The Future of Parallel Programming and Getting to Exascale

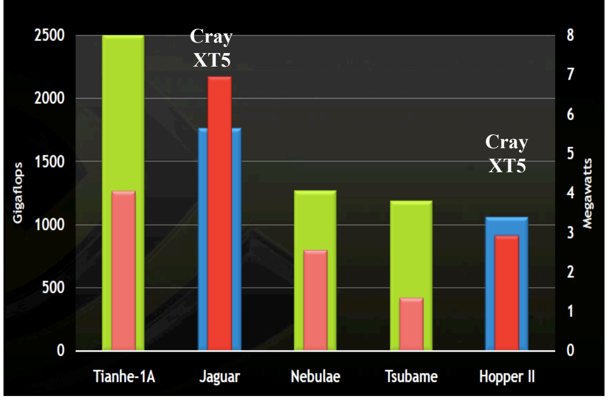
Last Lecture

The Fastest Supercomputers in the World

Name	Reign	Location	What processors?	How many?	How fast?
Titan	NEW	Oak Ridge National Laboratory, USA	AMD Opterons and Nvidia Keplers	560,640 cores, half GPUs	17.6 PFlops
Sequoia (IBM BG/Q)	2012	Lawrence Livermore National Laboratory, USA	IBM Power BGC (custom)	1,572,864 cores	16.3 PFlops
K	2011	Riken, JAPAN	SPARC64 processors	705,024 cores	10.5 PFlops
Tianhe-1A	2010	National Supercomputing Center, CHINA	Intel Xeon and Nvidia Fermis	186,368 cores	2.57 PFlops
Jaguar (Cray XT5)	2010	Oak Ridge National Laboratory, USA	AMD 6-core, dual-processor Opterons	~37,000 processor chips (224,162 cores)	1.76 PFlops
RoadRunner	2009	Los Alamos National Laboratory, USA	AMD Opterons, IBM Cell/BE (Playstations)	~19,000 processor chips (129,600 cores)	1.1 PFlops

See www.top500.org

Top 5 Performance and Power (3 use Nvidia GPUs) from "GPU Computing To Exascale and Beyond", Bill Dally, SC10



DARPA Exascale Technology Reports

Georgia Institute of Technology

See http://users.ece.gatech.edu/mrichard/ExascaleComputingStudyReports/ECS_reports.htm

Exascale Computing Study Report

eginning in mid-2007, DARPA/IPTO, contracting through AFRL, has sponsored a series of studies intended to understand the future course of mainstream computing technology and determine whether or not it would llow a 1,000X increase in the computational capabilities of computing systems by the 2015 time frame. Where current technology trends were deemed incapable of achieving such increases, the study was also charged with lentifying the major challenges and the areas where additional targeted research could lay the groundwork for overcoming them.

he following report of the first Exascale Computing Study has been publicly released. Reports of additional exascale computing studies will be made available here as they are also publicly released.

- ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems (September 28, 2008)

  Approved for Public Release, Distribution Unlimited

  The material in this document reflects the collective views, ideas, opinions and findings of the study participants only, and not those of any of the universities, corporations, or other institutions with which they are affiliated. Furthermore, the material in this document does not reflect the official views, ideas, opinions and/or findings of DARPA, the Department of Defense, or of the United States government.

- ExaScale Computing Software Study: Software Challenges in Extreme Scale Systems (September 14, 2009)

  Approved for Public Release, Distribution Unlimited

  The material in this document reflects the collective views, ideas, opinions and findings of the study participants only, and not those of any of the universities, corporations, or other institutions with which they are affiliated. Furthermore, the material in this document does not reflect the official views, ideas, opinions and/or findings of DARPA, the Department of Defense, or of the United States government.

Getting to Exascale

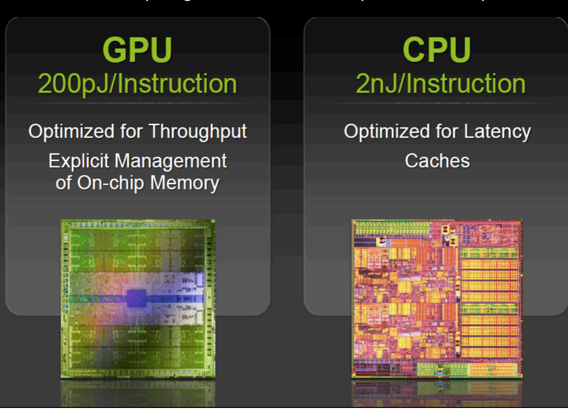
- Before 2020, exascale systems will be able to compute a quintillion operations per second!
- Scientific simulation will continue to push on system requirements:
  - To increase the precision of the result
  - To get to an answer sooner (e.g., climate modeling, disaster modeling)
- The U.S. will continue to acquire systems of increasing scale
  - For the above reasons
  - And to maintain competitiveness
- A similar phenomenon in commodity machines
  - More, faster, cheaper

