Ultra-Efficient Exascale Scientific Computing

Lenny Oliker, John Shalf, Michael Wehner

And other LBNL staff

Exascale is Critical to the DOE SC Mission

"…exascale computing (will) *revolutionize our approaches to global challenges in energy, environmental sustainability, and security.*"

E3 report

Green Flash: Ultra-Efficient Climate Modeling

- We present an alternative route to exascale computing
  - DOE SC exascale science questions are already identified.
  - Our idea is to target specific machine designs to each of these questions.
    - This is 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?"

Green Flash: Ultra-Efficient Climate Modeling

- We present an alternative route to exascale computing
  - DOE SC exascale science questions are already identified.
  - Our idea is to target specific machine designs to each of these questions.
    - This is 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?"
- Caveat:
  - We present here a feasibility design study.
  - Goal is to influence the HPC industry by evaluating a prototype design.

Global Cloud System Resolving Climate Modeling

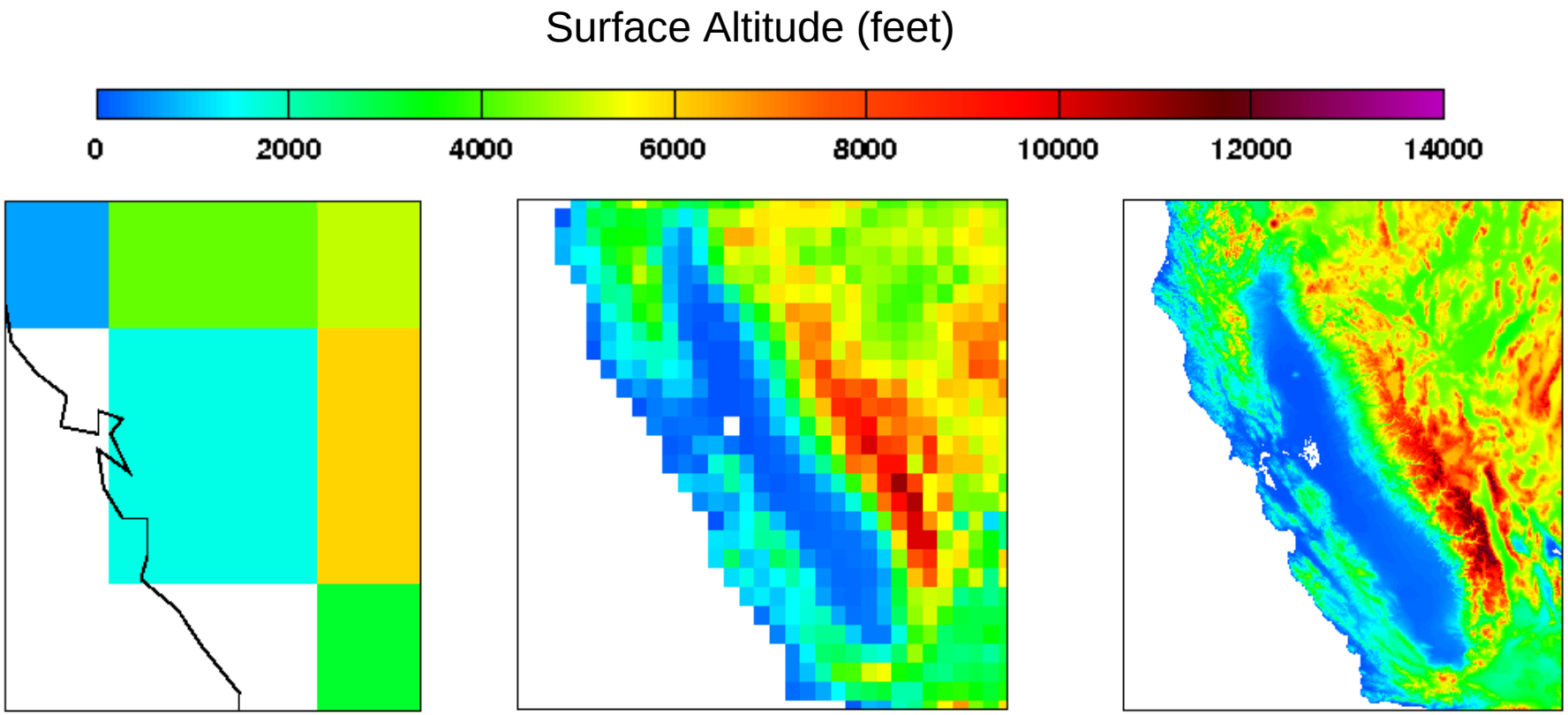
Individual cloud physics fairly well understood

Parameterization of mesoscale cloud statistics performs poorly.

Direct simulation of cloud systems in global models requires exascale!

