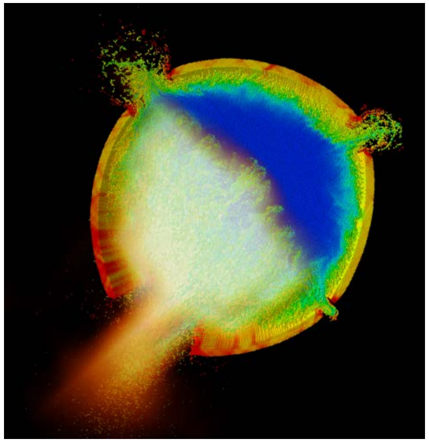
# Petascale Parallel Computing and Beyond

## General trends and lessons



Martin Berzins

1. Technology Trends
2. Towards Exascale
3. Trends in programming large scale systems

THE UNIVERSITY OF UTAH

**What kind of machine will you use in 2020?**

**What kind of problems will be solved**

**HPC power has increased by a factor of 1000 every decade.**

**Present state of the architectures see http://www.euroben.nl/reports/overview10.pdf**

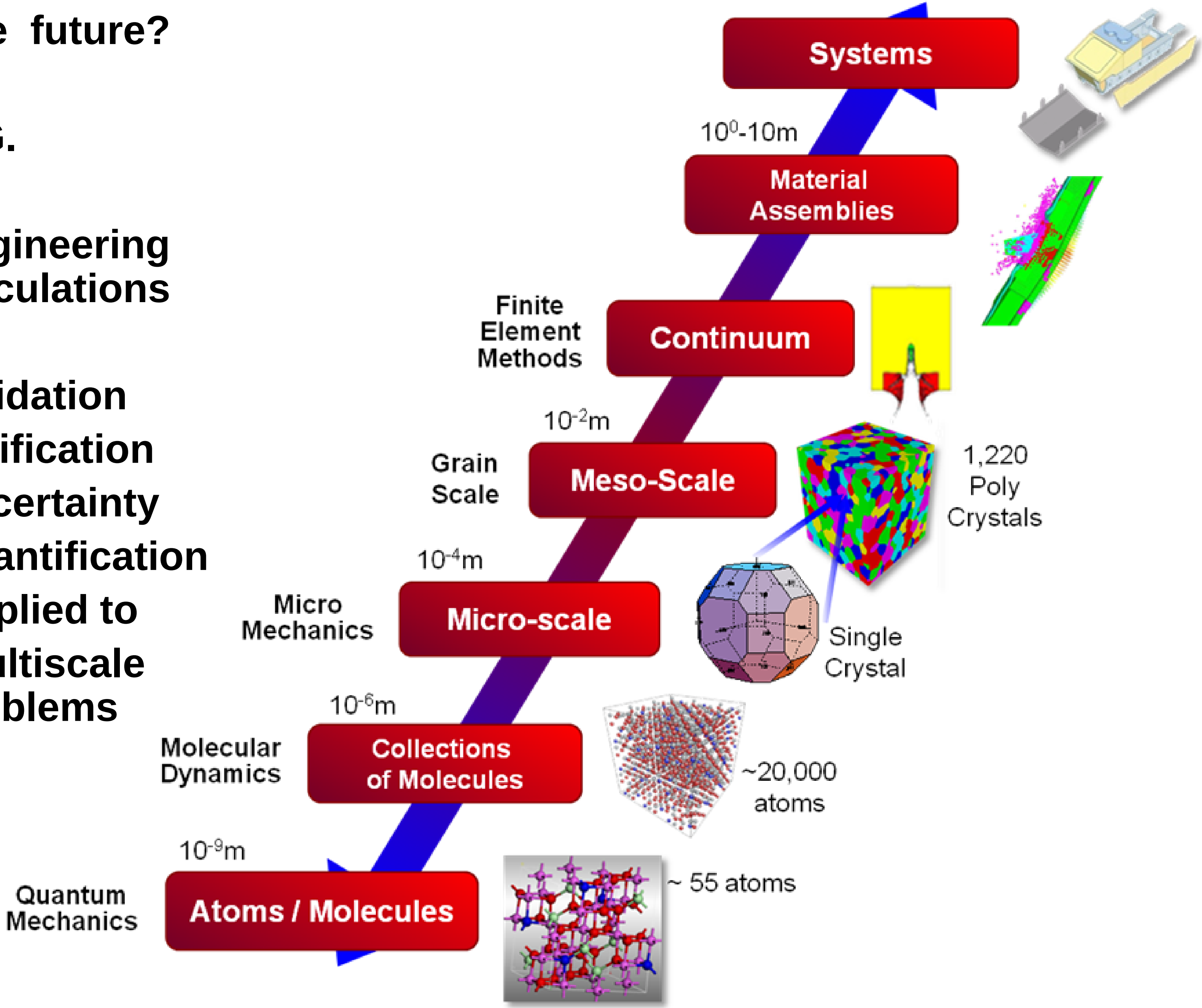
THE UNIVERSITY OF UTAH

# The future?

E.G.

Engineering calculations

Validation
Verification
Uncertainty
Quantification
applied to
multiscale
problems



Systems

$10^{0}$-10m

Material Assemblies

Finite Element Methods

Continuum

$10^{-2}$m

Grain Scale

Meso-Scale

1,220 Poly Crystals

$10^{-4}$m

Micro Mechanics

Micro-scale

Single Crystal

$10^{-6}$m

Molecular Dynamics

Collections of Molecules

~20,000 atoms

$10^{-9}$m

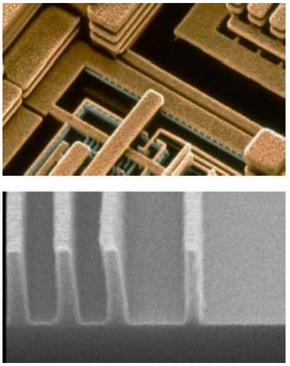
Quantum Mechanics

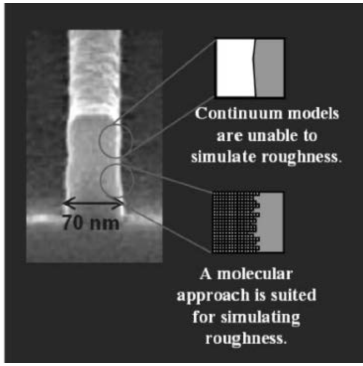
Atoms / Molecules

~ 55 atoms

# Predictive Computational Science [Oden Karniadakis]

## Predictive Computational Science is changing e.g. nano-maufacturing



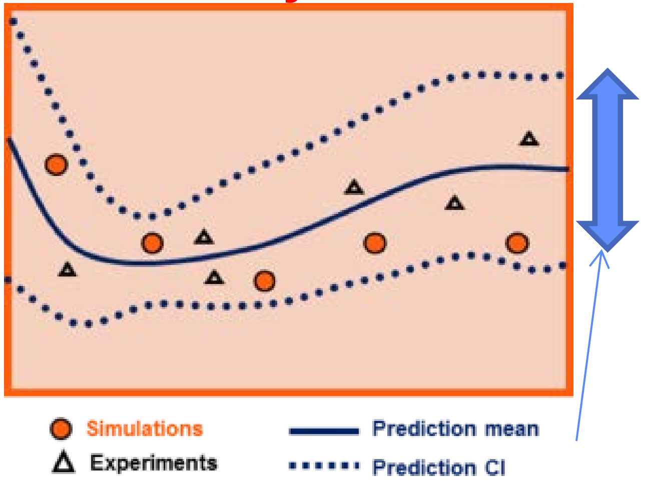

Continuum models are unable to simulate roughness.
70 nm
A molecular approach is suited for simulating roughness.

Science is based on subjective probability in which predictions must account for uncertainties in parameters, models, and experimental data . This involves many “experts” who are often wrong

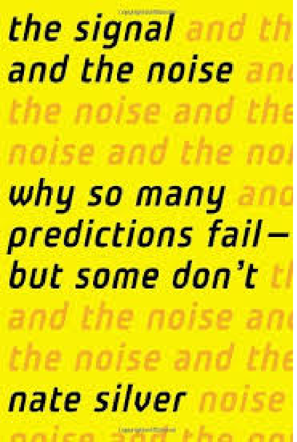
**We cannot deliver predictive engineering design over the next decade without quantifying uncertainty**

**Predictive Computational Science:**

Successful models are verified (codes) and validated (experiments) **(V&V)**. The uncertainty in computer predictions (the QoI’s) must be quantified if the predictions are used in important decisions. **(UQ)**



Confidence interval



***“Uncertainty is an essential and non-negotiable part of a forecast.***

***Quantifying uncertainty carefully and explicitly is essential to scientific progress.” Nate Silver***

# CURRENT DEVELOPMENTS IN PROCESSORS & HPC

Time of rapid technological change
Processors, parallel machines, graphics chips, cloud computing, networks, storage are all changing very quickly right now….

Petaflop reached by two DoE machines in 2009
17 Petaflop reached in 2012 Titan (GPU based).
33 Petaflop reached in 2013 Tianhe-2
(Intel MIC based).

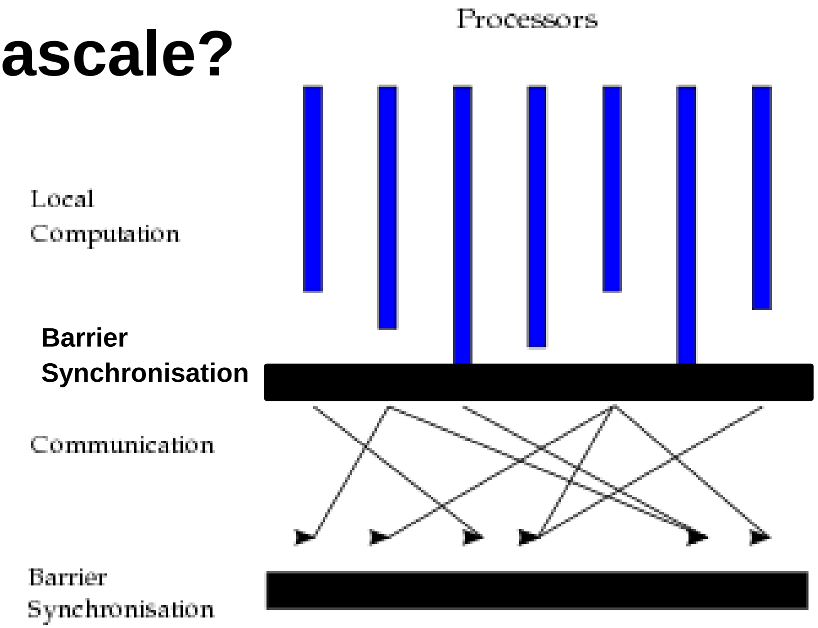
The moves are now to peak and sustained petascale performance and to begin to plan for the development of exascale machines
A major challenge is to build such a machine running at 20 MW

1 Teraflop = 10**12 flops = 1000 Gigaflops, 1 Gigaflop = 1000 megaflops, 10**9
1 Petaflop = 10**15 flops 1 Exaflop = 10**18 flops

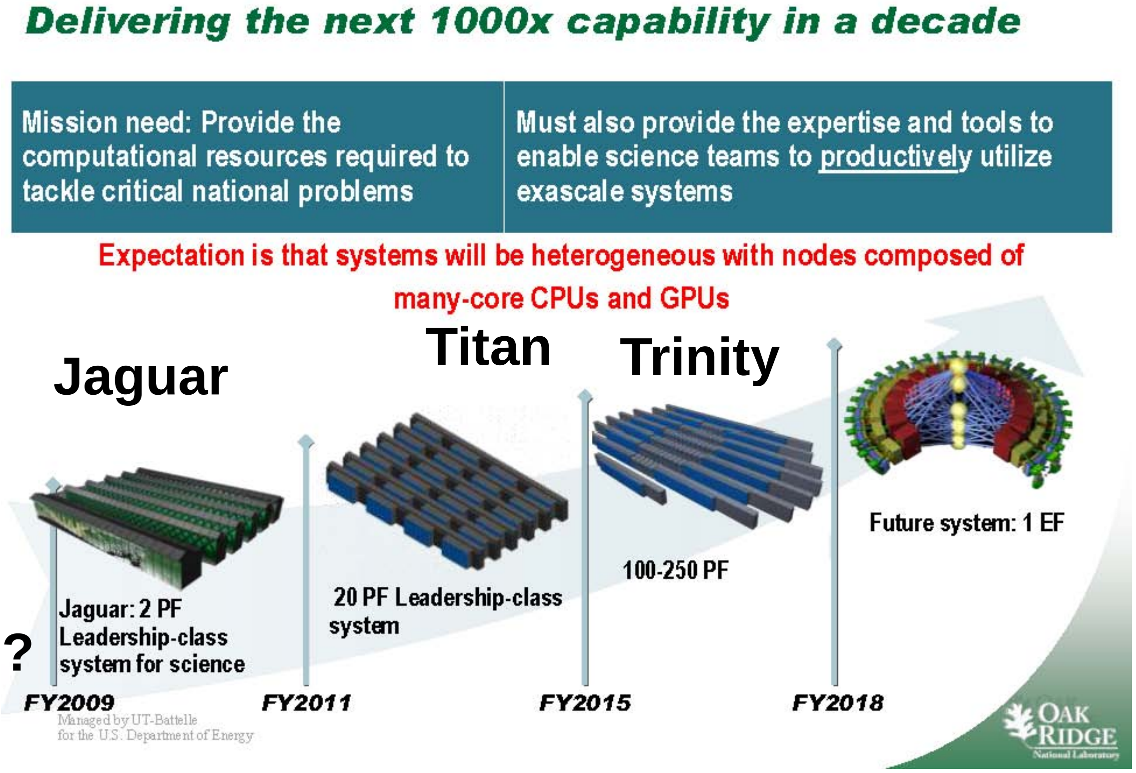
# Programming Models for Petascale?

