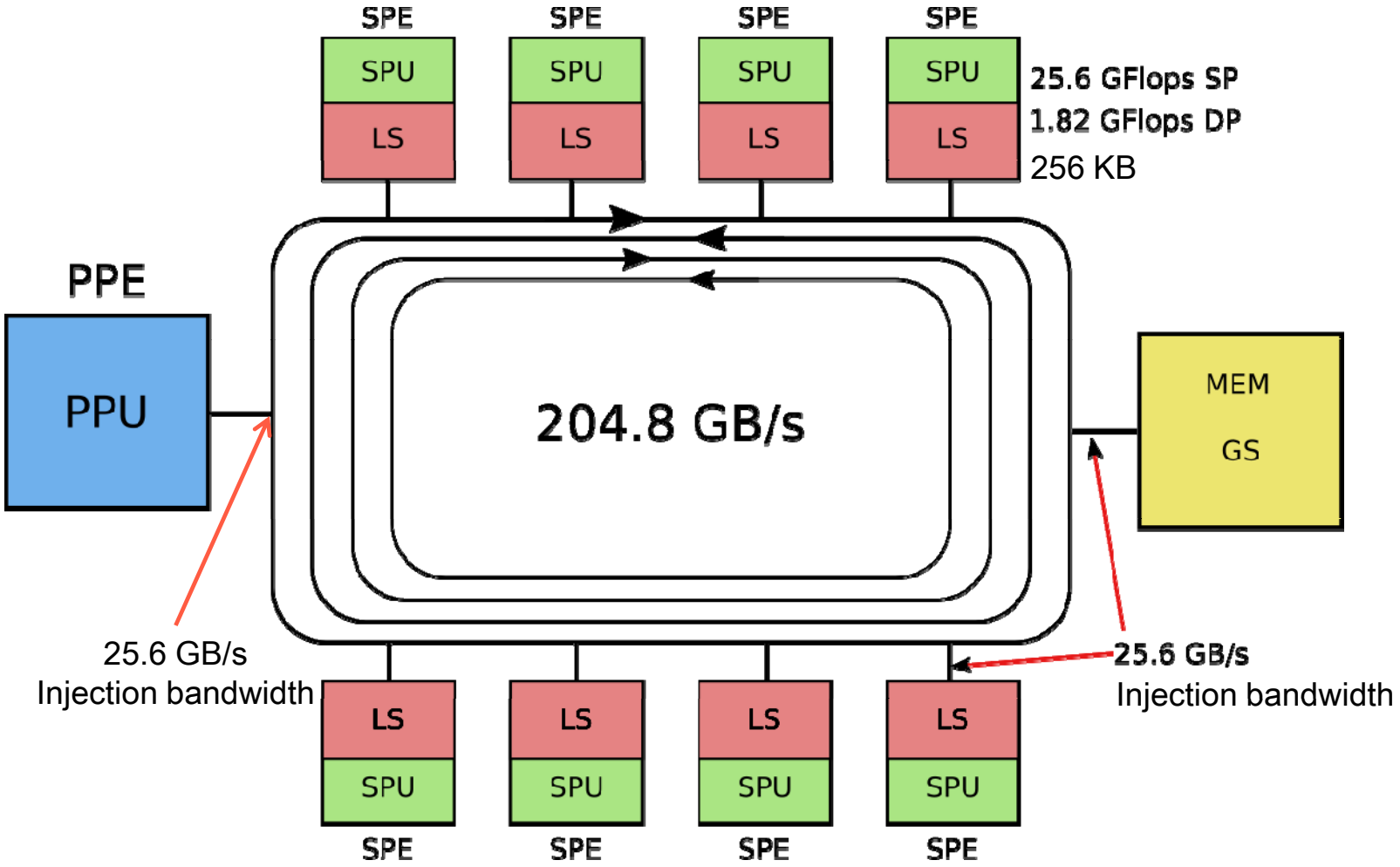
- Single precision is faster than DP because:
  - **Higher parallelism within vector units**
    - **4 ops/cycle (usually) instead of 2 ops/cycle**
  - **Reduced data motion**
    - **32 bit data instead of 64 bit data**
  - **Higher locality in cache**
    - **More data items in cache**

What about the Cell?



