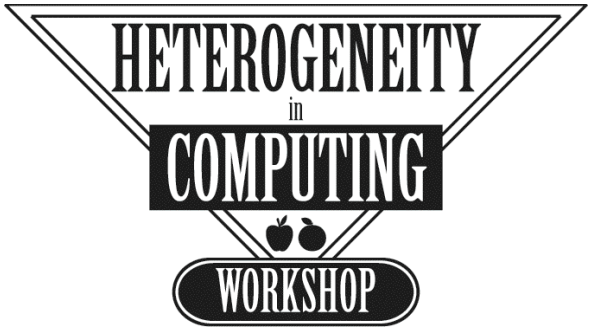


Emerging Heterogeneous Technologies for High Performance Computing

Jack Dongarra

University of Tennessee
Oak Ridge National Laboratory
University of Manchester

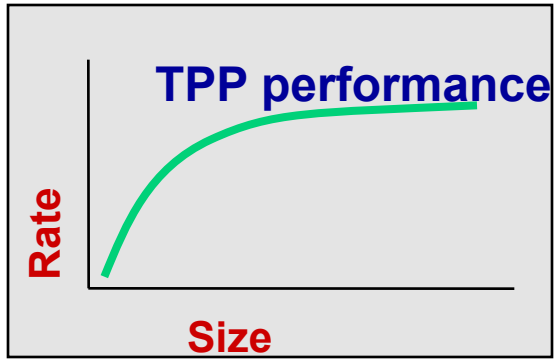
With a lot of help from

T. Dong, M. Gates, A. Haidar, J. Kurzak, P. Luszczek, S. Tomov, I. Yamazaki, H. Ltaief, H. Meuer, H. Simon, E. Stromairer

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

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

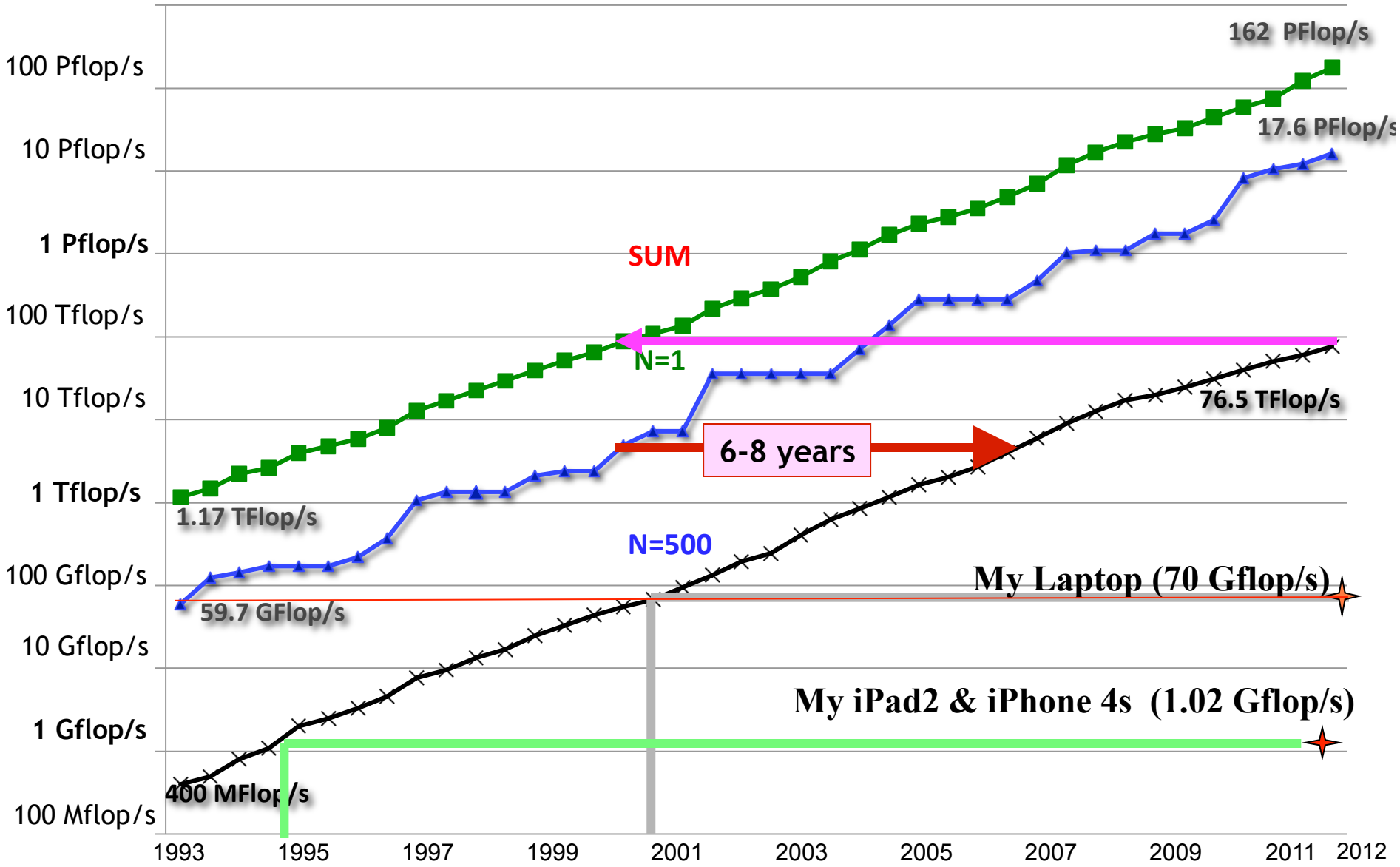
$Ax=b$, *dense problem*



- 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 of HPC Over the Last 20 Years



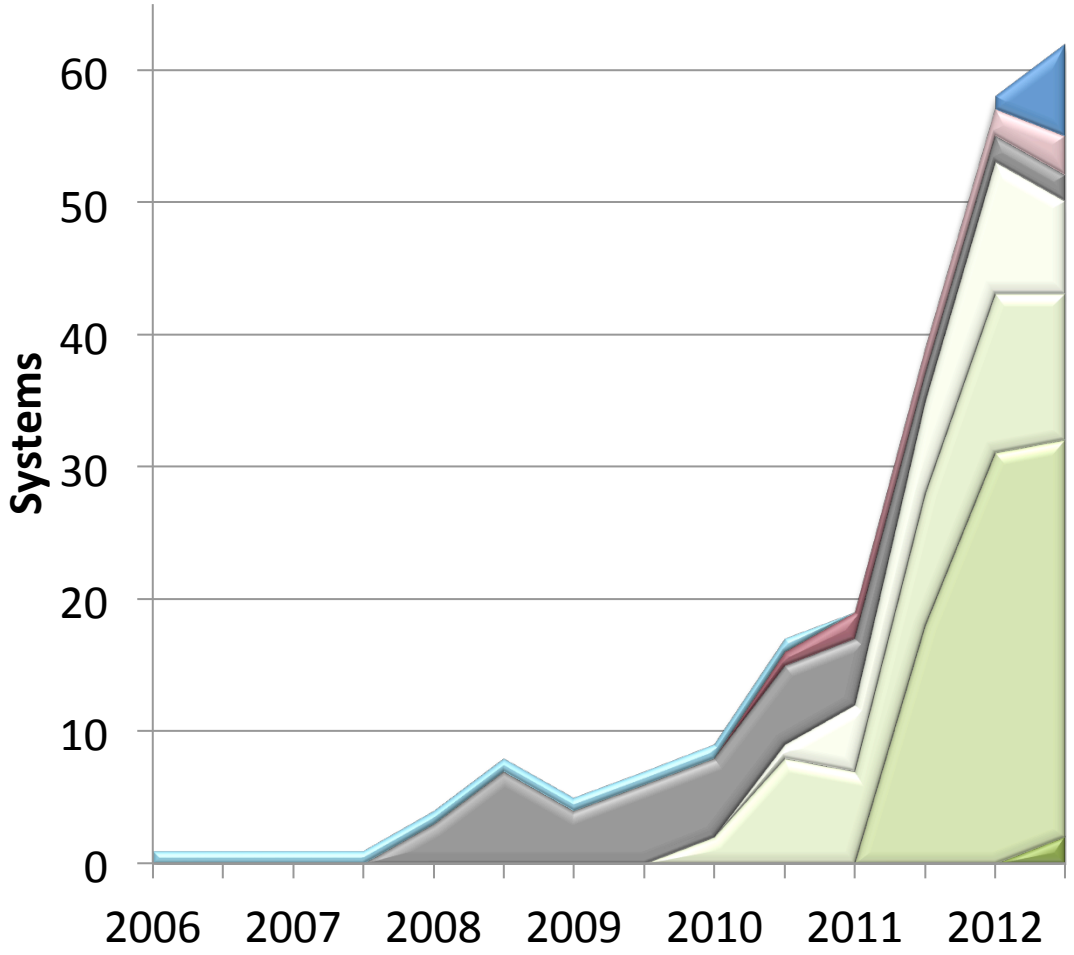
Pflop/s Club (23 systems; 6 Heterogeneous)

Name	Pflop/s	Country 10 4 2 2 2 2 1	
Titan	17.6	US	Cray: Hybrid AMD/Nvidia/Custom
Sequoia	16.3	US	IBM: BG-Q/Custom
K computer	10.5	Japan	Fujitsu: Sparc/Custom
Mira	8.16	US	IBM: BG-Q/Custom
JuQUEEN	4.14	Germany	IBM: BG-Q/Custom
SuperMUC	2.90	Germany	IBM: Intel/IB
Stampede	2.66	US	Dell: Hybrid Intel/Intel/IB
Tianhe-1A	2.57	China	NUDT: Hybrid Intel/Nvidia/Custom
Fermi	1.73	Italy	IBM: BG-Q/Custom
DARPA Trial Subset	1.52	US	IBM: IBM/Custom
Curie thin nodes	1.36	France	Bull: Intel/IB
Nebulae	1.27	China	Dawning: Hybrid Intel/Nvidia/IB
Yellowstone	1.26	US	IBM: Intel/IB
Pleiades	1.24	US	SGI: Intel/IB
Helios	1.24	Japan	Bull: Intel/IB
Blue Joule	1.21	UK	IBM: BG-Q/Custom
TSUBAME 2.0	1.19	Japan	HP: Hybrid Intel/Nvidia/IB
Cielo	1.11	US	Cray: AMD/Custom
Hopper	1.05	US	Cray: AMD/Custom
Tera-100	1.05	France	Bull: Intel/IB
Oakleaf-FX	1.04	Japan	Fujitsu: Sparc/Custom
Roadrunner	1.04	US	IBM: Hybrid AMD/Cell/IB (First one in '08)
DiRAC	1.04	UK	IBM: BG-Q/Custom

November 2012: The TOP10

Rank	Site	Computer	Country	Cores	Rmax [Pflops]	% of Peak	Power [MW]	MFlops /Watt
1	DOE / OS Oak Ridge Nat Lab	Titan, Cray XK7 (16C) + Nvidia Kepler GPU (14c) + custom	USA	560,640	17.6	66	8.3	2120
2	DOE / NNSA L Livermore Nat Lab	Sequoia, BlueGene/Q (16c) + custom	USA	1,572,864	16.3	81	7.9	2063
3	RIKEN Advanced Inst for Comp Sci	K computer Fujitsu SPARC64 VIIIfx (8c) + custom	Japan	705,024	10.5	93	12.7	827
4	DOE / OS Argonne Nat Lab	Mira, BlueGene/Q (16c) + custom	USA	786,432	8.16	81	3.95	2066
5	Forschungszentrum Juelich	JuQUEEN, BlueGene/Q (16c) + custom	Germany	393,216	4.14	82	1.97	2102
6	Leibniz Rechenzentrum	SuperMUC, Intel (8c) + IB	Germany	147,456	2.90	90*	3.42	848
7	Texas Advanced Computing Center	Stampede, Dell Intel (8) + Intel Xeon Phi (61) + IB	USA	204,900	2.66	67	3.3	806
8	Nat. SuperComputer Center in Tianjin	Tianhe-1A, NUDT Intel (6c) + Nvidia Fermi GPU (14c) + custom	China	186,368	2.57	55	4.04	636
9	CINECA	Fermi, BlueGene/Q (16c) + custom	Italy	163,840	1.73	82	.822	2105
10	IBM	DARPA Trial System, Power7 (8C) + custom	USA	63,360	1.51	78	.358	422
500	Slovak Academy Sci	IBM Power 7	Slovak Rep	3,074	.077	81		

Heterogeneous Systems – 62



- Intel Xeon Phi (7)
- Clearspeed CSX600 (0)
- ATI GPU (3)
- IBM PowerXCell 8i (2)
- NVIDIA 2070 (7)
- NVIDIA 2050 (11)
- NVIDIA 2090 (30)
- NVIDIA K20 (2)

32 US	1 Australia
6 China	1 Brazil
2 Japan	1 Canada
4 Russia	1 Saudi Arabia
2 France	1 South Korea
2 Germany	1 Spain
1 India	1 Switzerland
2 Italy	1 Taiwan
2 Poland	1 UK

Commodity plus Accelerator Today

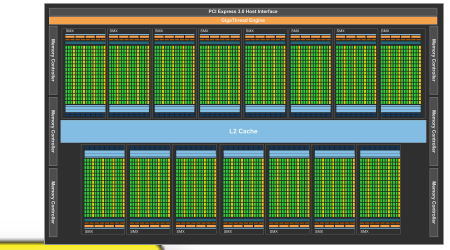
Commodity

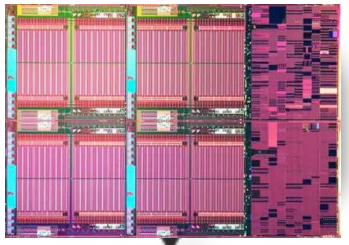
Intel Xeon
8 cores
3 GHz
8*4 ops/cycle
96 Gflop/s (DP)