**Harrod SC12: “today’s bulk synchronous (BSP), distributed memory, execution model is approaching an efficiency, scalability, and power wall.”**

- **Bulk synchronous approach**
- **MPI-only**
- **Costly global sync points**



**DOE ROADMAP [Geist]**

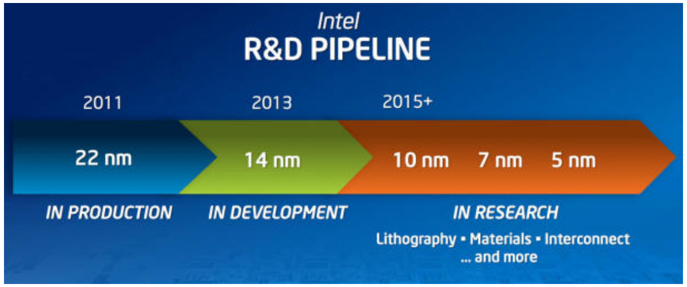

*Delivering the next 1000x capability in a decade*

Mission need: Provide the computational resources required to tackle critical national problems

Must also provide the expertise and tools to enable science teams to productively utilize exascale systems

Expectation is that systems will be heterogeneous with nodes composed of many-core CPUs and GPUs

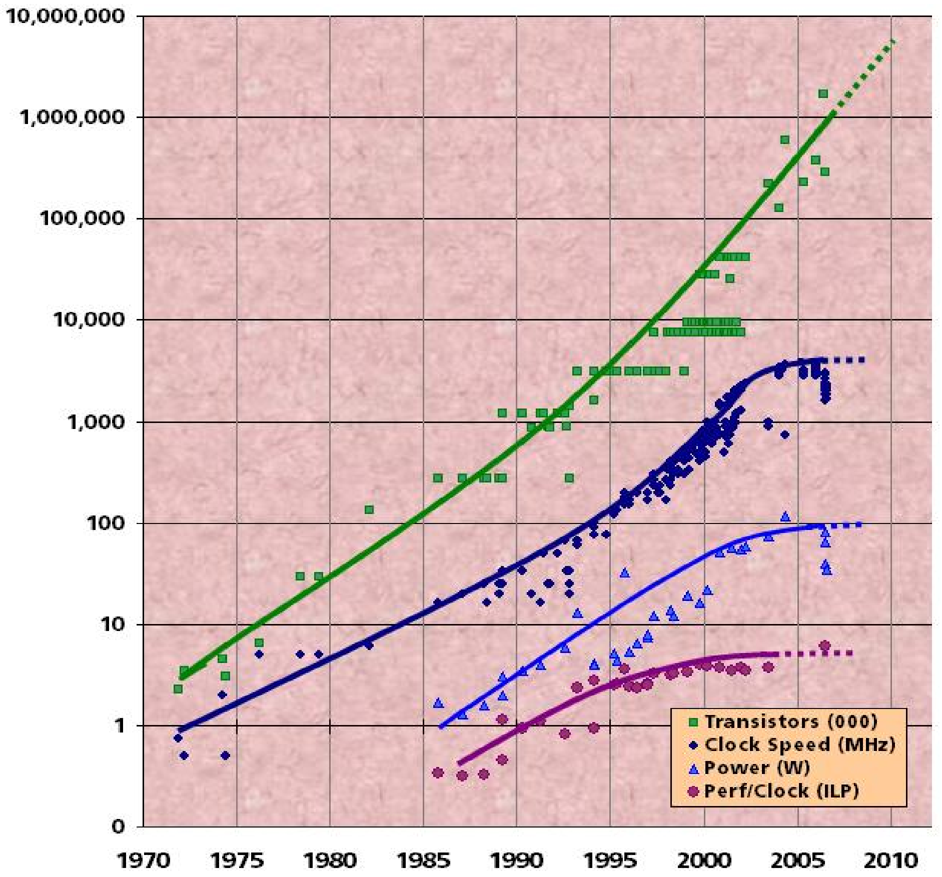
Jaguar
Titan
Trinity
Jaguar: 2 PF Leadership-class system for science
20 PF Leadership-class system
100-250 PF
Future system: 1 EF
FY2009
FY2011
FY2015
FY2018
Managed by UT-Battelle for the U.S. Department of Energy
Oak Ridge National Laboratory



**Possible but ?**

# Why worry about Parallel Computing ?

- **Energy problems** mean that processor clock speeds can't easily increase anymore.
- **Improved processes** mean that chips with feature sizes of 45nm (and below) are both here and possible  Feature size is half the distance between cells in a dynamic RAM memory chip.
- **Moore's Law** perhaps now means that the number of cores doubles every 18 months.

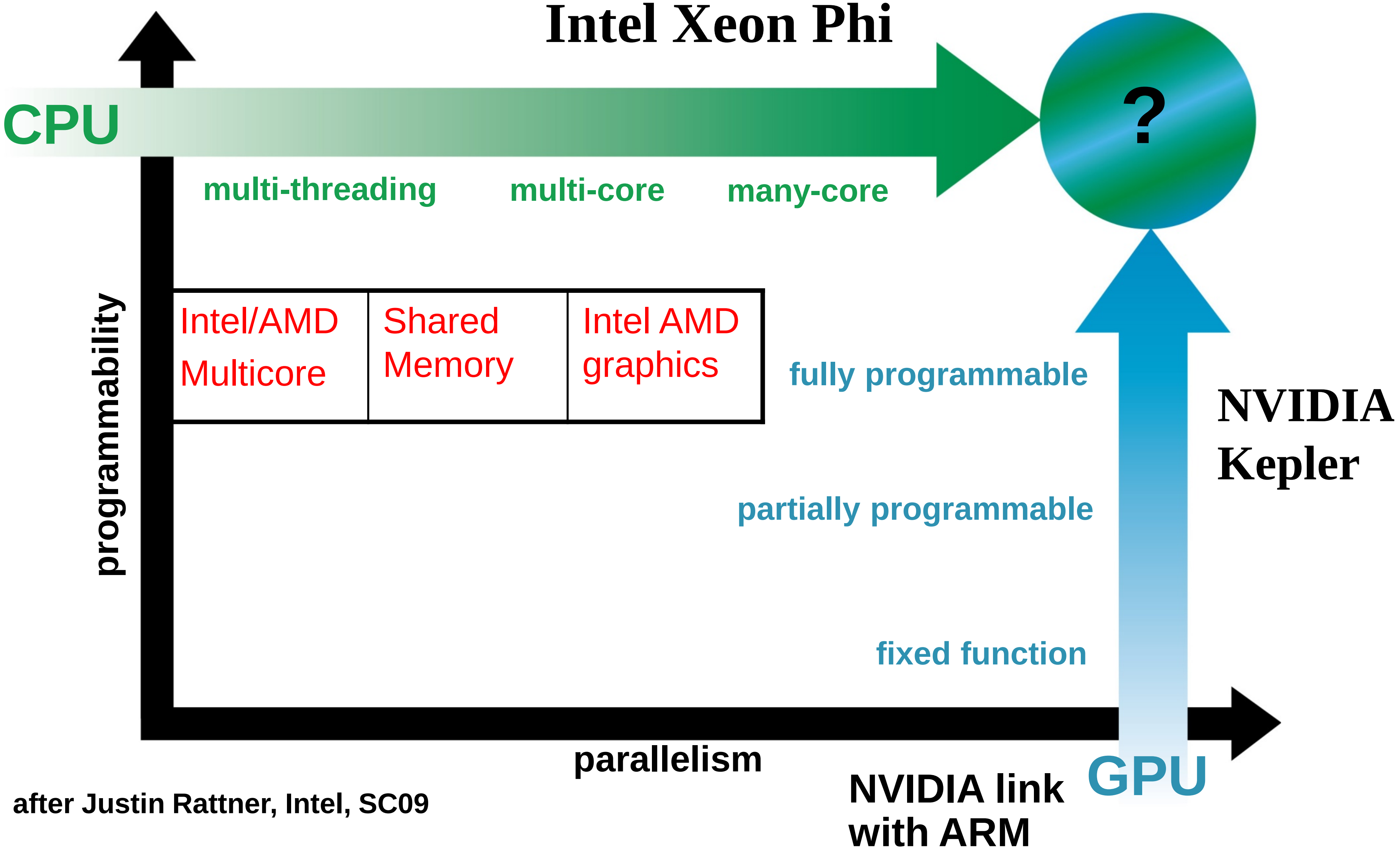


| High volume manufacturing | 2008 | 2010 | 2012 | 2014 | 2016 | 2018 | 2020 | 2022 |
|---|---|---|---|---|---|---|---|---|
| Feature size | 45 | 32 | 22 | 16 | 11 | 8 | 6 | 4 |
| Number of cores | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 |

**Are the commercial mass-market drivers there? Telemedicine?**

**New processes e.g GaN  7nm ?  Is this real?**

# Collision or Convergence?



Intel Xeon Phi

CPU

multi-threading

multi-core

many-core

?

| | | |
|---|---|---|
| Intel/AMD Multicore | Shared Memory | Intel AMD graphics |

