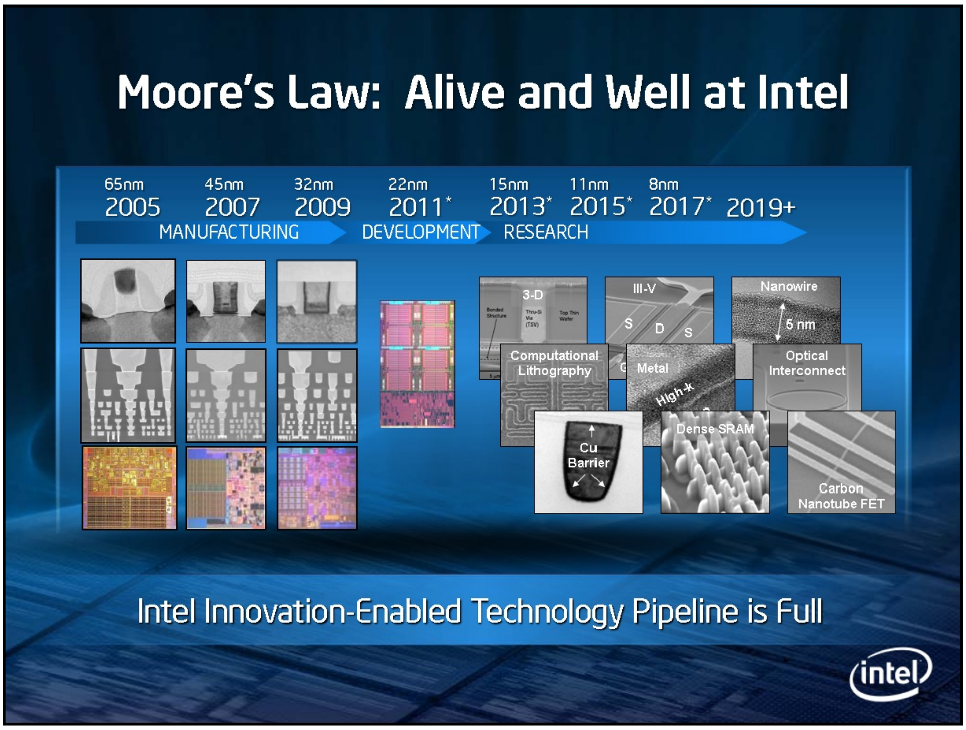
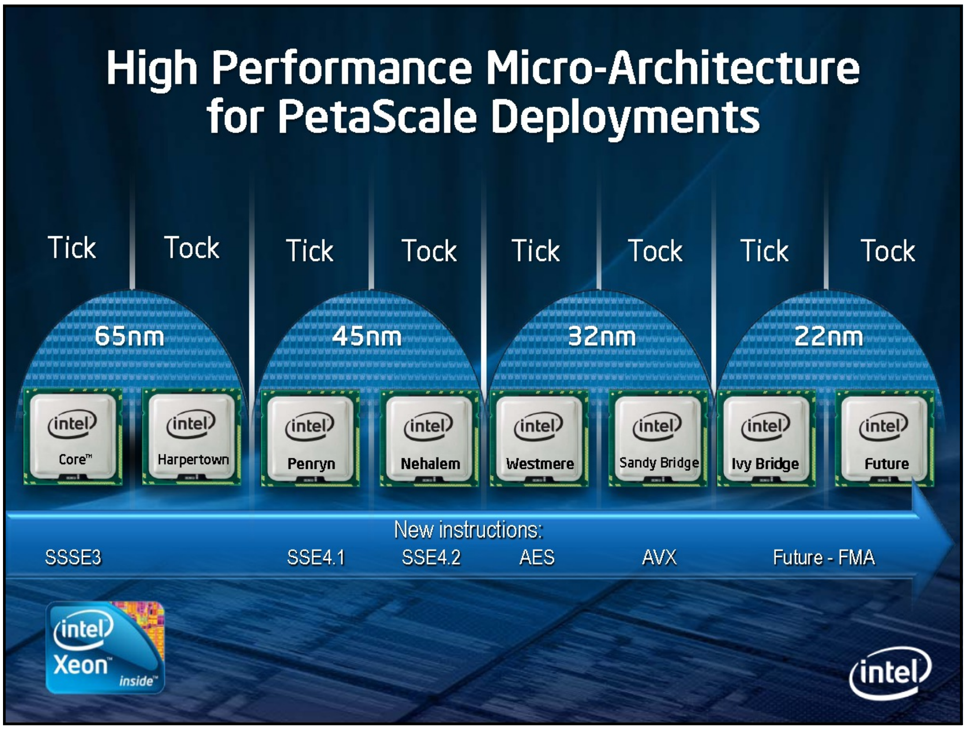