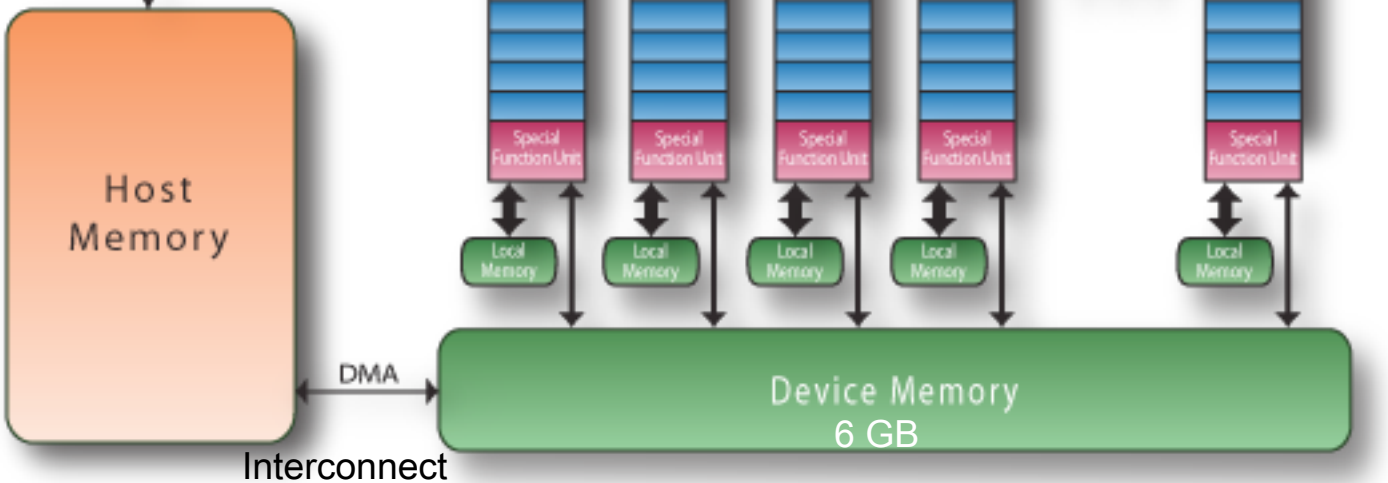
Accelerator (GPU)

Nvidia K20X “Kepler”
2688 “Cuda cores”
.732 GHz
2688*2/3 ops/cycle
1.31 Tflop/s (DP)

2688 “Cuda cores”
192 Cuda cores/SMX
14 Cores







Interconnect
PCI-X 16 lane
64 Gb/s (8 GB/s)
1 GW/s

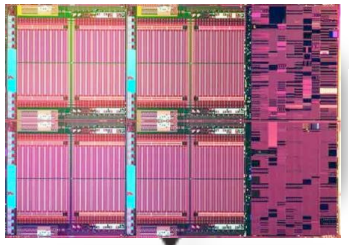
Commodity plus Accelerator Today

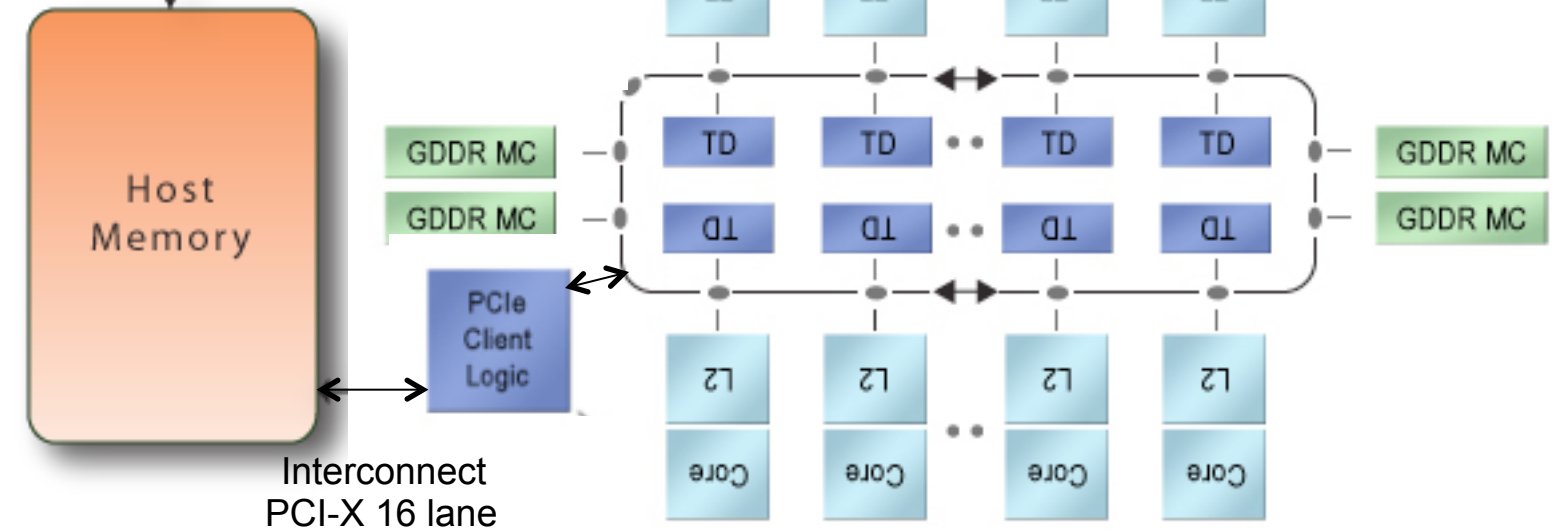
Commodity

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

Accelerator/Co-Processor

Intel Xeon Phi
244 “cores” (4 used by OS)
61 (60) FPU = 61 (60) cores
1.09 GHz
60*1.09*8*2 ops/cycle
1.046 Tflop/s (DP) or 2.092 Tflop/s (SP)



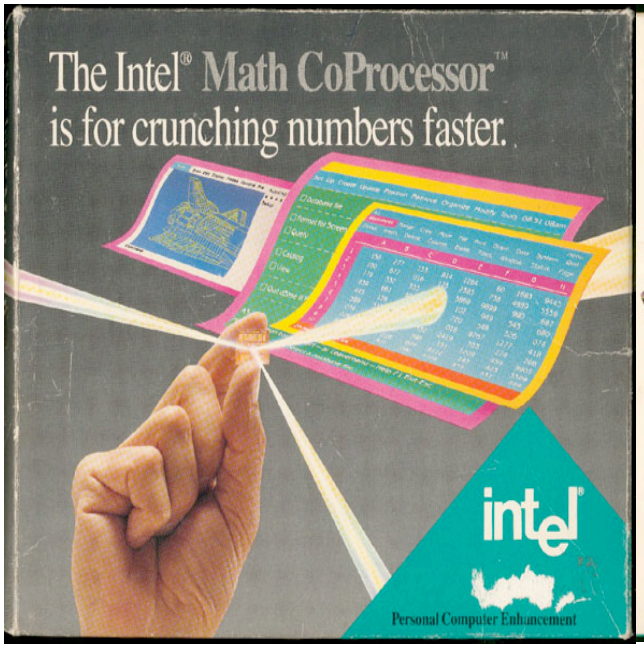


Interconnect
PCI-X 16 lane
64 Gb/s (8 GB/s)
1 GW/s

We Have Seen This Before

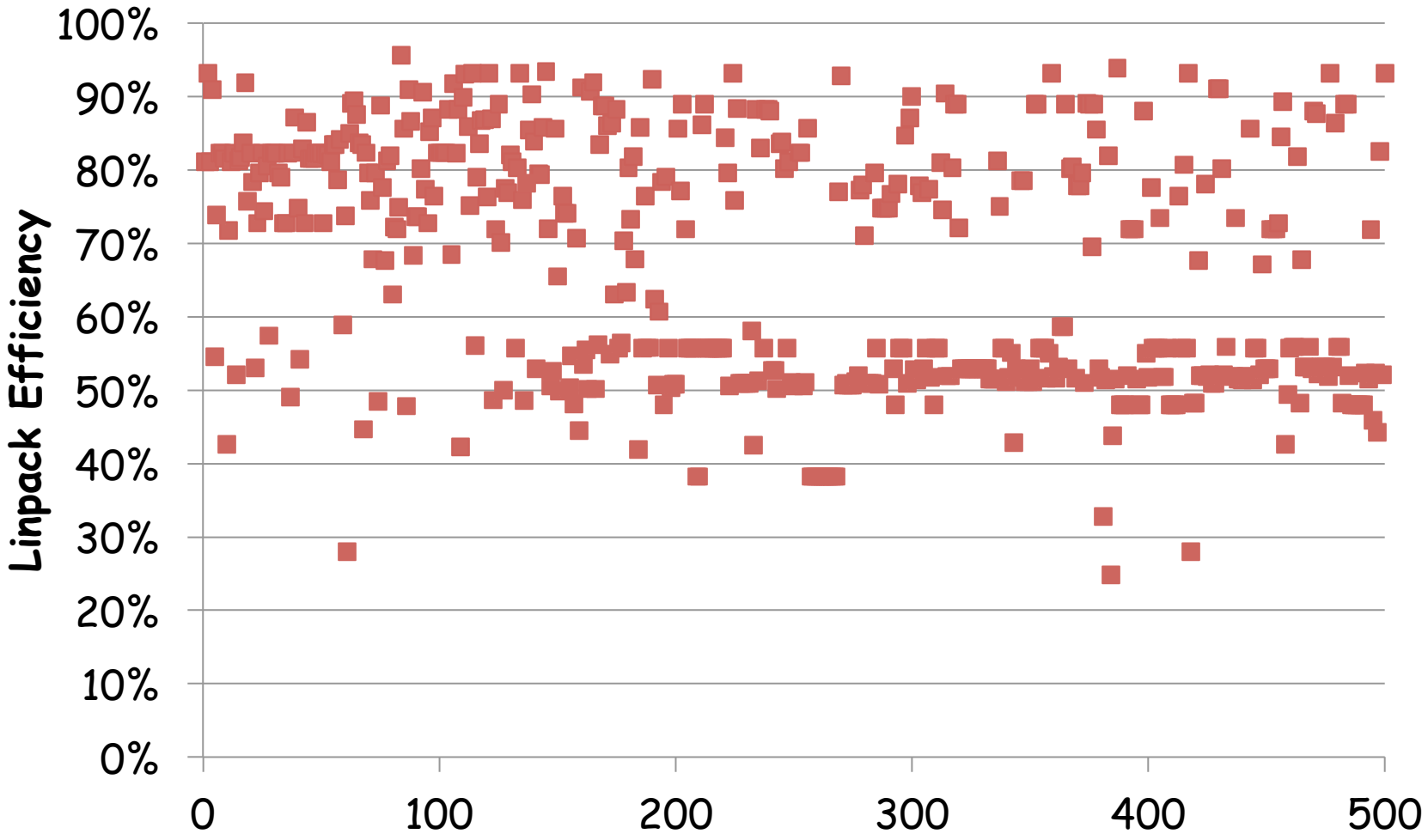
- Floating Point Systems FPS-164/ MAX Supercomputer (1976)
- Intel Math Co-processor (1980)
- Weitek Math Co-processor (1981)

