(Some) Answers to the Challenges of Petascale Computing

Rich Loft

Director, Technology Development

Computational and Information Systems Laboratory

National Center for Atmospheric Research

loft@ucar.edu

Computational & Information Systems Laboratory

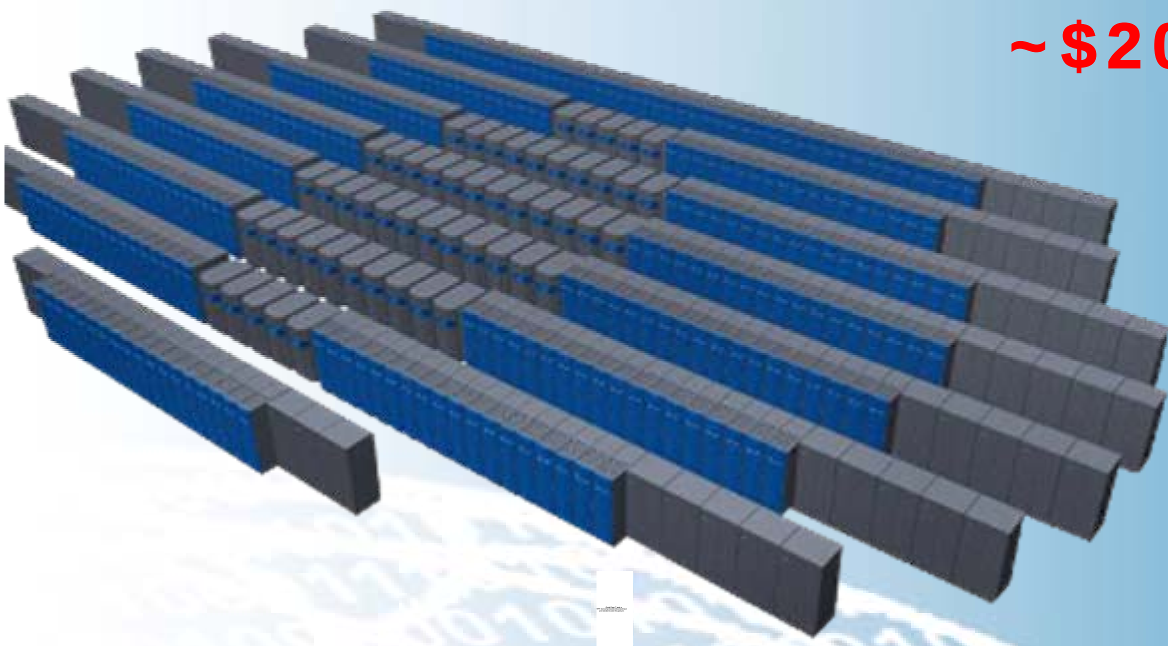
CISL

NCAR

A Petaflops Sustained System in 2011… will be big by any measure

10-20 MW

~$200 M



O(10^5 - 10^6) CPU's

Computational & Information Systems Laboratory

CISL

NCAR

The Petascale Challenges are Huge:

- **Petascale Computing Issues**
  - Stalled thread speeds
  - The challenge of parallelism (Amdahl)
  - Algorithmic scalability
- **Software Complexity Issues**
  - Interdisciplinarity of Earth Sciences
  - Increasingly Complex Models
- **Data Issues**
  - Data volumes that break tools
  - Complex workflows
- **People Issues**
  - Entraining them
  - Training them

Q. If you had a petascale computer what would you do with it?

A. Use it as a prototype of an exascale computer.

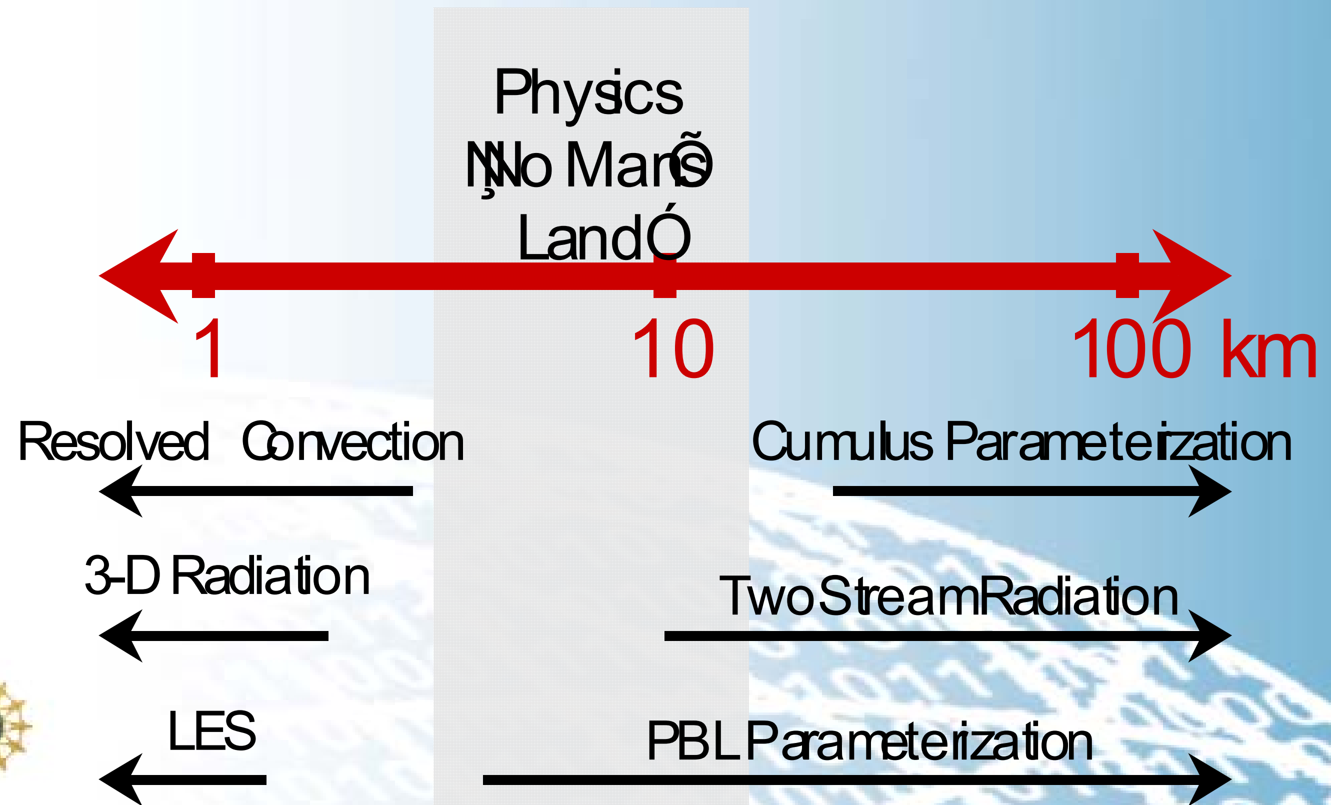
We know where we really want to go with Earth System Modeling: Unfortunately it is the exascale

Convective Scales in the Atmosphere are Tiny: basically O(1 km)

Modeling Trade-offs: Directly Resolving Convection is an Exascale (10^{18} FLOPS/s) Problem

Computational & Information Systems Laboratory

Challenges in High Resolution Numerical Weather Prediction



NCAR

The Exascale Earth System Model Vision

Coupled Ocean-Land-Atmosphere Model

~10 km x ~10 km (eddy-resolving)