Moore's Law: Alive and Well at Intel
65nm 2005
45nm 2007
32nm 2009
22nm 2011*
15nm 2013*
11nm 2015*
8nm 2017*
2019+
MANUFACTURING
DEVELOPMENT
RESEARCH
3-D
III-V
S D S
Nanowire
5 nm
Computational Lithography
Metal
High-k
Optical Interconnect
Cu Barrier
Dense SRAM
Carbon Nanotube FET
Intel Innovation-Enabled Technology Pipeline is Full
intel

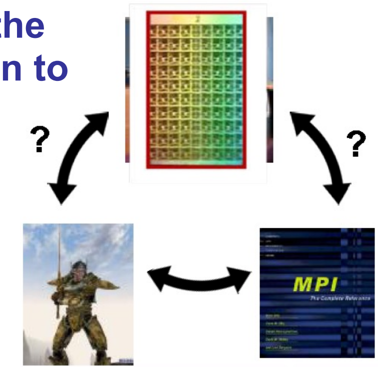
23


High Performance Micro-Architecture for PetaScale Deployments
Tick Tock Tick Tock Tick Tock Tick Tock
65nm
45nm
32nm
22nm
Core™
Harpertown
Penryn
Nehalem
Westmere
Sandy Bridge
Ivy Bridge
Future
New instructions:
SSSE3
SSE4.1
SSE4.2
AES
AVX
Future - FMA
intel Xeon™ inside™
intel

24

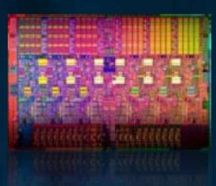
Impact on Software

- **We will need to rethink and redesign our software**
  - **Similar challenge as the 1990 to 1995 transition to clusters and MPI**



25

IA Programming Flexibility



Software

Instruction Parallelism	Data Parallelism	Thread Parallelism	Cluster / Process Parallelism

Serial Code Node Level
Fast Scalar performance, Optimized C/C++,FORTRAN, Threading and Performance Libraries, Debug / Analysis Tools

Parallel Node Level
Multi-core, Multi-Socket, SSE and AVX instructions, OpenMP, Threading Building Blocks, Performance Libraries, Thread Checker , Ct , Cilk

Multi-Node / Cluster Level
Cluster Tools, MPI Checker

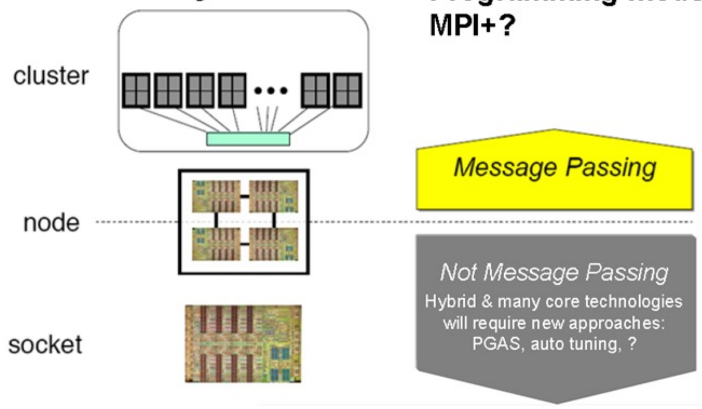
Programming choices and standards for range of parallel efficiency

26

A Likely Future Scenario

System: cluster + many core node

Programming model: MPI+?



after Don Grice, IBM, Roadrunner Presentation, ISC 2008

27

Why MPI will persist

- Obviously MPI will not disappear in five years
- Today we have more than 25 years of legacy software in MPI
- New systems are not sufficiently different to lead to a radical new programming model

28

What will be the "?" in MPI+?