Exascale Challenges Will Force Change in How We Write Software

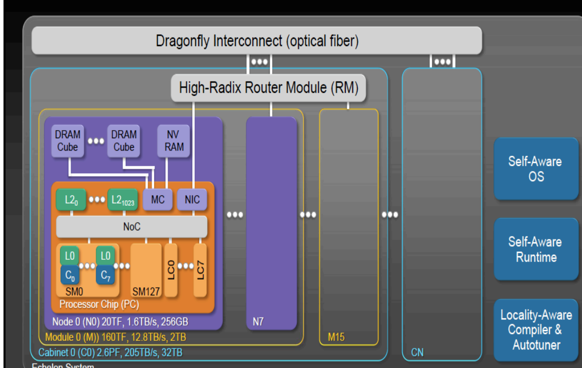
- Exascale architectures will be fundamentally different
  - Power management becomes fundamental
  - Reliability (h/w and s/w) increasingly a concern
  - Memory reduction to .01 bytes/flop
  - Hierarchical, heterogeneous
- Basic rethinking of software
  - Express and manage locality and parallelism for ~billion threads
  - Create/support applications that are prepared for new hardware (underlying tools map to h/w details)
  - Manage power and resilience
    - Locality is a big part of power/energy
    - Resilience should leverage abstraction changes

"Software Challenges in Extreme Scale Systems," V. Sarkar, B. Harrod and A. Snavely, SciDAC 2009, June, 2009. Summary of results from a DARPA study entitled, "Exascale Software Study," June 2008 through Feb. 2009.

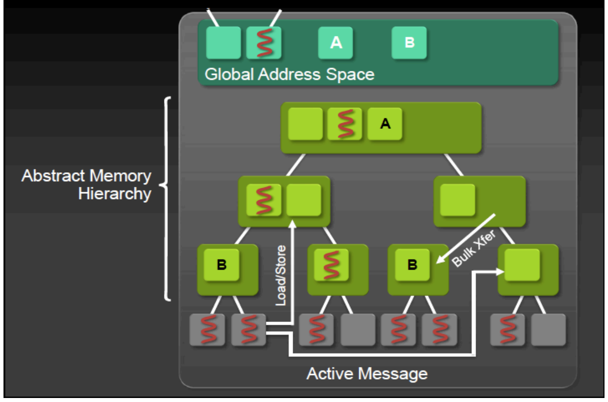
Differences in GPU/CPU for Power Consumption from "GPU Computing To Exascale and Beyond", Bill Dally, SC10



Echelon System Sketch from "GPU Computing To Exascale and Beyond", Bill Dally, SC10



Echelon Execution Model
from "GPU Computing To Exascale and Beyond", Bill Dally, SC10



Current PetaScale Systems, and other Upcoming Architectures

Name	Reign	Location	What processors?	How many?	How fast?
Titan	NEW	Oak Ridge National Laboratory, USA	AMD Opterons and Nvidia Keplers	560,640 cores, half GPUs	17.6 PFlops
Sequoia (IBM BG/Q)	2012	Lawrence Livermore National Laboratory, USA	IBM Power BGC (custom)	1,572,864 cores	16.3 PFlops
K	2011	Riken, JAPAN	SPARC64 processors	705,024 cores	10.5 PFlops
Tianhe-1A	2010	National Supercomputing Center, CHINA	Intel Xeon and Nvidia Fermis	186,368 cores	2.57 PFlops
Jaguar (Cray XT5)	2010	Oak Ridge National Laboratory, USA	AMD 6-core, dual-processor Opterons	~37,000 processor chips (224,162 cores)	1.76 PFlops
RoadRunner	2009	Los Alamos National Laboratory, USA	AMD Opterons, IBM Cell/BE (Playstations)	~19,000 processor chips (129,600 cores)	1.1 PFlops

What Makes a Parallel Programming Model Successful for High-End Computing

- Exposes architecture's ***execution model***, the principles of execution and what operations are supported well
- Must be possible to achieve high performance, even if it is painful
- Portable across platforms
- Easy migration path for existing applications, so nearby current approaches

What Makes a Parallel Programming Model Successful for the Masses

- Productivity
  - Programmer can express parallelism at a high level
  - Correctness is not difficult to achieve
- Portable across platforms
- Performance gains over sequential easily achievable

Future Parallel Programming

- It seems clear that for the next decade architectures will continue to get more complex, and achieving high performance will get harder.
- Most people in the research community agree that different kinds of parallel programmers will be important to the future of computing.
  - Programmers that understand how to write software, but are naïve about parallelization and mapping to architecture (***Joe programmers***)
  - Programmers that are knowledgeable about parallelization, and mapping to architecture, so can achieve high performance (***Stephanie programmers***)
  - Intel/Microsoft say there are three kinds (***Mort, Elvis and Einstein***)
- Programming abstractions will get a whole lot better by supporting specific users.

13

THE UNIVERSITY OF UTAH

A Broader View in 2012

Thanks to exascale reports and workshops

- Multiresolution programming systems for different users
  - Joe/Stephanie/Doug **[Pingali, UT]**
  - Elvis/Mort/Einstein **[Intel]**
- Specialization simplifies and improves efficiency
  - Target specific user needs with domain-specific languages/libraries
  - Customize libraries for application needs and execution context
- Interface to programmers and runtime/hardware
  - Seamless integration of compiler with programmer guidance and dynamic feedback from runtime
- Toolkits rather than monolithic systems
  - Layers support different user capability
  - Collaborative ecosystem
- Virtualization (over-decomposition)
  - Hierarchical, or flat but construct hierarchy when applicable?

14

THE UNIVERSITY OF UTAH