100 levels

Unstructured, adaptive grids

~100 m

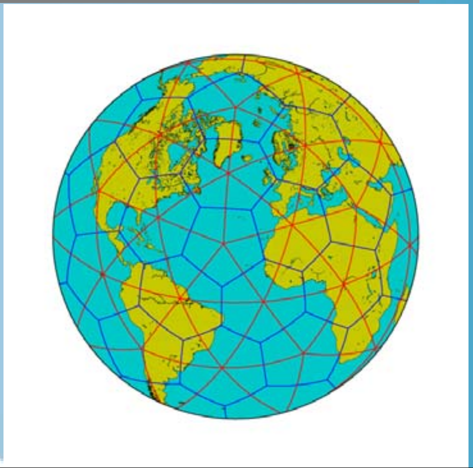
10 levels

Landscape-resolving

~1 km x ~1 km (cloud-resolving)

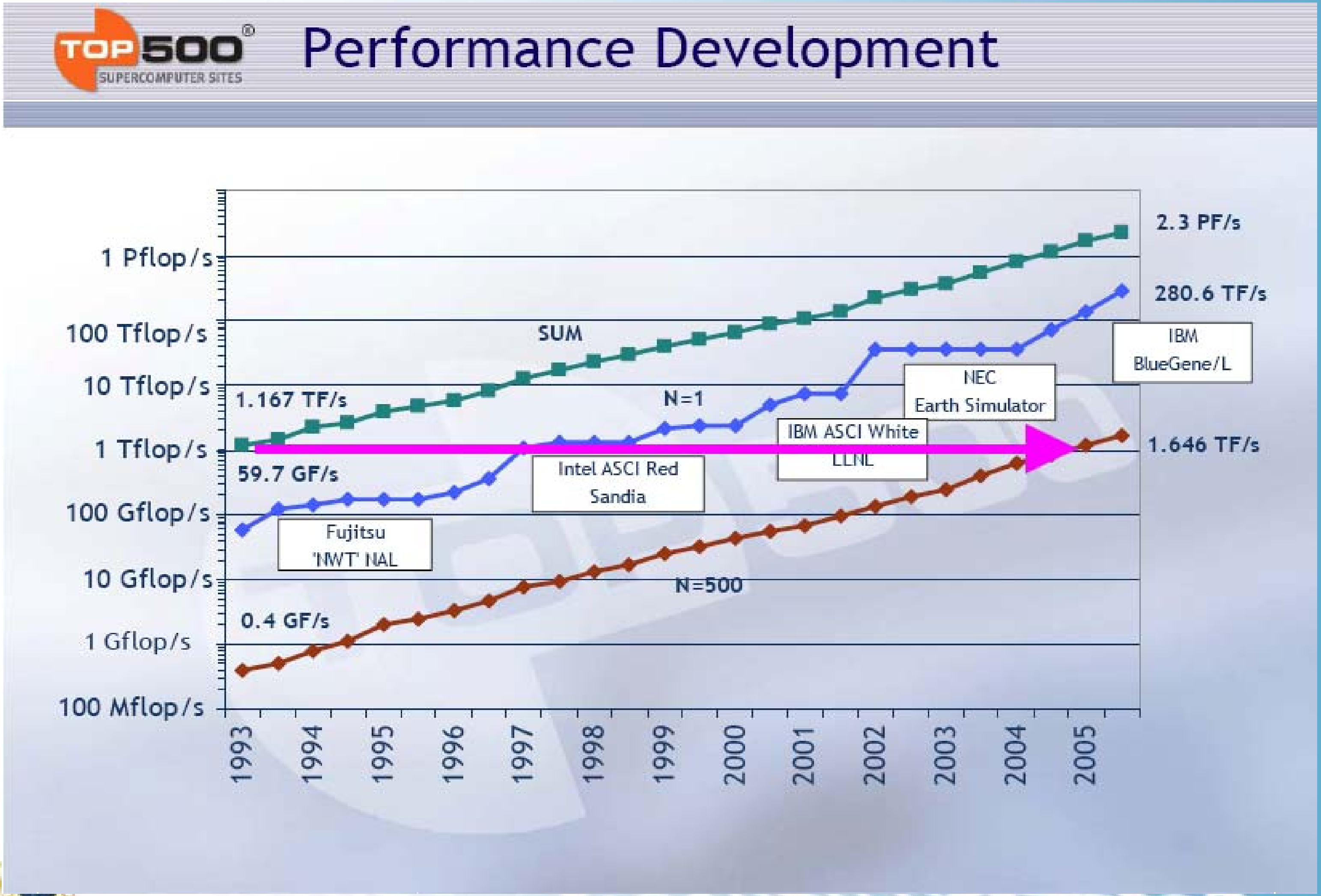
100 levels, whole atmosphere

Unstructured, adaptive grids



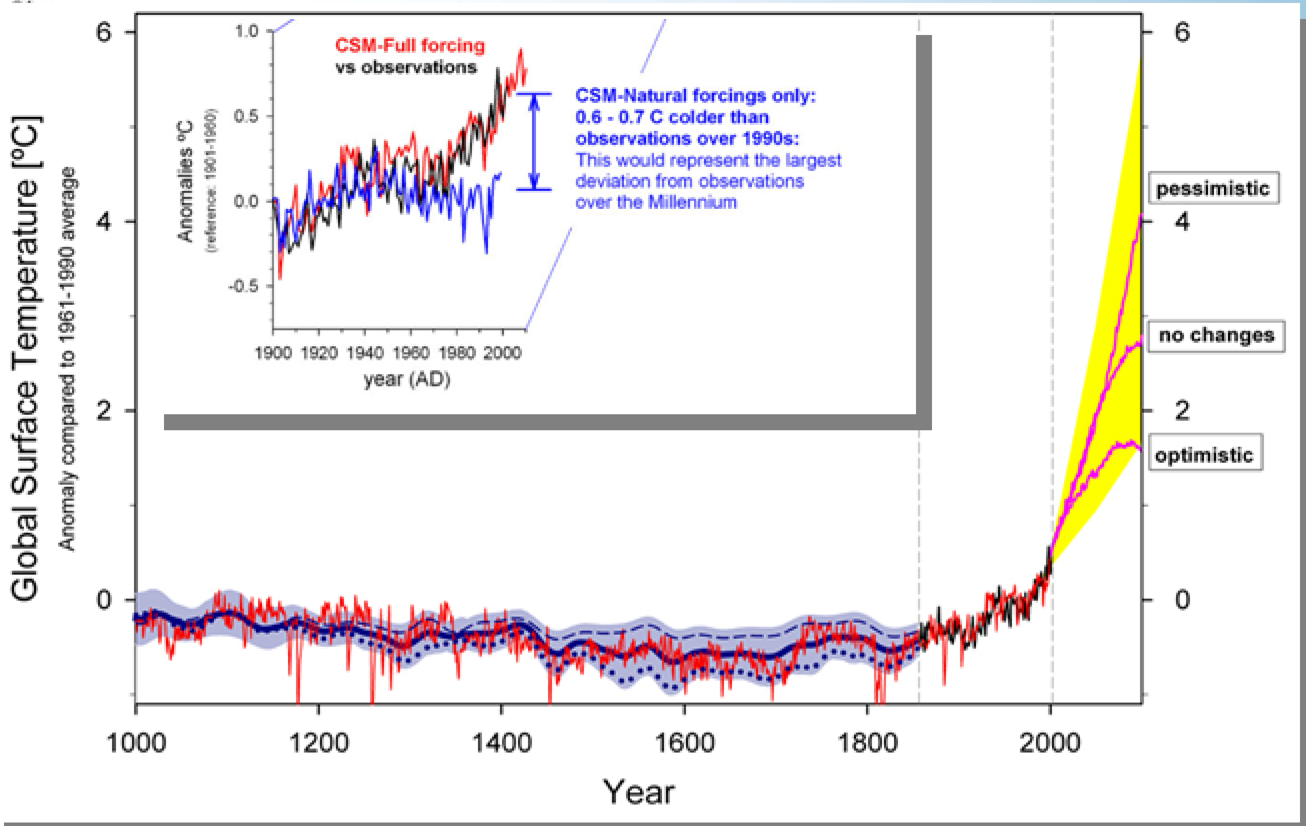
Requirement: Computing power enhancement by a factor of 10^4-10^6

Moore's Law is not fast enough!