fully programmable

partially programmable

fixed function

NVIDIA Kepler

programmability

parallelism

GPU

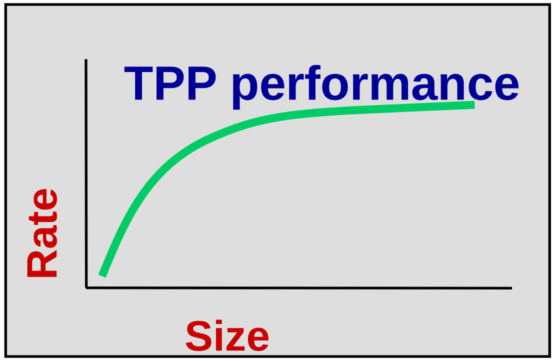
NVIDIA link with ARM

after Justin Rattner, Intel, SC09

## H. Meuer, H. Simon, E. Strohmaier, & J. Dongara

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

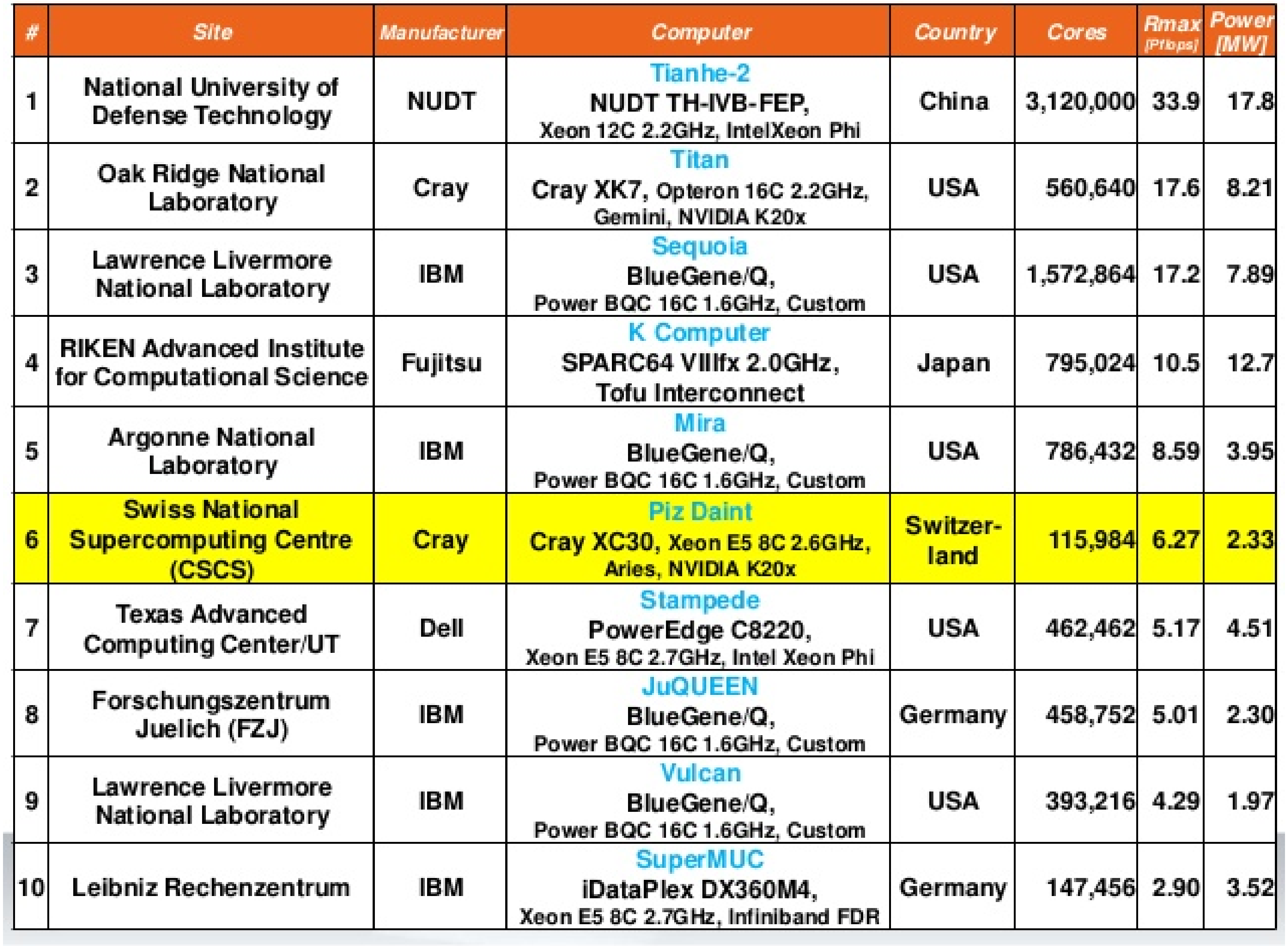
  $Ax=b$, *dense problem*



- Updated twice a year

SC‘xy in the States in November
Meeting in Germany in June

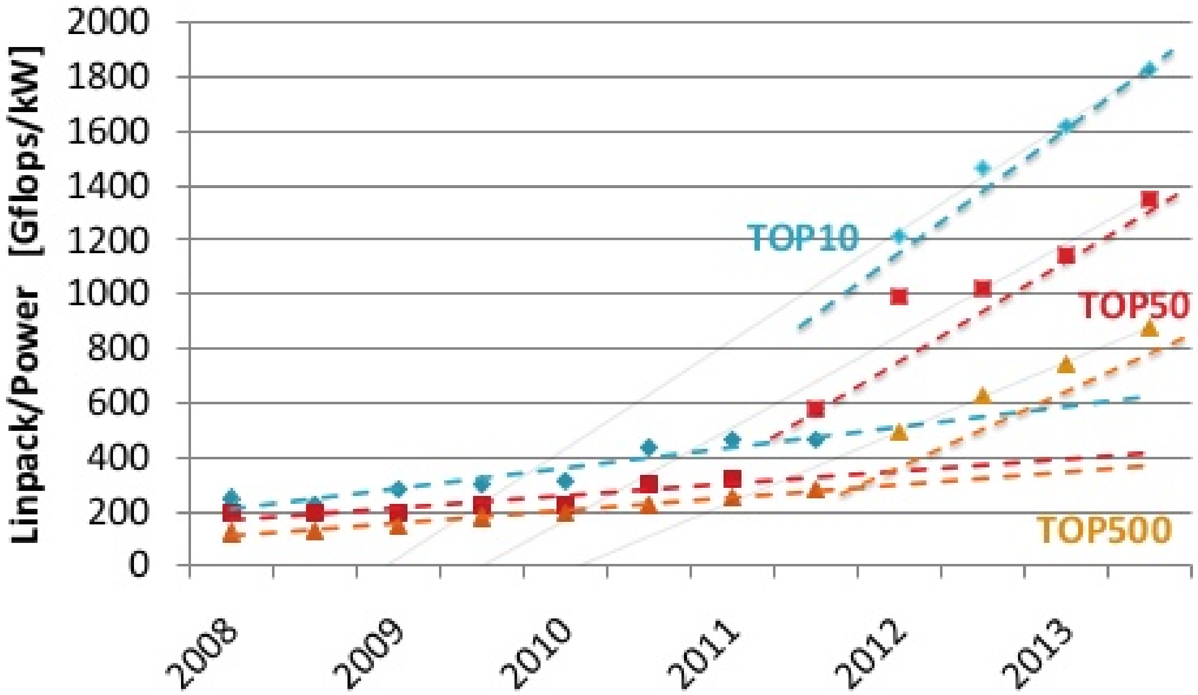
- All data available from **www.top500.org**



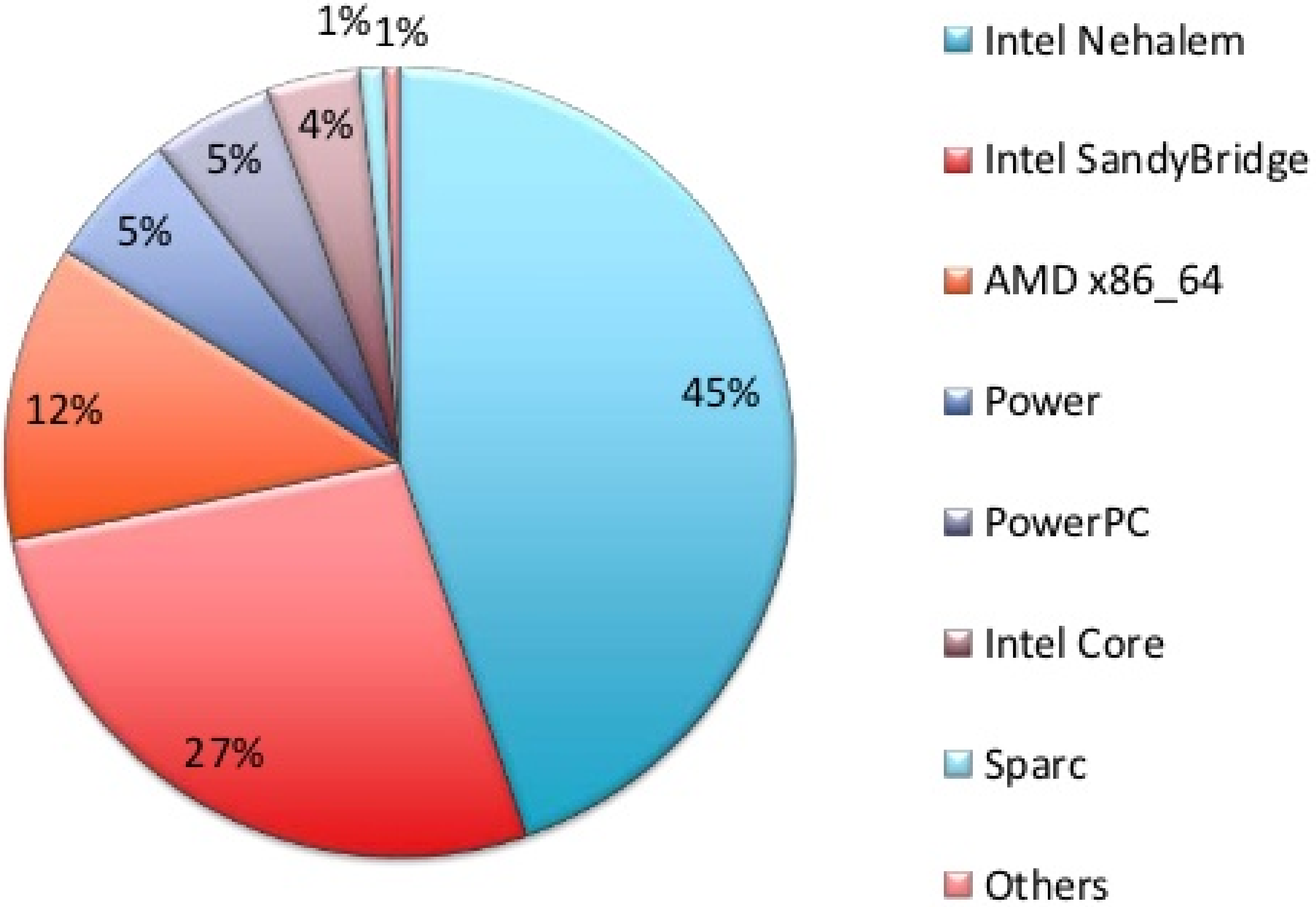
| # | Site | Manufacturer | Computer | Country | Cores | Rmax [Pflops] | Power [MW] |
|---|---|---|---|---|---|---|---|
| 1 | National University of Defense Technology | NUDT | Tianhe-2<br>NUDT TH-IVB-FEP,<br>Xeon 12C 2.2GHz, IntelXeon Phi | China | 3,120,000 | 33.9 | 17.8 |
| 2 | Oak Ridge National Laboratory | Cray | Titan<br>Cray XK7, Opteron 16C 2.2GHz,<br>Gemini, NVIDIA K20x | USA | 560,640 | 17.6 | 8.21 |
| 3 | Lawrence Livermore National Laboratory | IBM | Sequoia<br>BlueGene/Q,<br>Power BQC 16C 1.6GHz, Custom | USA | 1,572,864 | 17.2 | 7.89 |
| 4 | RIKEN Advanced Institute for Computational Science | Fujitsu | K Computer<br>SPARC64 VIIIfx 2.0GHz,<br>Tofu Interconnect | Japan | 795,024 | 10.5 | 12.7 |
| 5 | Argonne National Laboratory | IBM | Mira<br>BlueGene/Q,<br>Power BQC 16C 1.6GHz, Custom | USA | 786,432 | 8.59 | 3.95 |
| 6 | Swiss National Supercomputing Centre (CSCS) | Cray | Piz Daint<br>Cray XC30, Xeon E5 8C 2.6GHz,<br>Aries, NVIDIA K20x | Switzer-land | 115,984 | 6.27 | 2.33 |
| 7 | Texas Advanced Computing Center/UT | Dell | Stampede<br>PowerEdge C8220,<br>Xeon E5 8C 2.7GHz, Intel Xeon Phi | USA | 462,462 | 5.17 | 4.51 |
| 8 | Forschungszentrum Juelich (FZJ) | IBM | JuQUEEN<br>BlueGene/Q,<br>Power BQC 16C 1.6GHz, Custom | Germany | 458,752 | 5.01 | 2.30 |
| 9 | Lawrence Livermore National Laboratory | IBM | Vulcan<br>BlueGene/Q,<br>Power BQC 16C 1.6GHz, Custom | USA | 393,216 | 4.29 | 1.97 |
| 10 | Leibniz Rechenzentrum | IBM | SuperMUC<br>iDataPlex DX360M4,<br>Xeon E5 8C 2.7GHz, Infiniband FDR | Germany | 147,456 | 2.90 | 3.52 |