1976

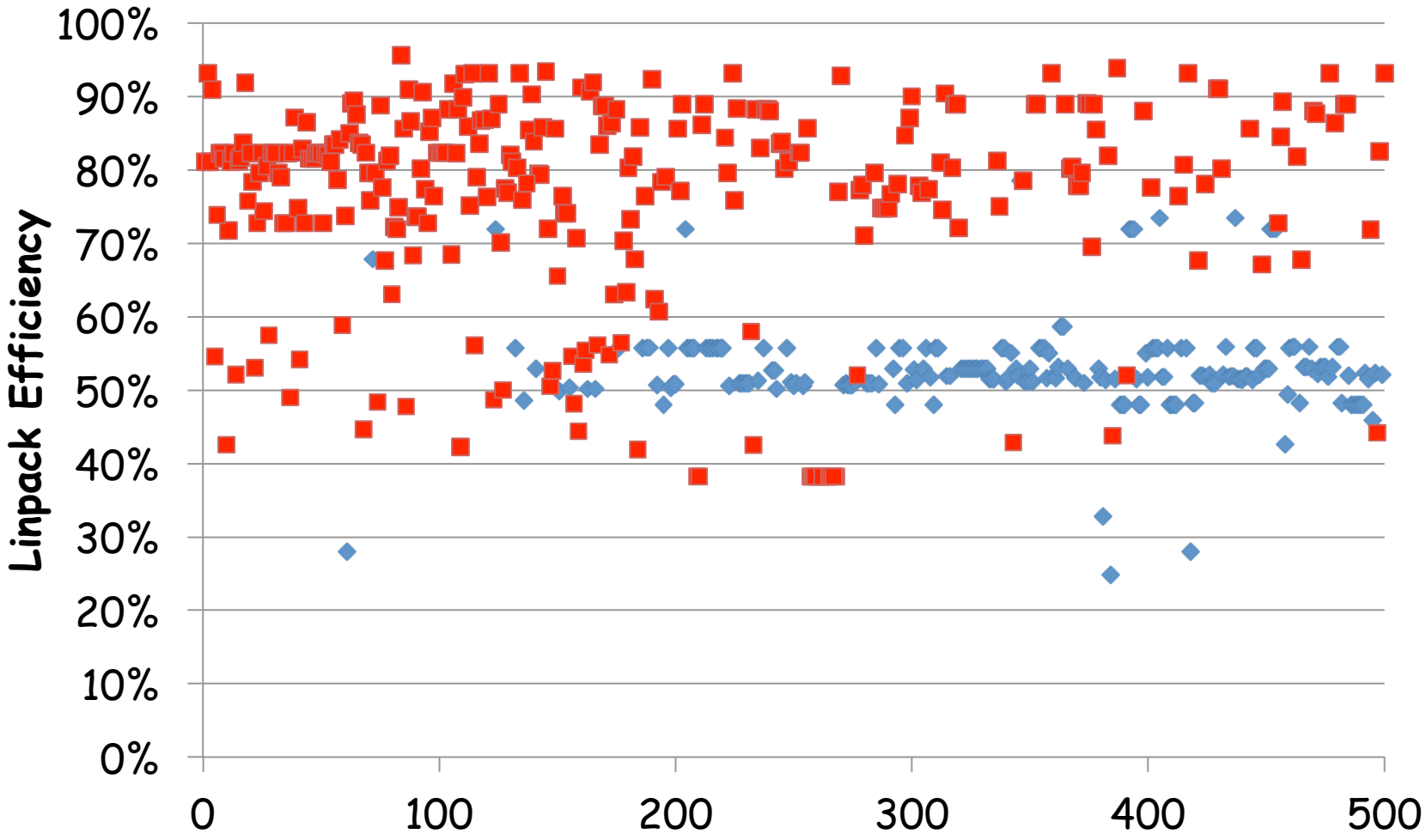


1980

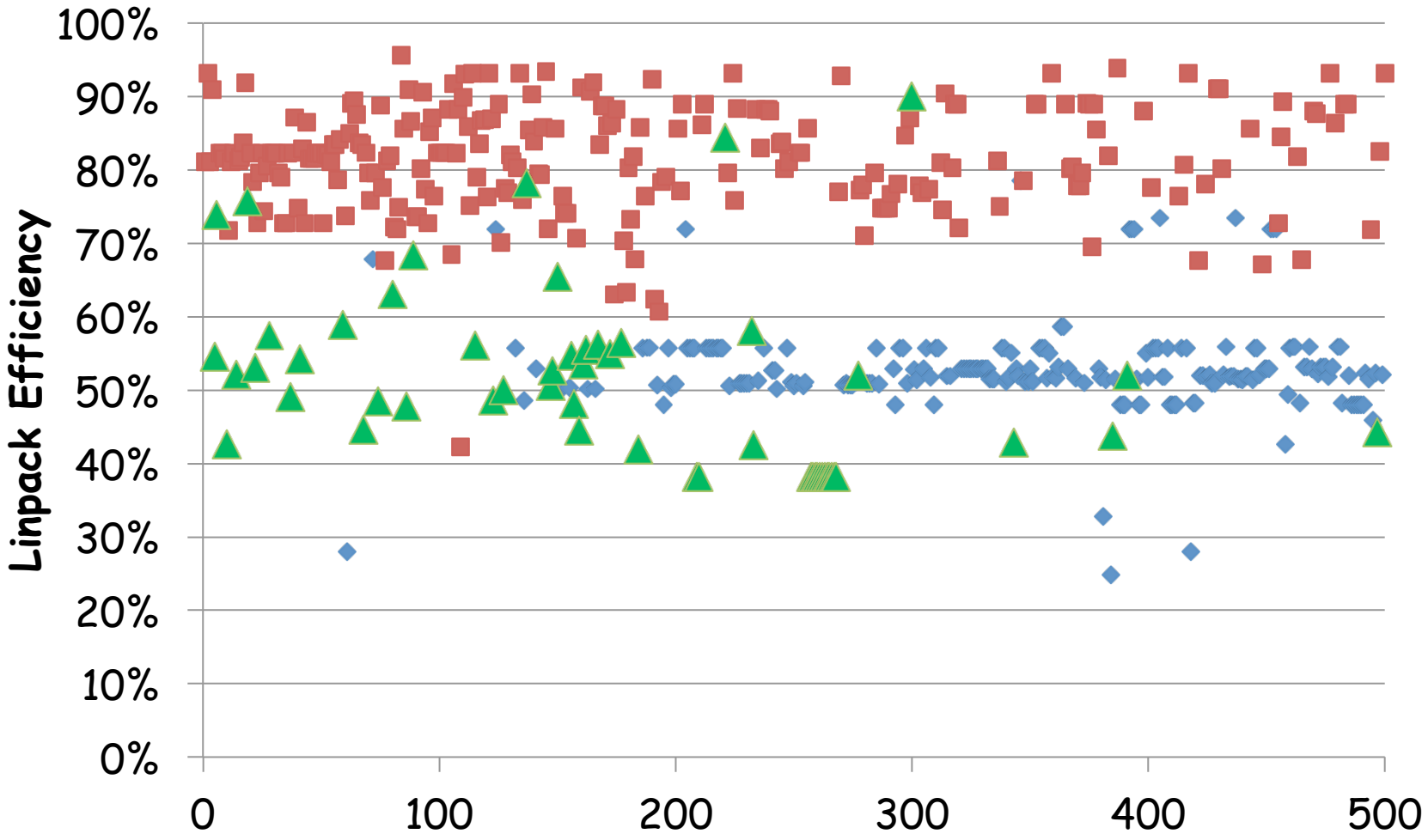
Linpack Efficiency



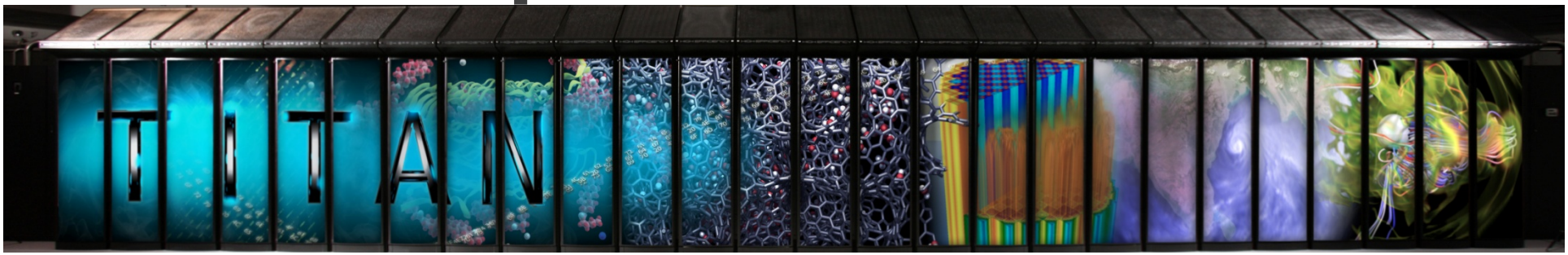
Linpack Efficiency

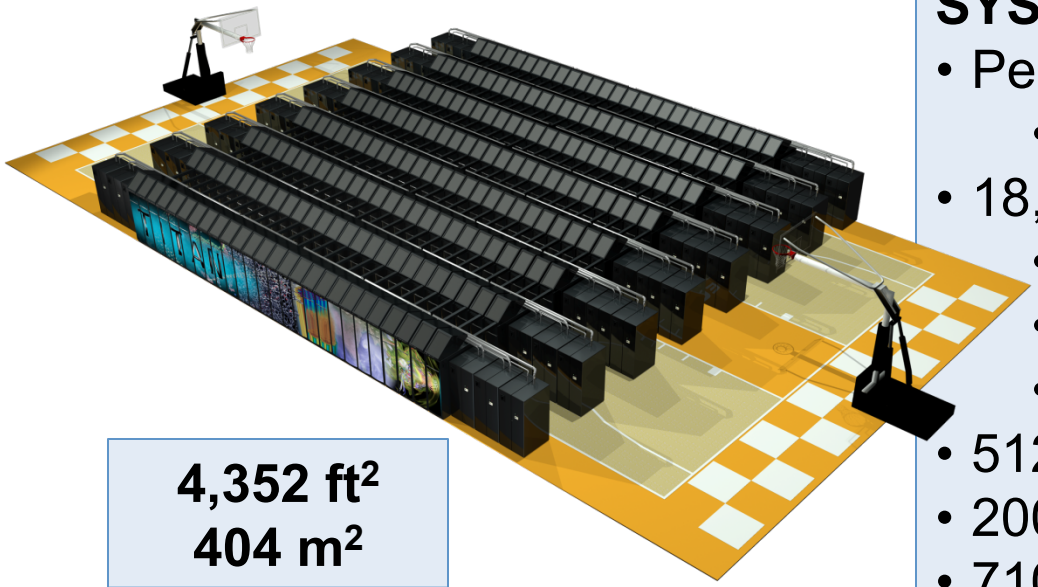


Linpack Efficiency



ORNL's "Titan" Hybrid System: Cray XK7 with AMD Opteron and NVIDIA Tesla processors





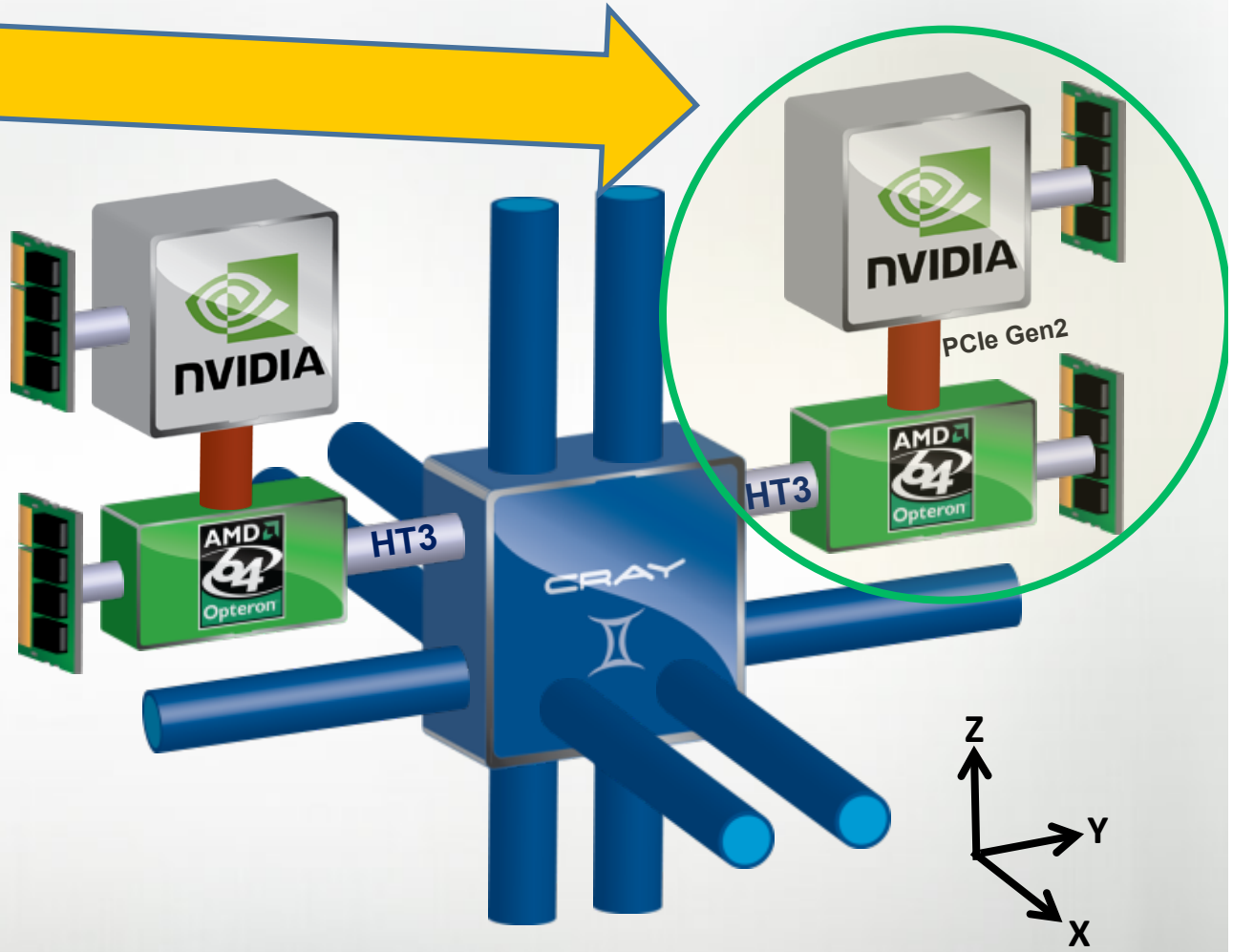
4,352 ft²
404 m²

SYSTEM SPECIFICATIONS:

- Peak performance of 27 PF
  - 24.5 Pflop/s GPU + 2.6 Pflop/s AMD
- 18,688 Compute Nodes each with:
  - 16-Core AMD Opteron CPU
  - 14-Core NVIDIA Tesla "K20x" GPU
  - 32 GB + 6 GB memory
- 512 Service and I/O nodes
- 200 Cabinets
- 710 TB total system memory
- Cray Gemini 3D Torus Interconnect
- 9 MW peak power

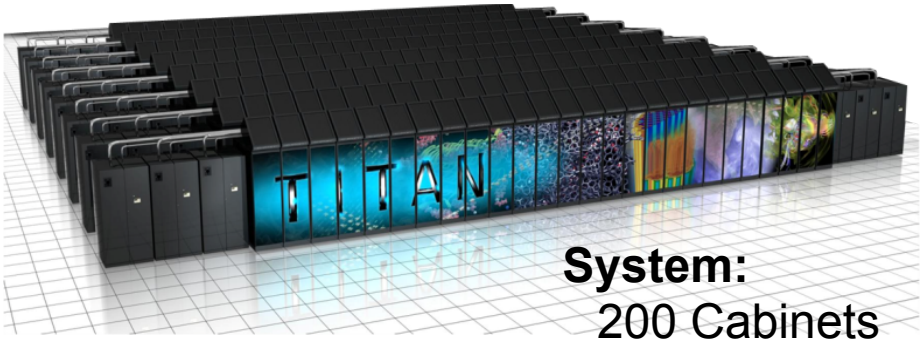
Cray XK7 Compute Node

XK7 Compute Node Characteristics
AMD Opteron 6274 Interlagos 16 core processor
Tesla K20x @ 1311 GF
Host Memory 32GB 1600 MHz DDR3
Tesla K20x Memory 6GB GDDR5
Gemini High Speed Interconnect

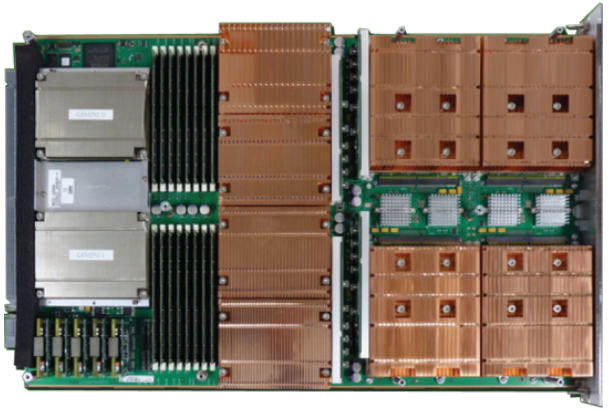


Slide courtesy of Cray, Inc.

