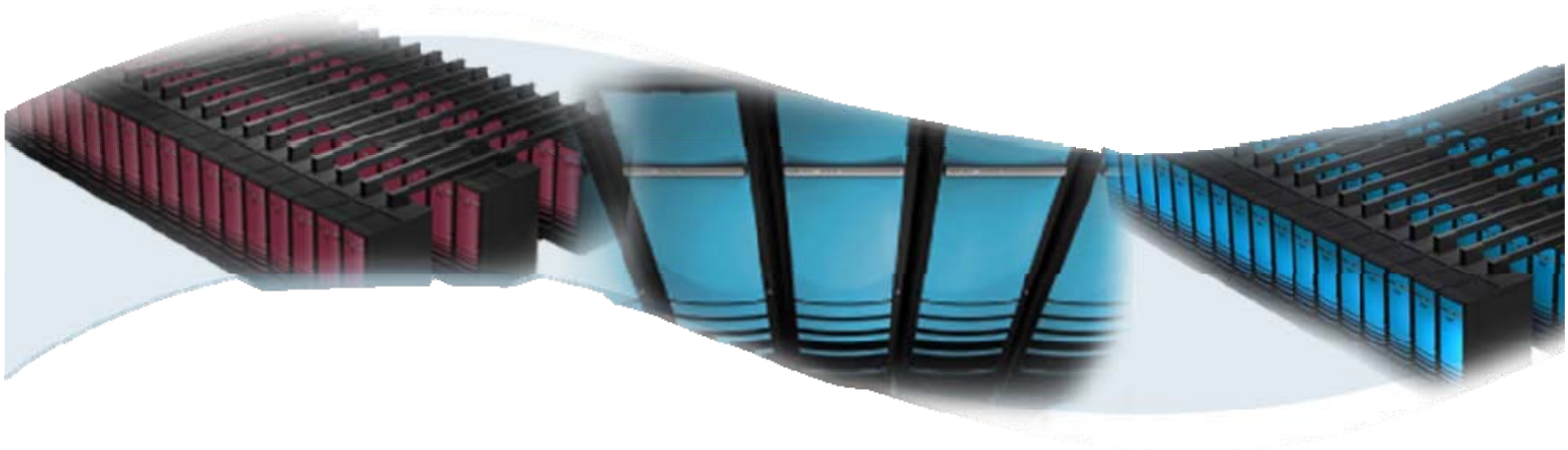
Building Effective, Power-Efficient Systems over the Next Decade