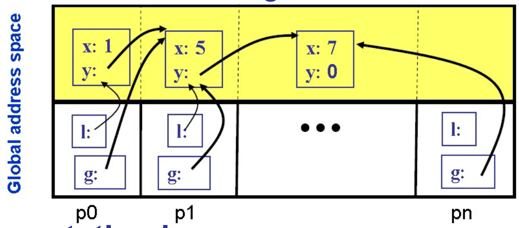
- Likely candidates are
  - PGAS languages
  - Autotuning
  - CUDA, OpenCL, OpenACC
  - A wildcard from the commercial space

What's Wrong with MPI Everywhere?

- One MPI process per core is wasteful of intra-chip latency and bandwidth
- **Weak scaling**: success model for the "cluster era" not enough memory per core
- **Heterogeneity**: MPI per CUDA thread-block?

PGAS Languages

- **Global address space**: thread may directly read/write remote data
- **Partitioned**: data is designated as local or global

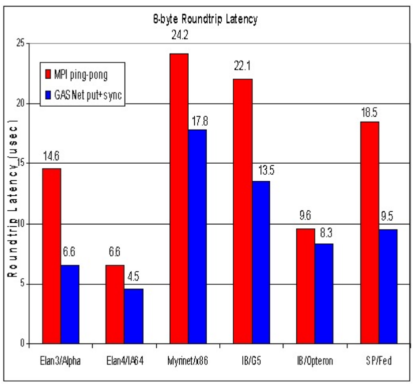


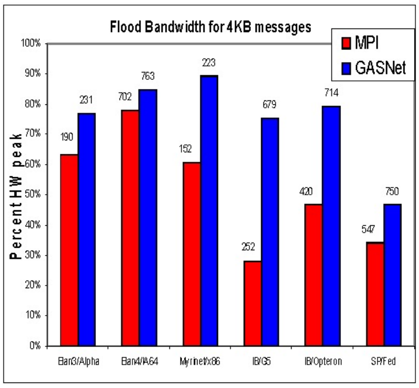
- **Implementation issues:**
  - Distributed memory: Reading a remote array or structure is explicit, not a cache fill
  - Shared memory: Caches are allowed, but not required
- No less scalable than MPI!
- Permits sharing, whereas MPI rules it out!

31

Performance Advantage of One-Sided Communication

- The put/get operations in PGAS languages (remote read/write) are one-sided (no required interaction from remote proc)
- This is faster for pure data transfers than two-sided send/receive