**Rmax is the achieved performance on the Benchmark**

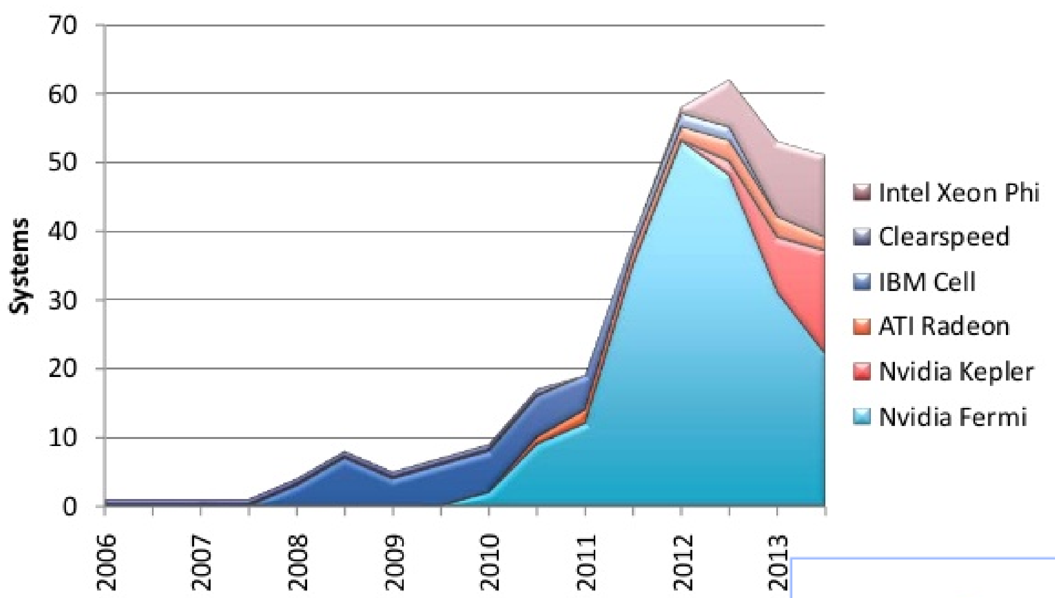
# Power Efficiency



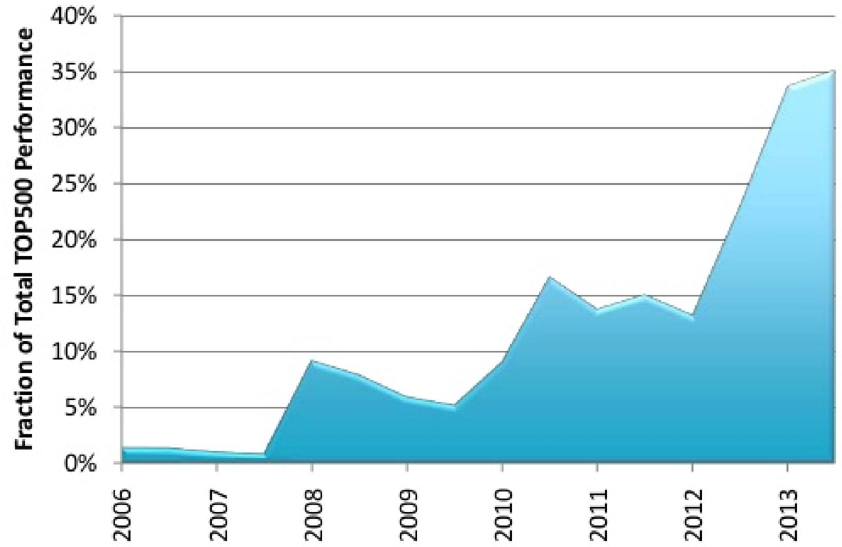
# Processors / Systems



## Accelerators

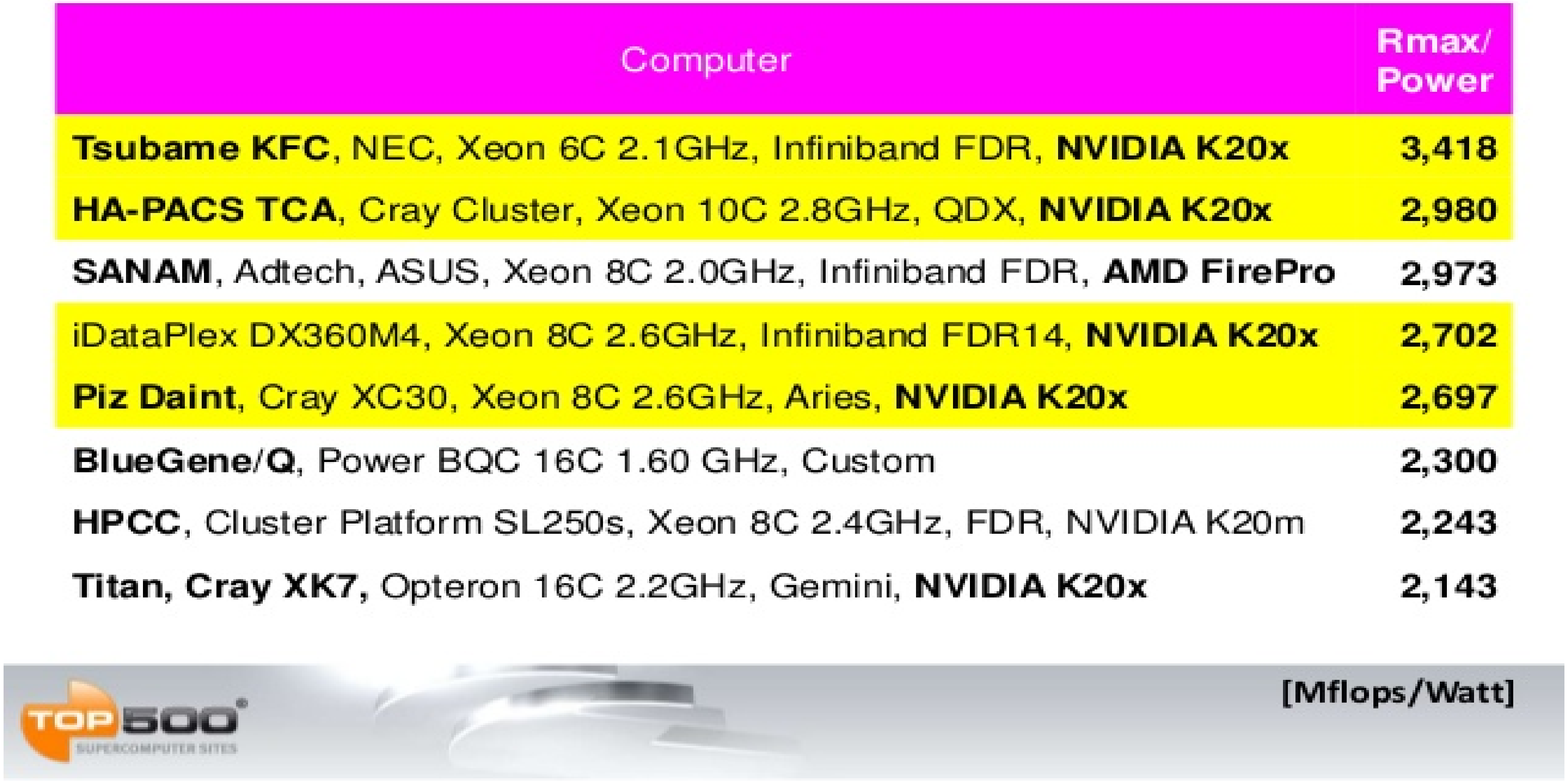


## Performance Share of Accelerators



**Impact of Accelerators**
**On the top 500**

# Most Power Efficient Architectures



| Computer | Rmax/ Power |
|---|---|
| **Tsubame KFC**, NEC, Xeon 6C 2.1GHz, Infiniband FDR, **NVIDIA K20x** | 3,418 |
| **HA-PACS TCA**, Cray Cluster, Xeon 10C 2.8GHz, QDX, **NVIDIA K20x** | 2,980 |
| **SANAM**, Adtech, ASUS, Xeon 8C 2.0GHz, Infiniband FDR, **AMD FirePro** | 2,973 |
| iDataPlex DX360M4, Xeon 8C 2.6GHz, Infiniband FDR14, **NVIDIA K20x** | 2,702 |
| **Piz Daint**, Cray XC30, Xeon 8C 2.6GHz, Aries, **NVIDIA K20x** | 2,697 |
| **BlueGene/Q**, Power BQC 16C 1.60 GHz, Custom | 2,300 |
| **HPCC**, Cluster Platform SL250s, Xeon 8C 2.4GHz, FDR, NVIDIA K20m | 2,243 |
| **Titan, Cray XK7,** Opteron 16C 2.2GHz, Gemini, **NVIDIA K20x** | 2,143 |

**[Mflops/Watt]**

**2.4 petaflops/megawatt** **exascale requires 50 petaflops / megawatt**

**ALL THE EFFICIENT MACHINES ARE ACCELERATOR BASED**

# The Challenge?

**Scalability of frameworks for complex multiscale multiphysics problems on Blue Waters, Sequoia, Titan and future machines?**

**"Exascale programming will require prioritization of critical-path and non-critical path tasks, adaptive directed acyclic graph scheduling of critical-path tasks, and adaptive rebalancing of all tasks……"[Brown et al. Exascale Report ]**

**Today's bulk synchronous (BSP), distributed memory, communicating sequental processes (CSP) based execution model is approaching an efficiency, scalability, and power wall. [Harrod SC12] – suggests….**

- **New Programming Models and DSLs**
- **Dynamic Runtime Systems: adapt to changing application goals and system conditions**

- **Locality-aware and Energy-efficient Strategies**
- **Language Interoperability**

# One solution?

Task-based apps code specifying connectivity to other tasks ( and data required outputs delivered etc)

Abstract treatment of communications via data warehouse  no MPI

Have a runtime system that distributes these tasks, load balances and rebalances these tasks and executes them efficiently on large parallel architectures.

## Example - Uintah Software

2003:code scales on 2K cores  -  2012: code scales on 200K cores

WITHOUT CHANGING A SINGLE LINE OF APPLICATIONS CODE (almost

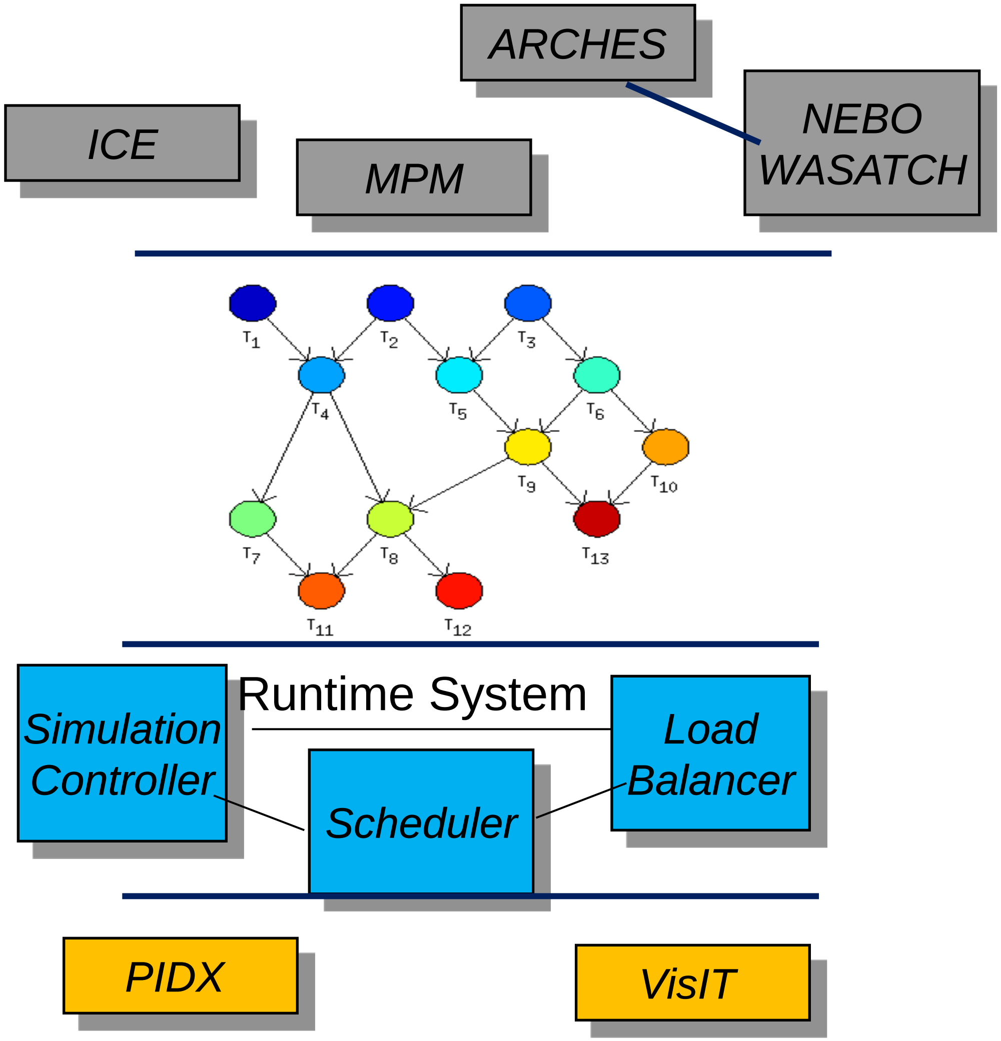
Work on   Titan, Stampede, BG/Q  ongoing  Wasatch DSL

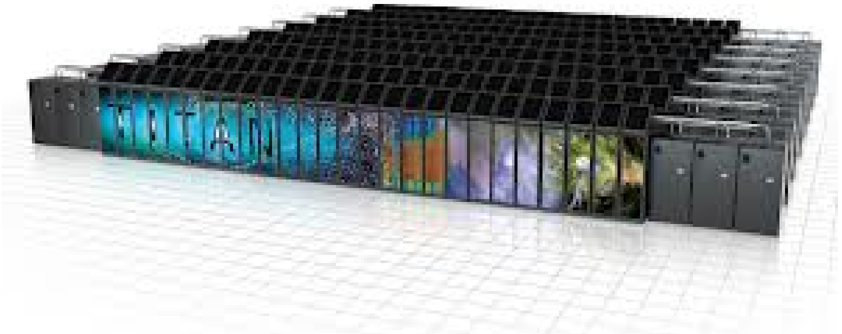
# Uintah Architecture

- **Application Specification** via ICE MPM ARCHES or NEBO/WASATCH DSL

- **Abstract task-graph** program that executes on:

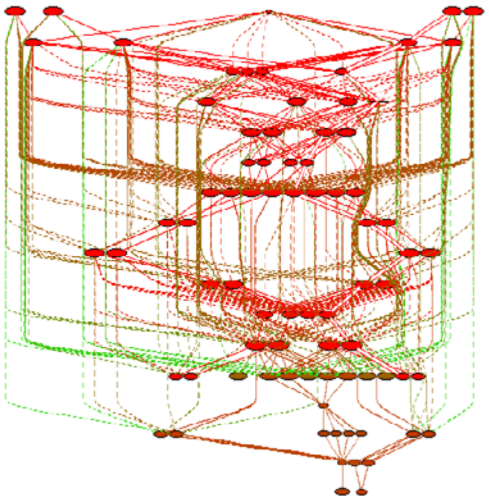
- **Runtime System** with:
- Asynchronous out-of-order execution
- Work stealing
- Overlap communication & computation
- Tasks running on cores and accelerators
- **Scalable I/O via Visus PIDX**
- **Viz using VisIt**





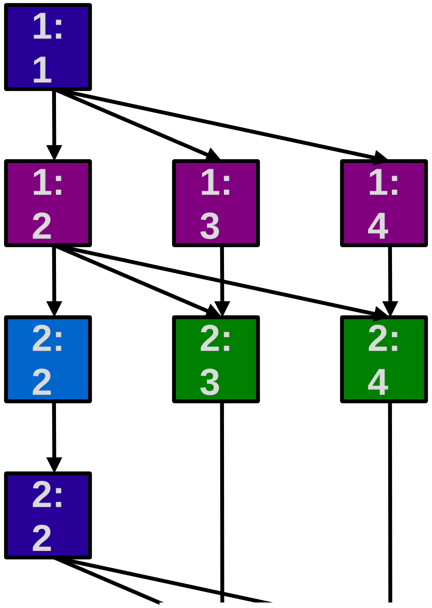
# Task Graph Based Languages/frameworks

**Uintah Taskgraph based PDE Solver**

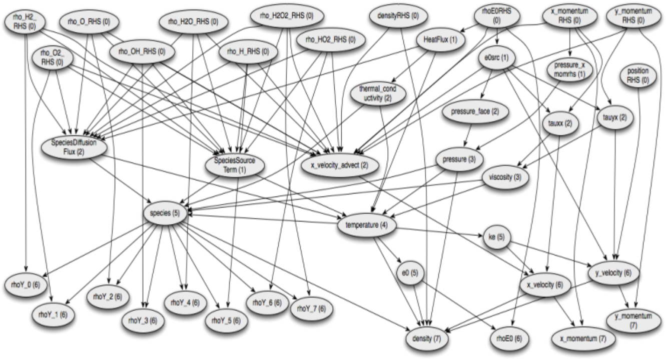


V. Sarkar
L. (S). Kale
S Parker
K. Knobe
J. Dongarra
etc

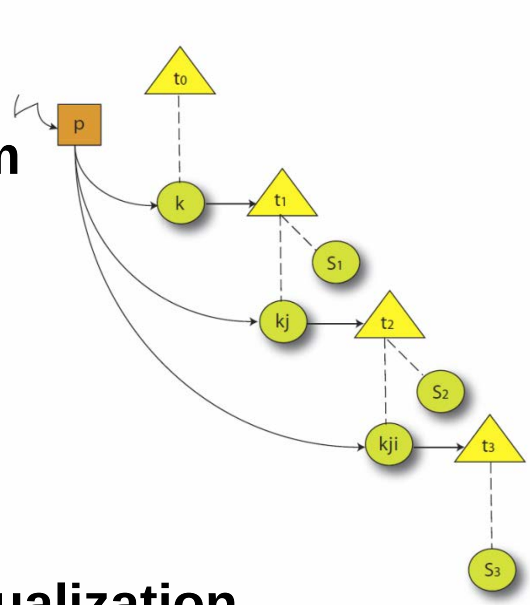
**Plasma (Dongarra): DAG based Parallel linear algebra software**