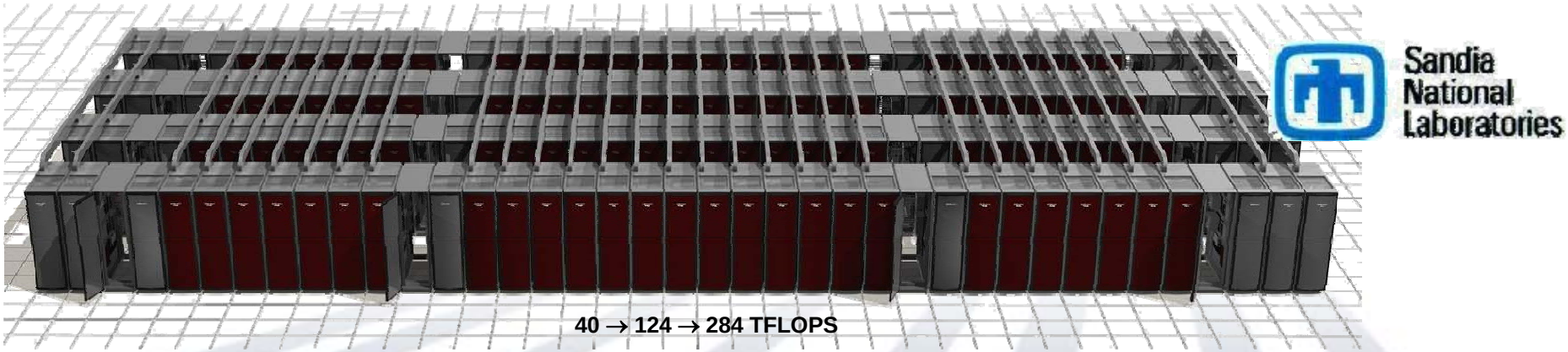
Steve Scott
Cray CTO

Workshop on Power Efficiency and the Path to Exascale Computing
SC'08

Some Current Large Systems from Cray

140 → 356 TFLOPS

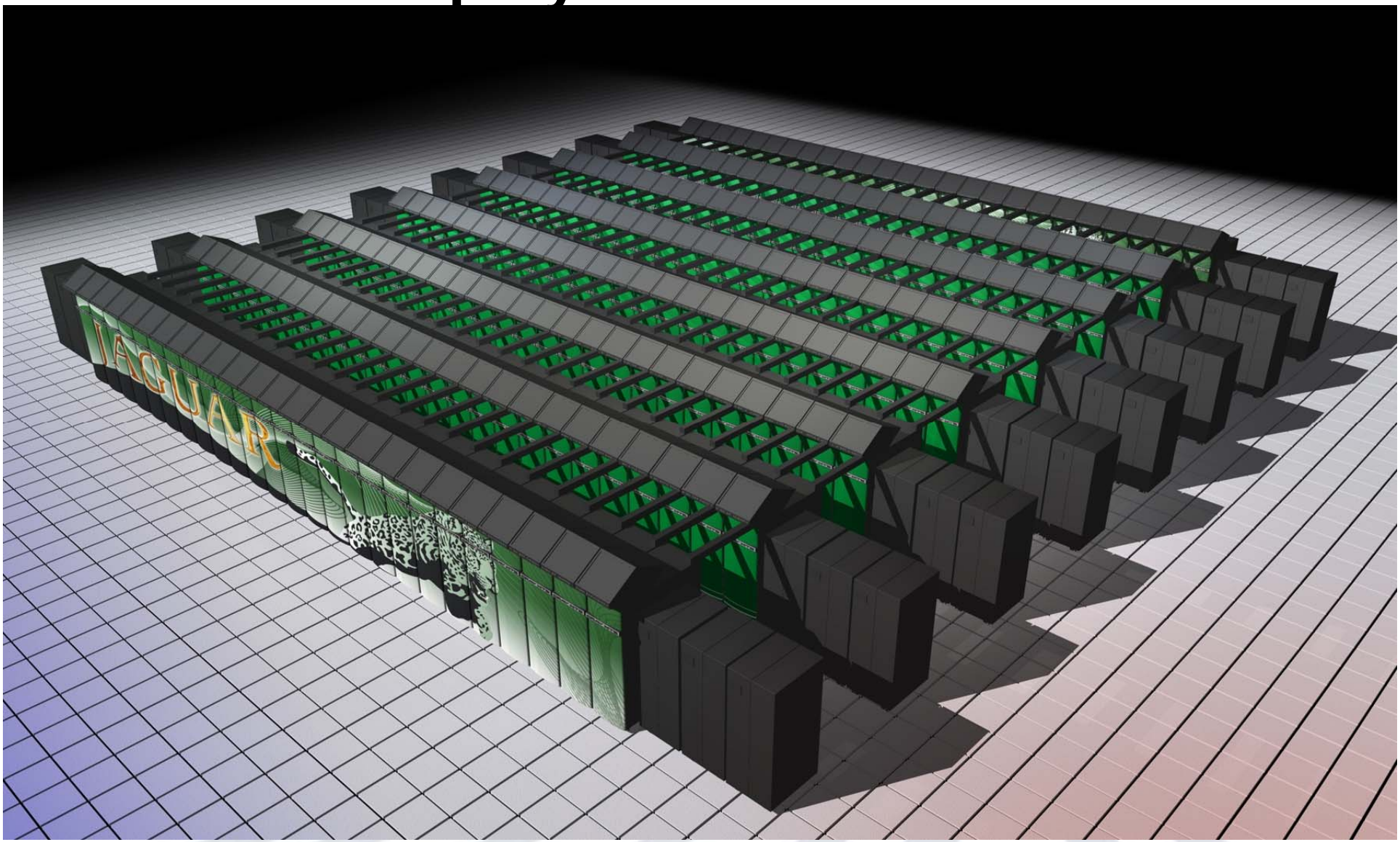
20 → 50 → 119 → 263 TF → 1.6 PF



40 → 124 → 284 TFLOPS

Copyright 2008, Cray Inc.

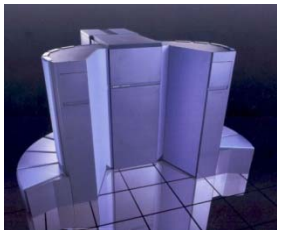
ORNL Petaflop System



Copyright 2008, Cray Inc.

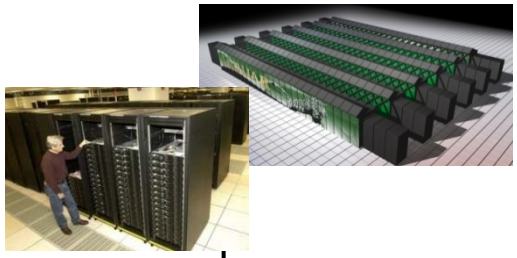
Milestones in Scalable Computing (Powers of 1000)

First application
Gigaflop, 1989



First application
Teraflop, 1998

First Linpack
Petaflop
2008



1990 2000 2010

First Linpack
Teraflop, 1996

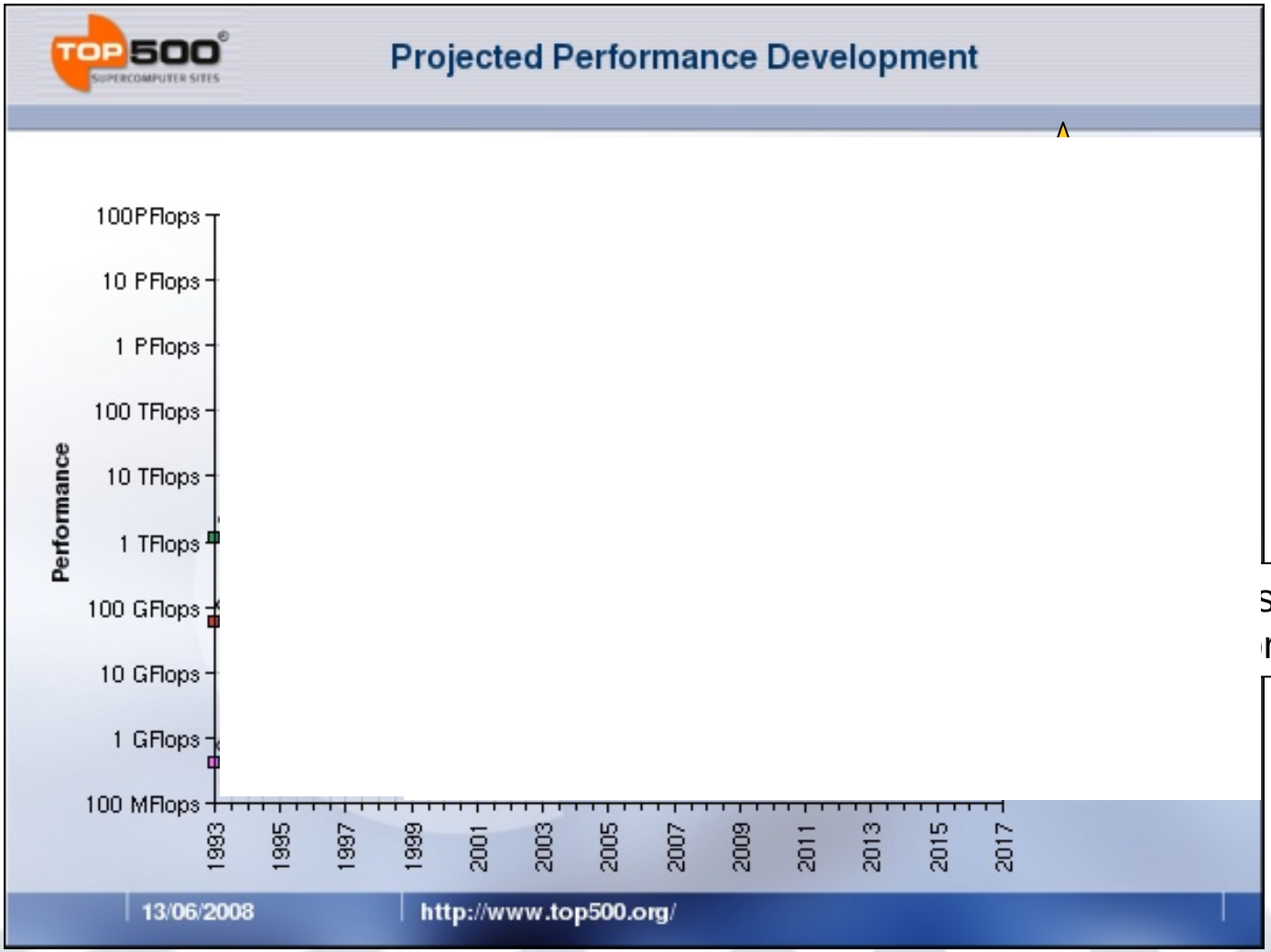
First peak (special purpose)
Petaflop, 2006

Next comes Exaflop, Zettaflop and Yottaflop…

Copyright 2008, Cray Inc.