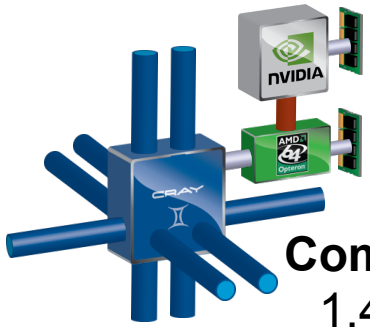
Titan:
Cray XK7 System



System:
200 Cabinets
18,688 Nodes
27 PF
710 TB



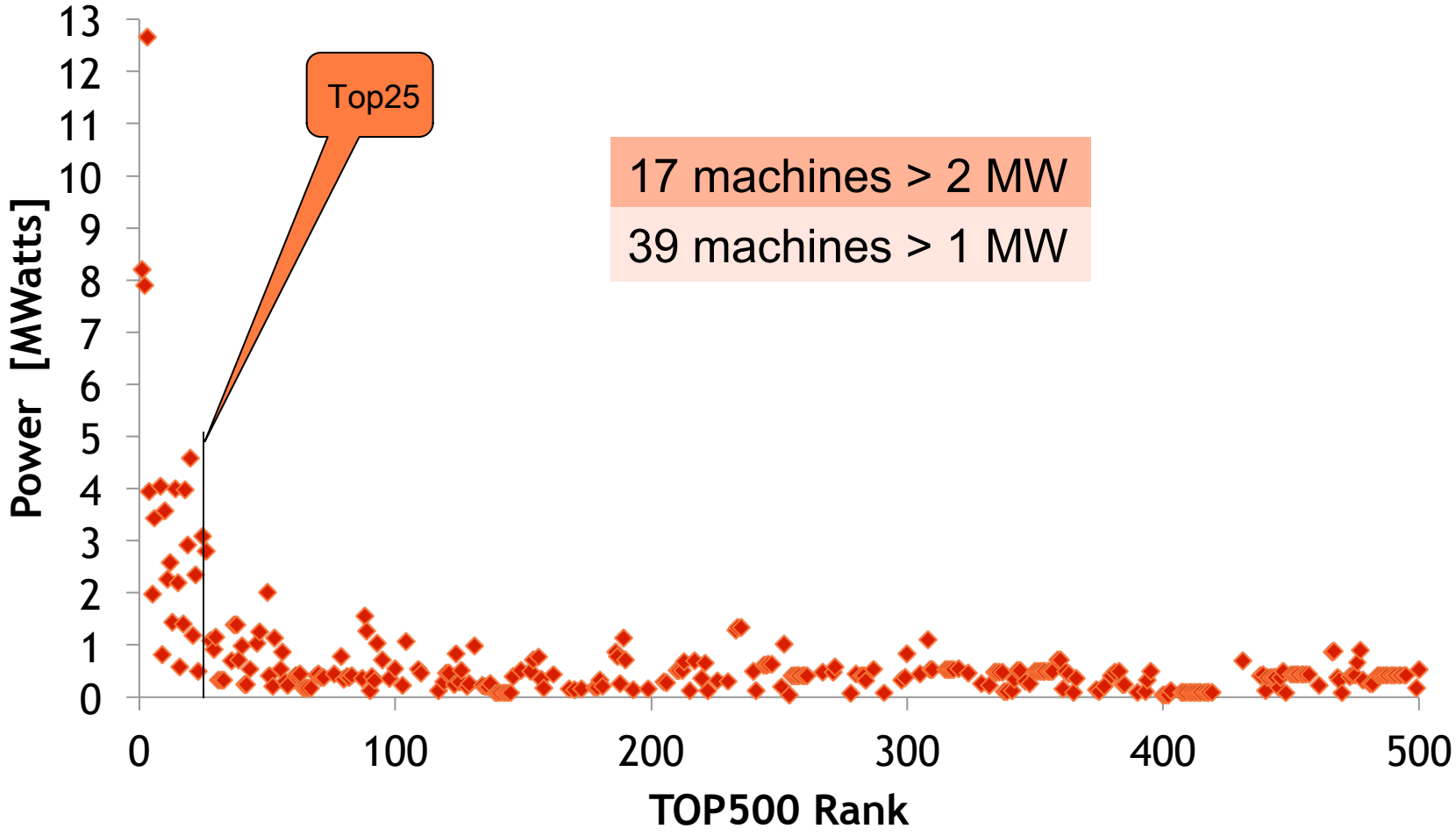
Cabinet:
24 Boards
96 Nodes
139 TF
3.6 TB



Compute Node:
1.45 TF
38 GB

Board:
4 Compute Nodes
5.8 TF
152 GB

Power Levels



The Green500 List

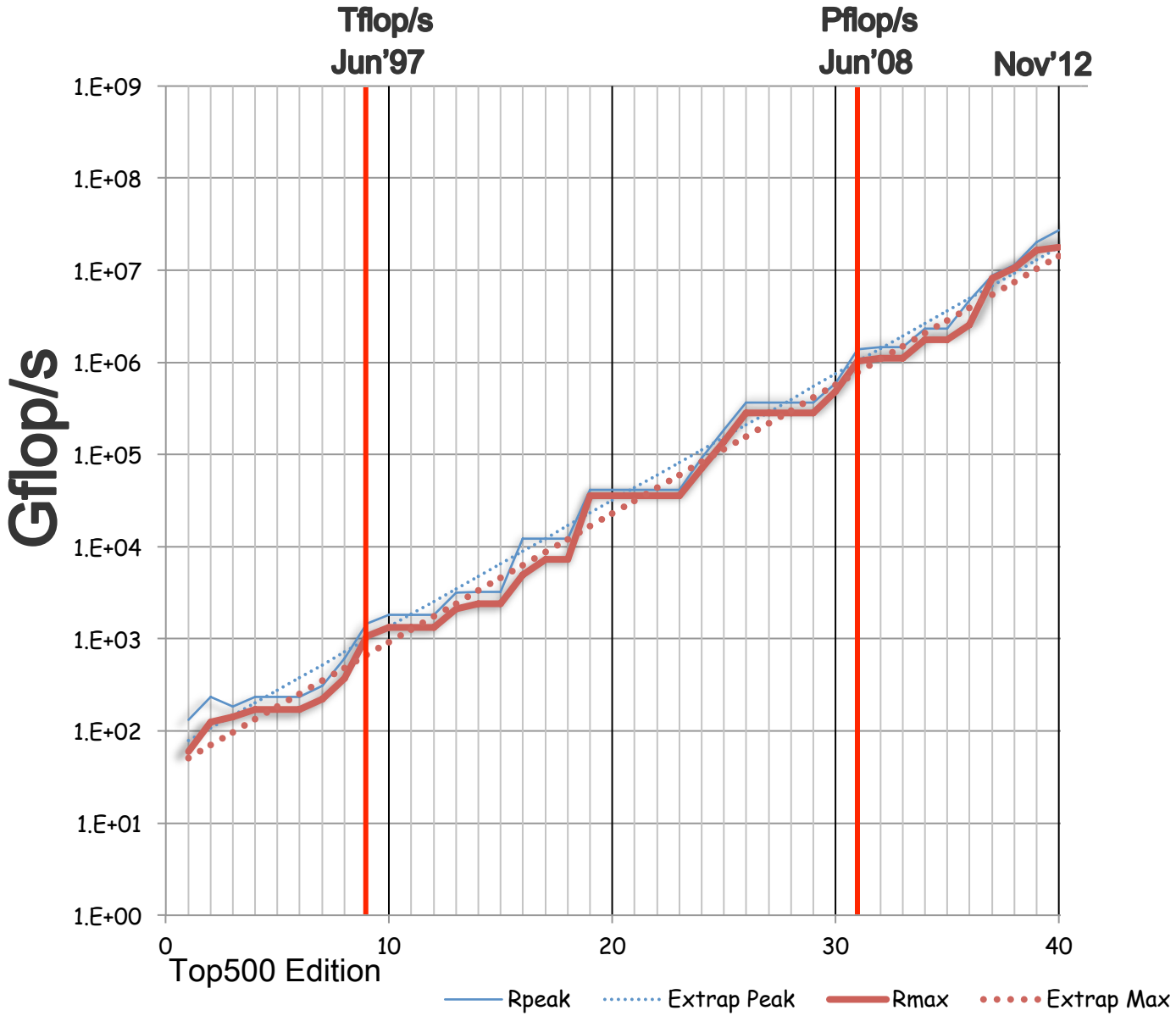
Listed below are the November 2012 The Green500's energy-efficient supercomputers ranked from 1 to 10.

Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1	2,499.44	National Institute for Computational Sciences/University of Tennessee	Beacon - Appro GreenBlade GB824M, Xeon E5-2670 8C 2.600GHz, Infiniband FDR, Intel Xeon Phi 5110P	44.89
2	2,351.10	King Abdulaziz City for Science and Technology	SANAM - Adtech ESC4000/FDR G2, Xeon E5-2650 8C 2.000GHz, Infiniband FDR, AMD FirePro S10000	179.15
3	2,142.77	DOE/SC/Oak Ridge National Laboratory	Titan - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x	8,209.00
4	2,121.71	Swiss Scientific Computing Center (CSCS)	Todi - Cray XK7 , Opteron 6272 16C 2.100GHz, Cray Gemini interconnect, NVIDIA Tesla K20 Kepler	129.00
5	2,102.12	Forschungszentrum Juelich (FZJ)	JUQUEEN - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect	1,970.00
6	2,101.39	Southern Ontario Smart Computing Innovation Consortium/University of Toronto	BGQdev - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect	41.09
7	2,101.39	DOE/NNSA/LLNL	rzuseq - BlueGene/Q, Power BQC 16C 1.60GHz, Custom	41.09
8	2,101.39	IBM Thomas J. Watson Research Center	BlueGene/Q, Power BQC 16C 1.60GHz, Custom	41.09
9	2,101.12	IBM Thomas J. Watson Research Center	BlueGene/Q, Power BQC 16C 1.60 GHz, Custom	82.19
10	2,101.12	Ecole Polytechnique Federale de Lausanne	CADMOS BG/Q - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect	82.19

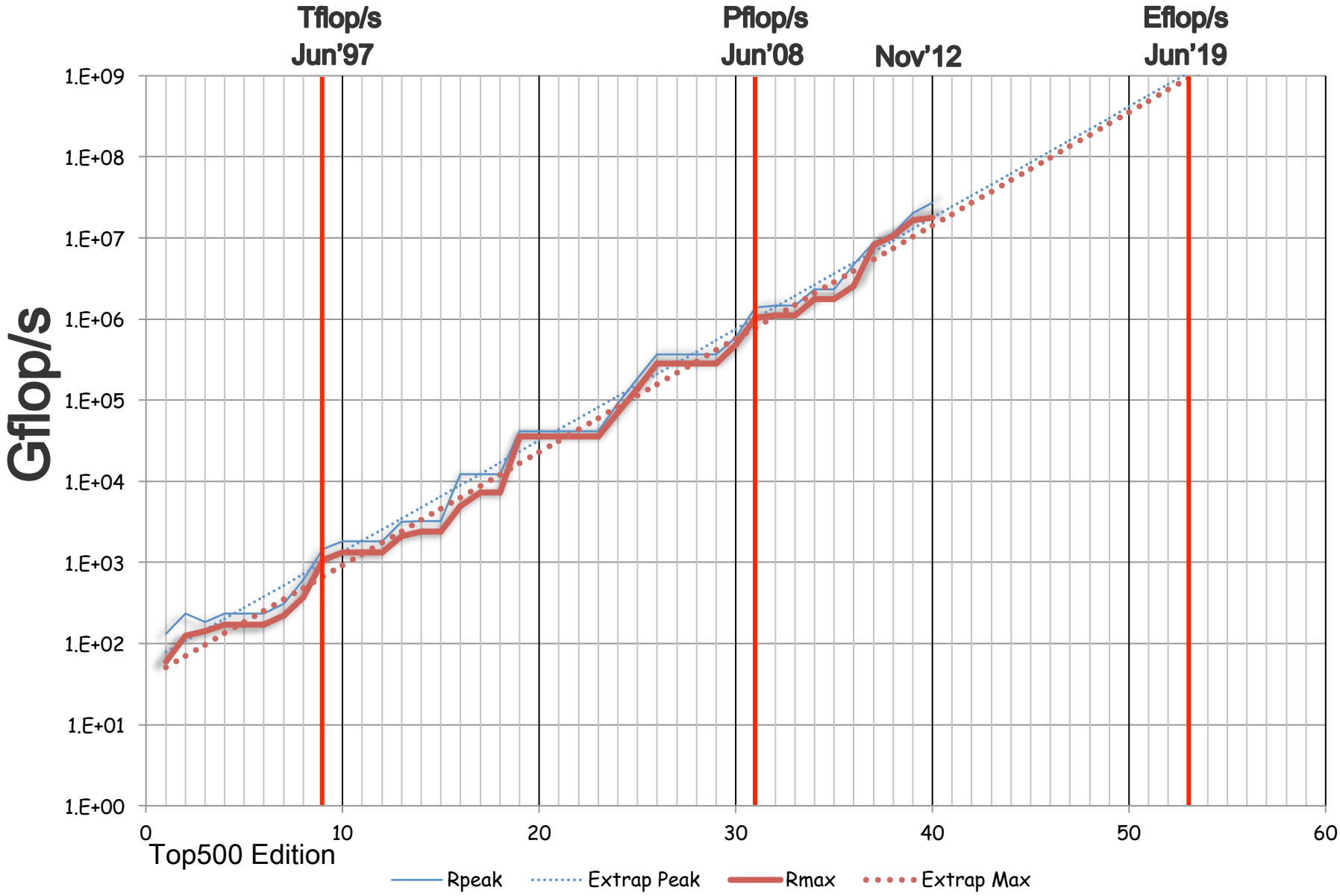
Most Power Efficient Hybrid Systems

Computer	Gflop/ Watt
Appro GreenBlade, Xeon 8C 2.6GHz, Infiniband FDR, **Intel Xeon Phi**	**2.45**
Cray XK7, Opteron 16C 2.1GHz, Gemini, **NVIDIA Kepler**	**2.24**
BlueGene/Q, Power BQC 16C 1.60 GHz, Custom	**2.10**
iDataPlex DX360M4, Xeon 8C 2.6GHz, Infiniband QDR, **Intel Xeon Phi**	**1.94**
RSC Tornado, Xeon 8C 2.9GHz, Infiniband FDR, **Intel Xeon Phi**	**1.69**
SGI Rackable, Xeon 8C 2.6GHz, Infiniband FDR, **Intel Xeon Phi**	**1.61**
Chundoong Cluster, Xeon 8C 2GHz, Infiniband QDR, **AMD Radeon HD**	**1.47**
Bullx B505, Xeon 6C 2.53GHz, Infiniband QDR, **NVIDIA 2090**	**1.27**
Intel Cluster, Xeon 8C 2.6GHz, Infiniband FDR, **Intel Xeon Phi**	**1.27**
Xtreme-X , Xeon 8C 2.6GHz, Infiniband QDR, **NVIDIA 2090**	**1.05**