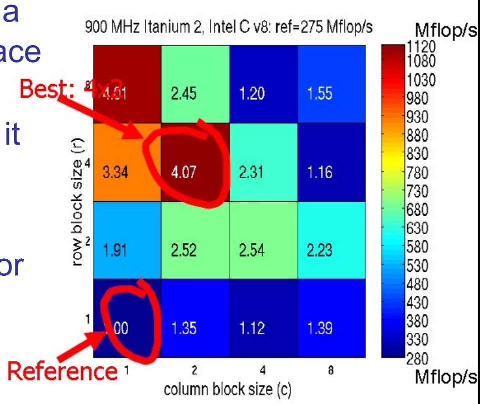


32

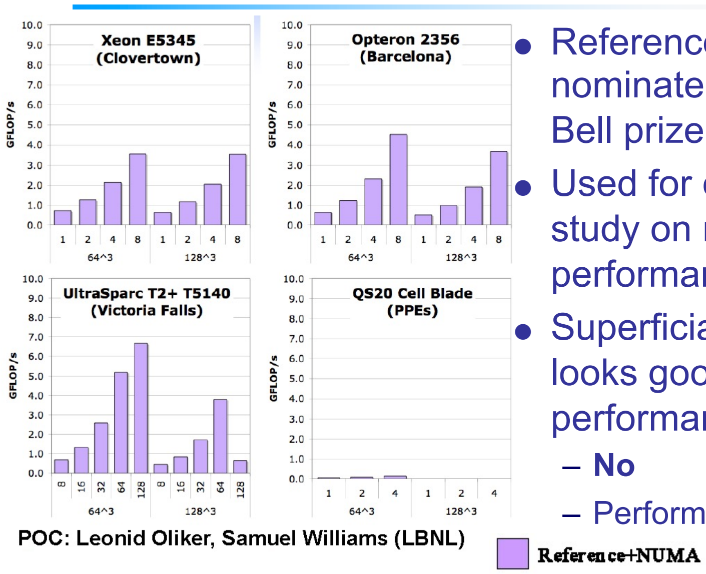
Autotuning

- Write programs that write programs
  - Automate search across a complex optimization space
  - Generate space of implementations, search it
  - Performance far beyond current compilers
  - Performance portability for diverse architectures!
  - Past success stories: PhiPAC, ATLAS, FFTW, Spiral, OSKI

Finite Element Problem
[Im, Yelick, Vuduc, 2005]



Performance

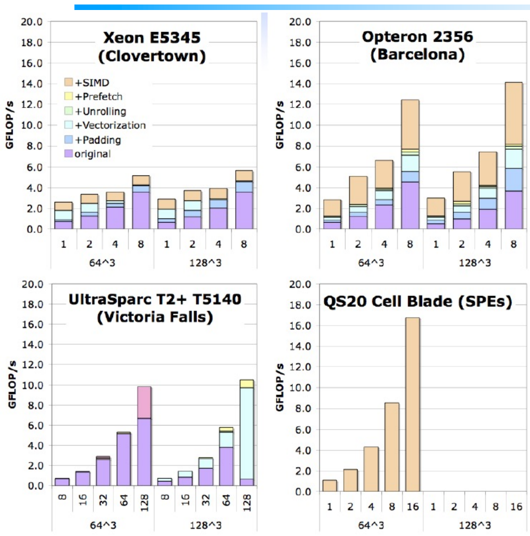


- Reference code was nominated for a Gordon Bell prize
- Used for out-of-box study on multicore performance
- Superficially, scalability looks good, but is performance good?
  - **No**
  - Performance model

POC: Leonid Oliker, Samuel Williams (LBNL)

35

Autotuning



- Auto-tuning dramatically improved performance
- Clearly no silver bullet
- SIMD partially breaks portability premise
- Note: Cell version was optimized, not auto-tuned

POC: Leonid Oliker, Samuel Williams (LBNL)

35

36

Autotuning

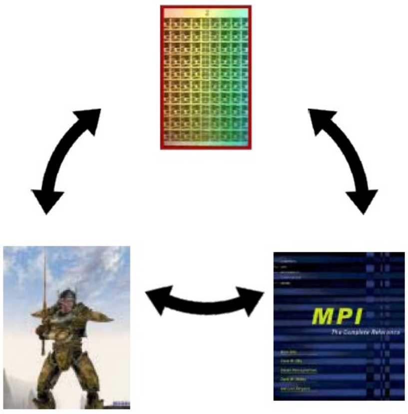
- Auto-tuning dramatically improved performance
- Clearly no silver bullet
- SIMD partially breaks portability premise
- Note: Cell version was optimized, not auto-tuned

POC: Leonid Oliker, Samuel Williams (LBNL)

36

The Likely HPC Ecosystem from 202x+

CPU + GPU = future many-core driven by commercial applications



MPI+(autotuning, PGAS, ??)