March to a ~~Teraflop~~ ~~Petaflop~~ Exaflop

(…Zettaflop, Yottaflop, Lottaflop!)



stant cost,
r power

Copyright 2008, Cray Inc.

DARPA Exascale Study Team

May – December 2007

Objective: understand the course of mainstream technology and determine the primary challenges to reaching a 1000x increase in computing capability by 2015.

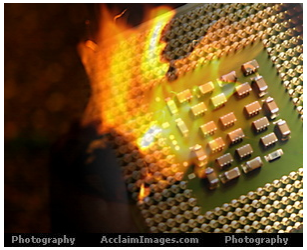
NAME	Affiliation	NAME	Affiliation
Keren Bergman	**Columbia**	**Steve Keckler**	**UT-Austin**
Shekhar Borkar	**Intel**	**Dean Klein**	**Micron**
Dan Campbell	**GTRI**	**Peter Kogge**	**Notre Dame (chair)**
Bill Carlson	**IDA**	**Bob Lucas**	**USC/ISI**
Bill Dally	**Stanford**	**Mark Richards**	**Georgia Tech**
Monty Denneau	**IBM**	**Al Scarpeli**	**AFRL**
Paul Franzon	**NCSU**	***Steve Scott***	***Cray***
Bill Harrod	**DARPA**	**Allan Snavely**	**SDSC**
Kerry Hill	**AFRL**	**Thomas Sterling**	**LSU**
Jon Hiller	**STA**	**Stan Williams**	**HP**
Sherman Karp	**STA**	**Kathy Yelick**	**UC-Berkeley**

Copyright 2008, Cray Inc.

Key Challenges to Get to the Exascale

- Okay, all we need is a system with 5-10 TF/chip and 100-200K chips…

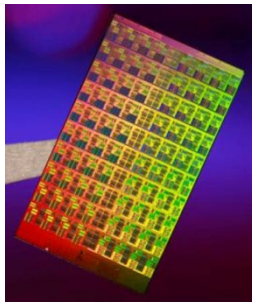
- Power
  - If 2014 FPU is ~10-20 pJ/flop, then 1 EF = 10-20MW for flops alone!
  - Memory bandwidth: Eref/sec * 10% miss * 100 bits * 3pJ/bit = 30 MW
  - Rotating storage @ ~5 MW/EB could easily add tens of MW
- System and application resiliency
  - 500 FIT processor (very aggressive!) has an expected lifetime of > 100 years
  - 200,000 processors → over four failures per day
  - Now throw in a network and over 10M memory parts
  - Forget trying to make MTBF acceptable (*deal* with failures)
  - Applications are the *really* hard part
    - CP/RS is a start, but bandwidth isn't scaling (SSD's?)
    - Need to consider resilient, transactional models (automated?)



- Local memory bandwidth technologies
  - Last 30 years: DRAM density has outpaced bandwidth by ~75 times
  - Memory BW is limiting performance of future designs
  - Processors + DIMMs has to go
  - 3D chip stacking, direct optical memory connections
  - Nano-structure memory; phase-change memory; holographic memory…

Copyright 2008, Cray Inc.

Key Challenges to Get to the Exascale (2)



- Processor microarchitecture to exploit locality
  - Tension between efficiency and programmability
  - Need a new microarchitecture and execution model
    - Much lower control overhead relative to computation
    - Much more aggressive exploitation of locality
    - Explicit control of data movement (?)
  - Must *not* burden the programmer with this

- Finding enough concurrency in the applications
  - One billion operations executed each *clock cycle*
  - 200,000 sockets * 200 GB/s * 100ns = 4 billion refs in flight!
  - Will require *huge* problems
  - Must exploit threading, vectorization/streaming and multi-level parallelism
- Programming difficulty
  - MPI is the defacto standard, but is a low-productivity programming model
    - What's more, it's not *really* portable, in that it is unsuitable for machines with global memory, advanced latency hiding mechanisms, and low overhead synchronization
  - Time is right for a high productivity language
    - Partitioned global address space (PGAS) languages a good start
    - Chapel, X10, or other new language would be better yet

Copyright 2008, Cray Inc.

Cray's Approach to Extreme Scaling

- Strong focus on power efficiency
  - Leverage processor power saving features and technology improvements
  - Drive the system PUE down close to 1 by eliminating chillers and CRACs
- Design processors that…
  - …reduce control overhead per flop (not just standard multi-core)
  - …can exploit multi-level parallelism (critical to achieve needed concurrency)
  - …are likely heterogeneous (fast serial threads and efficient parallel threads)
  - …are highly latency tolerant (so as to leave fewer applications behind)
- Use a high productivity system architecture
  - More on that in a moment…
- Enable and support better programming models and programming tools
  - Continue to provide excellent MPI performance for mainstream HPC codes
  - Enable optimization and new development with Shmem, PGAS and Chapel
  - Scalable debuggers and performance tools, expert analysis, auto-tuning, etc.
- Manage *on-node* parallelism via compiler and runtime
  - User primarily thinks about parallelizing over nodes
  - Compiler and runtime think about hierarchical parallelism
  - (Of course allow sophisticated users to provide hints or do restructuring)

Copyright 2008, Cray Inc.

Improving System Power Efficiency

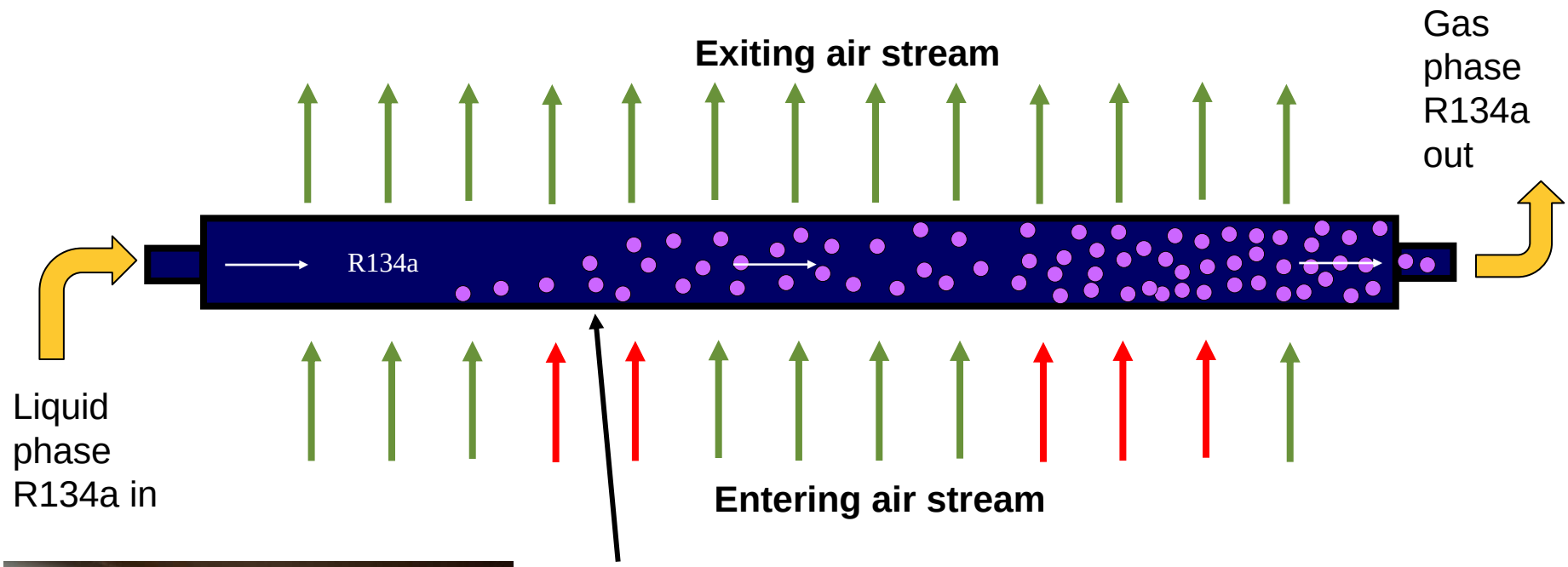
Copyright 2008, Cray Inc.