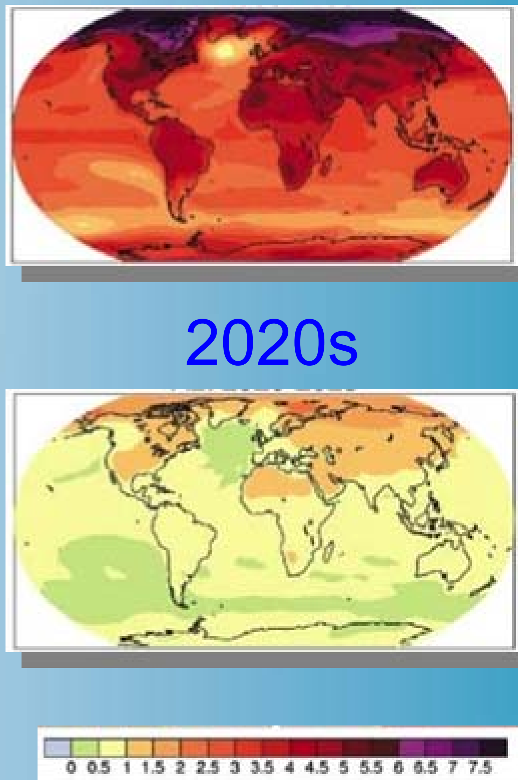
…suggests 10^4 to 10^6 improvement will take 20 years

Meanwhile the climate is changing…



Ammann et al., 2007

A2: 2090s



IPCC, 2007

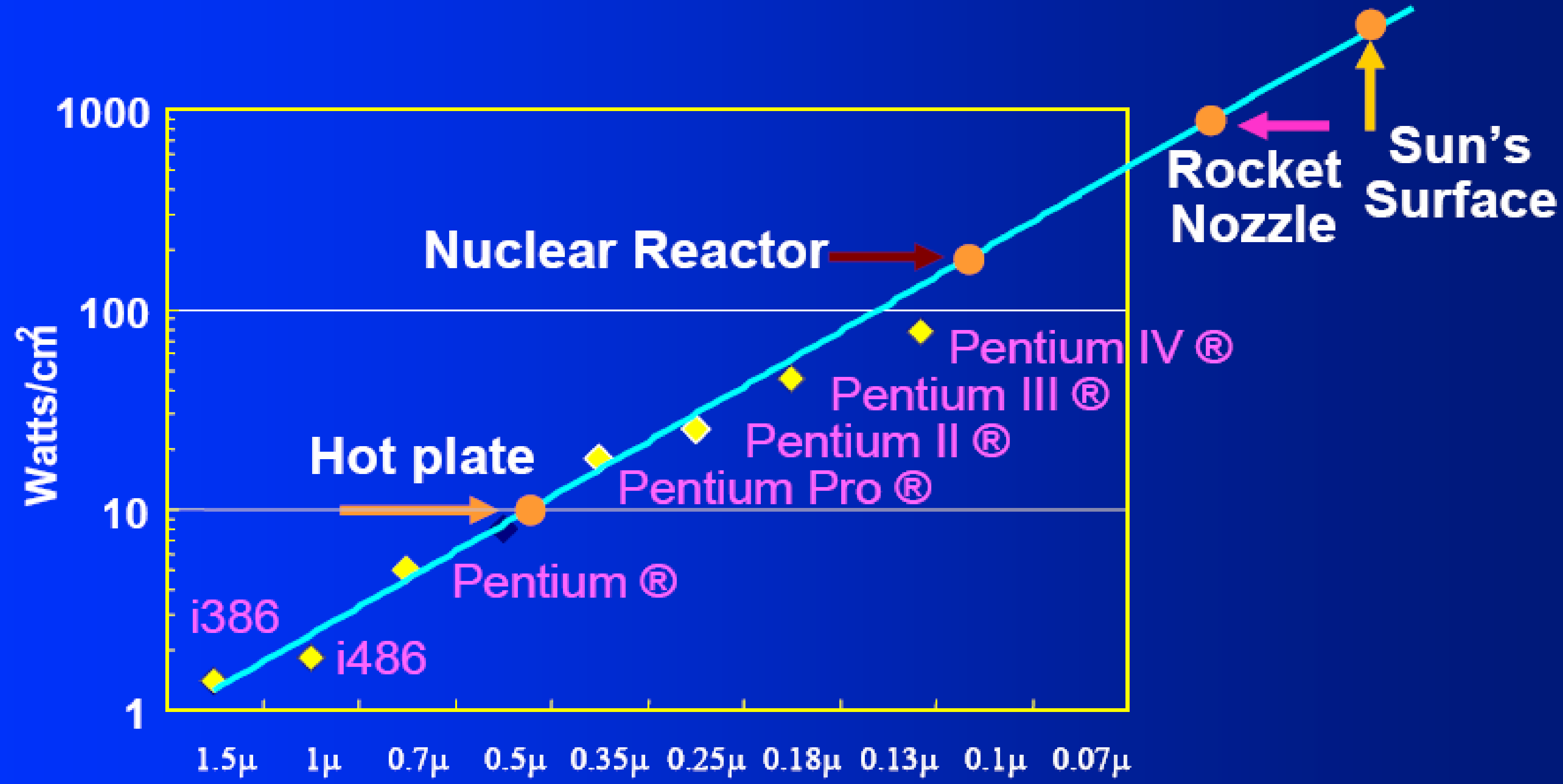
NCAR

And the ice is melting…

QuickTime™ and a
YUV420 codec decompressor
are needed to see this picture.