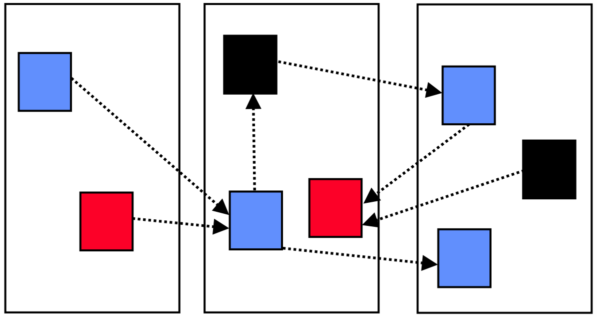


**Wasatch Taskgraph**



**Intel CnC: new language for graph based parallelism**





**Charm++: Object-based Virtualization**

# Why does Dynamic Execution of Directed Acyclic Graphs Work Well?

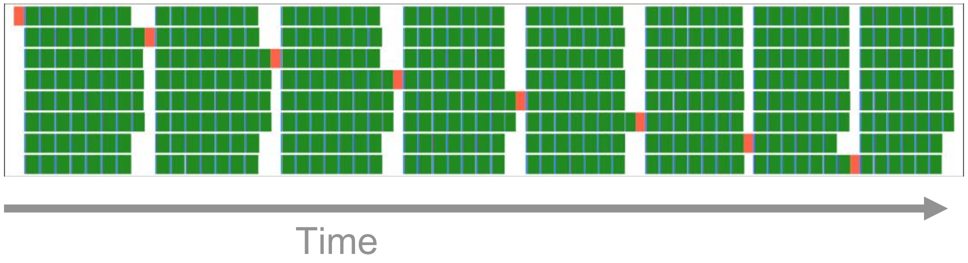
- **Eliminate spurious synchronizations points e.g.**

- **Have multiple task-graphs per multicore (+ gpu) node – provides excess parallelism - slackness**

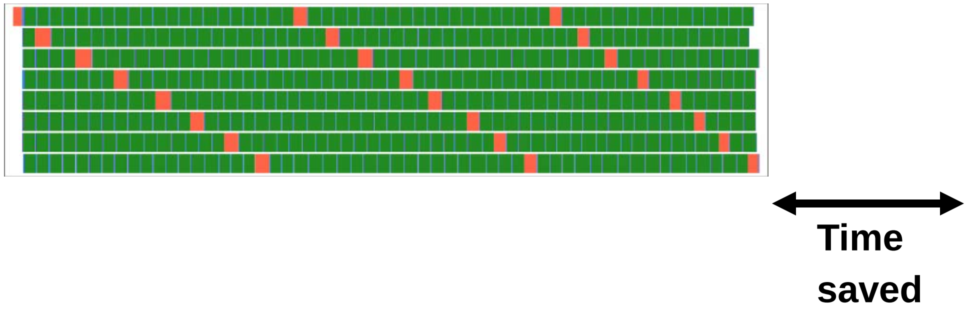
- **Overlap communication with computation by executing tasks as they become available – avoid waiting (use out-of order execution).**

- **Load balance complex workloads by having a sufficiently rich mix of tasks per multicore node that load balancing is done per node**

**Fork-Join – parallel BLAS (Dongarra)**



**DAG-based – dynamic scheduling**



**DATA FLOW APPROACH - SPECIFY ORDER OF EXECUTION ONLY**

# DARPA Exascale Software Study

- DARPA public report by (Vivek Sarkar et al.)
- Silver model for exascale software which must:
  - Have abstraction for high degree of concurrency for directed dynamic graph structured calculations.
  - Enable latency hiding by overlapping computation and communications
  - Minimize synchronization and other overheads
  - Support adaptive resource scheduling
  - Unified approach to heterogeneous procesing
- Silver model is a graph-based asynchronous-task work queue model.
- Some instances of this type of approach in use now. CnC, Charm++, Plasma, Uintah Very disruptive technology - forces us to rethink programming model

# Specific Programming Challenges

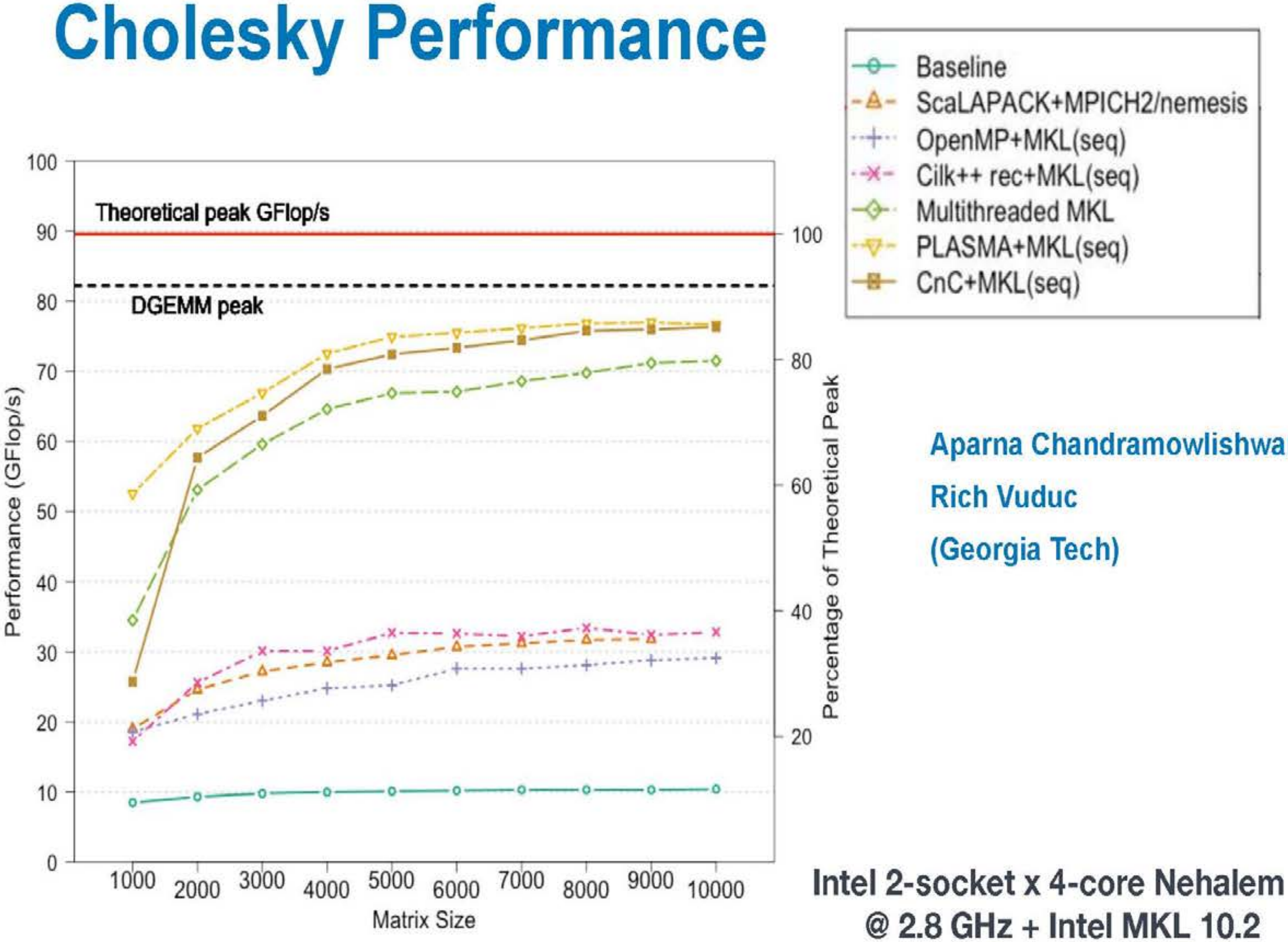
- Explicit management of resources
  - This data on that processor+this work on that processor
- Analogy: memory management
  - We declare arrays, and malloc dynamic memory chunks
  - Do not specify memory addresses
- As usual, indirection is the key
  - Programmer:
    - This data, partitioned into these pieces
    - This work divided that way
  - System: automatic mapping of data and work to processors

**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**

# Concurrent Collections CnC

- A new language for expressing graph based parallelism [Knobe] Intel
- Separates out specification of task-graph from its execution.
- Combines ideas from tuple-space (Linda) streaming and data flow languages.
- Implemented by HP, Intel Rice GaTech on distributed and shared memory
- Static/dynamic distribution scheduling

# Cholesky Performance



Aparna Chandramowlishwaran

Rich Vuduc

(Georgia Tech)

Intel 2-socket x 4-core Nehalem
@ 2.8 GHz + Intel MKL 10.2

*Intel and the Intel logo are registered trademarks of Intel Corporation. Other brands and names are the property of their respective owners

# Parallel Scalability Metrics

ISOEFFICIENCY: How fast does the problem size have to grow as the number of processors grows to maintain constant efficiency.

ISOTIME: How does the number of processors and/or problem size have to change to deliver a solution in constant time.

ISOMEMORY: How does the memory usage change with problem size and processor numbers

Weak scalability: constant time with constant load per/core- needs isomemory and isoefficiency and isotime

Strong scalability: fixed problem size time reduced according to number of cores – needs all of above and very low overheads!

Data structures and algorithms cannot depend on P – the number of processors- everything must be local and linear wrt processors

# Uintah Parallel Computing Framework

- **Uintah uses NSF ( Ranger Kraken) DOE parallel computers, typical run – 2K to 98K cores $10^7$ cells, $10^7$ particles**

- Uintah [1998-2005] - **far-sighted design by Steve Parker**:
  **Solution of broad class of fluid-structure interaction problems**
  **Patch-based AMR using particles and mesh-based fluid solver**

  **Automated task-graph generation for scheduling parallelism**
  **Automated load balancing**
  **Asynchronous communication**
  **User only writes “serial” code for a hexahedral patch**

- Uintah has “legacy” code aspects –original design sound
- MANY COMPONENTS OF THE CODE HAVE BEEN REWRITTEN

**How do we apply Uintah to model Developing Detonations?**
**How do we start to think about scaling to petascale and beyond?**

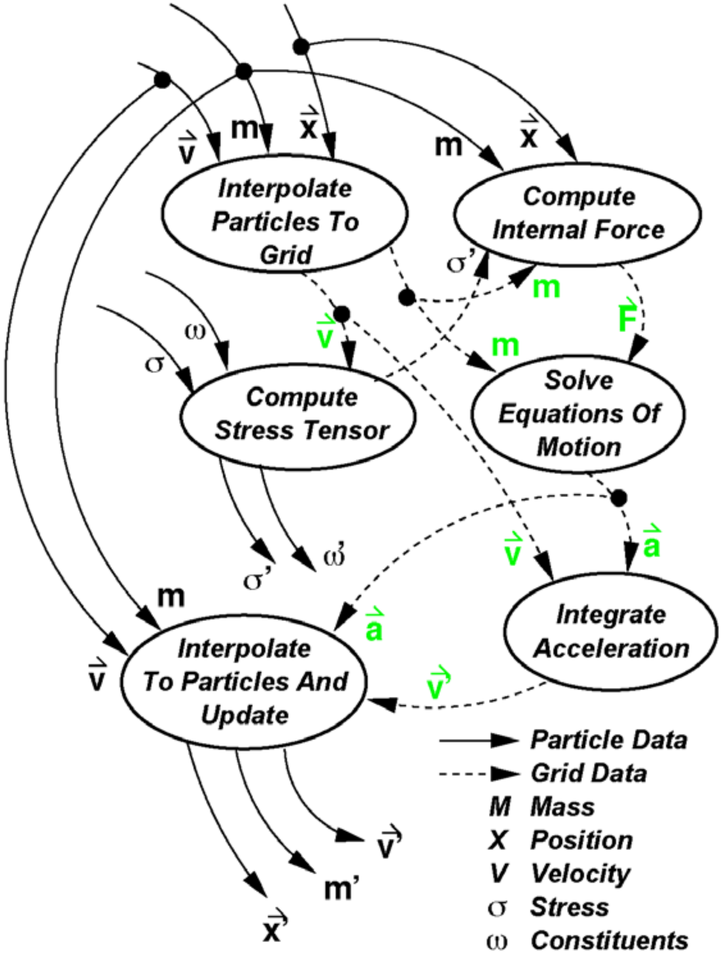
# Directed Acyclic Graphs

- Each task defines its computation with required inputs and outputs

- Uintah uses this information to create a task graph of computation (nodes) + communication (along edges)

- Similar to Charm++ TBlas, CnC DAG approach increasingly popular for efficient parallelism with irregular communications