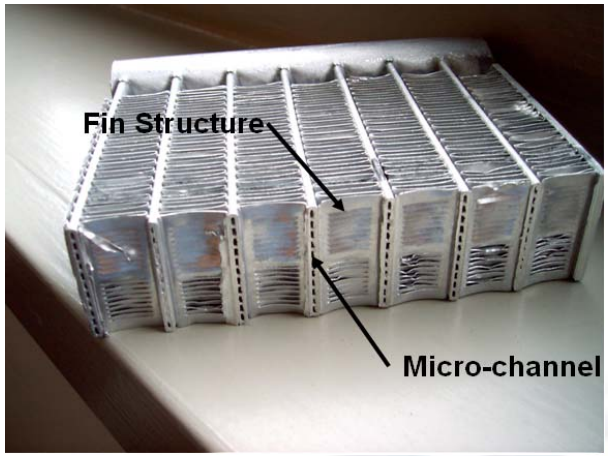
Cray Baker HE Cabinet

- Pre-cooler and after-cooler evaporators
- Capable of removing 60kW at 20C ambient air temps
- Allows 18C (65F) water
- Liquid-Cooled kit is an option and can be retrofitted in the field
- Other improvements include
  - Improved backplane for better signal integrity
  - Shielded interconnect cables to support higher frequencies
- Cabinets will accommodate future blades and processors through 2011+

Copyright 2008, Cray Inc.

PHase change Liquid EXchange(PHLEX)

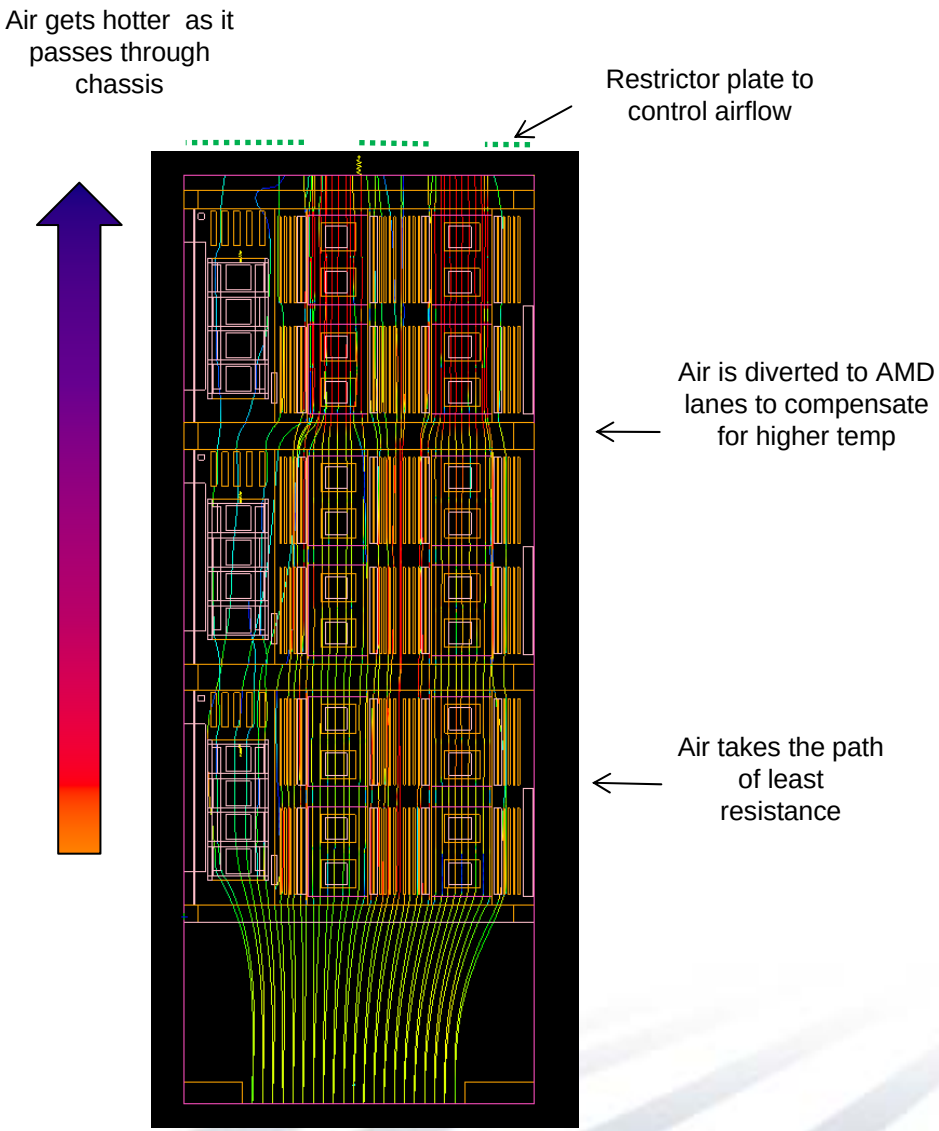




Over 10 x more effective than a water coil of similar size (phase change much more effective method to remove heat)

Copyright 2008, Cray Inc.