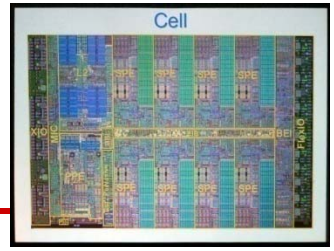
- **Power PC at 3.2 GHz**
  - **DGEMM at 5 Gflop/s**
  - **Altivec peak at 25.6 Gflop/s**
    - **Achieved 10 Gflop/s SGEMM**
- **8 SPUs**
  - **204.8 Gflop/s peak!**
  - **The catch is that this is for 32 bit floating point; (Single Precision SP)**
  - **And 64 bit floating point runs at 14.6 Gflop/s total for all 8 SPEs!!**
    - **Divide SP peak by 14; factor of 2 because of DP and 7 because of latency issues**

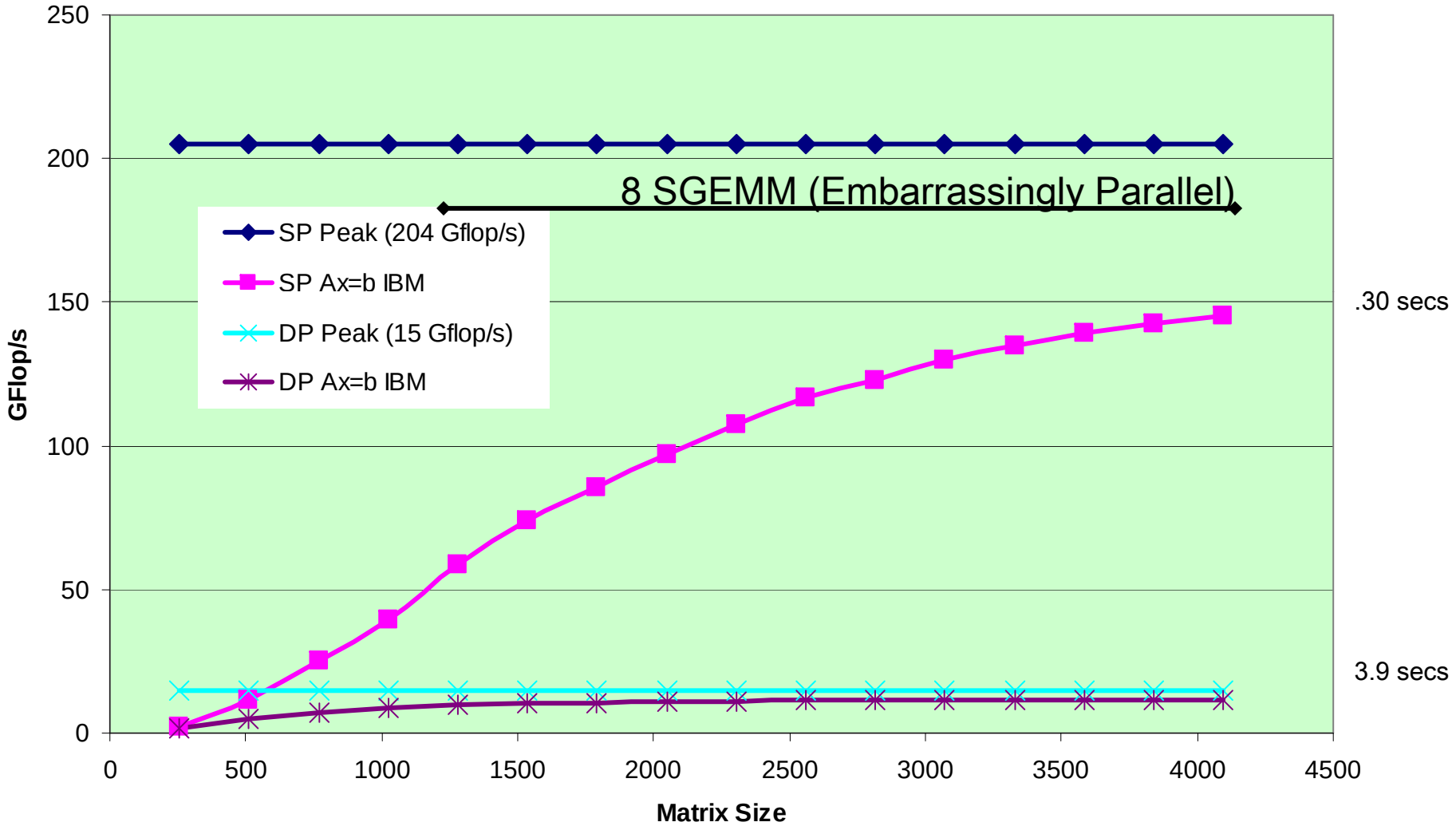
Moving Data Around on the Cell



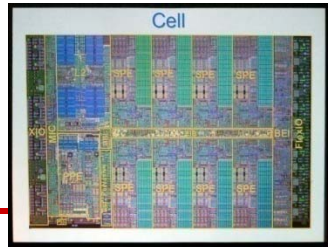
Worst case memory bound operations (no reuse of data)
3 data movements (2 in and 1 out) with 2 ops (SAXPY)
For the cell would be 4.6 Gflop/s (25.6 GB/s*2ops/12B)