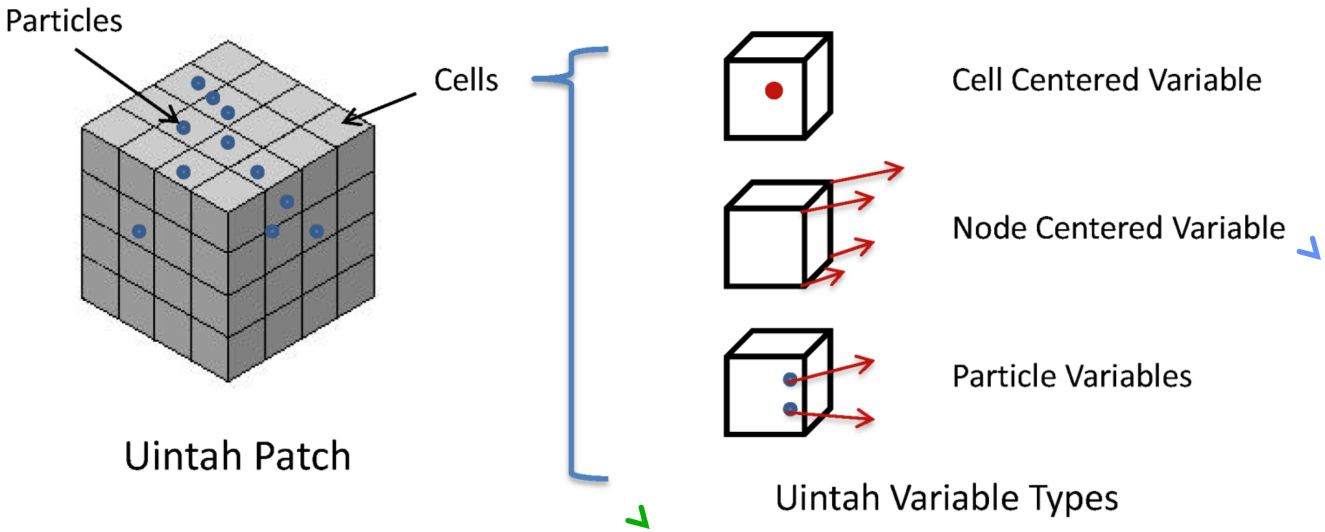
- Slow static execution replaced by asynchronous and out-of-order execution by keeping MULTIPLE VERSIONS of TASK INPUTS



# Uintah Methods Patch and Variables

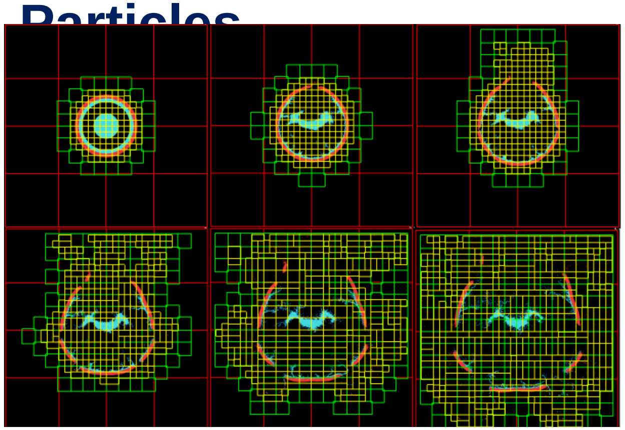
**ICE is a cell-centered finite volume method for Navier Stokes equations**

**ARCHES is a combustion code using several different radiation models and linear solvers**



- **Structured Grid Variable (for Flows) are Cell Centered Nodes, Face Centered Nodes.**
- **Unstructured Points (for Solids) are Particles**

**MPM is a novel method that uses particles and nodes**

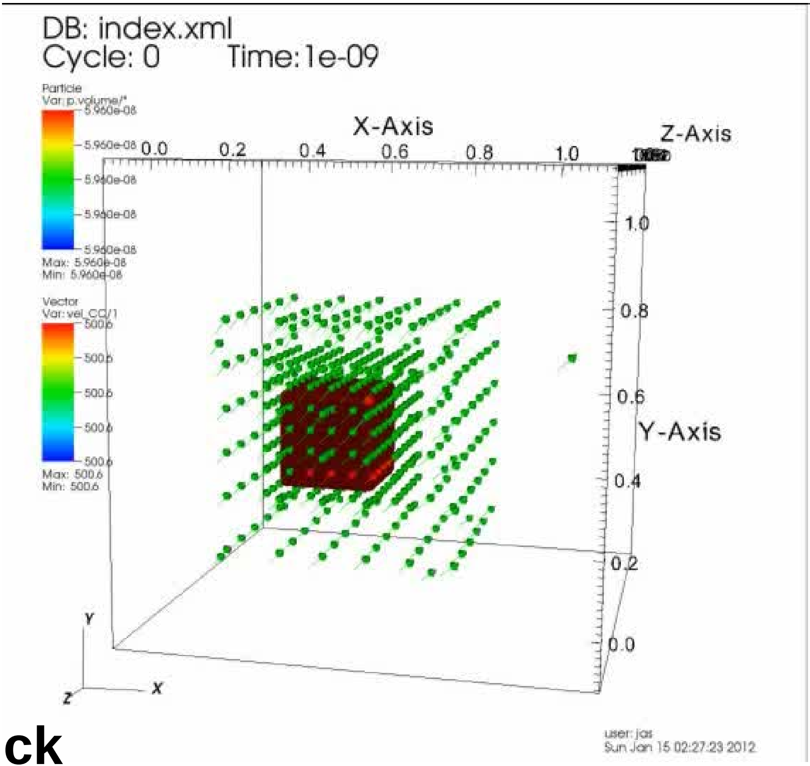


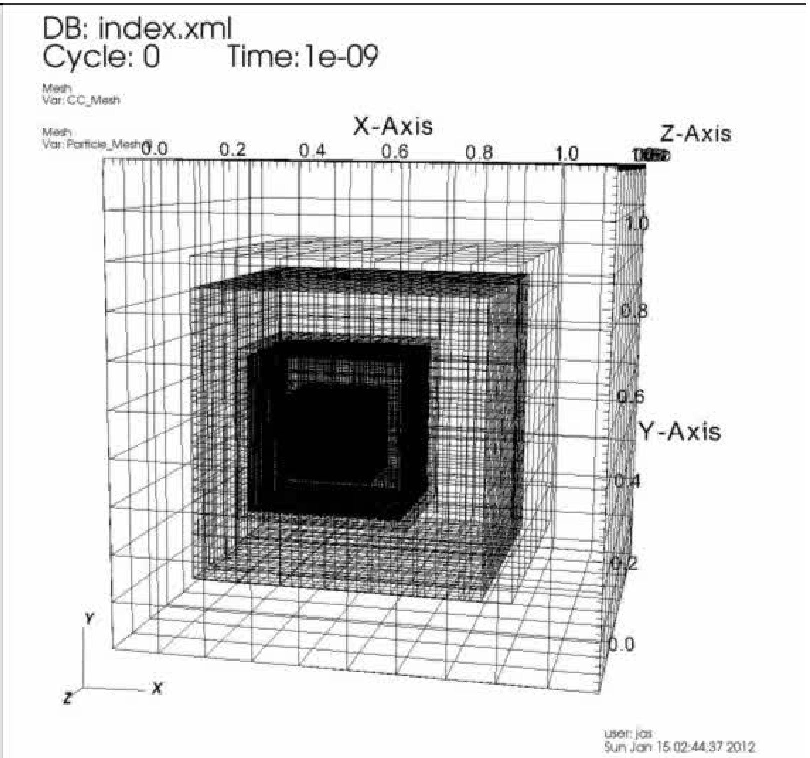
- Structured Grid + Unstructured Points
- Patch-based Domain Decomposition
- Adaptive Mesh Refinement

- Dynamic Load Balancing
  - Profiling + Forecasting Model
  - Parallel Space Filling Curves

# Fluid Structure Interaction Example:
# AMR MPMICE

**A PBX explosive flow pushing a piece of its metal container**





**Click**

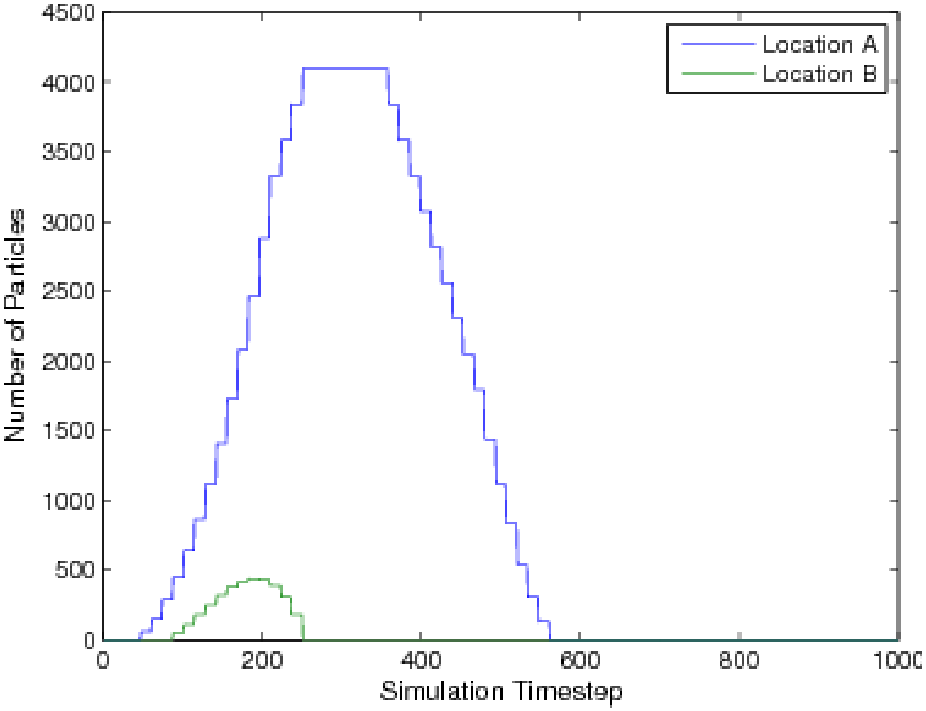
**Flow velocity and particle volume** **Computational grids and particles**

**Grid Variables: Fixed number per patch, relative easy to balance**

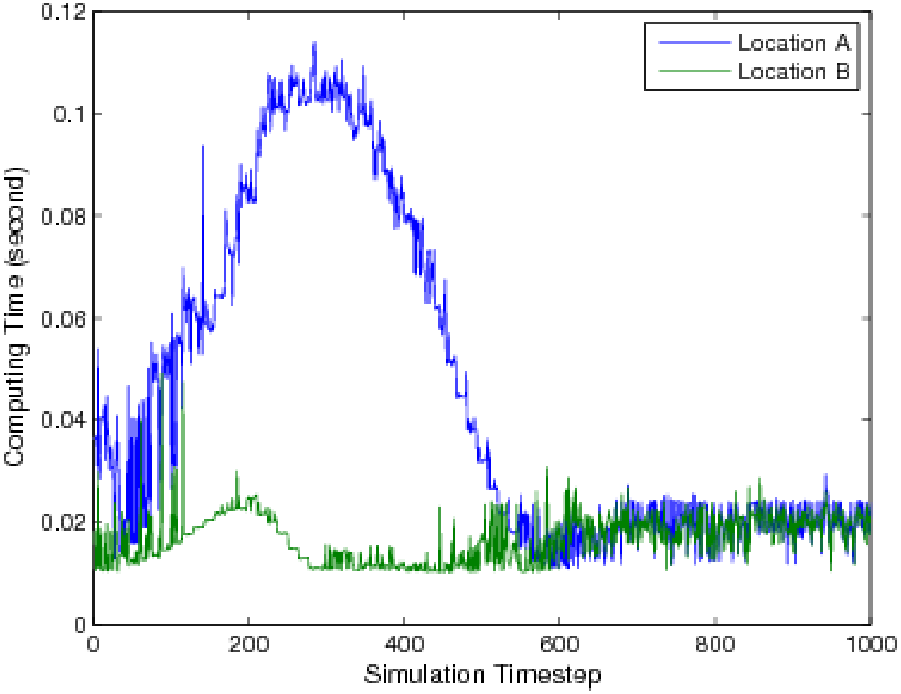
**Particle Variables: Variable number per patch, hard to load balance**