TOP500 Editions (40 so far, 20 years)

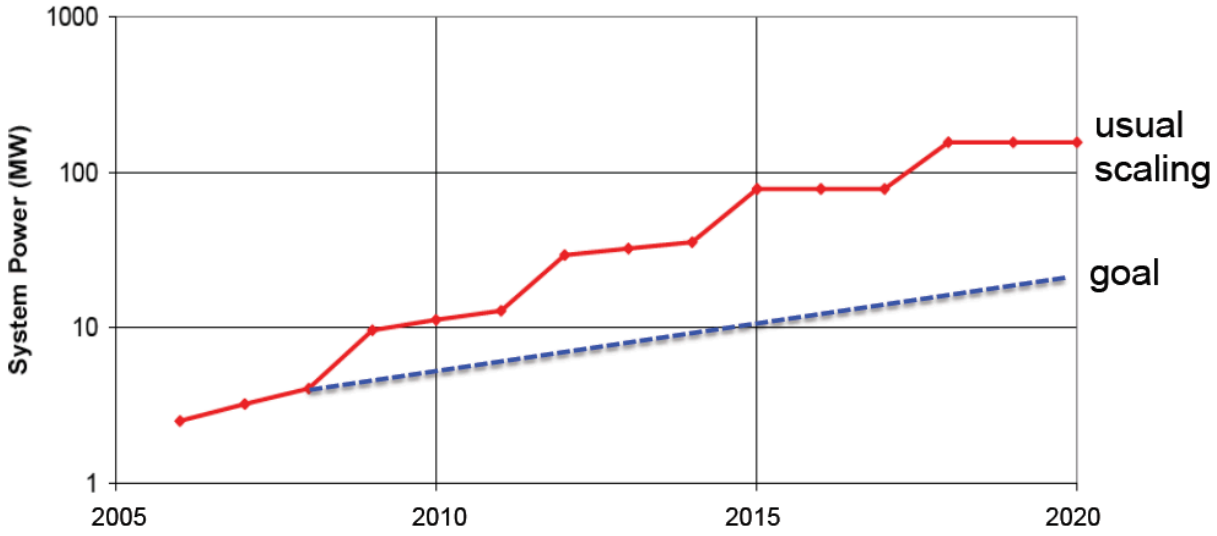


TOP500 Editions (53 edition, 26 years)



Energy Cost Challenge

- At ~$1M per MW energy costs are substantial
  - 10 Pflop/s in 2011 uses ~10 MWs
  - 1 Eflop/s in 2018 > 100 MWs



  - DOE Target: 1 Eflop/s around 2020-2022 at 20 MWs
  - 50 Gflops/W today around 2 Gflops/W

The High Cost of Data Movement

- Flop/s or percentage of peak flop/s become much less relevant

Approximate power costs (in picoJoules)

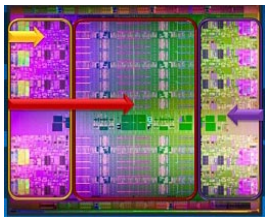
	2011
DP FMADD flop	100 pJ
DP DRAM read	4800 pJ
Local Interconnect	7500 pJ
Cross System	9000 pJ

Source: John Shalf, LBNL

- Algorithms & Software: minimize data movement; perform more work per unit data movement.

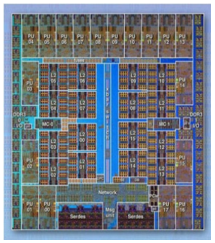
Technology Paths to Exascale

- **Multicore:** Maintain complex cores, and replicate (x86, SPARC, Power7) [#3, 6, and 10]



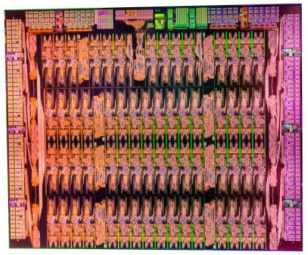
Intel Xeon E7
(10 cores)

- **Manycore/Embedded:** Use many simpler, low power cores from embedded (BlueGene, future ARM) [#2, 4, 5, and 9]



IBM BlueGene/Q
(16 +2 cores)

- **GPU/Coprocessor/Accelerator**: Use highly specialized processors from graphics market space (NVidia Fermi, Intel Xeon Phi, AMD) [# 1, 7, and 8]



Intel Xeon Phi
(60 cores)

Potential System Architecture

Systems	2012 Titan Computer
System peak	**27 Pflop/s**
Power	**8.3 MW** **(2 Gflops/W)**
System memory	710 TB (38*18688)
Node performance	1,452 GF/s (1311+141)
Node memory BW	232 GB/s (52+180)
Node concurrency	16 cores CPU 2688 CUDA cores
Total Node Interconnect BW	8 GB/s
System size (nodes)	18,688
Total concurrency	50 M
MTTF	?? unknown

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

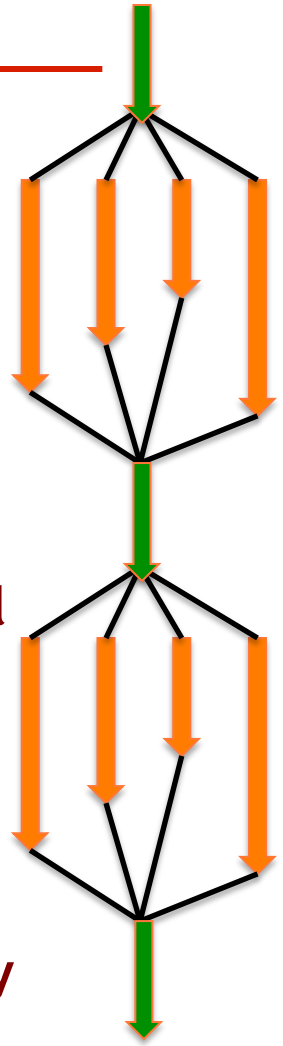
Systems	2012 Titan Computer	2022	Difference Today & 2022
System peak	**27 Pflop/s**	**1 Eflop/s**	**O(100)**
Power	**8.3 MW (2 Gflops/W)**	**~20 MW (50 Gflops/W)**	**O(10)**
System memory	710 TB (38*18688)	32 - 64 PB	O(100)
Node performance	1,452 GF/s (1311+141)	1.2 or 15TF/s	O(10)
Node memory BW	232 GB/s (52+180)	2 - 4TB/s	O(10)
Node concurrency	16 cores CPU 2688 CUDA cores	O(1k) or 10k	O(100) – O(10)
Total Node Interconnect BW	8 GB/s	200-400GB/s	O(100)
System size (nodes)	18,688	O(100,000) or O(1M)	O(10) – O(100)
Total concurrency	50 M	O(billion)	O(100)
MTTF	?? unknown	O(<1 day)	O(?)

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.

Motivation

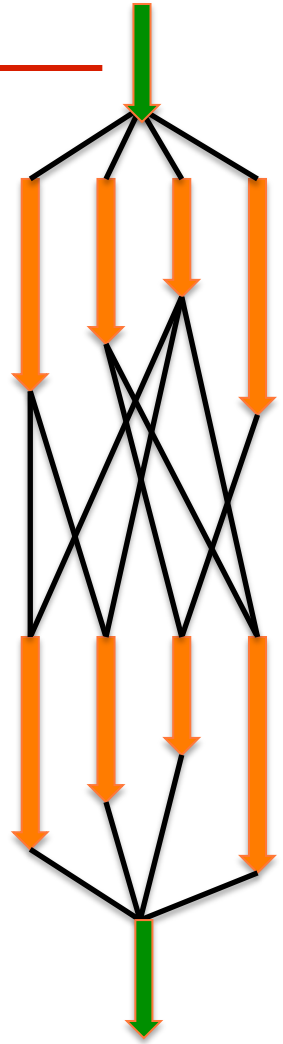
- Today software developers face systems with
  - > 1 TFLOP/s of compute power per node
  - 32+ of cores, 100+ hardware threads
  - Highly heterogeneous architectures (cores + specialized cores + accelerators/coprocessors)
  - Deep memory hierarchies
  - Today, we deal with thousands of them (plan to deal with millions)
  - ➔ systemic load imbalance / decreasing use of the resources
- How to harness these devices productively?
  - SPMD produces choke points, wasted wait times
  - We need to improve efficiency, power and reliability



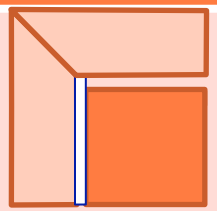
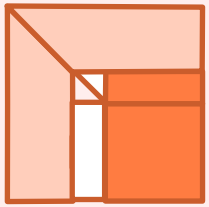
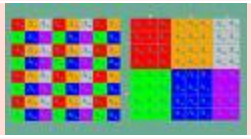
How to Program

- Threads & synchronization | Processes & Messages
  - Hand written Pthreads, compiler-based OpenMP, Chapel, UPC, MPI, hybrid
- Very challenging to find parallelism, to debug, to maintain and to get good performance
  - *Portably*
  - *With reasonable development efforts*

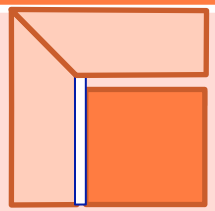
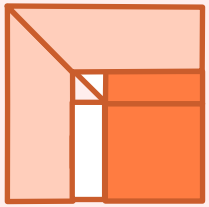
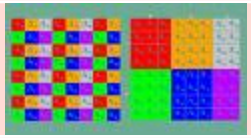
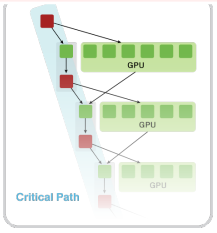
  When is it time to redesign a software?
- Increasing gaps between the capabilities of today's programming environments, the requirements of emerging applications, and the challenges of future parallel architectures