Progressive Airflow Technology



The management of airflow;
supply it only where needed

Originally developed and implemented in the XT5 cabinet

Allows to cool with very low air flow rates (2x the power of XT4 with same airflow)

Results in a low air side pressure requirements

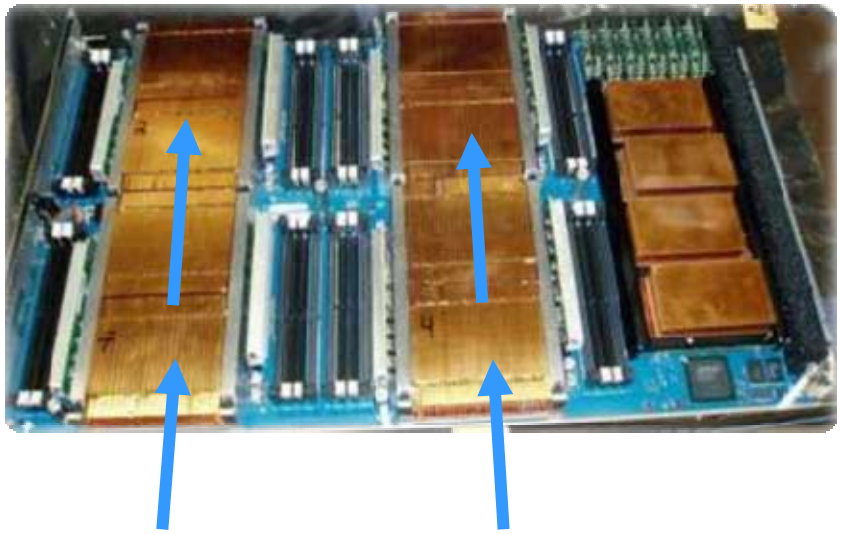
Gives a very hot exhaust temperature (120F+)

Gives a more uniform chip temp from the bottom to top cage

Copyright 2008, Cray Inc.

PAT at the Board Level

Graduated fin design delivers constant operating temperatures



Air Flow

42 fins

37 fins

24 fins

18 fins

Copyright 2008, Cray Inc.

But best Linpack per watt is NOT the goal…

Sustained application performance per watt and ease of programming are.

Copyright 2008, Cray Inc.

Petaflop Jaguar XT5 System

Nine Application World Records Set in First Two Weeks!

Science Area	Code	Contact	Cores	% of Peak	Total Perf	Scaling
Materials	DCA++	Schulthess	150144	97%	1.3 PF*	Weak
Materials	LSMS/WL	ORNL	149580	76.40%	1.05 PF	Weak
Seismology	SPECFEM3D	UCSD	149784	12.60%	165 TF	Weak
Weather	WRF	Michalakes	70000	5.60%	36 TF	Strong
Climate	POP	Jones	18000	3.00%	5 TF	Strong
Combustion	S3D	Chen	144000	6.00%	83 TF	Weak
Fusion	GTC	PPPL	102000		20 B particles pushed	Weak
Chemistry	Madness	Harrison	150000			
Materials	LS3DF	Lin-Wang Wang	147456	32%	442 TF	Weak

* Mixed Precision – 626 TF at 128K cores in 64 bit only

Copyright 2008, Cray Inc.

Cascade

CRAY
THE SUPERCOMPUTER COMPANY

High Productivity Computing Systems

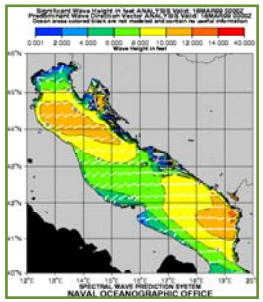
Critical to National Security

- Develop a **new generation** of **economically viable** high productivity computing systems for national security and industrial user communities (2011)
- **Ensure U.S. lead, dominance, and control** in this critical technology

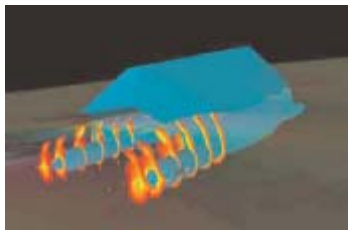
- **Performance** (time-to-solution): speedup critical apps by **10X to 40X**
- **Programmability** (idea-to-first-solution): dramatically reduce cost and time for developing apps
- **Portability** (transparency): insulate software from system specifics
- **Robustness** (reliability): continue operating in the presence of failures

Applications:

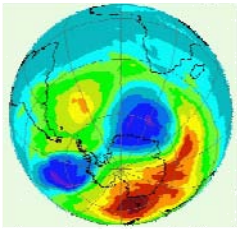
Weather Prediction



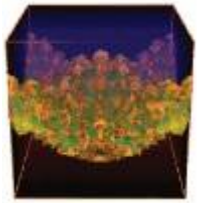
Ocean/wave Forecasting



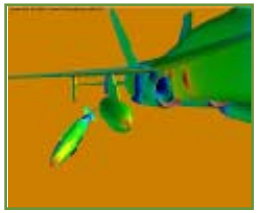
Ship Design



Climate Modeling



Nuclear Stockpile Stewardship



Weapons Integration

Fill the Critical DoD Need for:

Operational weather and ocean forecasting, weapons design and analysis, airborne contaminant modeling, intelligence / surveillance / reconnaissance, cryptanalysis, etc.

Productivity Begins with the Architecture

- Global shared address space with one-sided data transfers
  - *So that code can easily reference and access objects held in remote nodes without involvement of code running on those nodes*
- High bandwidth, low granularity network
  - *So that programs can be written with far less concern about how and when communicaiton takes place*
- Latency-tolerant processors
  - *So that compute capabilities do not go idle waiting for data, and programmers do not have to stage data and computation*
- Plentiful threading with efficient, lightweight synchronization
  - *So that parallelism can be dynamically exploited at multiple levels in the code, and programmers need to worry less about load balancing and synchronization*
- Adaptive processing capabilities
  - *So that idioms that would benefit from vectorization, streaming, fine-grain multithreading, or fast sequential processing can execute efficiently, and the programmer does not have to change the code to fit the paradigm*

Copyright 2008, Cray Inc.