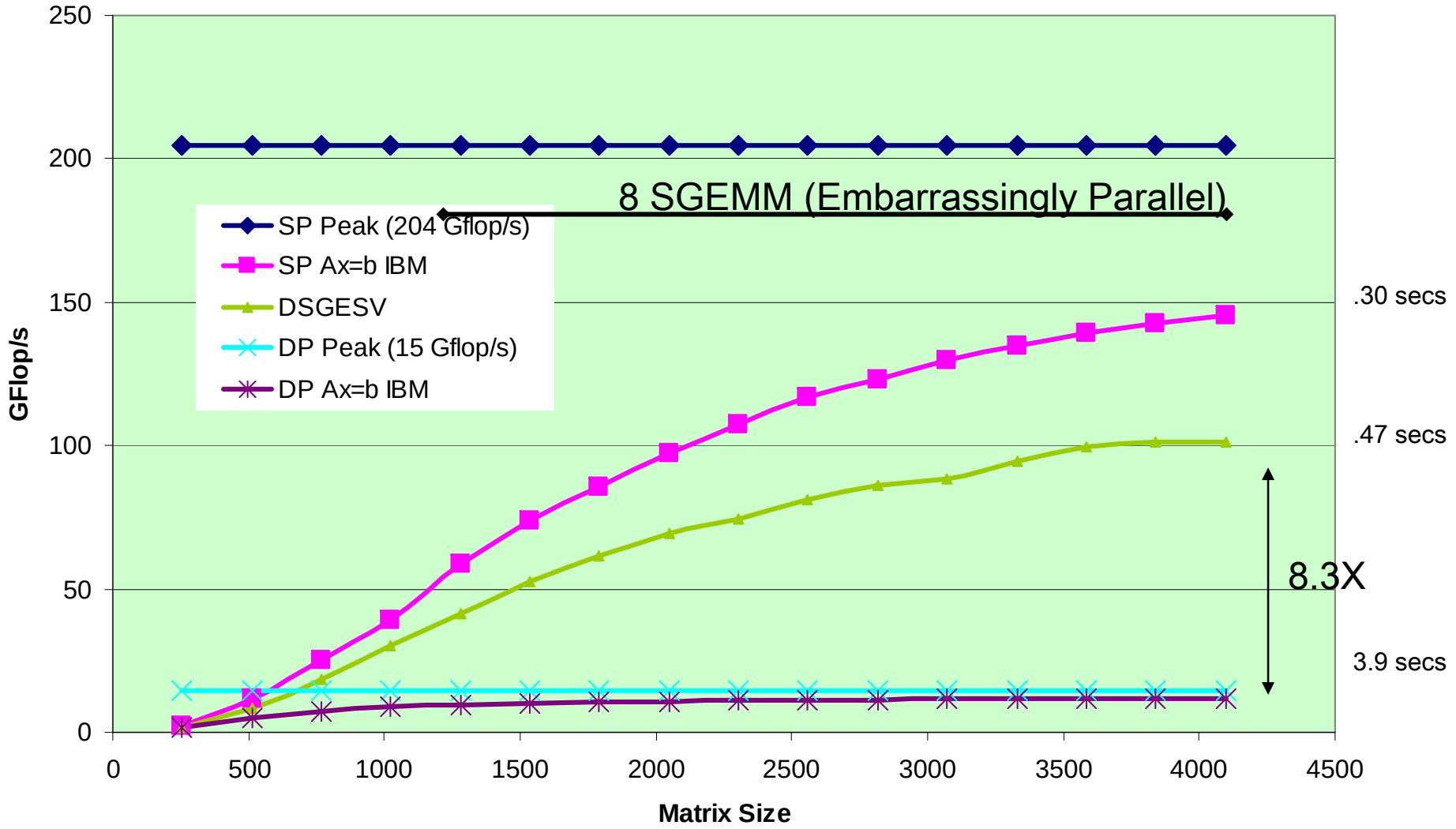
IBM Cell 3.2 GHz, Ax = b



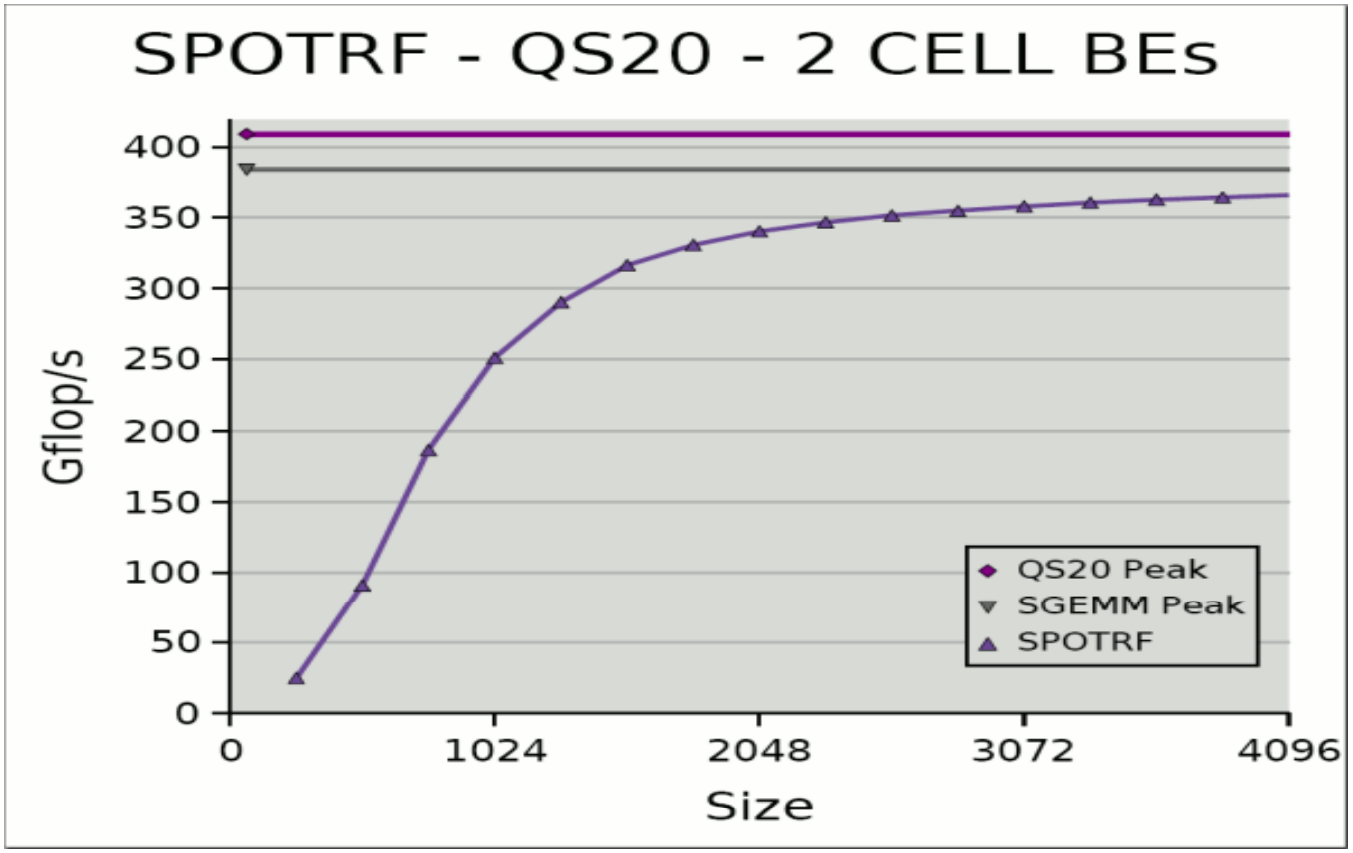


IBM Cell 3.2 GHz, Ax = b





Cholesky - Using 2 Cell Chips



Intriguing Potential

- **Exploit lower precision as much as possible**
  - **Payoff in performance**
    - **Faster floating point**
    - **Less data to move**
- **Automatically switch between SP and DP to match the desired accuracy**
  - **Compute solution in SP and then a correction to the solution in DP**
- **Potential for GPU, FPGA, special purpose processors**
  - **What about 16 bit floating point?**
    - **Use as little you can get away with and improve the accuracy**
- **Applies to sparse direct and iterative linear systems and Eigenvalue, optimization problems, where Newton's method is used.**

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

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

Correction = - A\(b – Ax)

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

Alfredo Buttari, UTK
Julien Langou, UColorado
Julie Langou, UTK
Piotr Luszczek, MathWorks
Jakub Kurzak, UTK
Stan Tomov, UTK