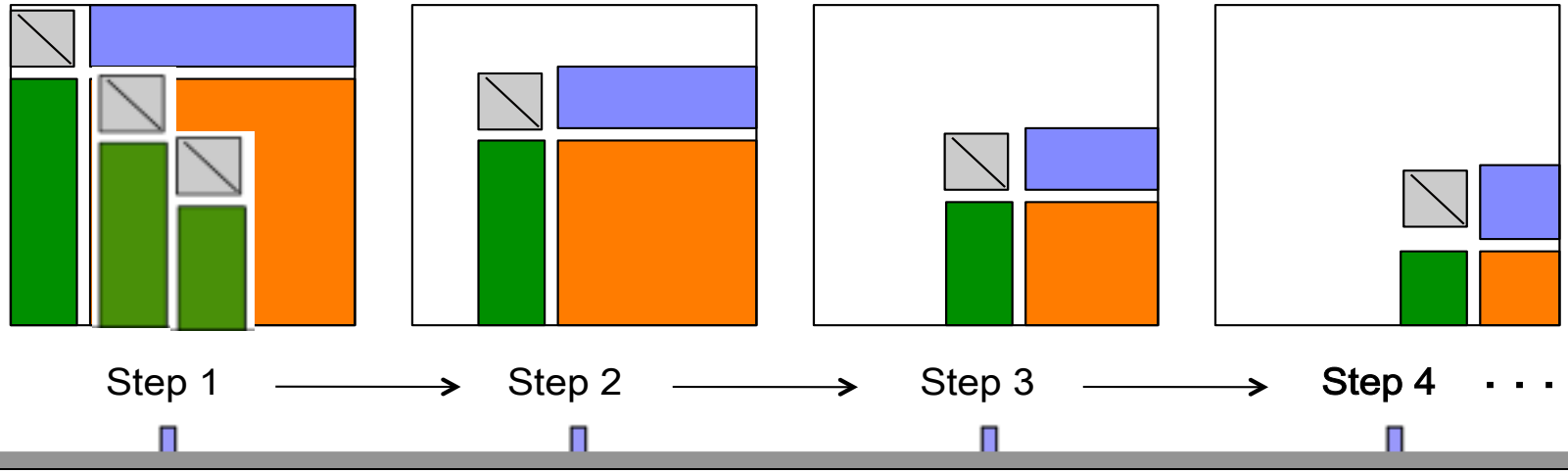
A New Generation of DLA Software

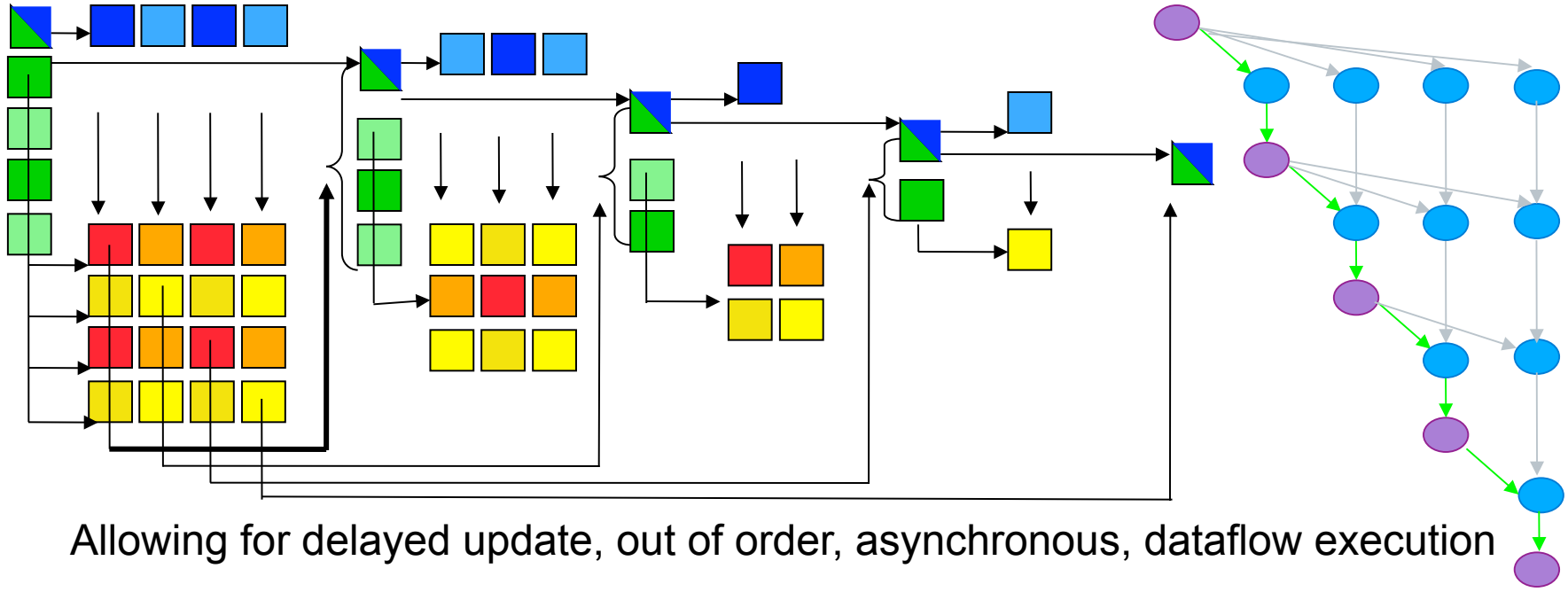
Software/Algorithms follow hardware evolution in time		
LINPACK (70's) (Vector operations)		Rely on - Level-1 BLAS operations
LAPACK (80's) (Blocking, cache friendly)		Rely on - Level-3 BLAS operations
ScaLAPACK (90's) (Distributed Memory)		Rely on - PBLAS Mess Passing

A New Generation of DLA Software

Software/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 New Algorithms (many-core friendly)		Rely on - a DAG/scheduler - block data layout - some extra kernels
MAGMA Hybrid Algorithms (heterogeneity friendly)		Rely on - hybrid scheduler - hybrid kernels

Synchronization (in LAPACK LU)

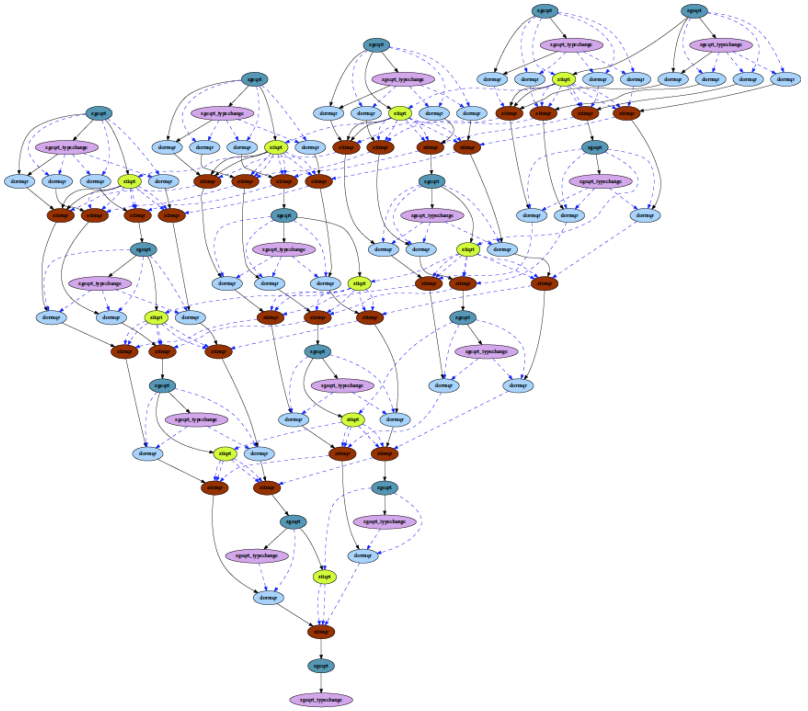


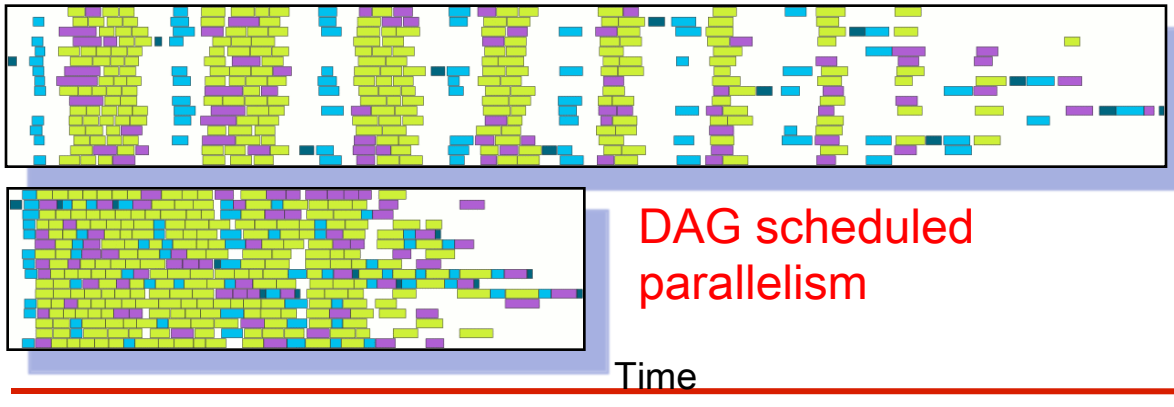


Allowing for delayed update, out of order, asynchronous, dataflow execution

Parallel Linear Algebra s/w for Multicore/Hybrid Architectures

- **Objectives**
  - High utilization of each core
  - Scaling to large number of cores
  - Synchronization reducing algorithms
- **Methodology**
  - Dynamic DAG scheduling (QUARK)
  - Explicit parallelism
  - Implicit communication
  - Fine granularity / block data layout
- **Arbitrary DAG with dynamic scheduling**



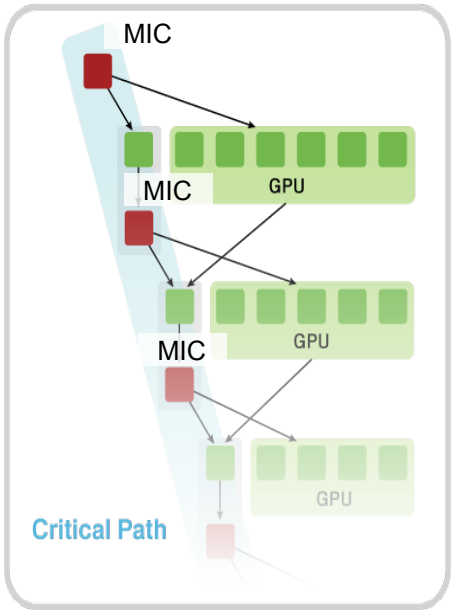


Methodology overview

A methodology to use all available resources:

- MAGMA MIC uses hybridization methodology based on
  - Representing linear algebra algorithms as collections of tasks and data dependencies among them
  - Properly scheduling tasks' execution over multicore CPUs and manycore coprocessors

Hybrid CPU+MIC algorithms (small tasks for multicores and large tasks for MICs)



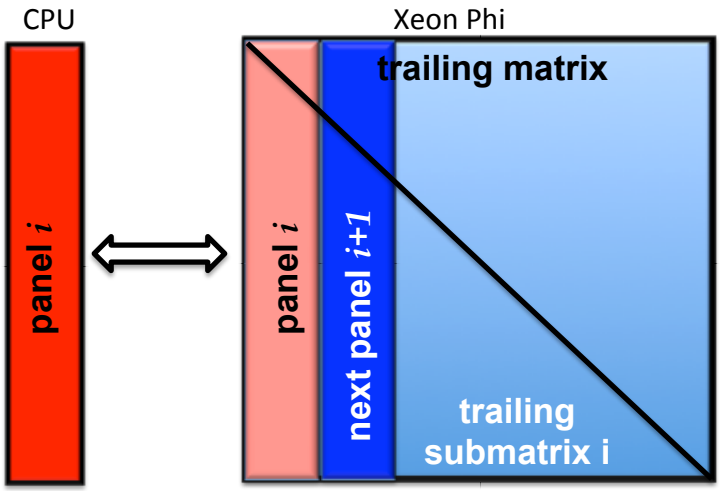
- Successfully applied to fundamental linear algebra algorithms
  - One- and two-sided factorizations and solvers
  - Iterative linear and eigensolvers

- Productivity
  1) High level;
  2) Leveraging prior developments;
  3) Exceeding in performance homogeneous solutions

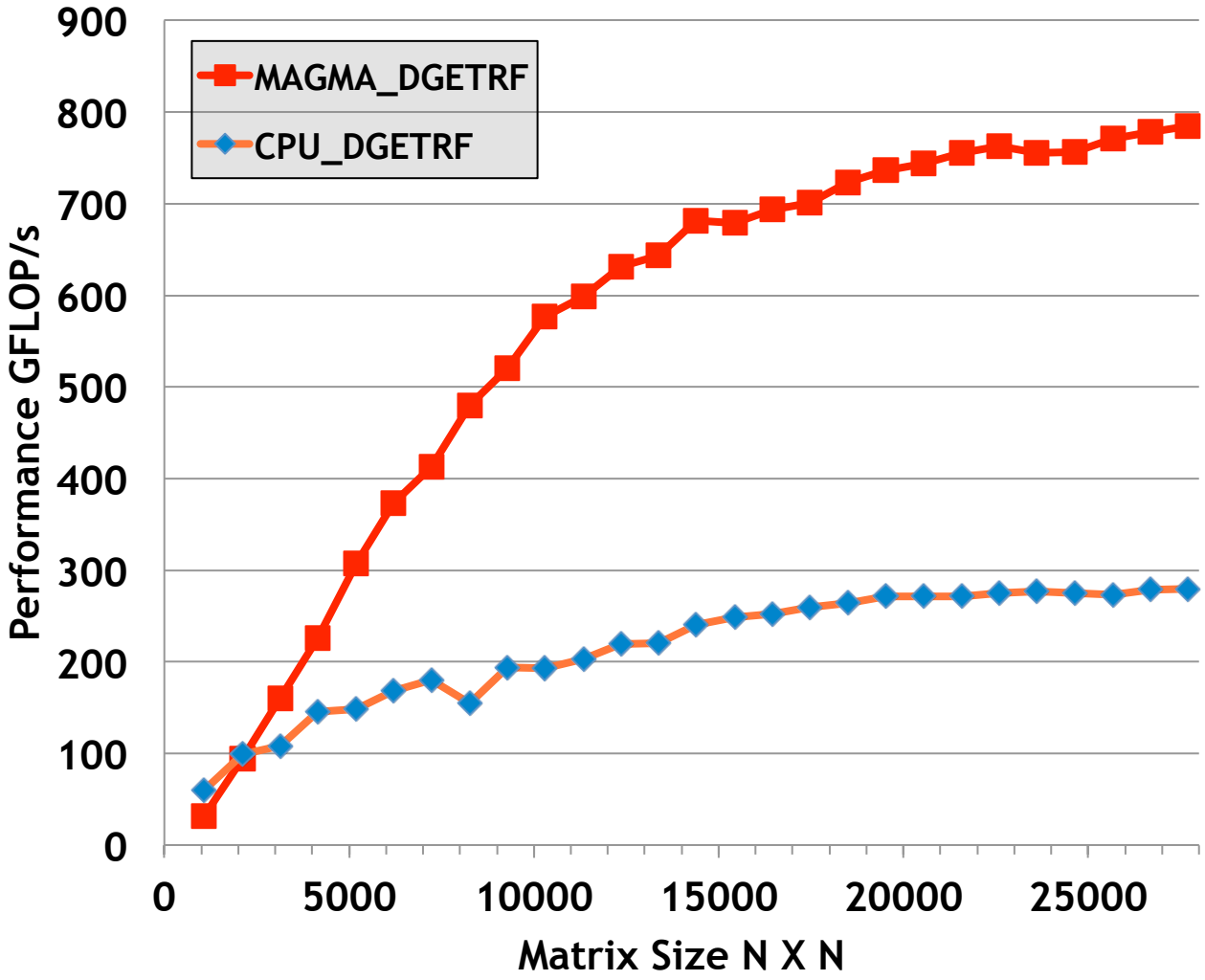
Hybrid Algorithms

One-Sided Factorizations (LU, QR, and Cholesky)

- **Hybridization**
  - **Panels (Level 2 BLAS) are factored on CPU using LAPACK**
  - **Trailing matrix updates (Level 3 BLAS) are done on the Accelerator using "look-ahead"**



MAGMA MIC Performance (LU)



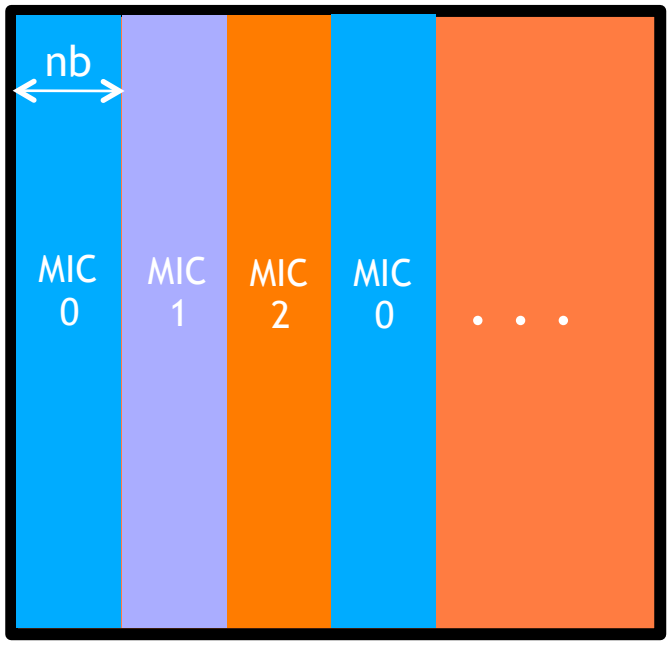
Host
Sandy Bridge (2 x 8 @2.6 GHz)
DP Peak 332 GFlop/s