- Direct simulation of cloud systems replacing statistical parameterization.
  - This approach recently was called for by the 1st WMO Modeling Summit.
- Championed by Prof. Dave Randall, Colorado State University

Global Cloud System Resolving Models are a Transformational Change



200km
Typical resolution of IPCC AR4 models

25km
Upper limit of climate models with cloud parameterizations

1km
Cloud system resolving models

1km-Scale Global Climate Model Requirements

Simulate climate 1000x faster than real time

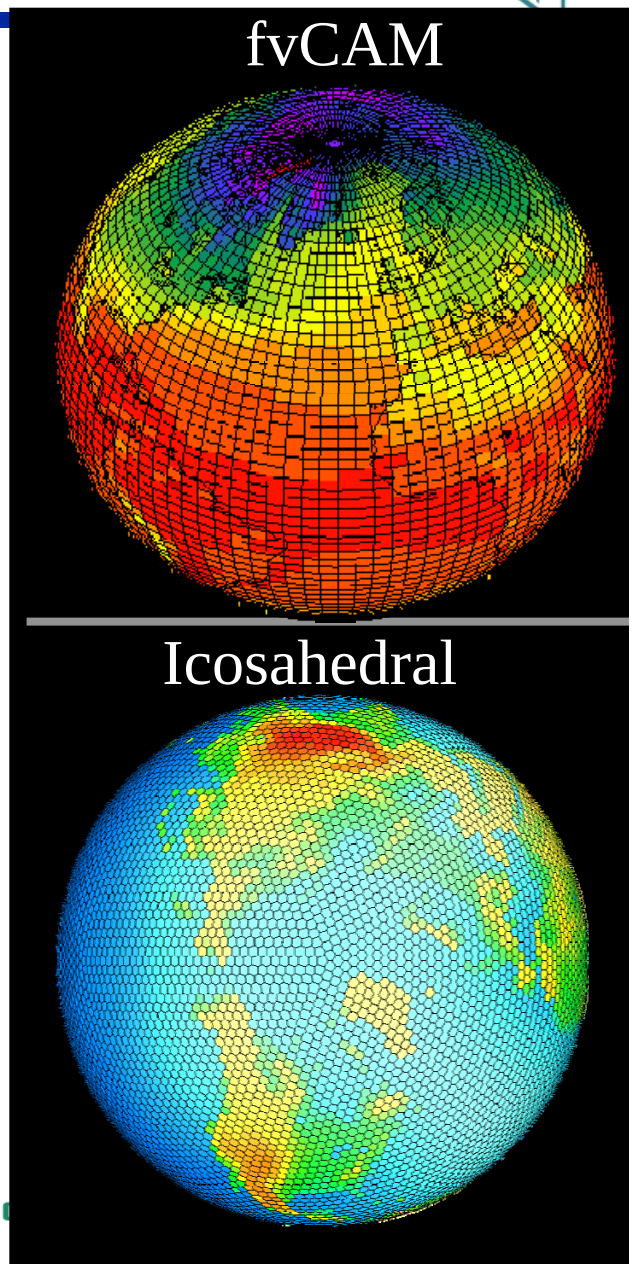
10 Petaflops sustained per simulation
(~200 Pflops peak)

10-100 simulations (~20 Exaflops peak)

Truly exascale!

Some specs:

- **Advanced dynamics algorithms: icosahedral, cubed sphere, reduced mesh, etc.**
- **~20 billion cells → Massive parallelism**
- **100 Terabytes of Memory**
- **Can be decomposed into ~20 million total subdomains**