In addition, there’s been an underlying paradigm shift in how progress in HPC technology is occurring

Relentless rise of power density

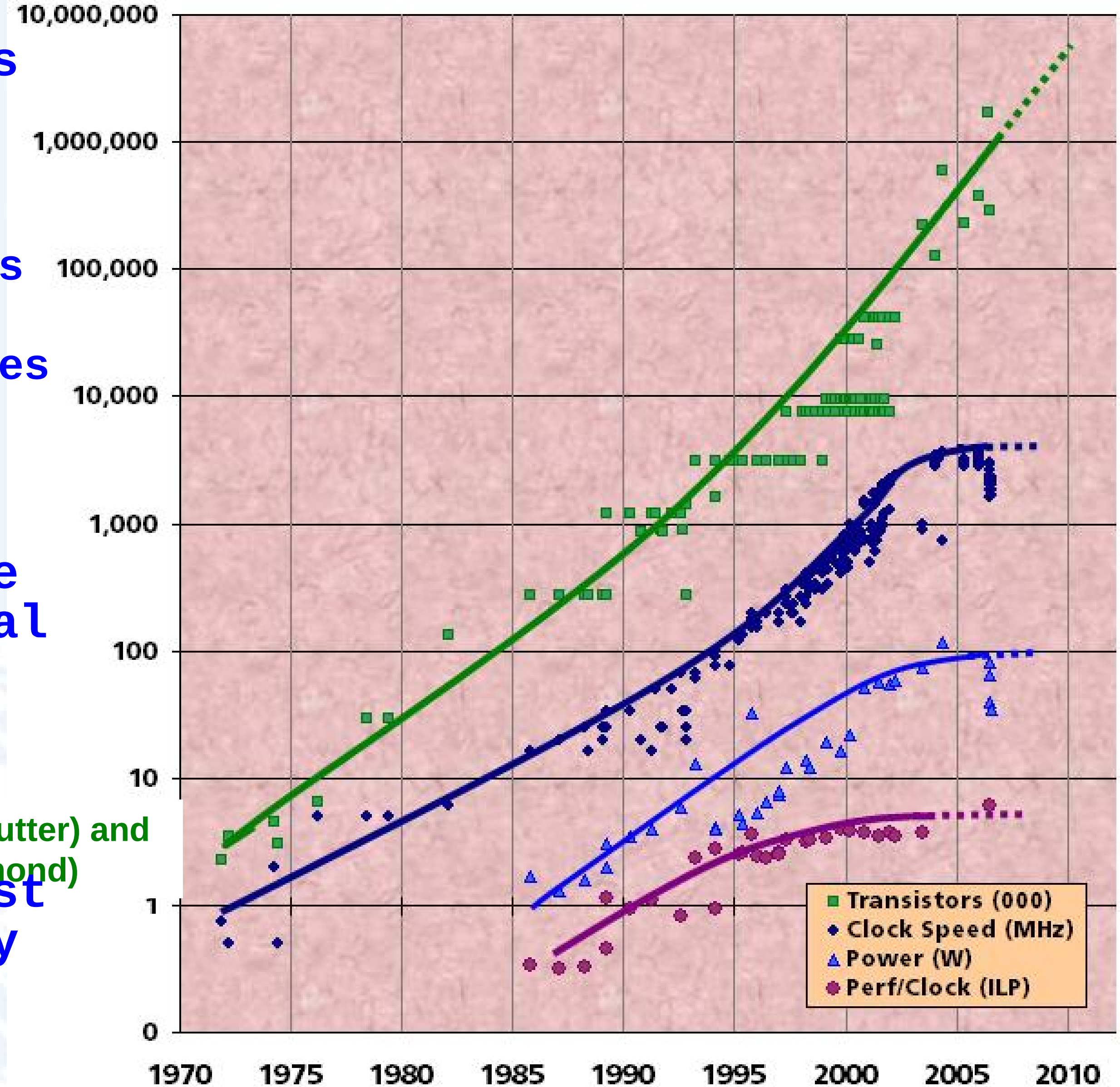


- 80% increase in power density/generation
- Voltage scales by ~0.8
- 225% increase in current consumption/unit area !

Source: Shekhar Borkar, Intel

Chip Level Trends

- Chip density is continuing increase ~2x every 2 years
  - Clock speed is not
  - Number of cores are doubling instead
- There is little or no additional hidden parallelism (ILP)
- Parallelism must be exploited by software



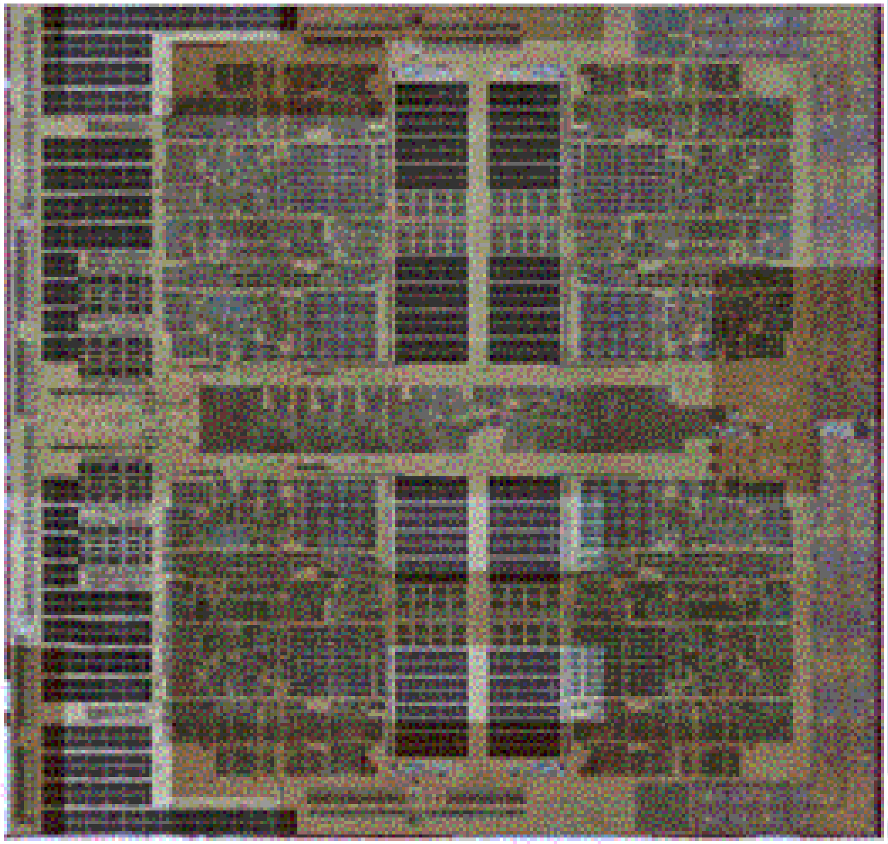
Source: Intel, Microsoft (Sutter) and Stanford (Olukotun, Hammond)