Next generation "clusters" with many-core or hybrid nodes

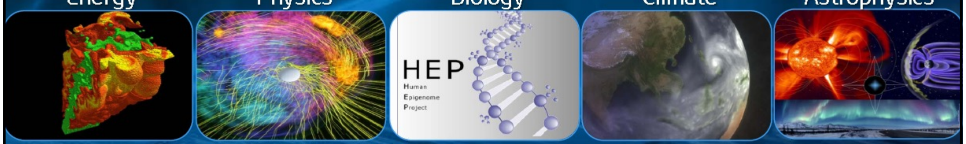
37

Exascale: *The Next Frontier*

Challenges

- Power – energy / operation of computation, data transport, memory
- Threading software to millions/billions of threads
- Memory/Storage capacity and bandwidth
- Managing high-node count systems in the existence of failures (MTBF)
- Affordability

Energy | Physics | Biology | Climate | Astrophysics



Intel committed to solving the challenges of Exascale

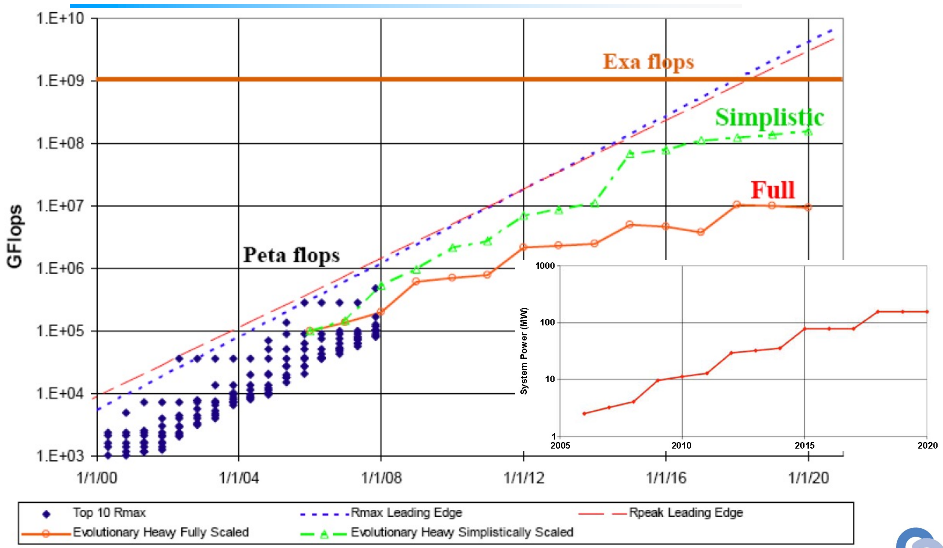
38

DARPA Exascale Study

- Commissioned by DARPA to explore the challenges for Exaflop computing
- Two models for future performance growth
  - **Simplistic**:
    - ITRS (International Technology Roadmap for Semiconductors) roadmap
    - Power for memory grows linear with # of chips
    - Power for interconnect stays constant
  - **Fully scaled**: same as simplistic, but memory and router power grow with peak flops per chip

39

Won't reach Exascale this way



From Peter Kogge, DARPA ExascaleStudy

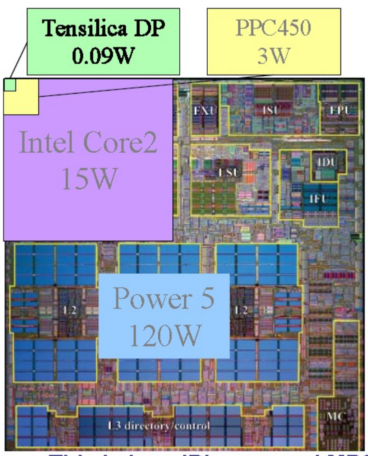
40

Extrapolating to Exaflop/s

	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. 10Gbps now possible in both copper and optics.
Power in network	100 KW	20 MW	4 MW	10 mW/Gbps assumed. 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 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. 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: 1) fast global sum (2 per iteration) 2) hardware offload for messaging (Driverless messaging)

Source: David Turek, IBM

Design for Low Power: More Concurrency

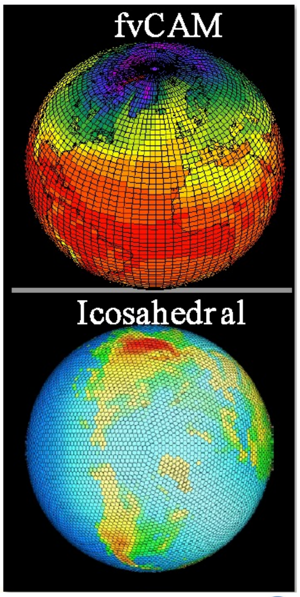


- Cubic power improvement with lower clock rate due to V^2F
- Slower clock rates enable use of simpler cores
- Simpler cores use less area (lower leakage) and reduce cost
- Tailor design to application to <u>reduce waste</u>