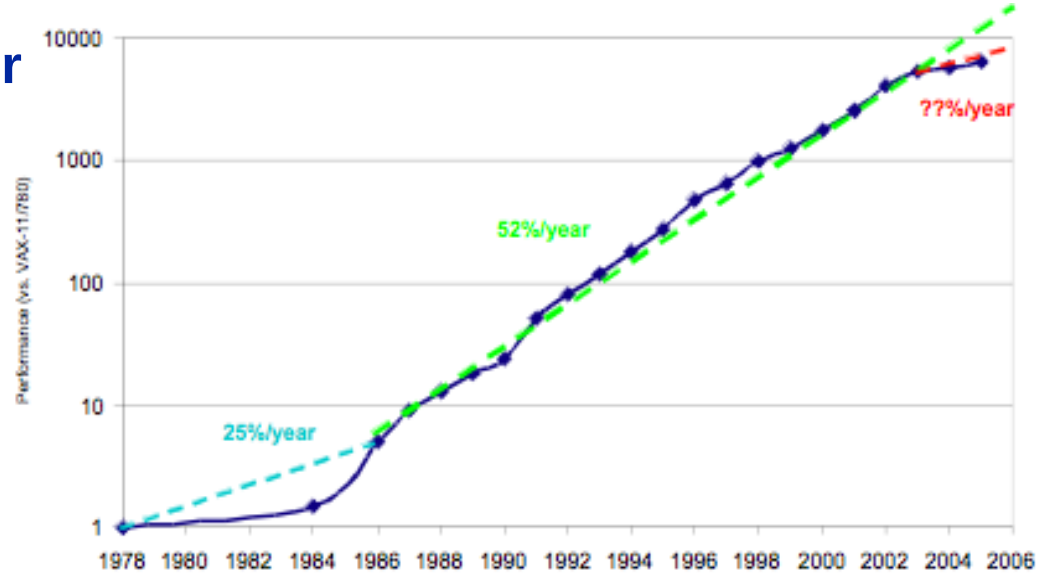
Proposed
Ultra-Efficient Computing

- **Cooperative "science-driven system architecture" approach**

- **Radically change HPC system development via application-driven hardware/software co-design**
  - *Achieve 100x power efficiency over mainstream HPC approach for targeted high impact applications, at significantly lower cost*
  - **Accelerate development cycle for exascale HPC systems**
  - **Approach is applicable to numerous scientific areas in the DOE Office of Science**

- **Research activity to understand feasibility of our approach**

Primary Design Constraint: *POWER*



- **Transistors still getting smaller**
  - **Moore's Law: alive and well**
- **Power efficiency and clock rates no longer improving at historical rates**
- **Demand for supercomputing capability is accelerating**
- **E3 report considered an Exaflop system for 2016**
- **Power estimates for exascale systems based on extrapolation of current design trends range up to 179MW**
  - DOE E3 Report 2008
  - DARPA Exascale Report *(in production)*
  - LBNL IJHPCA Climate Simulator Study 2008 (Wehner, Oliker, Shalf)

Need fundamentally new approach to computing designs

Our Approach

- **Identify high-impact Exascale scientific applications important to DOE Office of Science (E3 report)**

- **Tailor system to requirements of target scientific problem**
  - Use design principles from embedded computing
  - Leverage commodity components in novel ways - not full custom design

- **Tightly couple hardware/software/science development**
  - Simulate hardware before you build it (RAMP)
  - Use applications for validation, not kernels
  - Automate software tuning process (Auto-Tuning)

Path to Power Efficiency

Reducing Waste in Computing

- **Examine methodology of embedded computing market**
  - **Optimized for low power, low cost, and high computational efficiency**

*"Years of research in low-power embedded computing have shown only one design technique to reduce power: **reduce waste.**"*

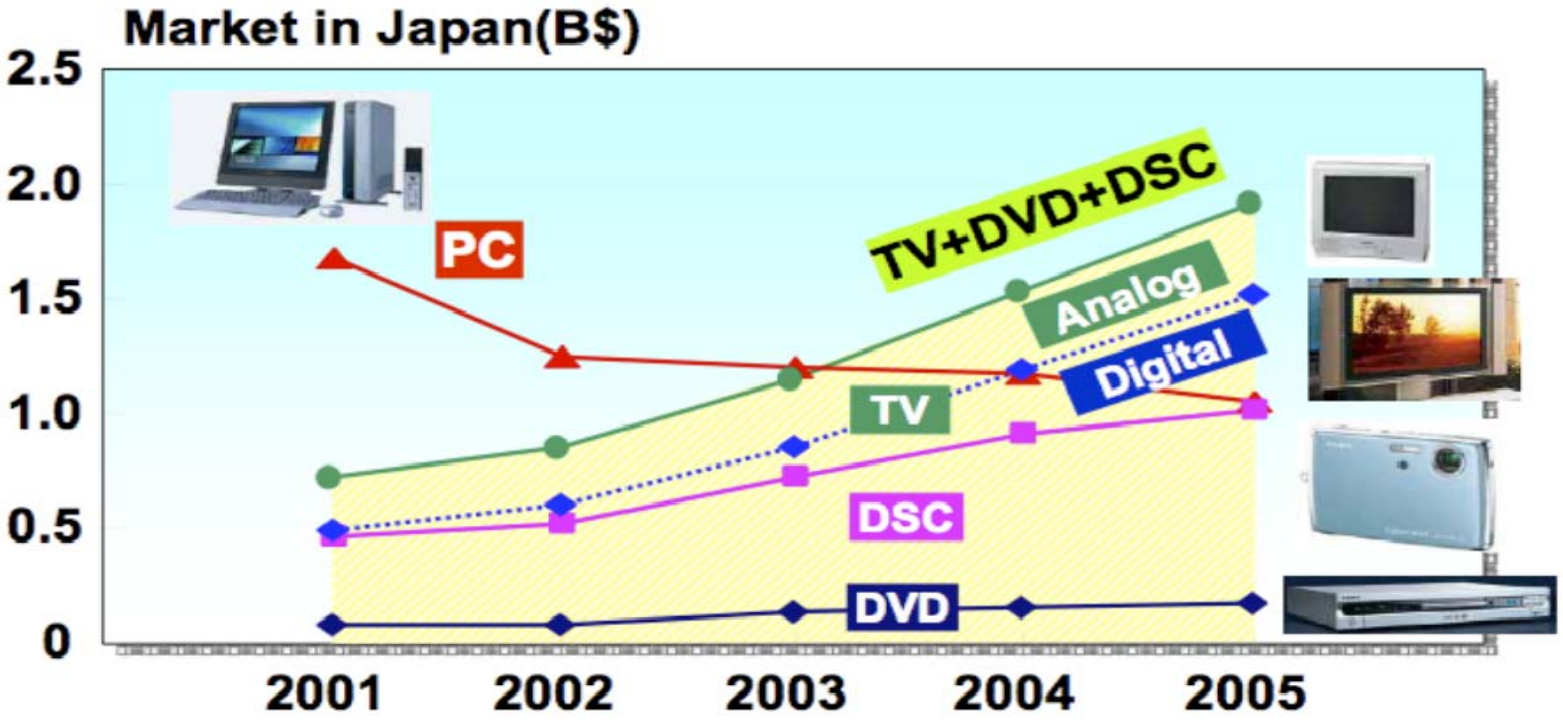
—Mark Horowitz, Stanford University & Rambus Inc.