Coprocessor
Intel Xeon Phi (60 @ 1.09 GHz)
DP Peak 1046 GFlop/s

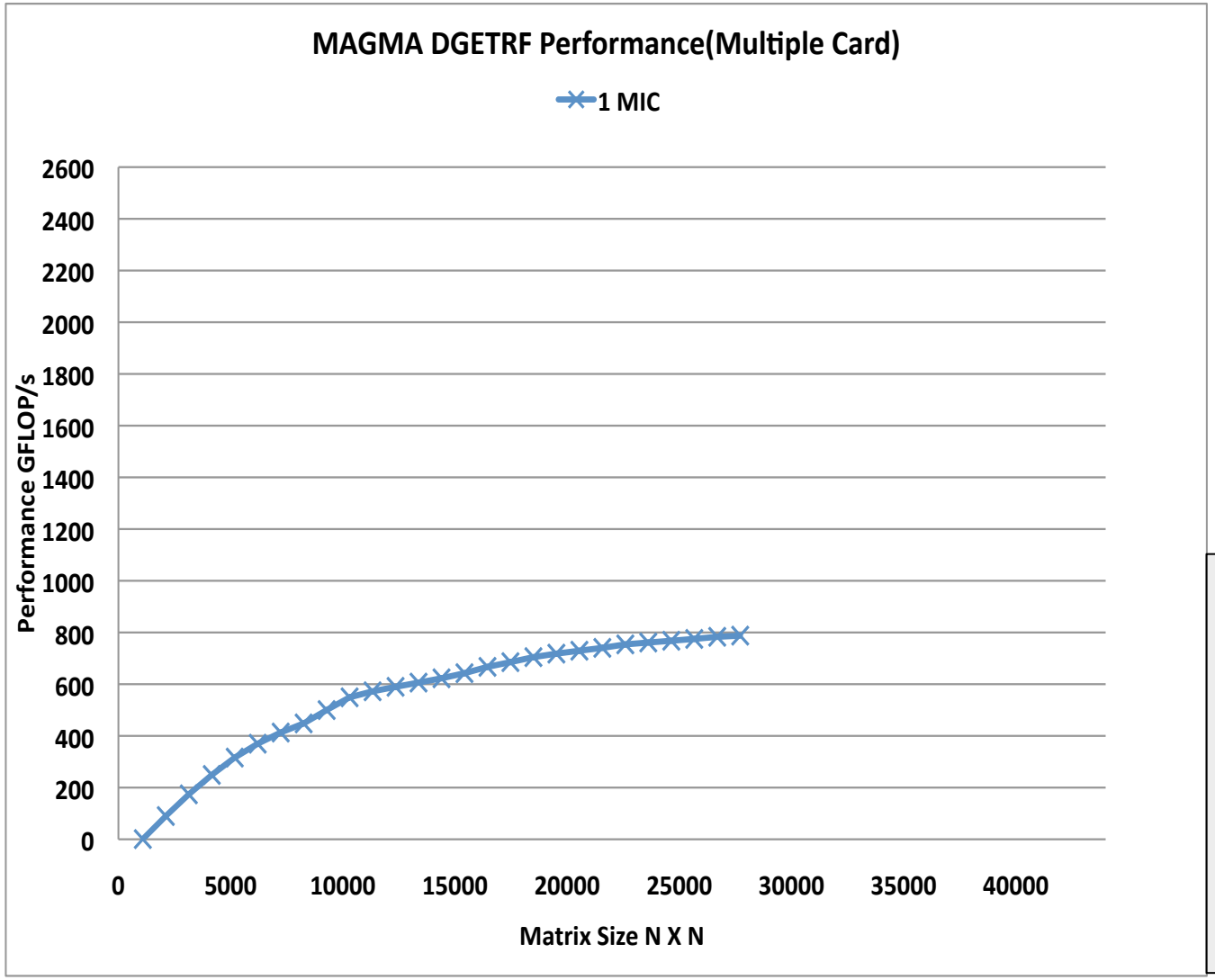
System DP Peak 1378 GFlop/s
MPSS 2.1.4346-16
compiler_xe_2013.1.117

From Single to MultiMIC Support

- Data distribution
  - 1-D block-cyclic distribution
- Algorithm
  - MIC holding current panel is sending it to CPU
  - All updates are done in parallel on the MICs
  - Look-ahead is done with MIC holding the next panel