This is how iPhones and MP3 players are designed to maximize battery life and minimize cost

Green Flash: Ultra-Efficient Climate Modeling

- An alternative route to exascale computing
  - Exascale science questions are identified
  - The idea is to target specific machine designs to each of these questions
    - Possible because of new technologies driven by the consumer market
- We want to turn the process around
  - Ask "What machine do we need to answer a question?"
  - Not "What can we answer with that machine?"
- Goal: influence the HPC industry by evaluating a prototype design



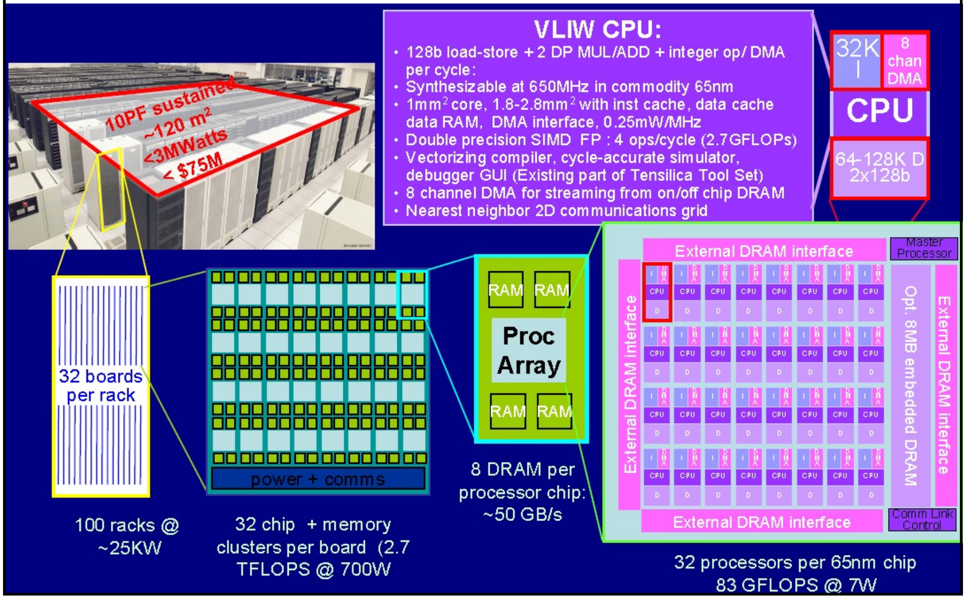
Green Flash Strawman System Design

We examined three different approaches (in 2008 technology)
Computation.015oX.02oX100L: 10 PFlops sustained, ~200 PFlops peak

- **AMD Opteron**: Commodity approach, lower efficiency for scientific applications offset by cost efficiencies of mass market
- **BlueGene**: Generic embedded processor core and customize system-on-chip (SoC) to improve power efficiency for scientific applications
- **Tensilica XTensa**: Customized embedded CPU w/SoC provides further power efficiency benefits but maintains programmability

Processor	Clock	Peak/ Core (Gflops)	Cores/ Socket	Sockets	Cores	Power	Cost 2008
AMD Opteron	2.8GHz	5.6	2	890K	1.7M	179 MW	$1B+
IBM BG/P	850MHz	3.4	4	740K	3.0M	20 MW	$1B+
Green Flash / Tensilica XTensa	650MHz	2.7	32	120K	4.0M	3 MW	$75M

Climate System Design Concept



Summary on Green Flash

- Exascale computing is vital for numerous key scientific areas
- We propose a new approach to high-end computing that enables transformational changes for science
- Research effort: study feasibility and share insight w/ community
- This effort will augment high-end general purpose HPC systems
  - Choose the science target first (climate in this case)
  - Design systems for applications (rather than the reverse)
  - Leverage power efficient embedded technology
  - Design hardware, software, scientific algorithms together using hardware emulation and auto-tuning
  - Achieve exascale computing sooner and more efficiently

Applicable to broad range of exascale-class applications

47

What about 1 million cores?