- **Sources of Waste**
  - **Wasted transistors (surface area)**
  - **Wasted computation (useless work/speculation/stalls)**
  - **Wasted bandwidth (data movement)**
  - **Designing for serial performance**

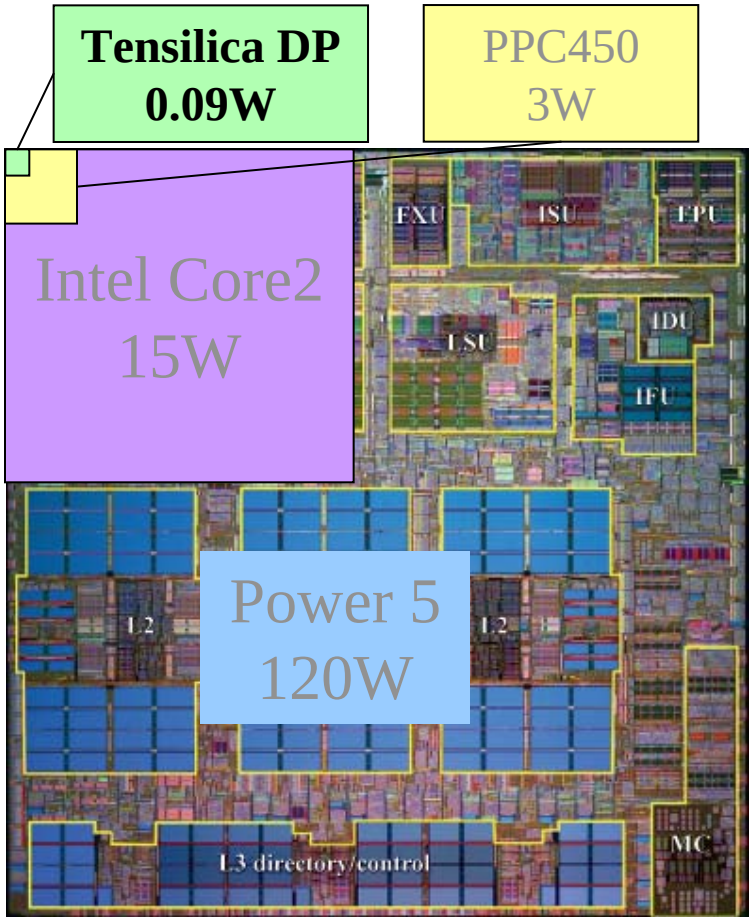
- **Technology now favors parallel throughput over peak sequential performance**

Processor Technology Trend

- **1990s - R&D computing hardware dominated by desktop/COTS**
  - Had to learn how to use COTS technology for HPC
- **2010 - R&D investments moving rapidly to consumer electronics/ embedded processing**
  - Must learn how to leverage embedded processor technology for future HPC systems