Computational & Information Systems Laboratory

CISL

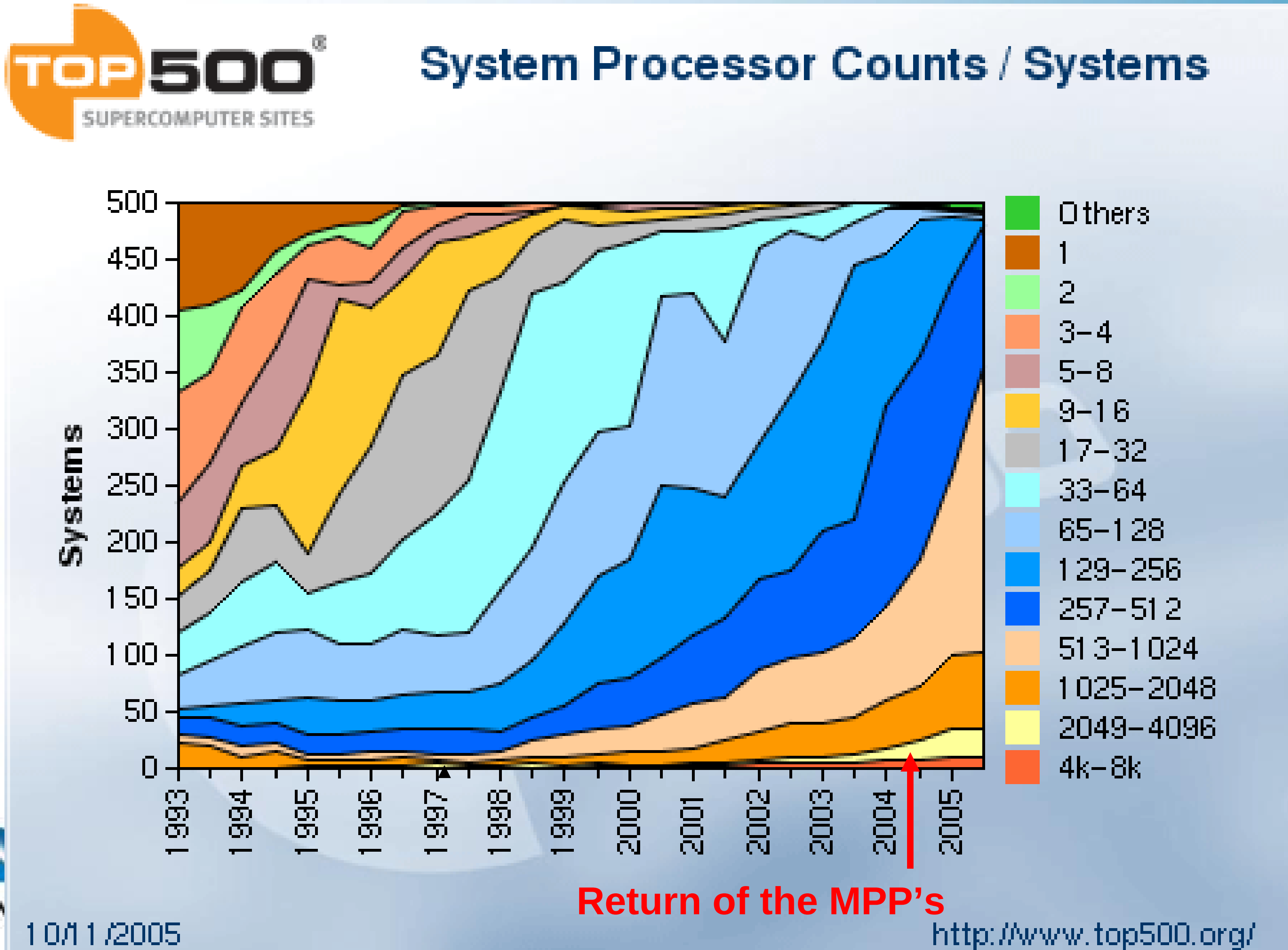
NCAR

Moore's Law = More Cores: Quad Core "Barcelona" AMD Processor…



Can 8, 16, 32 cores be far behind?

The history of parallelism in supercomputing…

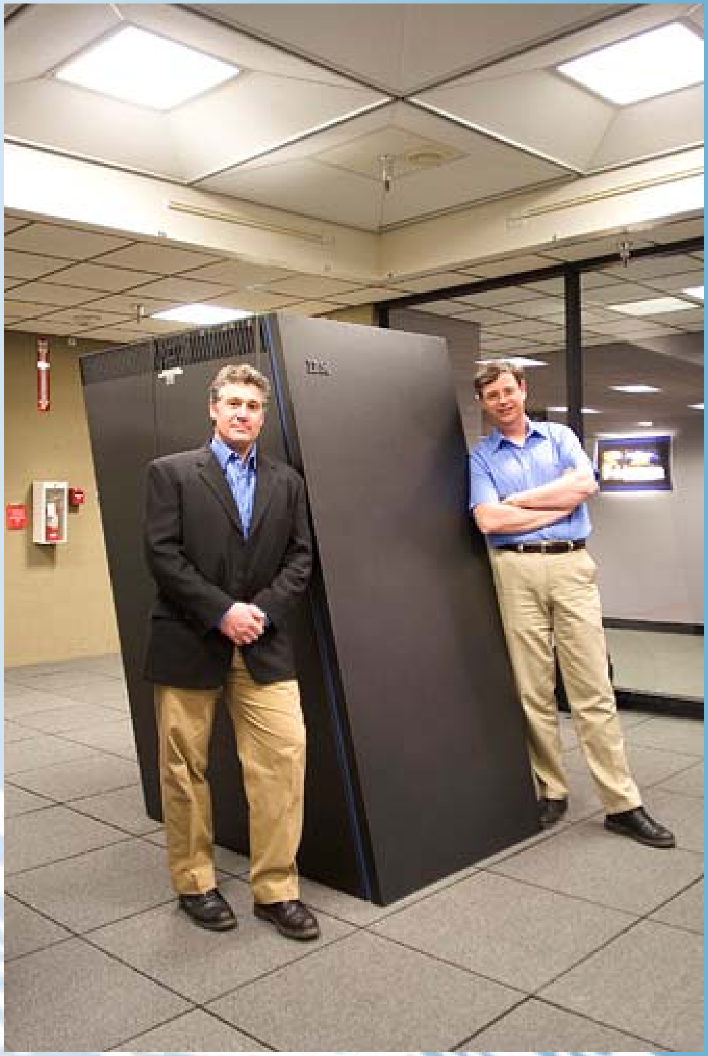


Computational & Information Systems Laboratory

NCAR and University Colorado Partner to Experiment with Blue Gene/L

Characteristics:

- 2048 Processors/5.7 TF
- PPC 440 (750 MHz)
- Two processors/node
- 512 MB memory per node
- 6 TB file system



Dr. Henry Tufo and myself with "frost" (2005)