MAGMA MIC Scalability
LU Factorization Performance in DP

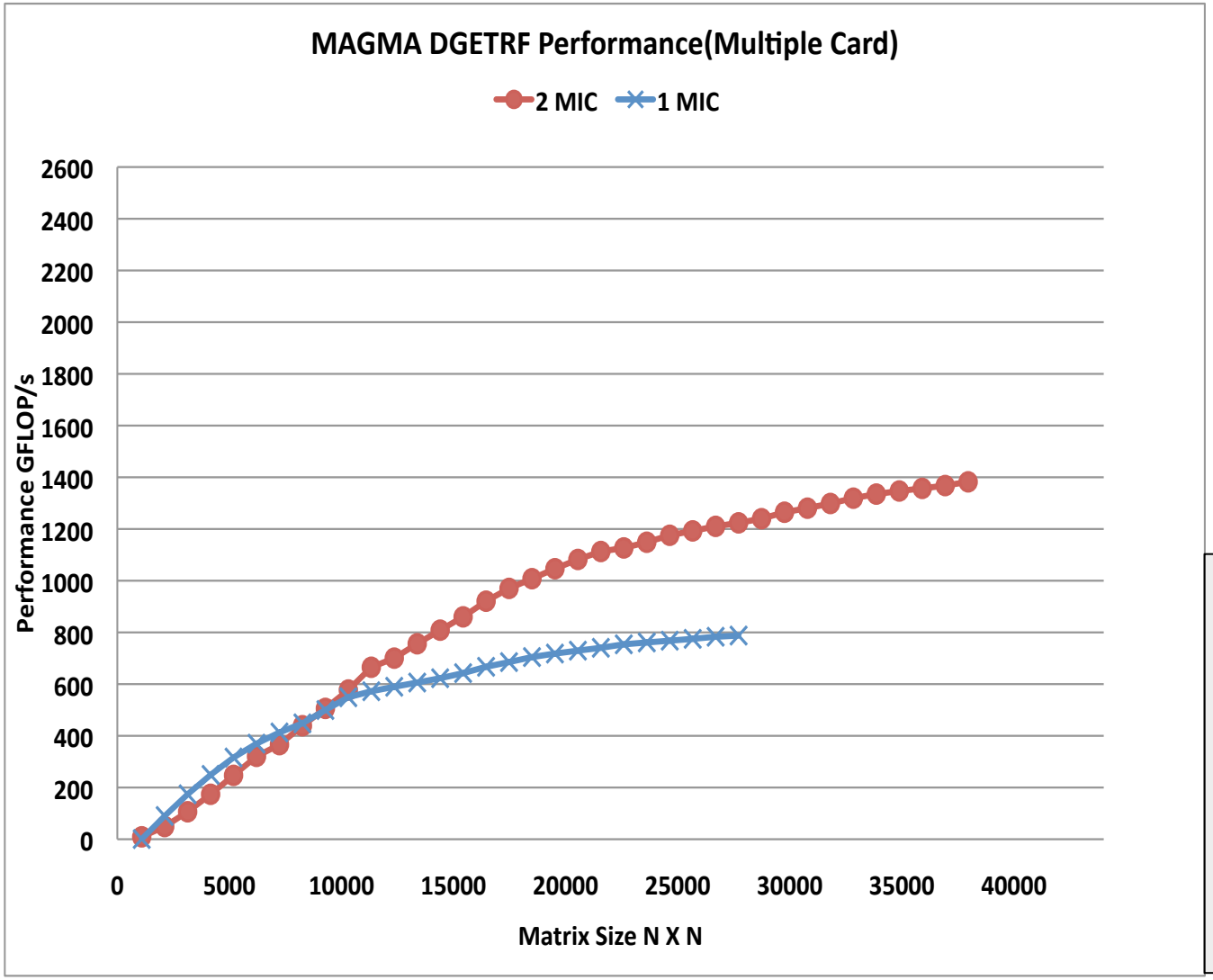


Host
Sandy Bridge (2 x 8 @2.6 GHz)
DP Peak 332 GFlop/s

Coprocessor
Intel Xeon Phi (60 @ 1.09 GHz)
DP Peak 1046 GFlop/s

System DP Peak 1378 GFlop/s
MPSS 2.1.4346-16
compiler_xe_2013.1.117

MAGMA MIC Scalability
LU Factorization Performance in DP

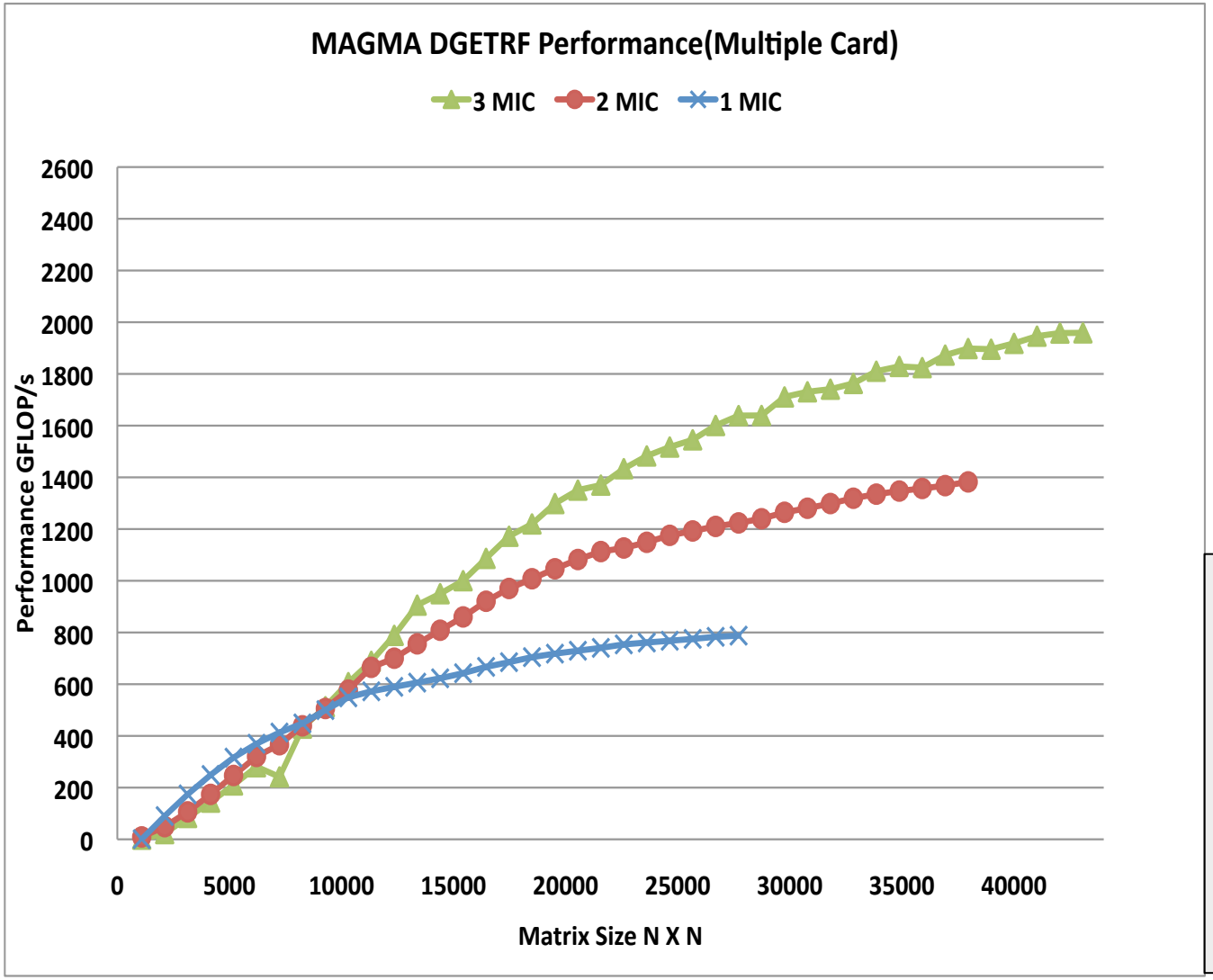


Host
Sandy Bridge (2 x 8 @2.6 GHz)
DP Peak 332 GFlop/s

Coprocessor
Intel Xeon Phi (60 @ 1.09 GHz)
DP Peak 1046 GFlop/s

System DP Peak 1378 GFlop/s
MPSS 2.1.4346-16
compiler_xe_2013.1.117

MAGMA MIC Scalability
LU Factorization Performance in DP

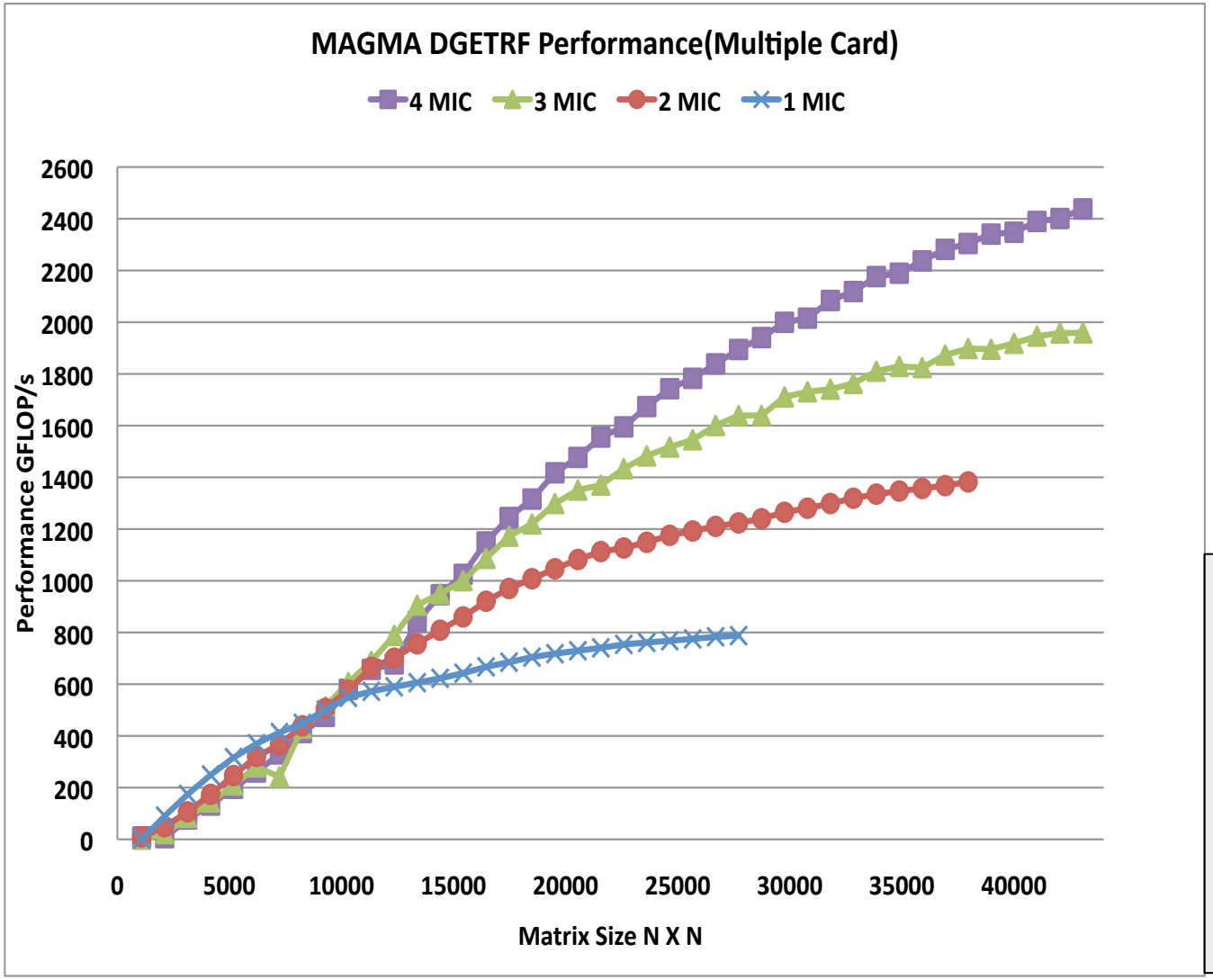


Host
Sandy Bridge (2 x 8 @2.6 GHz)
DP Peak 332 GFlop/s

Coprocessor
Intel Xeon Phi (60 @ 1.09 GHz)
DP Peak 1046 GFlop/s

System DP Peak 1378 GFlop/s
MPSS 2.1.4346-16
compiler_xe_2013.1.117

MAGMA MIC Scalability
LU Factorization Performance in DP



Host
Sandy Bridge (2 x 8 @2.6 GHz)
DP Peak 332 GFlop/s

Coprocessor
Intel Xeon Phi (60 @ 1.09 GHz)
DP Peak 1046 GFlop/s

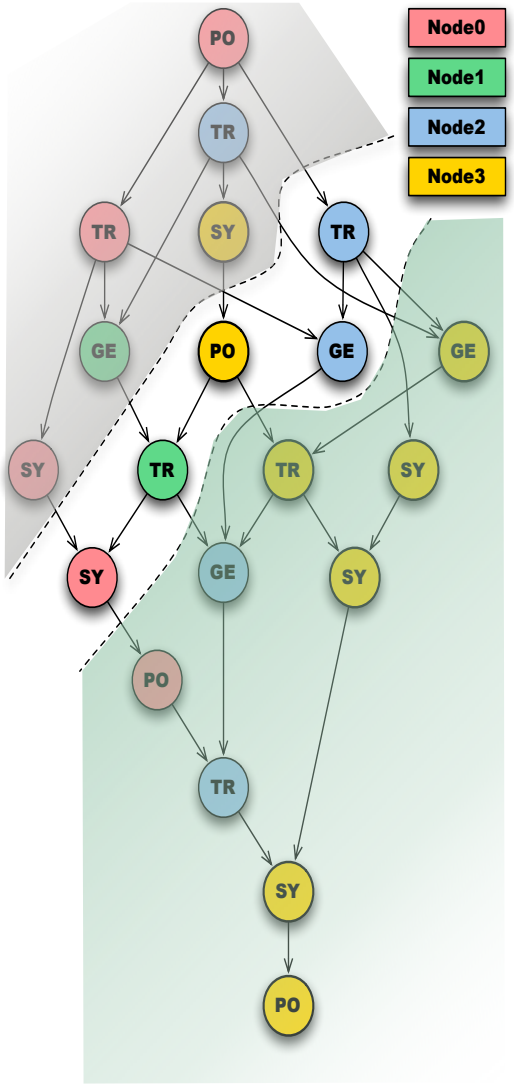
System DP Peak 1378 GFlop/s
MPSS 2.1.4346-16
compiler_xe_2013.1.117

Distributed Memory Runtime System

- **Parallel Runtime Scheduler & Execution Control**
  - Executes a dataflow representation of a program
  - Scheduler provides
    - Automatic load-balance between cores
    - Harness the power of accelerators (GPU, Mic, etc)
  - Works on large scale distributed memory machines
    - Communications are implicit, overlapped
    - user defined Communication pattern and data-distribution

Prominent feature: *Parameterized Task Graph*

Runtime DAG scheduling



- Every process has the symbolic DAG representation
  - Only the (node local) frontier of the DAG is considered
  - Distributed Scheduling based on remote completion notifications
- Background remote data transfer automatic with overlap
- NUMA / Cache aware Scheduling
  - Work Stealing and sharing based on memory hierarchies

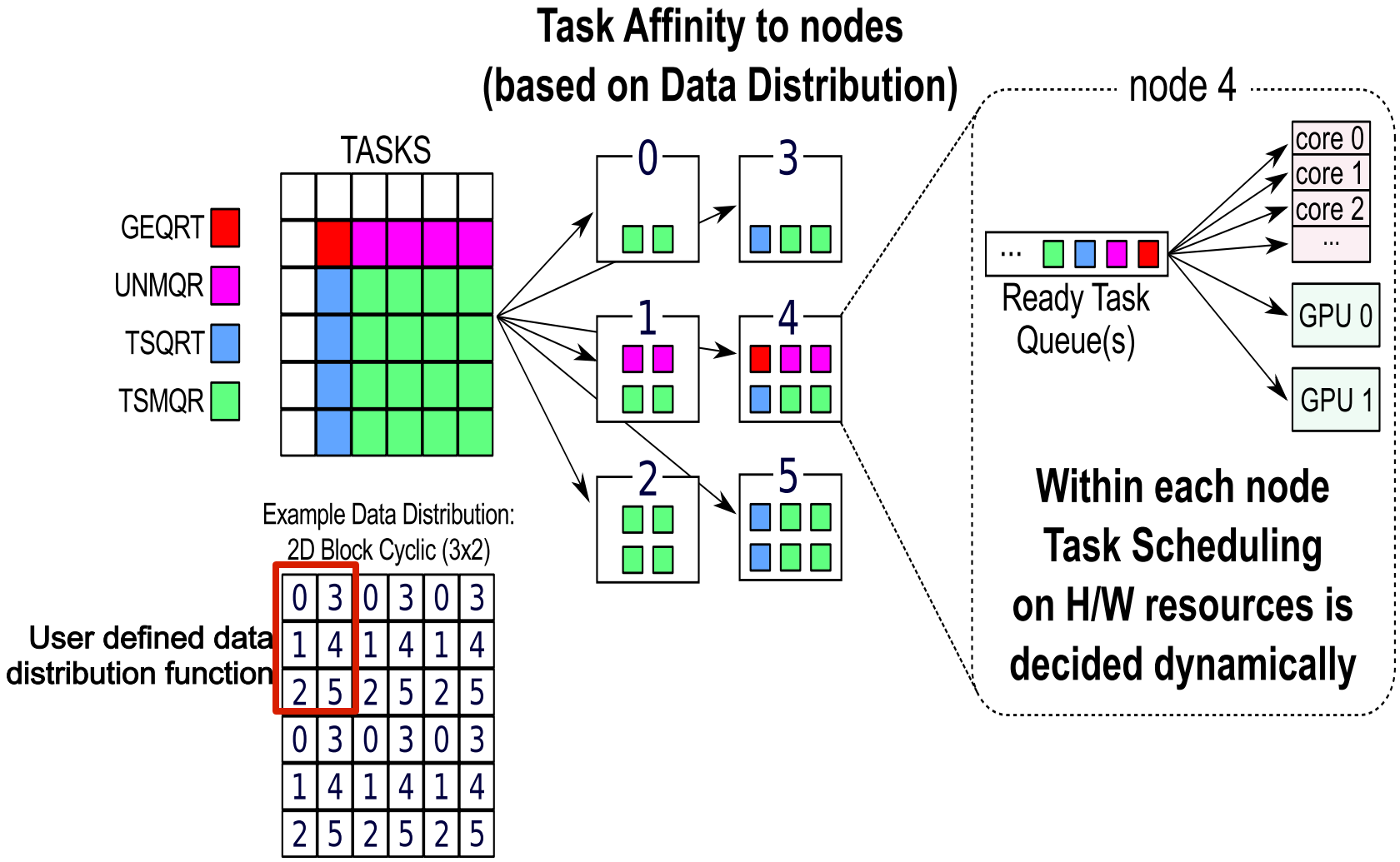
Related Work

	PaRSEC	SMPss	StarPU	Charm ++	FLAME	QUARK	Tblas	PTG
Scheduling	Distr. (1/core)	Repl (1/node)	Repl (1/node)	Distr. (Actors)	w/ SuperMatrix	Repl (1/node)	Centr.	Centr.
Language	Internal or Seq. w/ Affine Loops or w/ add_task	Seq. w/ add_task	Seq. w/ add_task	Msg-Driven Objects	Internal (LA DSL)	Seq. w/ add_task	Seq. w/ add_task	Internal
Accelerator	GPU	GPU	GPU		GPU	GPU		
Availability	Public	Public	Public	Public	Public	Public	Not Avail.	Not Avail.

Early stage: ParalleX
Non-academic: Swarm, MadLINQ, CnC

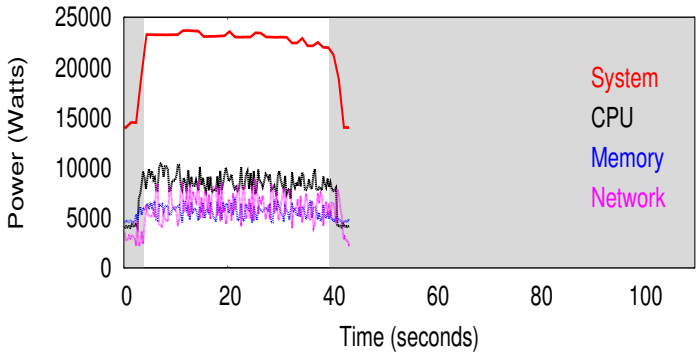
All projects support Distributed and Shared Memory
(QUARK with QUARKd; FLAME with Elemental)

Task Affinity in PaRSEC

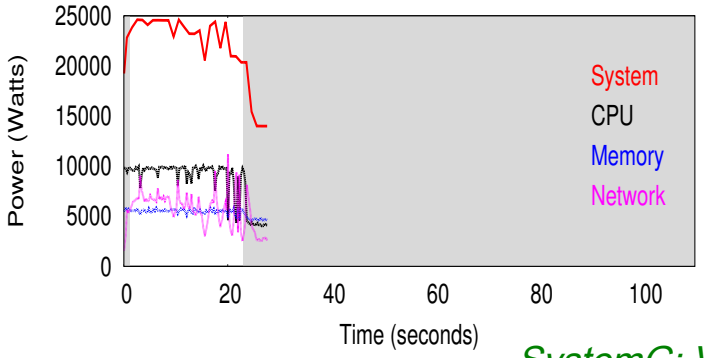


Energy efficiency

QR factorization (256 cores)



(a) ScaLAPACK.



(b) DPLASMA.

Total energy consumption

# Cores	Library	Cholesky	QR
128	ScaLAPACK	192000	672000
	DPLASMA	128000	540000
256	ScaLAPACK	240000	816000
	DPLASMA	96000	540000
512	ScaLAPACK	325000	1000000
	DPLASMA	125000	576000

Work in progress with Hatem Ltaief

- **Energy used depending on the number of cores**
- **Up to 62% more energy efficient while using a high performance tuned scheduling**
  - **Power efficient scheduling**

SystemG: Virginia Tech Energy Monitored cluster (ib40g, intel, 8cores/node)

Summary

- Major Challenges are ahead for extreme computing
  - Parallelism O(10^9)
    - Programming issues
  - Hybrid
    - Peak and HPL may be very misleading
    - No where near close to peak for most apps
  - Fault Tolerance
    - Today Sequoia BG/Q node failure rate is 1.25 failures/day
  - Power
    - 50 Gflops/w (today at 2 Gflops/w)
- We will need completely new approaches and technologies to fully embrace the Exascale level

Collaborators / Software / Support

- **PLASMA**
  http://icl.cs.utk.edu/plasma/
- **MAGMA**
  http://icl.cs.utk.edu/magma/
- **Quark (RT for Shared Memory)**
- **http://icl.cs.utk.edu/quark/**
- **PaRSEC(**Parallel Runtime Scheduling and Execution Control**)**
- **http://icl.cs.utk.edu/parsec/**

- Collaborating partners
  University of Tennessee, Knoxville
  University of California, Berkeley
  University of Colorado, Denver

  INRIA, France
  KAUST, Saudi Arabia

These tools are being applied to a range of applications beyond dense LA:
Sparse direct, Sparse iterations methods and Fast Multipole Methods