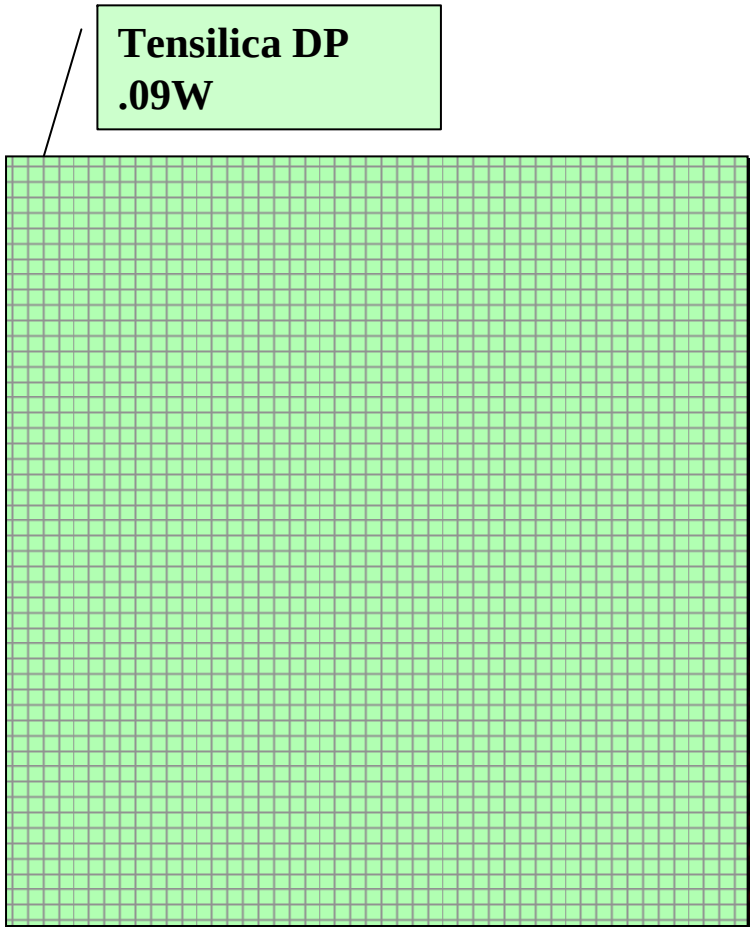
Design for Low Power: More Concurrency



- 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 reduce waste☐☐

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

Low Power Design Principles

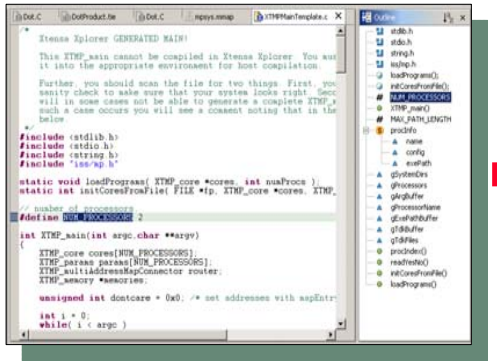


- IBM Power5 (server)
  - 120W@1900MHz
  - Baseline
- Intel Core2 sc (laptop) :
  - 15W@1000MHz
  - *4x more FLOPs/watt than baseline*
- IBM PPC 450 (BG/P - low power)
  - 0.625W@800MHz
  - 90x more
- Tensilica XTensa (Moto Razor) :
  - 0.09W@600MHz
  - 400x more

Even if each core operates at 1/3 to 1/10th efficiency of largest chip, you can pack 100s more cores onto a chip and consume 1/20 the power

Embedded Design Automation

(Example from Existing Tensilica Design Flow)



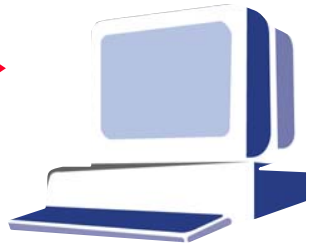
Processor configuration

1. Select from menu
2. Automatic instruction discovery (XPRES Compiler)
3. Explicit instruction description (TIE)

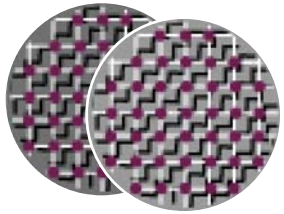
Processor Generator (*Tensilica*)

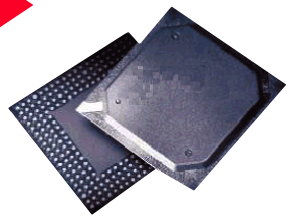
Application-optimized processor implementation (RTL/Verilog)

Base CPU		OCD
Apps Datapaths	Cache	Timer
Extended Registers		FPU



Tailored SW Tools: Compiler, debugger, simulators, Linux, other OS Ports *(Automatically generated together with the Core)*





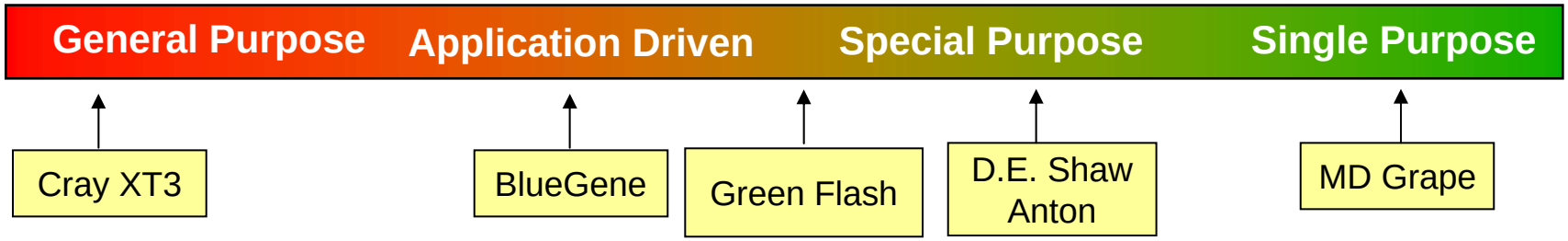
Build with any process in any fab

Advanced Hardware Simulation (RAMP)