Computational & Information Systems Laboratory

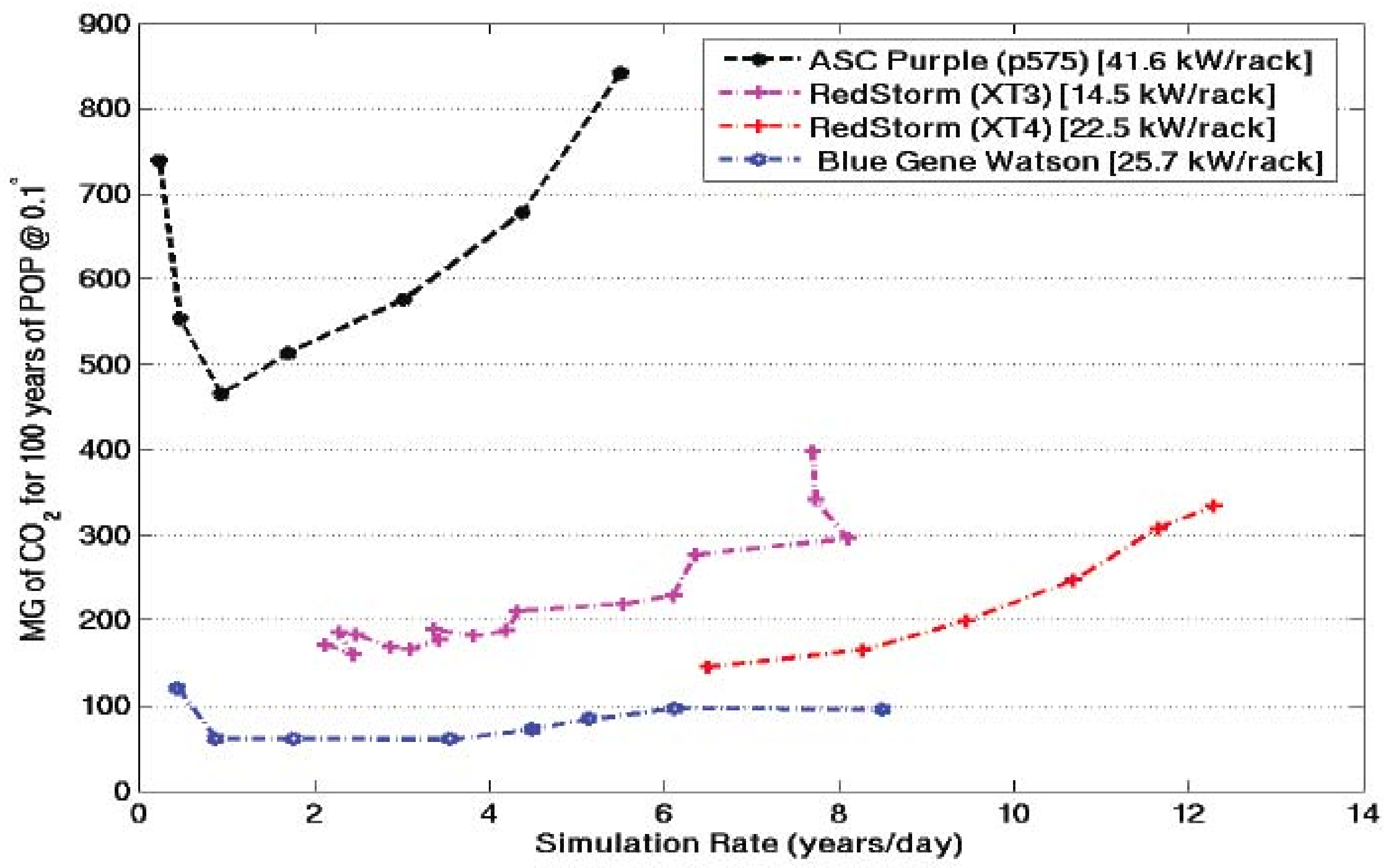
CISL

NCAR

Petascale System Design Issues: Performance Means Heat

- However, achievable performance has been increasingly gated by the memory hierarchy performance not CPU peak
  - Peak is basically a poor predictor of application performance
- Aggregate memory bandwidth =
  - Signaling rate/pin x pins/socket x sockets
- To increase aggregate bandwidth you can increase
  - signaling rate - fundamental technology issue
  - pins/socket - packaging technology
  - sockets - more communications
- Consequences
  - More heat
  - Higher heat density
  - More heat from the interconnect
- System power requirements
  - Track-1 O(10 MW)
  - Mid next-decade exascale system- O(100 MW)

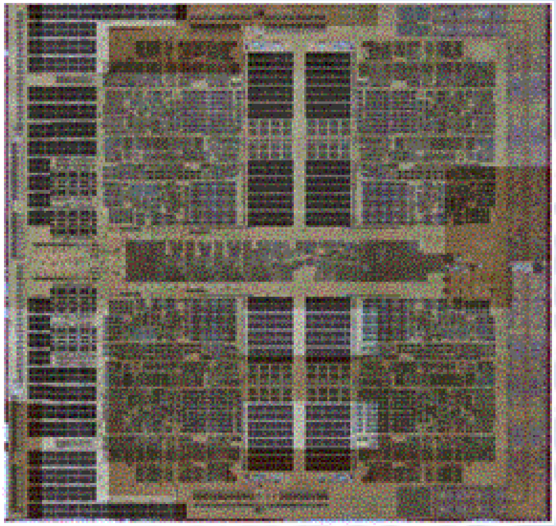
Not all systems have the same carbon footprint



Getting applications to scale is green!

A Thought Experiment

- The road we're on says we'll get:
  - 2x CPU's every 18 months
  - But stagnant thread speed
- Suppose these idealized conditions exist:
  - Perfectly scalable system
  - Its infinite extensibility (for a price)