# THE PARTICLES AND AMR CAUSE SIGNIFICANT AND UNPREDICTABLE LOAD IMBALANCES

**Particle number at two locations vs time**

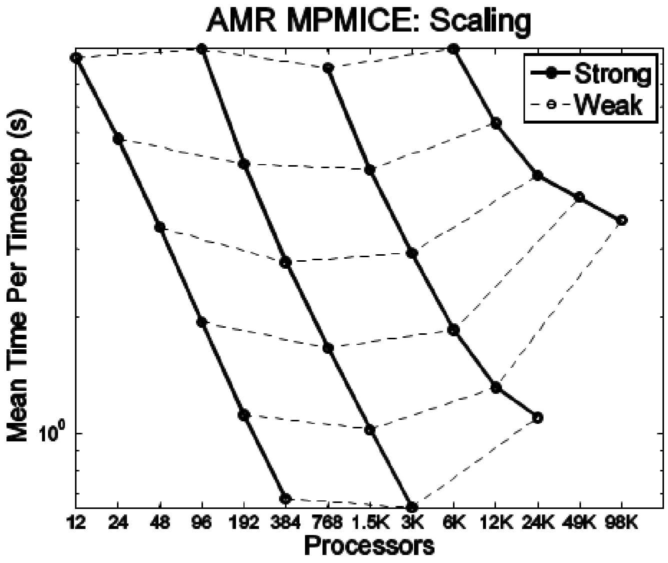


**Time per patch at two locations vs time**

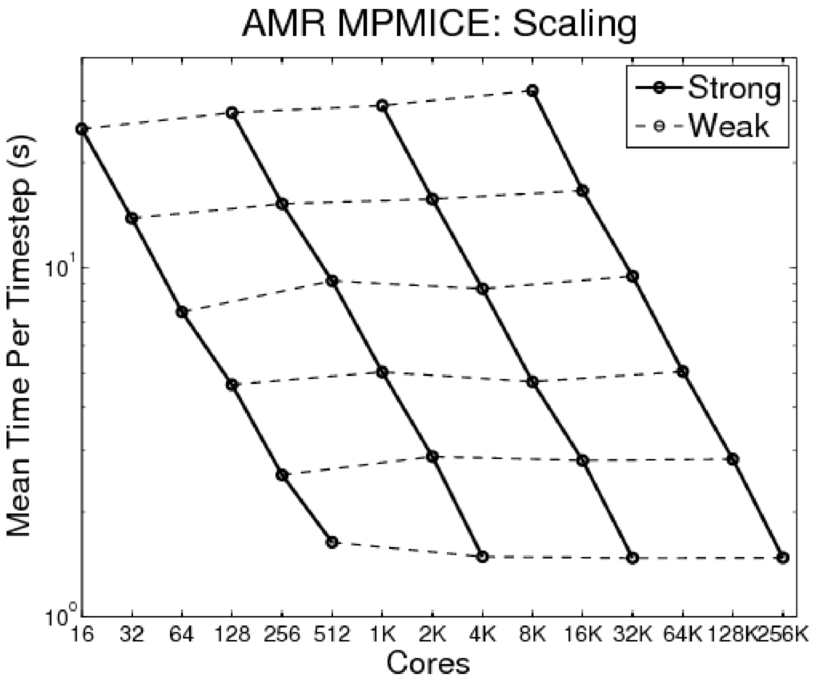


OLD

# Scalability on Titan CPUs



Scaling Breakdown



- Poor scalability up to 98K cores (Kraken, NICS)
- Issues:
  - Out of memory with 98K cores
  - AMR MPMICE scaling, Load Imbalance
- Solution: New runtime system with Hybrid thread/MPI

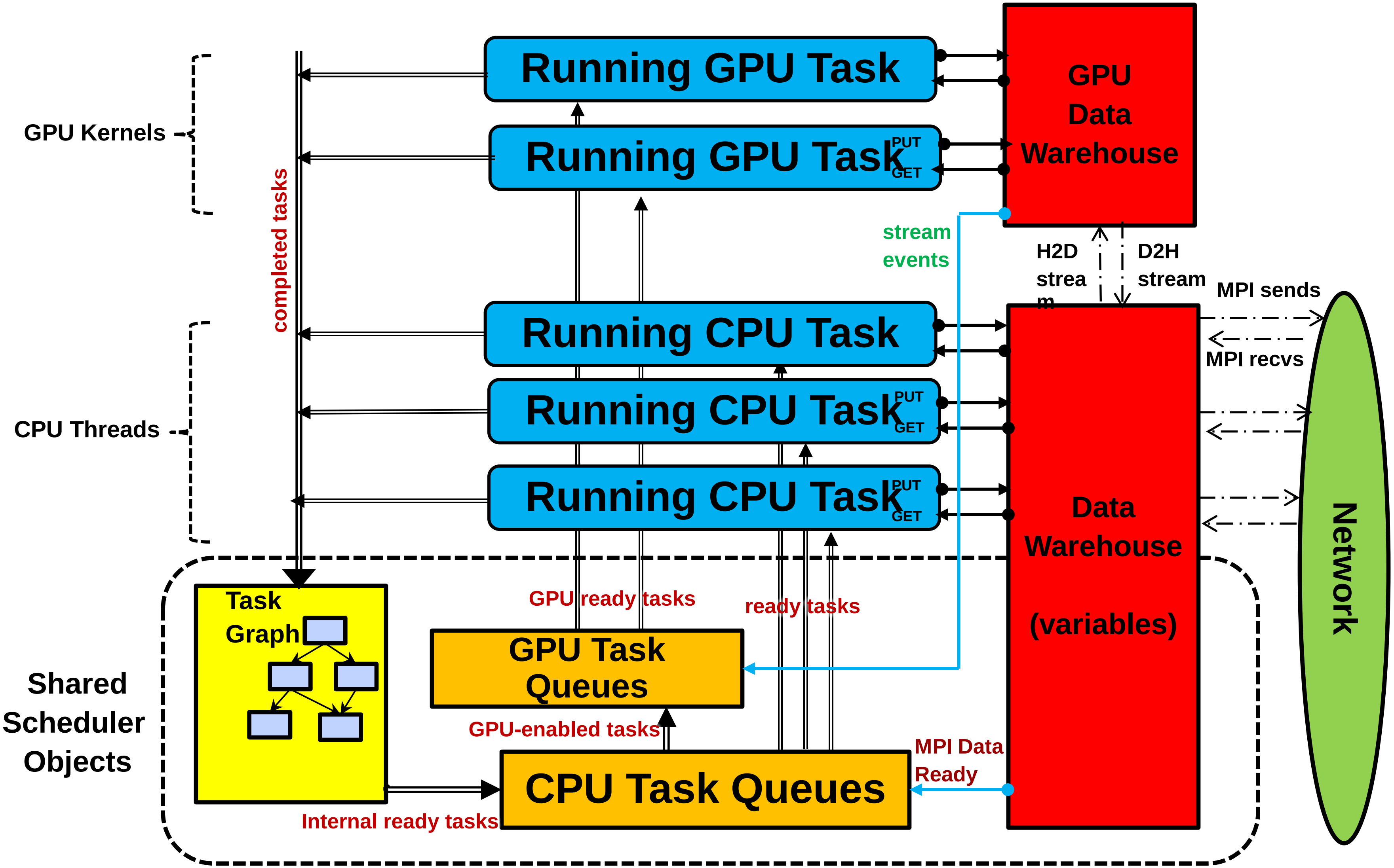
**One flow with particles moving**

**3-level AMR MPM ICE 70% efficiency**

**At 256K cores vs 16K cores**

**Distributed Controller**

# Unified Heterogeneous Scheduler & Runtime



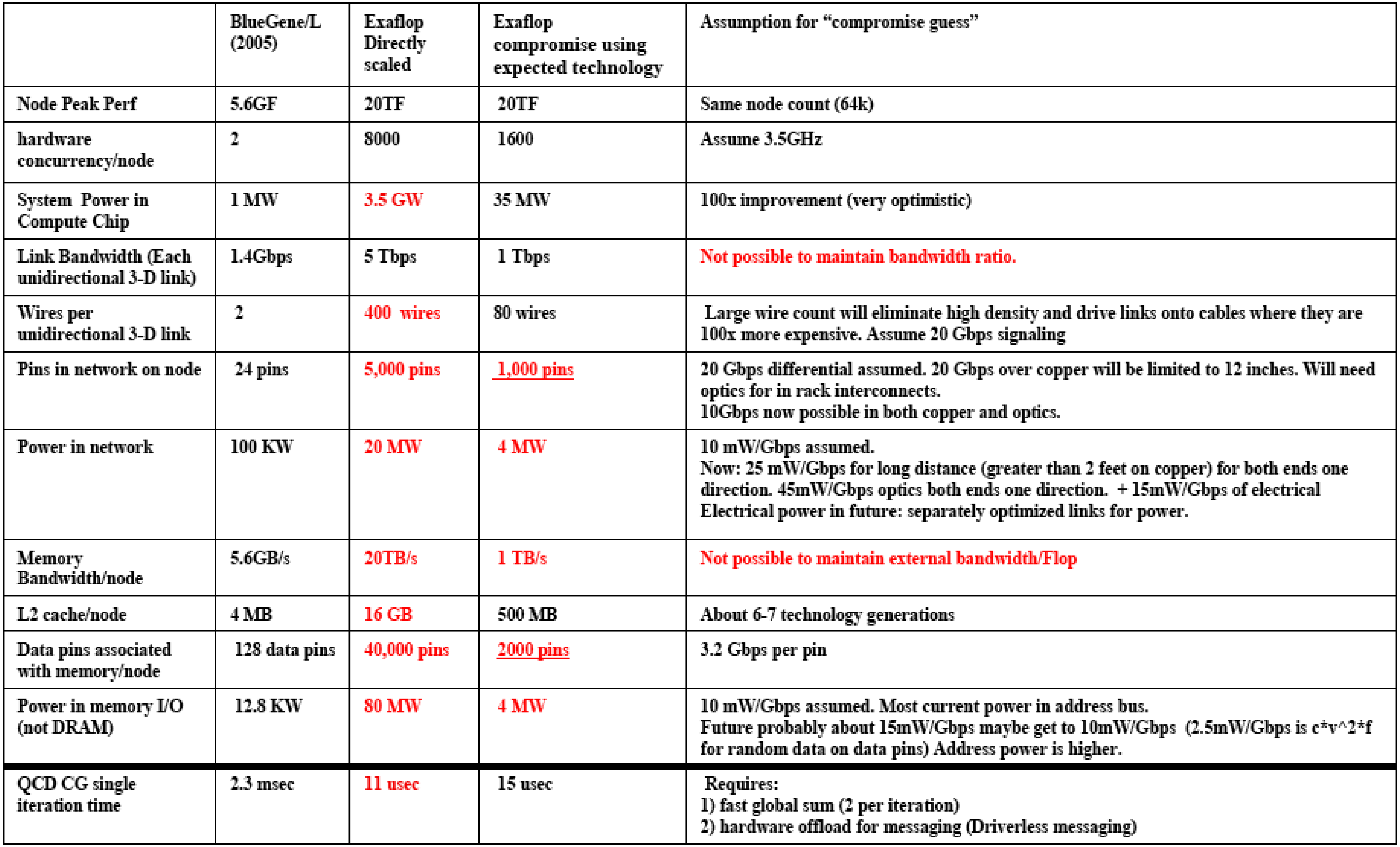
**One MPI process and warehouse per multi-core/gpu node – 10% of memory**

# DARPA Exascale Hardware Study

- **DARPA public report (Peter Kogge et al.)**
- **Describes Challenges in going to Exascale at national level and petascale at University level.**
- **Exascale machine Aggressive Strawman:**
  - **742 cores per socket, 12 sockets per node, 32 nodes per rack**
  - **166,113,024 cores, 223,872 sockets**
  - **4 flops per cycle per core @1.5Ghz, 1.029 PFlops**
  - **Power 67MW! DoE aims for just 25MW**
- **Novel technologies considered e.g. t**
  - **On chip optics (ongoing e.g. HP)**
  - **Phase change or Holographic memory**
- **Extraordinary concurrency is the only game in town**
- **Power, fault tolerance, programmability are key**

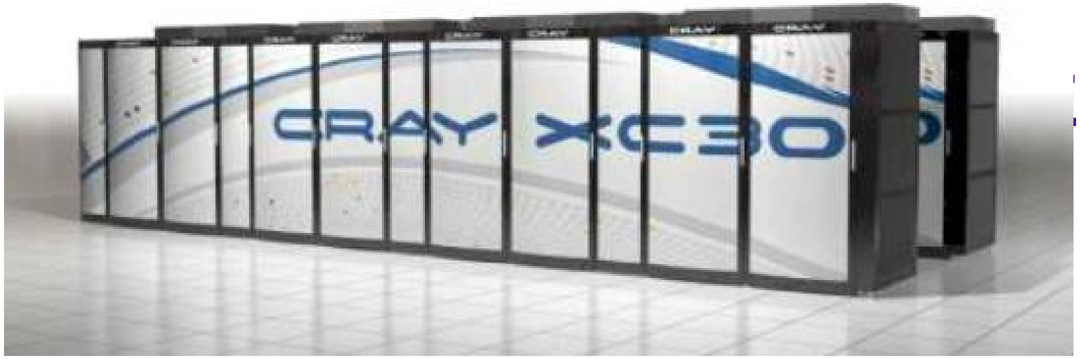
**IMPLICATION IS PETASCALE AT LOCAL LEVEL – terascale laptops!**