A New Processor Partnership

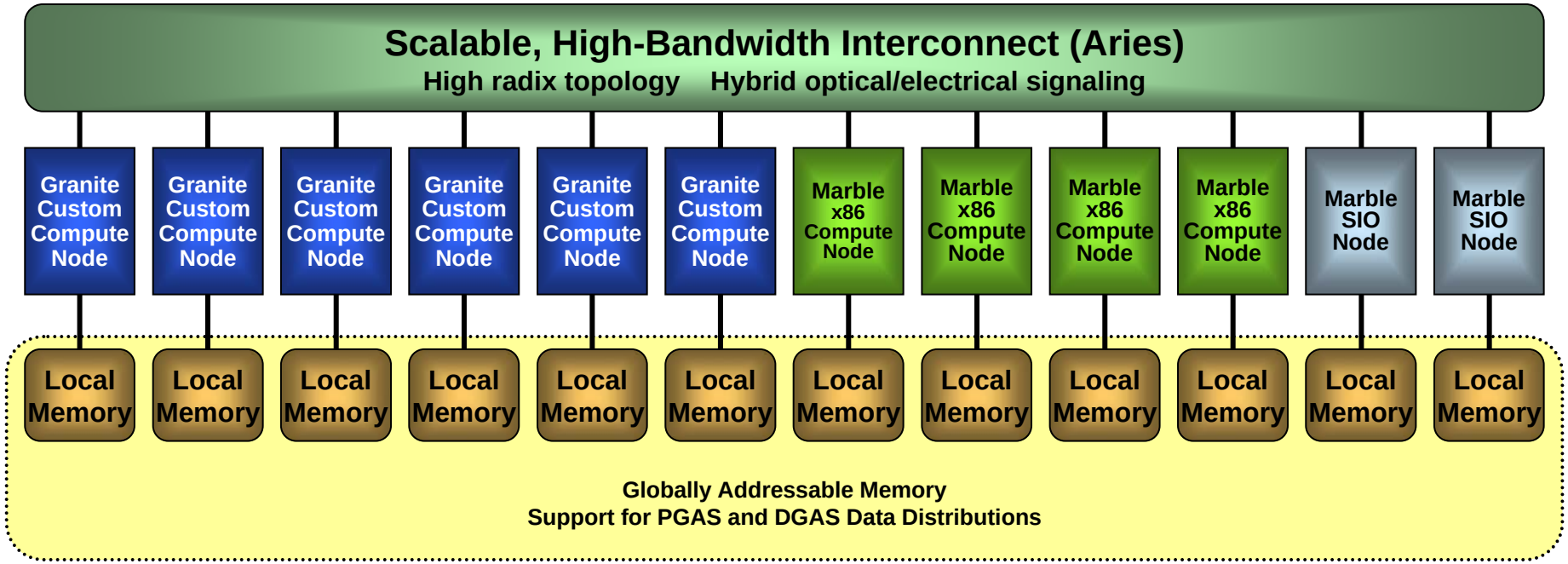
- Collaborative agreement to pursue high-end HPC technologies and opportunities
  - Intel's world-class IC process and processor architecture
  - Cray's strength in HPC architecture & large systems
  - Cray working with Intel on a new high-end processor
- Gives customers access to the best processors at any point in time

- Our strong partnership with AMD continues
- Cray will continue to develop and upgrade our existing XT product line past the end of the decade

Copyright 2008, Cray Inc.

Cray Cascade System

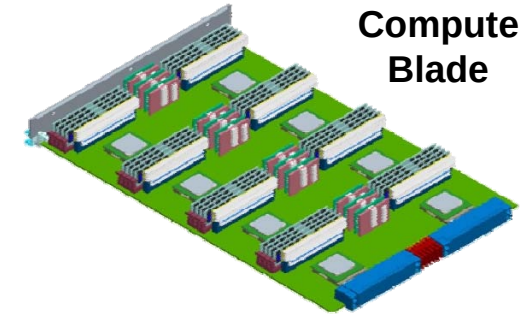
DARPA High Productivity Computing Systems



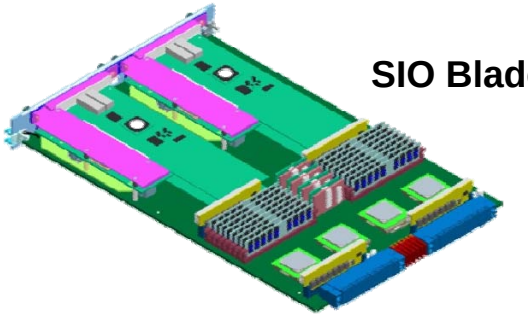
- Tightly integrated hybrid computing
- Hybrid electrical/optical network
- Configurable network, memory, processing and I/O
- Globally addressable memory
- Very high performance communication and synchronization

Copyright 2008, Cray Inc.

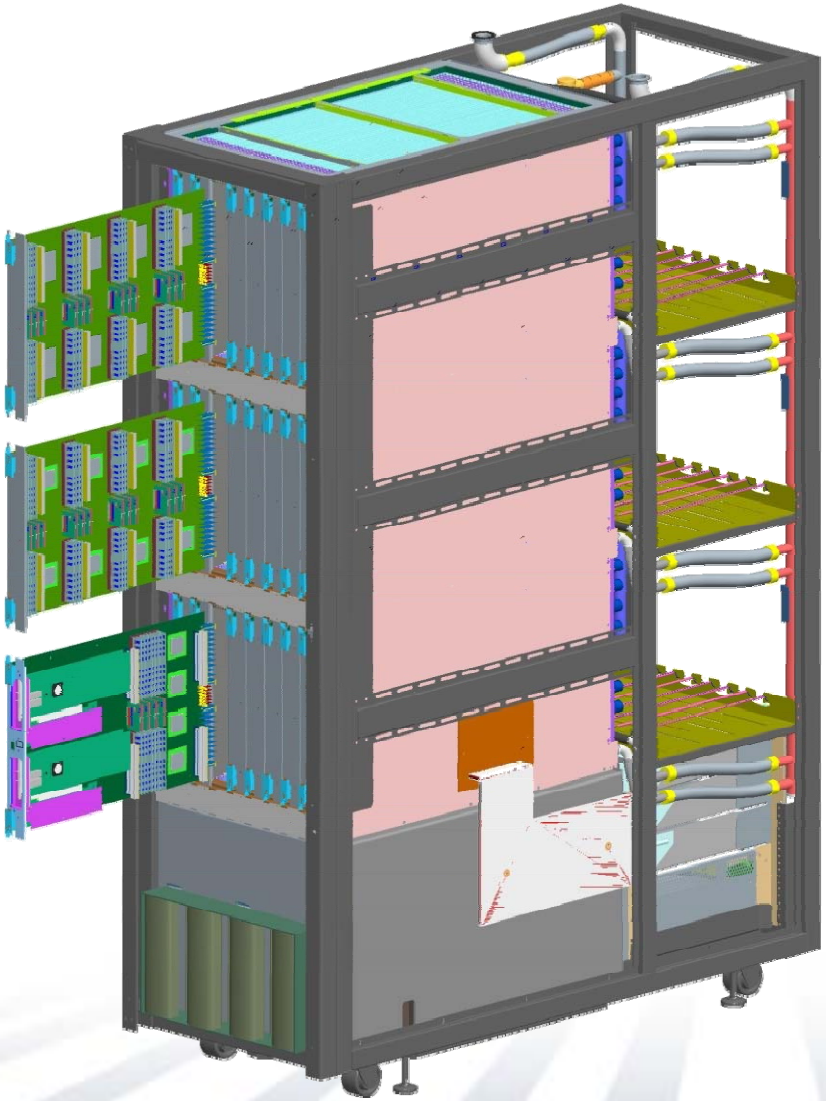
Cascade Packaging Overview



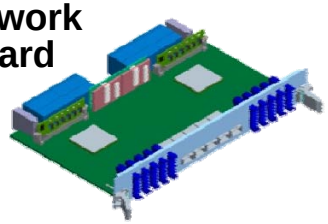
Granite or Marble Compute Blade



SIO Blade



Aries network card



Copyright 2008, Cray Inc.