- **Research Accelerator for Multi-Processors (RAMP)**
  - **Utilize FGPA boards to emulate large-scale multicore systems**
  - **Simulate hardware before it is built**
  - **Break slow feedback loop for system designs**
  - **Allows fast performance validation**
  - **Enables tightly coupled hardware/software/science co-design** *(not possible using conventional approach)*

- **Technology partners:**
  - **UC Berkeley: John Wawrzynek, Jim Demmel, Krste Asanovic, Kurt Keutzer**
  - **Stanford University / Rambus Inc.: Mark Horowitz**
  - **Tensilica Inc.: Chris Rowen**

Customization Continuum: Green Flash



- Application-driven does NOT necessitate a special purpose machine
- MD-Grape: Full custom ASIC design
  - 1 Petaflop performance for one application using 260 kW for $9M
- D.E. Shaw Anton System: Full and Semi-custom design
  - Simulate 100x–1000x timescales vs any existing HPC system (~200kW)
- Application-Driven Architecture (Green Flash): Semicustom design
  - Highly programmable core architecture using C/C++/Fortran
  - Goal of 100x power efficiency improvement vs general HPC approach
  - Better understand how to build/buy application-driven systems
  - **Potential:** 1km-scale model (~200 Petaflops peak) running in O(5 years)

Green Flash Strawman System Design

We examined three different approaches (in 2008 technology)

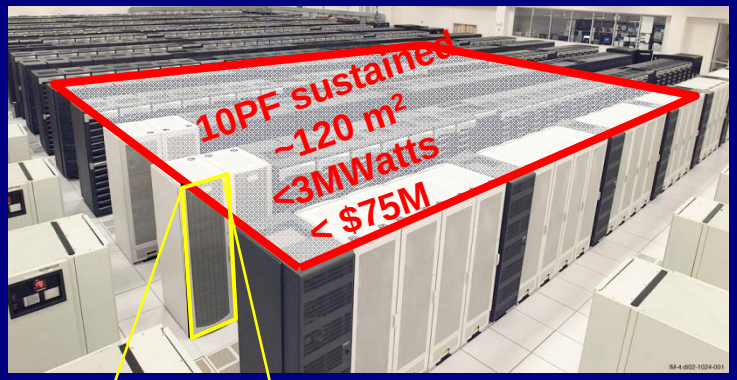
Computation .015°X.02°X100L: 10 PFlops sustained, ~200 PFlops peak

- **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

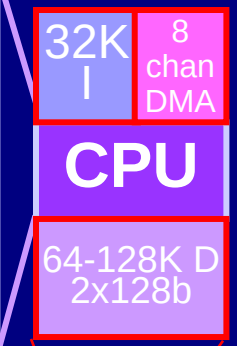
Climate System Design Concept

Strawman Design Study

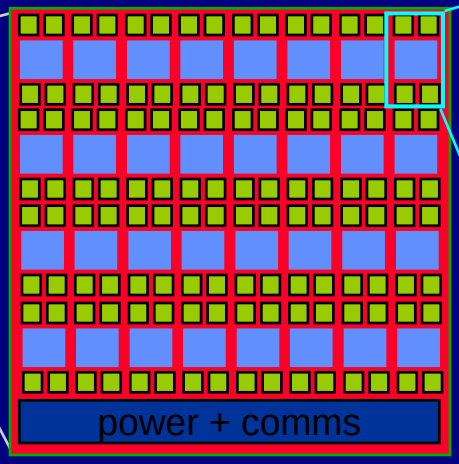


VLIW CPU:

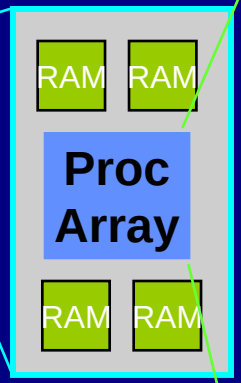
- 128b load-store + 2 DP MUL/ADD + integer op/ DMA per cycle:
- Synthesizable at 650MHz in commodity 65nm
- 1mm² core, 1.8-2.8mm² with inst cache, data cache data RAM, DMA interface, 0.25mW/MHz
- Double precision SIMD FP : 4 ops/cycle (2.7GFLOPs)
- Vectorizing compiler, cycle-accurate simulator, debugger GUI (Existing part of Tensilica Tool Set)
- 8 channel DMA for streaming from on/off chip DRAM
- Nearest neighbor 2D communications grid