- What are applications developers concerned about?
- … but before we answer this question, the more interesting question is …
  - 1000 cores on the laptop ?
- What are commercial applications developers going to do with it?

47

48

Summary

- Major Challenges are ahead for extreme computing
  - Power
  - Parallelism
  - …and many others not discussed here
- We will need completely new approaches and technologies to reach the Exascale level
- This opens up a unique opportunity for science applications to lead extreme scale systems development

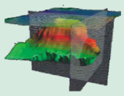
48

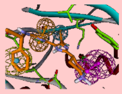
Table of Contents

Objective

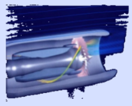
- Building up the ability to translate parallel computing power into science and engineering breakthroughs
  - Identify applications whose computing structures are suitable for
  - These applications can be revolutionized by 100X more computing power
  - You have access to expertise needed to tackle these applications
- Develop algorithm patterns that can result in both better efficiency as well as better HW utilization
  - To share with the community of application developers

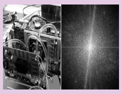
Future Science and Engineering Breakthroughs Hinge on Computing

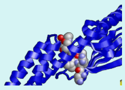

Computational Geoscience

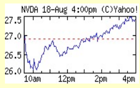

Computational Chemistry


Computational Medicine


Computational Modeling


Computational Physics


Computational Biology



Computational Finance

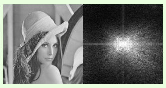


Image Processing

51

A major shift of paradigm

- In the 20th Century, we were able to understand, design, and manufacture what we can ***measure***
  - Physical instruments and computing systems allowed us to see farther, capture more, communicate better, understand natural processes, control artificial processes…
- In the 21st Century, we are able to understand, design, create what we can ***compute***
  - Computational models are allowing us to see even farther, going back and forth in time, relate better, test hypothesis that cannot be verified any other way, create safe artificial processes

52

53

Examples of Paradigm Shift

20th Century

- Small mask patterns and short light waves
- Electronic microscope and Crystallography with computational image processing
- Anatomic imaging with computational image processing
- Teleconference

21st Century

- Computational optical proximity correction
- Computational microscope with initial conditions from Crystallography
- Metabolic imaging sees disease before visible anatomic change
- Tele-emersion

53

54

Faster is not "just Faster"

- 2-3X faster is "just faster"
  - Do a little more, wait a little less
  - Doesn't change how you work
- 5-10x faster is "significant"
  - Worth upgrading
  - Worth re-writing (parts of) the application
- 100x+ faster is "fundamentally different"
  - Worth considering a new platform
  - Worth re-architecting the application
  - Makes new applications possible
  - Drives "time to discovery" and creates fundamental changes in Science

54

55

How much computing power is enough?

- Each jump in computing power motivates new ways of computing
  - Many apps have approximations or omissions that arose from limitations in computing power
  - Every 100x jump in performance allows app developers to innovate
  - Example: graphics, medical imaging, physics simulation, etc.

Application developers do not take us seriously until they see real results.

55

56

Why didn't this happen earlier?

- Computational experimentation is just reaching critical mass
  - Simulate large enough systems
  - Simulate long enough system time
  - Simulate enough details
- Computational instrumentation is also just reaching critical mass
  - Reaching high enough accuracy
  - Cover enough observations