Main Software Productivity Enhancements

- Compilers
  - Common user environment across Marble and Granite blades
  - Integrated support for CAF, UPC
  - Incremental compilation, runtime profiling, improved user feedback
  - Fully automatic multi-level parallelism (shared memory, multi-threading, vectorization)
- Programming Tools
  - Comparative debugging and dual-code debugging
  - Automatic performance analysis
  - Environment setup (modules)
- Scientific Libraries
  - Auto tuning (Cray Adaptive Sparse Kernels)
  - Adaptivity (Cray Adaptive FFT, Cray Adaptive Sparse Kernels)
- Programming Languages
  - Support for traditional languages
  - Integrated support for UPC, CAF, OpenMP
  - Chapel

Copyright 2008, Cray Inc.

Chapel

A new parallel language developed by Cray for HPCS

Themes

- Raise level of abstraction, generality compared to SPMD approaches
- Support prototyping of parallel codes + evolution to production-grade
- Narrow gap between parallel and mainstream languages

Chapel's Productivity Goals

- Vastly improve **programmability** over current languages/models
- Support **performance** that matches or beats MPI
- Improve **portability** over current languages/models (actually better than MPI)
- Improve code **robustness** via better abstractions and semantics

```
...
forall (_, r) in (Updates, RAStream()) do
  on T( r & indexMask ) do
    T( r & indexMask ) ^= r;
```

Status

- Draft language specification available
- Portable prototype implementation underway
- Most effort to date has been focused on functionality and feature evaluation
- Early releases to ~90 users at ~30 sites (academic, government, industry)
- Public release planned for SC08

Copyright 2008, Cray Inc.

Longer Term…

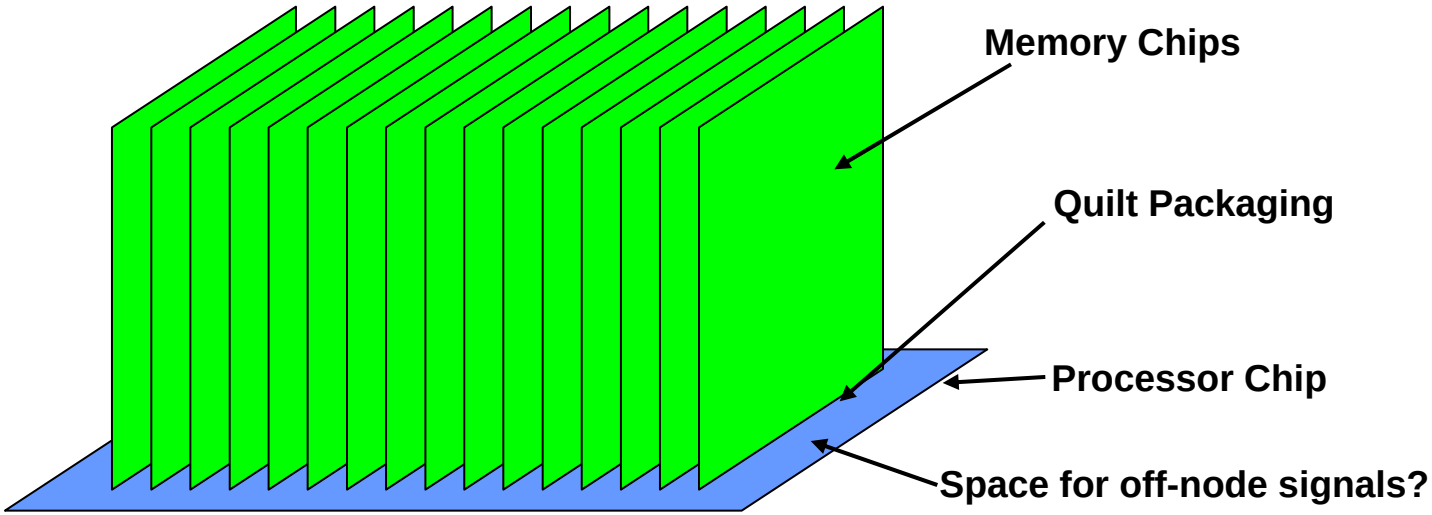
Copyright 2008, Cray Inc.

One Suggestion for Packaging Granularity

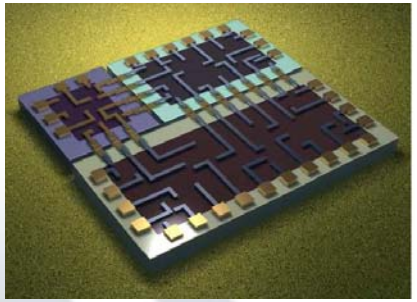
- Put as much memory as you can *directly attached* to the processor
  - Call this a node
  - It's fairly small
- Build as good a network as you can between nodes
  - Optimize the network around packaging constraints
    - Lots more bandwidth on board than off board
  - Provide support for board-level domain of "flat" shared memory
  - Distributed memory model beyond the board
- Layer on a good system architecture
  - Globally addressable memory
  - Scalable address translation and synchronization
  - Latency tolerant processor architecture

Copyright 2008, Cray Inc.