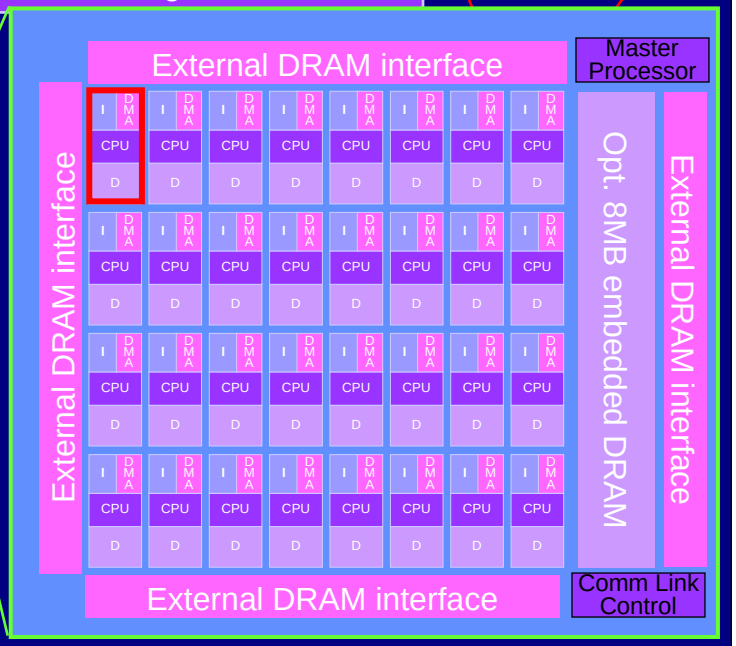
100 racks @ ~25KW



32 chip + memory clusters per board (2.7 TFLOPS @ 700W



8 DRAM per processor chip: ~50 GB/s



32 processors per 65nm chip
83 GFLOPS @ 7W

Portable Performance for Green Flash

- **Challenge: Our approach would produce multiple architectures, each different in the details**
  - Labor-intensive user optimizations for each specific architecture
  - Different architectural solutions require vastly different optimizations
  - Non-obvious interactions between optimizations & HW yield best results

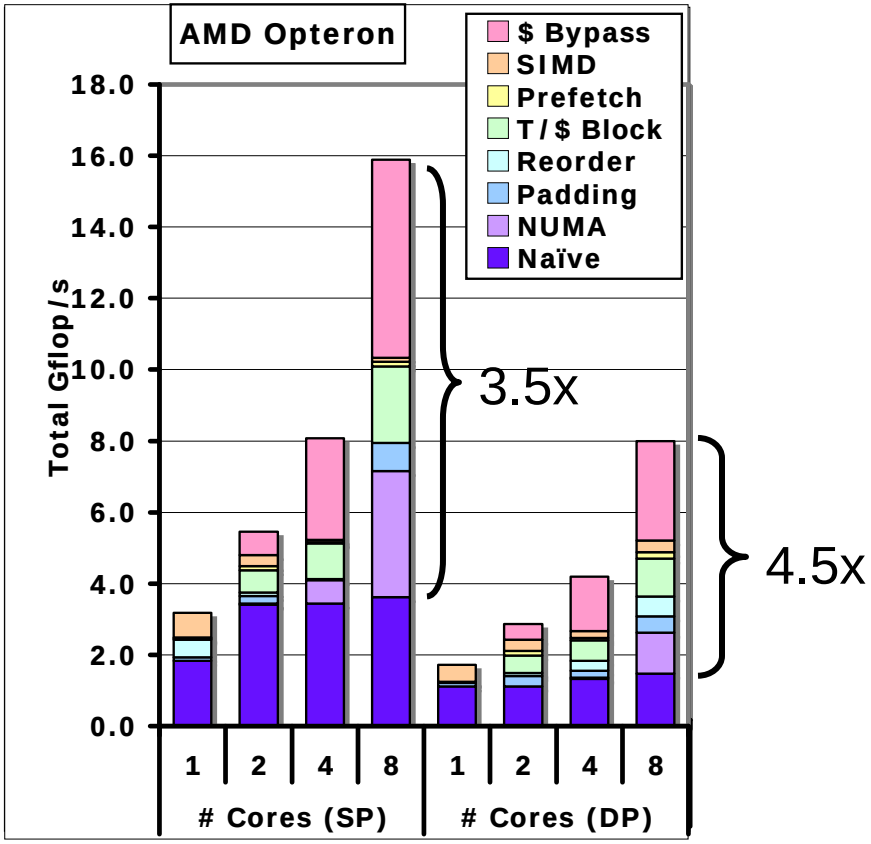
- **Our solution: Auto-tuning**
  - **Automate search across a complex optimization space**
  - **Achieve performance far beyond current compilers**
  - **Attain performance portability for diverse architectures**