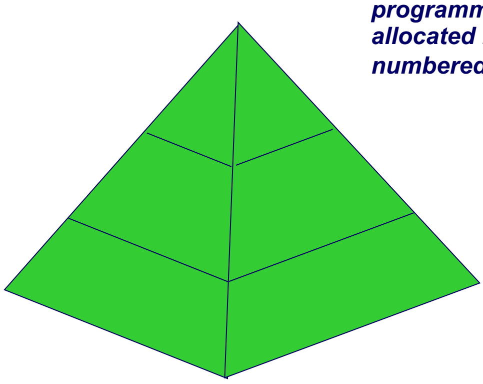
56

A Great Opportunity for Many

- New massively parallel computing is enabling
  - Drastic reduction in “time to discovery”
  - 1st principle-based simulation at meaningful scale
  - New, 3rd paradigm for research: computational experimentation
- The “democratization” of power to discover
  - $2,000/Teraflops SPFP in personal computers today
  - $5,000,000/Petaflops DPFP in clusters in 2-3 years
  - HW cost will no longer be the main barrier for big science
- This is once-in-a-career opportunity for many!

57

The Pyramid of Parallel Programming



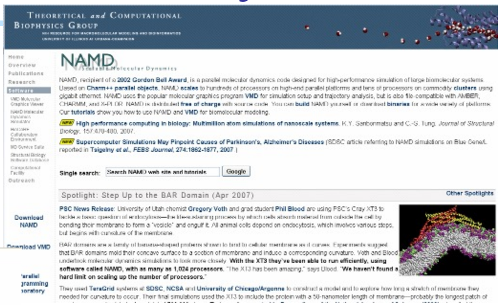
Thousand-node systems with MPI-style programming, >100 TFLOPS, $M, allocated machine time (programmers numbered in hundreds)

Hundred-core systems with CUDA-style programming, 1-5 TFLOPS, $K, machines widely availability (programmers numbered in 10s of thousands)

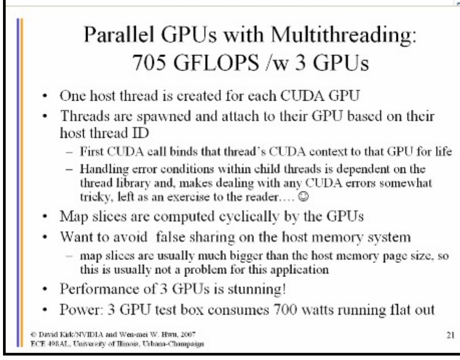
Hundred-core systems with MatLab-style programming, 10-50 GFLOPS, $K, machines widely available (programmers numbered in millions)

58

VMD/NAMD Molecular Dynamics



http://www.ks.uiuc.edu/Research/vmd/projects/ece498/lecture/



Parallel GPUs with Multithreading:
705 GFLOPS /w 3 GPUs

- One host thread is created for each CUDA GPU
- Threads are spawned and attach to their GPU based on their host thread ID
  - First CUDA call binds that thread's CUDA context to that GPU for life
  - Handling error conditions within child threads is dependent on the thread library and, makes dealing with any CUDA errors somewhat tricky, left as an exercise to the reader.... ☺
- Map slices are computed cyclically by the GPUs
- Want to avoid false sharing on the host memory system
  - map slices are usually much bigger than the host memory page size, so this is usually not a problem for this application
- Performance of 3 GPUs is stunning!
- Power: 3 GPU test box consumes 700 watts running flat out

© David Kirk/NVIDIA and Wen-mei W. Hwu, 2007
ECE 498AL, University of Illinois, Urbana-Champaign
21

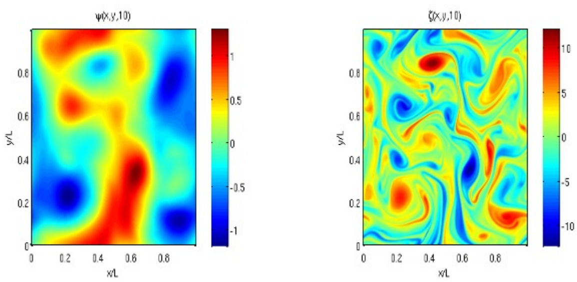
- 240X speedup
- Computational biology

59

Matlab: Language of Science

15X with MATLAB CPU+GPU

http://developer.nvidia.com/object/matlab_cuda.html



Pseudo-spectral simulation of 2D Isotropic turbulence

http://www.amath.washington.edu/courses/571-winter-2006/matlab/FS_2Dturb.m

60