Computational & Information Systems Laboratory

CISL

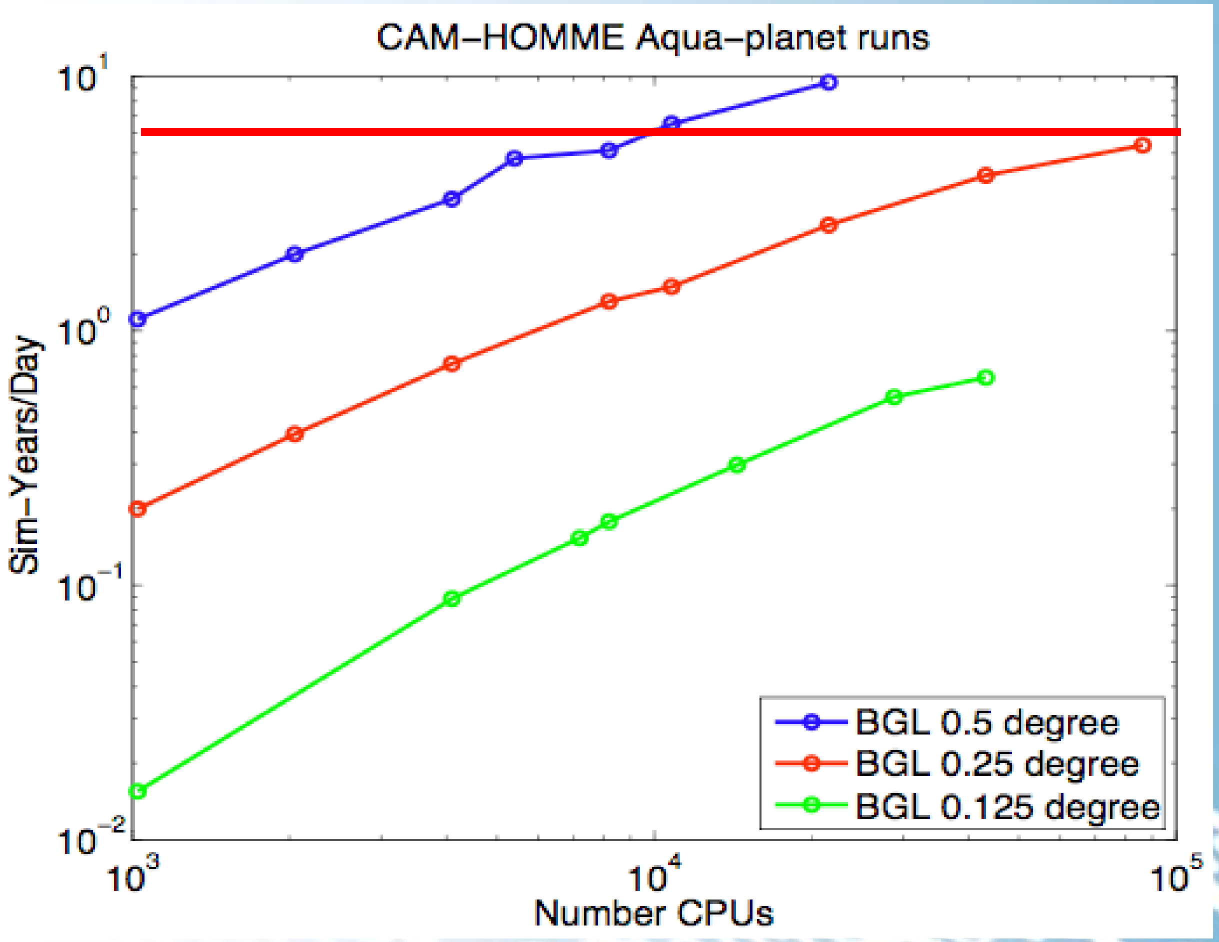
NCAR

Merciless Effects of CFL

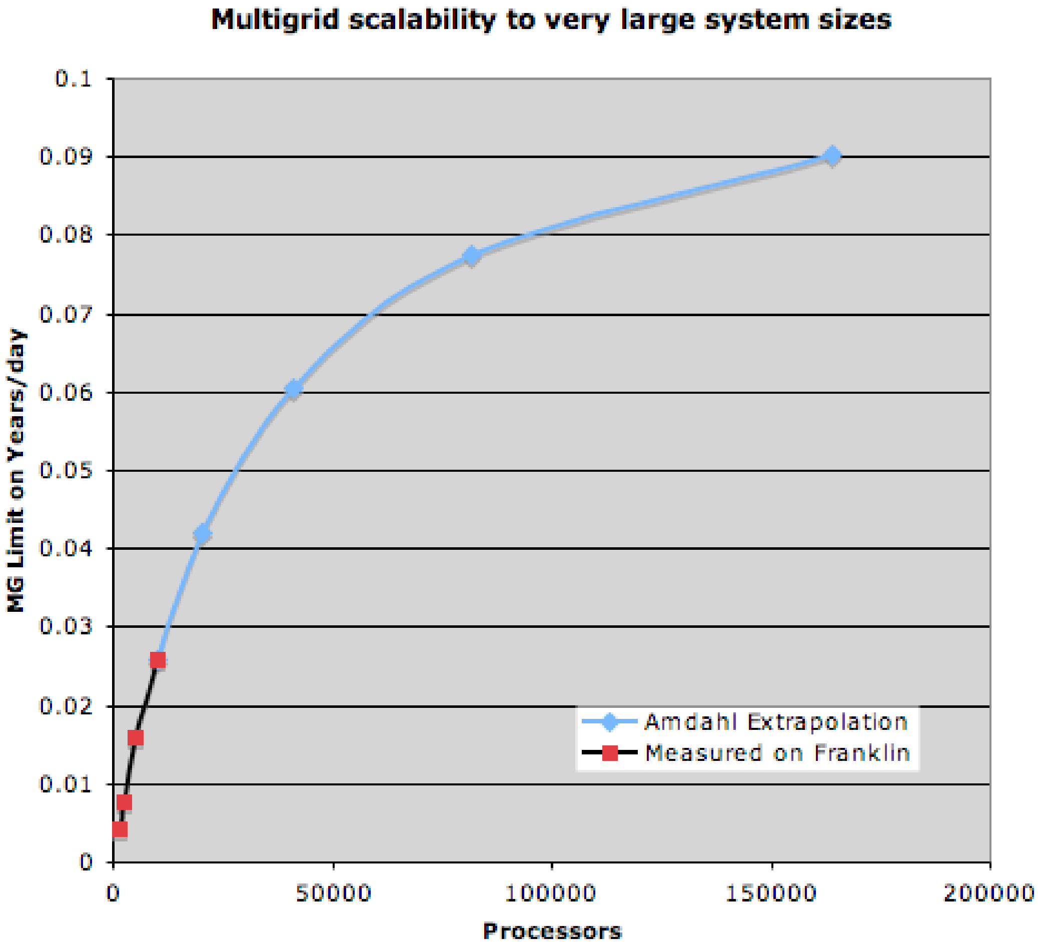
- Dynamics timestep goes like N^{-1}
  - The cost of dynamics relative to physics increases as N
  - e.g. if dynamics takes 20% at 25 km it will take 86% of the time at 1 km
- Option 1: Look at Algorithmic Acceleration
  - Semi-Lagrangian Transport
    - cannot ignore CFL with impunity
    - Increasingly non-local and dynamic communication patterns
  - Implicit or semi-implicit time integration - solvers
    - Non-local/quasi-local communications
  - Adaptive methods
- Option 2: Faster threads - find more parallelism in code
  - Architecture - old tricks, new tricks… magic tricks
    - Vector units, GPU's, FPGA's
  - device innovations (high-K)

Example: Aqua-Planet Experiment with CAM/HOMME Dycore

Integration Rate Drops of as Resolution Increases



Option 1. Applied Math vs Amdahl's Law- Could Solver Scalability Also Limit Integration Rate?

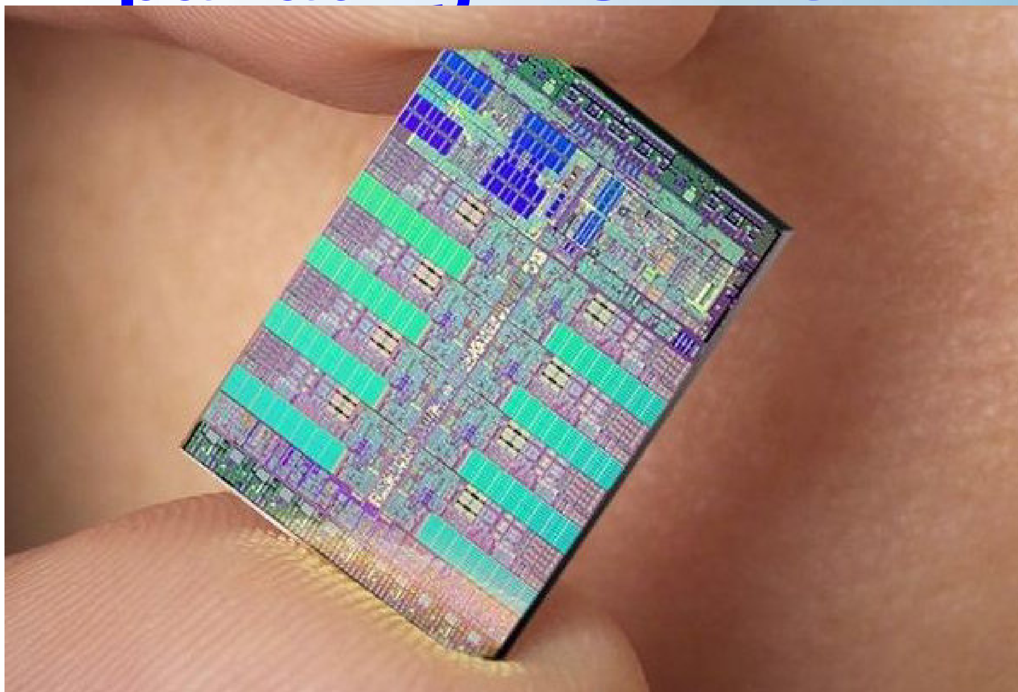


Computational & Information Systems Laboratory

CISL

NCAR

Option 2: an architectural paradigm shift?



- **IBM Cell Processor - 8 cores**
- **Intel "concept chip" 1 TFLOPS 80 cores/socket**
- **Paradigm shift?**
  - **GP-GPU - 128 graphics pipes**
    - **Measured 20x on WRF microphysics**
  - **FPGA (data flow model)**
    - **Simulated 21.7x on Xilinx V5 CAM sw-radiation code.**

Architecture is Important (Again)!