Auto-Tuning for Multicore

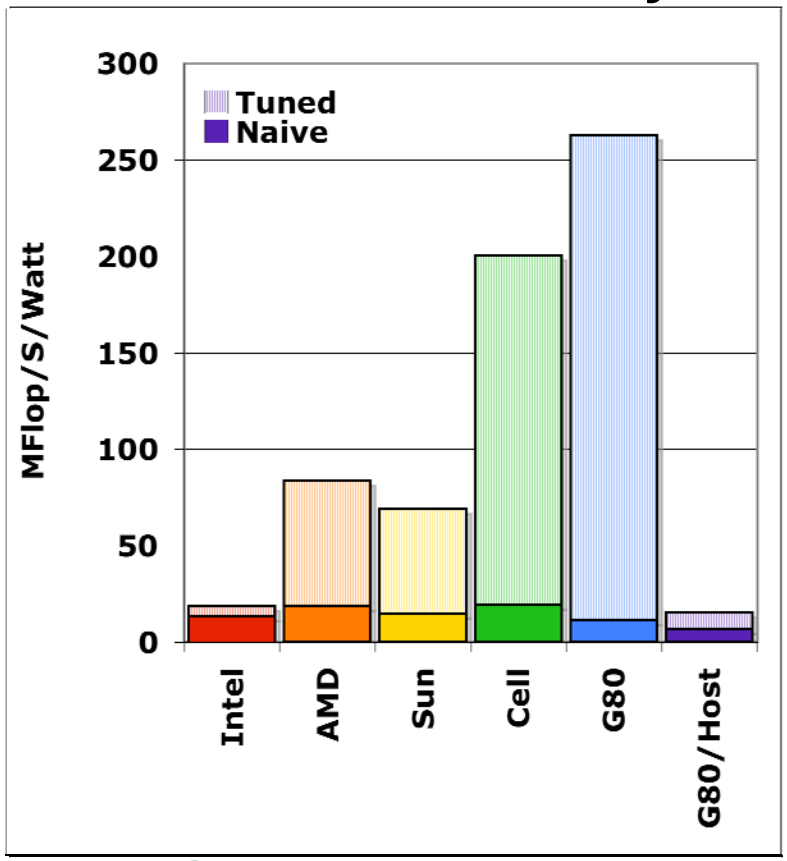
(finite-difference computation)

- Take advantage of unique multicore features via auto-tuning
- Attains performance portability across different designs
- Only requires basic compiling technology
- Achieve high serial performance, scalability, and optimized power efficiency

Performance Scaling



Power Efficiency

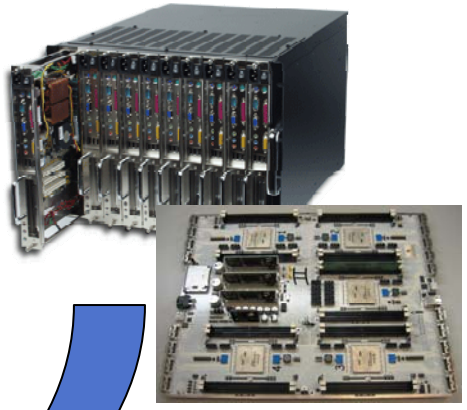


Traditional New Architecture Hardware/Software Design

How long does it take for a full scale application to influence architectures?

Design New System (2 year concept phase)

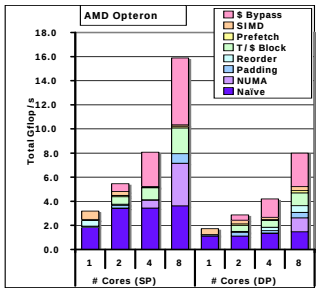
Build Hardware (2 years)



Cycle Time 4-6+ years

Port Application

Tune Software (2 years)

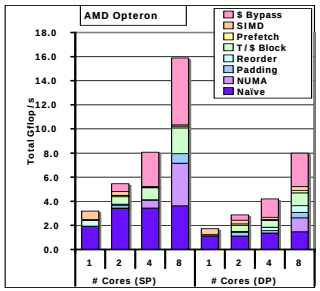


Proposed New Architecture Hardware/Software Co-Design

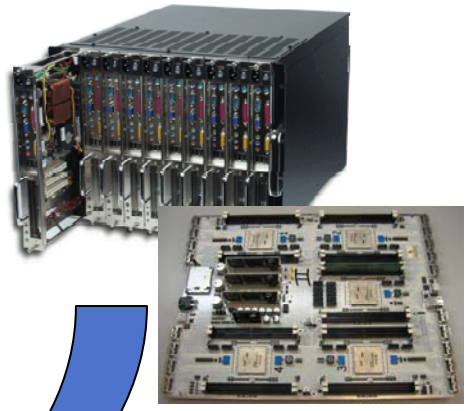
How long does it take for a full scale application to influence architectures?

Synthesize SoC (hours)

Autotune Software (Hours)



Cycle Time 1-2 days



Emulate Hardware (RAMP) (hours)

Build application

Summary

- **Exascale computing is vital to the DOE SC mission**
- **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 DOE applications