3D Node: Processor + Orthogonal Memory Chips

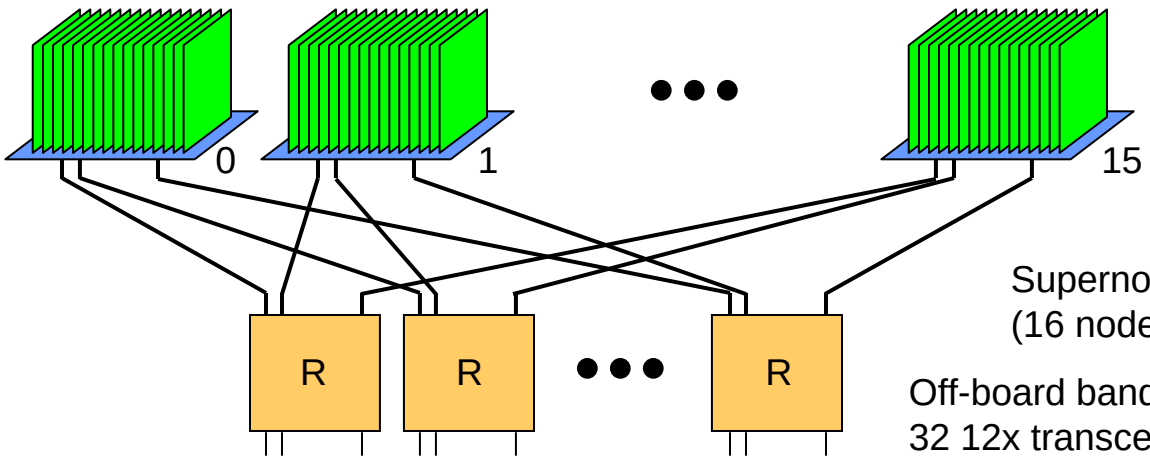


- Capacity
  - 8-32 memory chips @ 1 GB each = 8-32 GB per node
- Bandwidth
  - 5 μm pitch wires (10 μm per diff signal), 15mm edge ⇒ 1500 signals per memory chip
  - Need to keep signaling rates to < 10 Gbps with memory periphery transistors
  - Assume 512 bits/dir @ 8.25 Gbps, packetized protocol, 80% read efficiency
    - ⇒ 320 GB/s read bandwidth per memory chip (1.28W at 0.5 pJ/bit)
    - ⇒ 2.5-10 TB/s read bandwidth per node with 8-32 memory chips
  - Could nicely feed a 5-10 TF node
  - Probably still too much power in memory chips to support this…
- Interesting power/cooling challenges



Copyright 2008, Cray Inc.

Example Board Architecture



Shown as fat-tree. Could consider other topology for on-board or off-board links.

Aggregate node bandwidth:
(16 nodes)*(4 TB/s/node) = 80 TB/s

Supernode bandwidth:
(16 nodes)*(64 sigs/node)*(25 Gbps) = 6.4 TB/s

Off-board bandwidth:
32 12x transceivers @ 16 Gbps = 768 GB/s

- Can treat as 16 nodes for highest aggregate memory bandwidth
- Could combine into 2, 4, 8 or 16-node "super-nodes"
  - Flat addressing, latency and bandwidth
  - Hashed to avoid bank conflicts
  - Would still want compiler to exploit locality within a single node
    - Either via explicit local segments or via caching (possibly in main memory)
- Inter-node signaling shown using conservative technology extrapolations
  - Could also consider high-bandwidth on-board technologies (quilting, capacitive coupling, optics?, etc.) to boost super-node bandwidth even further
- Stresses "traditional" notions of balance
  - But respects economic/technology balance

Copyright 2008, Cray Inc.

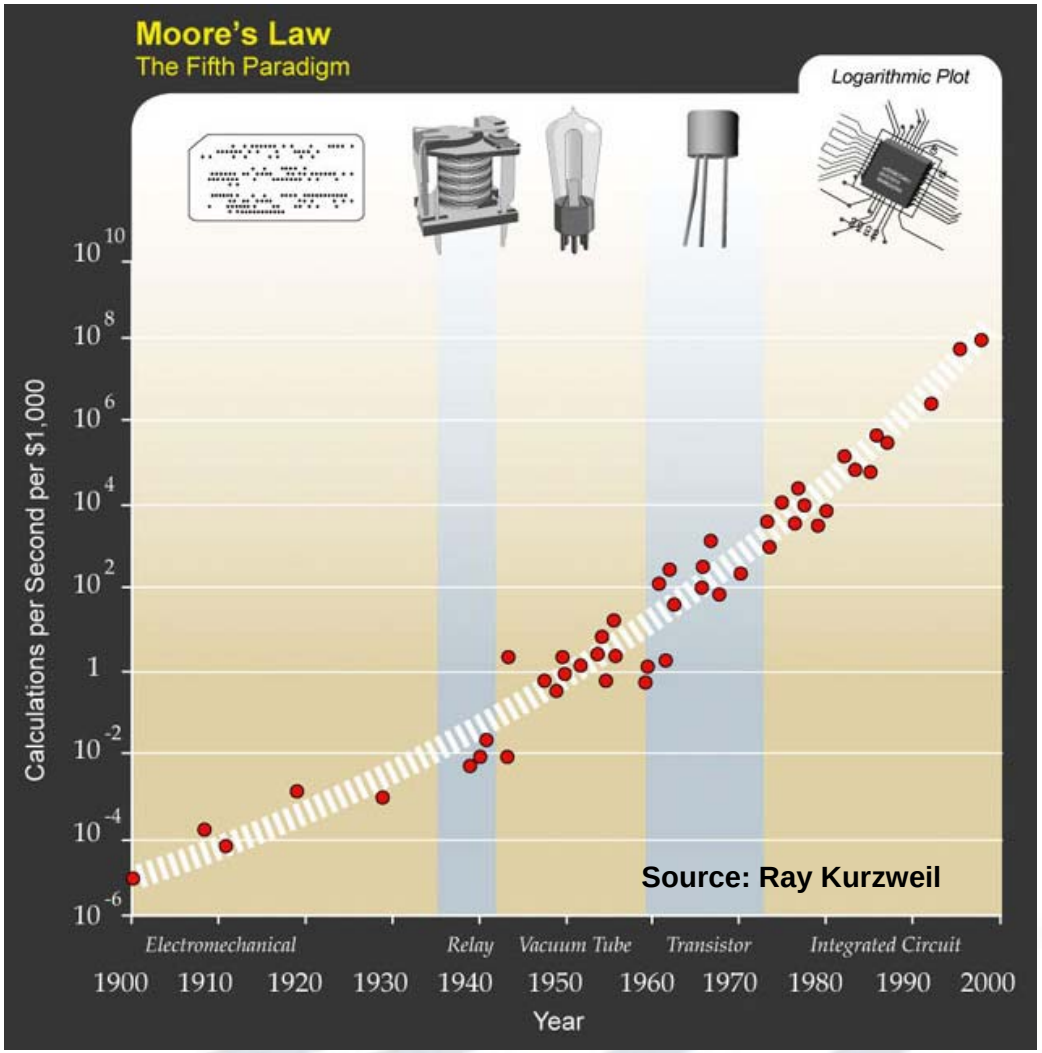
Takeaway Points

- Need aggressive exploitation of locality to acknowledge physics
  - Sub-chip, node, board
- Need more of our transistors used for computation, less for control
  - Vector/SIMD/streaming ("accelerator" technologies)
  - Explicit control of storage and data movement
- Need to hide the complexity of this from the users
  - Compiler and Runtime need to manage node parallelism
  - Better tools and programming models
- Need to solve the application resiliency problem, or high end machines will be useless in a decade

Copyright 2008, Cray Inc.

What about the Zettascale? (20 years out?)

- CMOS will *never* get there (due to power, density, defects, voltage scaling, etc.)



- Computational power has increased exponentially across multiple technology transitions
- What's next?
  - Carbon nanotubes?
  - Quantum? Bio? Optical?
  - May need *reversible* logic to overcome energy loss
- Will *not* look like today's architecture
- Will almost certainly involve new hw/sw execution model
- Need to understand technology before understanding architecture

Copyright 2008, Cray Inc.

Thank you.

Questions?

Copyright 2008, Cray Inc.