# Extrapolating to Exaflop/s in 2018?!?!

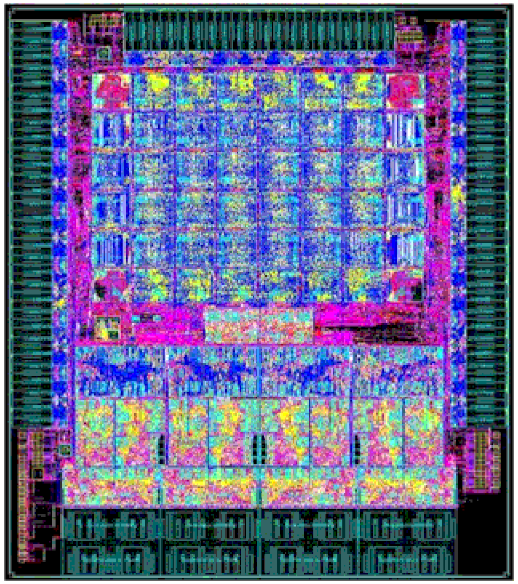


| | BlueGene/L (2005) | Exaflop Directly scaled | Exaflop compromise using expected technology | Assumption for "compromise guess" |
|---|---|---|---|---|
| Node Peak Perf | 5.6GF | 20TF | 20TF | Same node count (64k) |
| hardware concurrency/node | 2 | 8000 | 1600 | Assume 3.5GHz |
| System Power in Compute Chip | 1 MW | 3.5 GW | 35 MW | 100x improvement (very optimistic) |
| Link Bandwidth (Each unidirectional 3-D link) | 1.4Gbps | 5 Tbps | 1 Tbps | Not possible to maintain bandwidth ratio. |
| Wires per unidirectional 3-D link | 2 | 400 wires | 80 wires | Large wire count will eliminate high density and drive links onto cables where they are 100x more expensive. Assume 20 Gbps signaling |
| Pins in network on node | 24 pins | 5,000 pins | 1,000 pins | 20 Gbps differential assumed. 20 Gbps over copper will be limited to 12 inches. Will need optics for in rack interconnects.<br>10Gbps now possible in both copper and optics. |
| Power in network | 100 KW | 20 MW | 4 MW | 10 mW/Gbps assumed.<br>Now: 25 mW/Gbps for long distance (greater than 2 feet on copper) for both ends one direction. 45mW/Gbps optics both ends one direction. + 15mW/Gbps of electrical<br>Electrical power in future: separately optimized links for power. |
| Memory Bandwidth/node | 5.6GB/s | 20TB/s | 1 TB/s | Not possible to maintain external bandwidth/Flop |
| L2 cache/node | 4 MB | 16 GB | 500 MB | About 6-7 technology generations |
| Data pins associated with memory/node | 128 data pins | 40,000 pins | 2000 pins | 3.2 Gbps per pin |
| Power in memory I/O (not DRAM) | 12.8 KW | 80 MW | 4 MW | 10 mW/Gbps assumed. Most current power in address bus.<br>Future probably about 15mW/Gbps maybe get to 10mW/Gbps (2.5mW/Gbps is c*v^2*f for random data on data pins) Address power is higher. |
| QCD CG single iteration time | 2.3 msec | 11 usec | 15 usec | Requires:<br>1) fast global sum (2 per iteration)<br>2) hardware offload for messaging (Driverless messaging) |

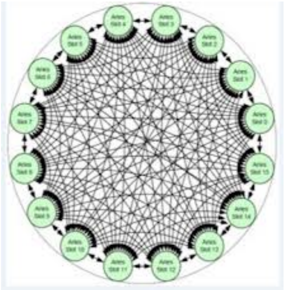
Source: David Turek, IBM



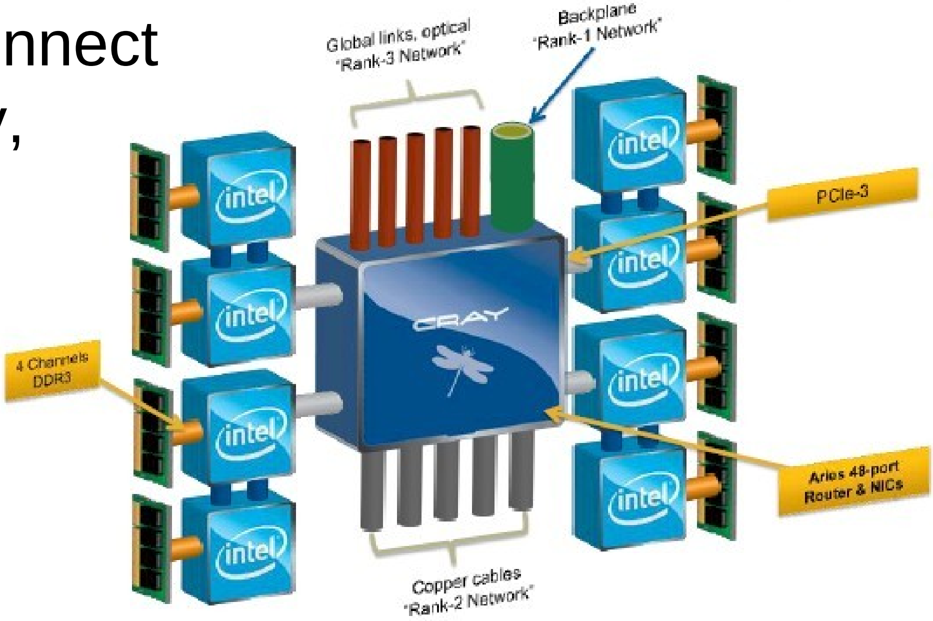
# – Next generation



- Sustained application performance codes: 236 Tflop/s  Aggregate memory: 333 TB
- 5,200 computes nodes with 64 GB memory per node
- Cray Aries high-speed interconnect (0.25 µs to 3.7 µs MPI latency, ~8GB/sec MPI bandwidth)
- Prototype 100pF machine



**Dragonfly Network**



# *Sketch of Nvidia Echelon research system*

**Echelon design incorporates a large number (~1024) of stream cores and a smaller (~8) number of latency-optimized CPU-like cores on a single chip, sharing a common memory system.**

**Eight stream cores will form a streaming multiprocessor (SM) and 128 of SMs will forum the large pool of throughput-optimized processing elements.**

**Such a chip could deliver 20 TeraFLOPS with double precision and a number of them will form a 2.6 PetaFLOPS rack. At present Nvidia Fermi (GF110) chip 512 with stream processors operating at 1544MHz can deliver 0.79TFLOPS of DP compute performance.**

**Considerint the 25 times difference in performance, it is highly likely that the Echelon will employ post-Maxwell (~2013 ~ 2014) Nvidia GPU design.**

**In order to keep power consumption of such a chip relatively low, stream processors have to process a double-precision floating point operation using just 10 picojoules of power, down from 200 picojoules on Nvidia's current Fermi chips,**

**Current AMD INTEL processors use 200 nanojoules per flop  a thousand times as much**

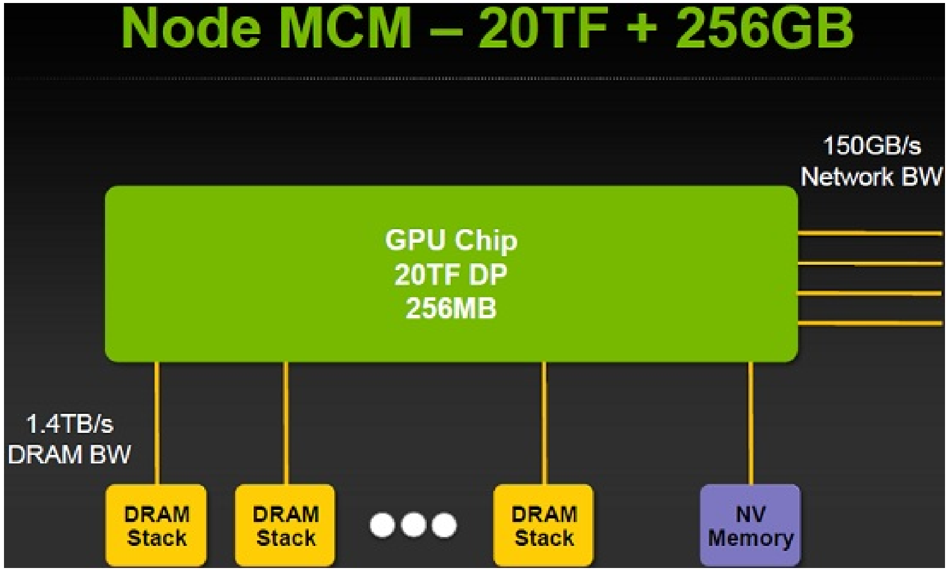
# NVIDIAs Exascale Vision

Node

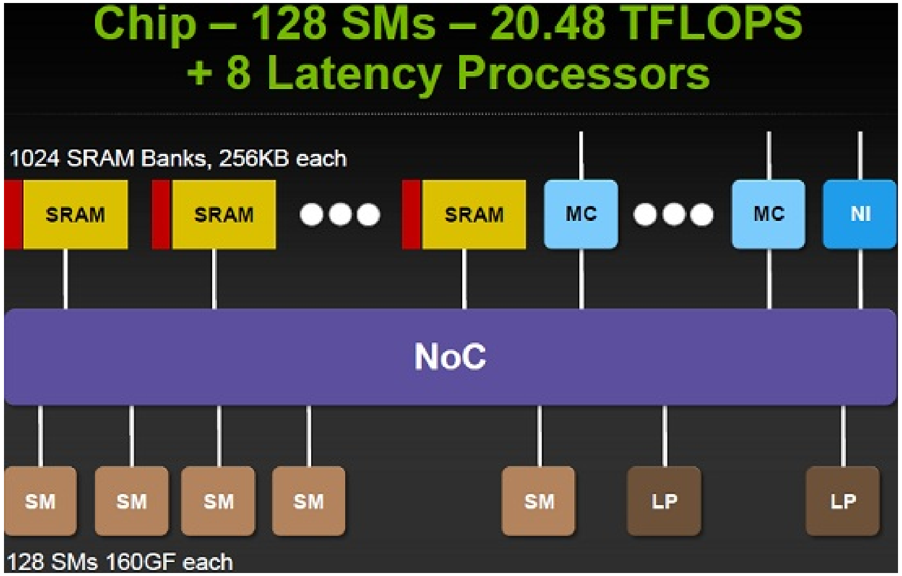
Petaflop

Chip

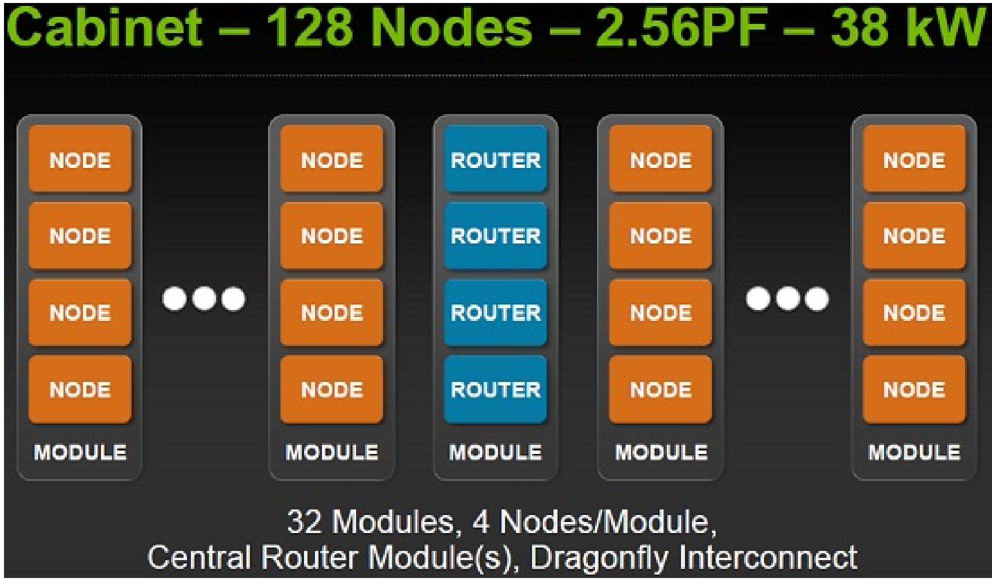
Cabinet



Node MCM – 20TF + 256GB



Chip – 128 SMs – 20.48 TFLOPS + 8 Latency Processors



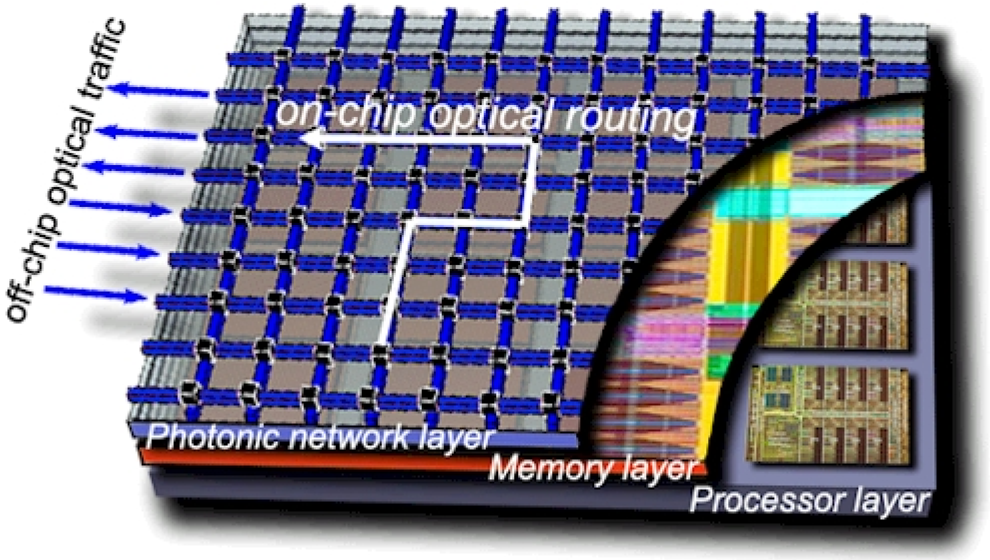
Cabinet – 128 Nodes – 2.56PF – 38 kW

32 Modules, 4 Nodes/Module, Central Router Module(s), Dragonfly Interconnect

## Examples of the technology to be used

- (i) Stacked chips
- (ii) On chip optical routing
- (iii) Very large numbers of cores per chip
- (iv) Extra memory for fault tolerance etc

IBM Stacked Chip



We do not know what exascale machines will look  like.
China’s Tianhe 2 is  an interesting addition

# Summary

- Petascale computing is here
- Rapid developments with GPUs
- Much new technology being developed
- New architecture and software models needed for 100M cores
- This is a great time to work in HPC!