- Improvements in clock rates trumped architecture for 15 years
- Clock rates stall out -> architecture is back
- Accelerator space is wide open and poised for rapid increases in performance
- How do we exploit this?

Computational & Information Systems Laboratory

CISL

NCAR

Computational Intensity (CI)

- Compute Intensity:

  CI = Total Operations/(Input + Output data)
- GFLOPS = CI*Bandwidth
- Bandwidth expensive, flops cheap
- The higher the CI, the better we're able to exploit this state of affairs

Computational Intensity: Examples

- Saxpy: C= aX[]+B[], a = scalar, X, B vectors
  - CI = 1/3
- Matrix-Vector Multiply (N large)
  - CI = (2*N-1)*N/(N*(N+2)) ~ 2
- Radix 2 FFT -
  - CI = (5*log2(N)*N)/(2*N) = 2.5*log2(N)
  - 6.6 GFLOPS (low compute intensity)
- NxN - Matrix Multiply
  - CI = $(2*N^2-1)*N/(3*N*N) \sim 2*N/3$
  - 167 GFLOPS nVidia (high compute intensity)

Here Come the Accelerators: GPUs

- GPUs
  - SIMD find-grained parallelism
  - Also multi-level concurrency
  - Very fast, peak 520 GF/s
  - Cheap (< $500) commodity plug-in coprocessor for ordinary desktop systems
  - Programmability? Better tools on the way
- Approach used in WRF NWP Model
  - Incremental adoption of acceleration, module by module
  - Cloud microphysics (WSM5 testbed)
    - 25% of run time, < 1% of lines of code
    - **10x boost in microphysics**
    - 20% increase in App performance overall versus high-end AMD opteron
  - Ongoing, adapt more of code to GPU

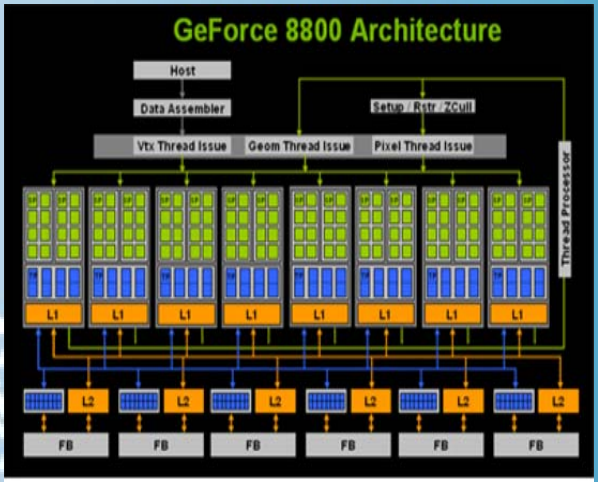


Figure 1. GeForce 8800 GTX block diagram

Computational & Information Systems Laboratory

CISL

NCAR

Here Come the Accelerators: WSM5 Kernel Performance

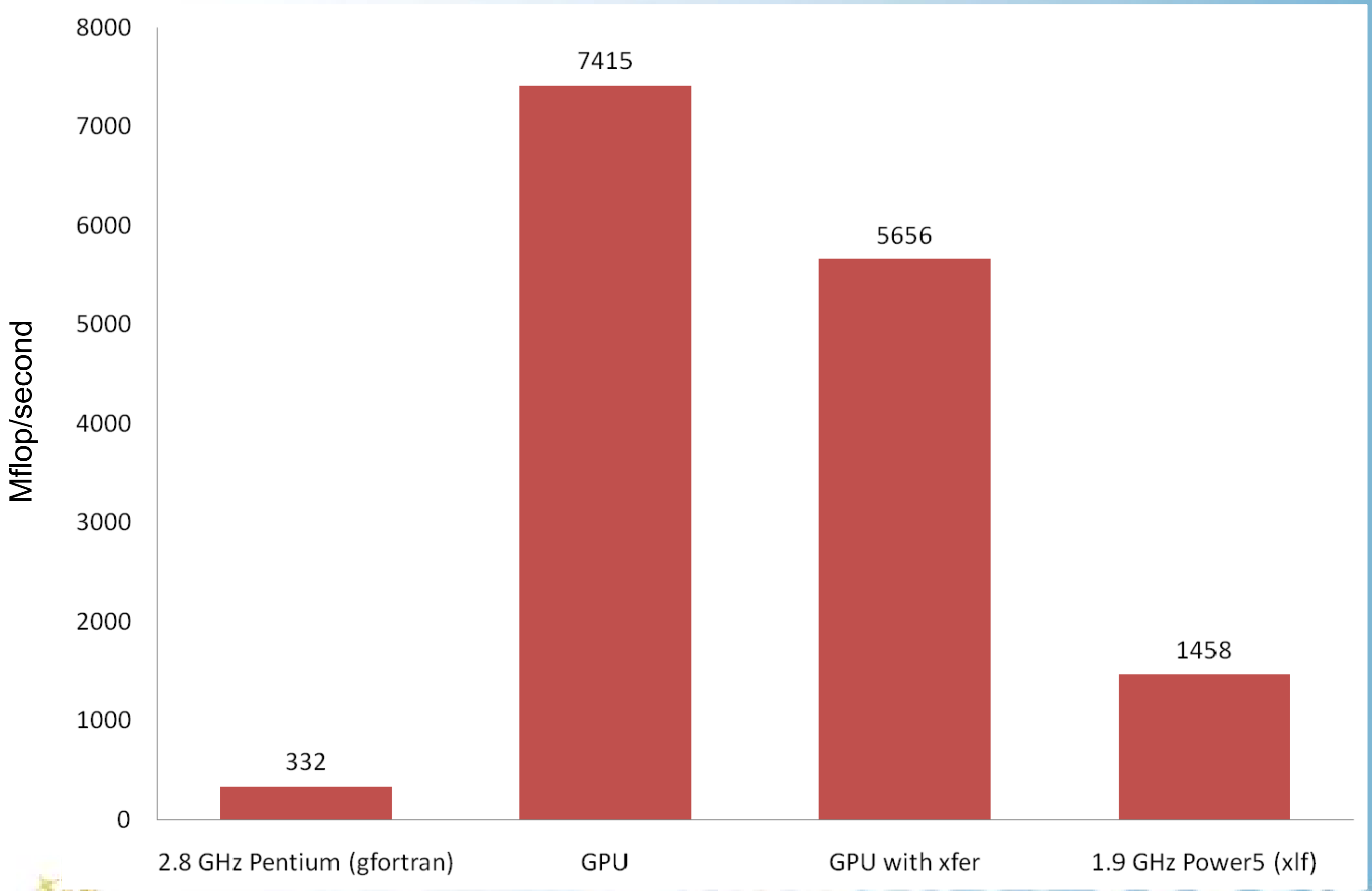
- **Stand-alone microphysics testbed**
- **Workload: Eastern U.S. "Storm of Century" case**
  - **74 x 61 (4500) threads**
  - **28 cells/column**
  - **~300 Mflop/invocation**
  - **5 MB footprint**
  - **Moving 2 MB host<-> GPU in 15 milliseconds (130MB/sec)**

Computational & Information Systems Laboratory

CISL

NCAR

Here Come the Accelerators: WSM5 Kernel Performance

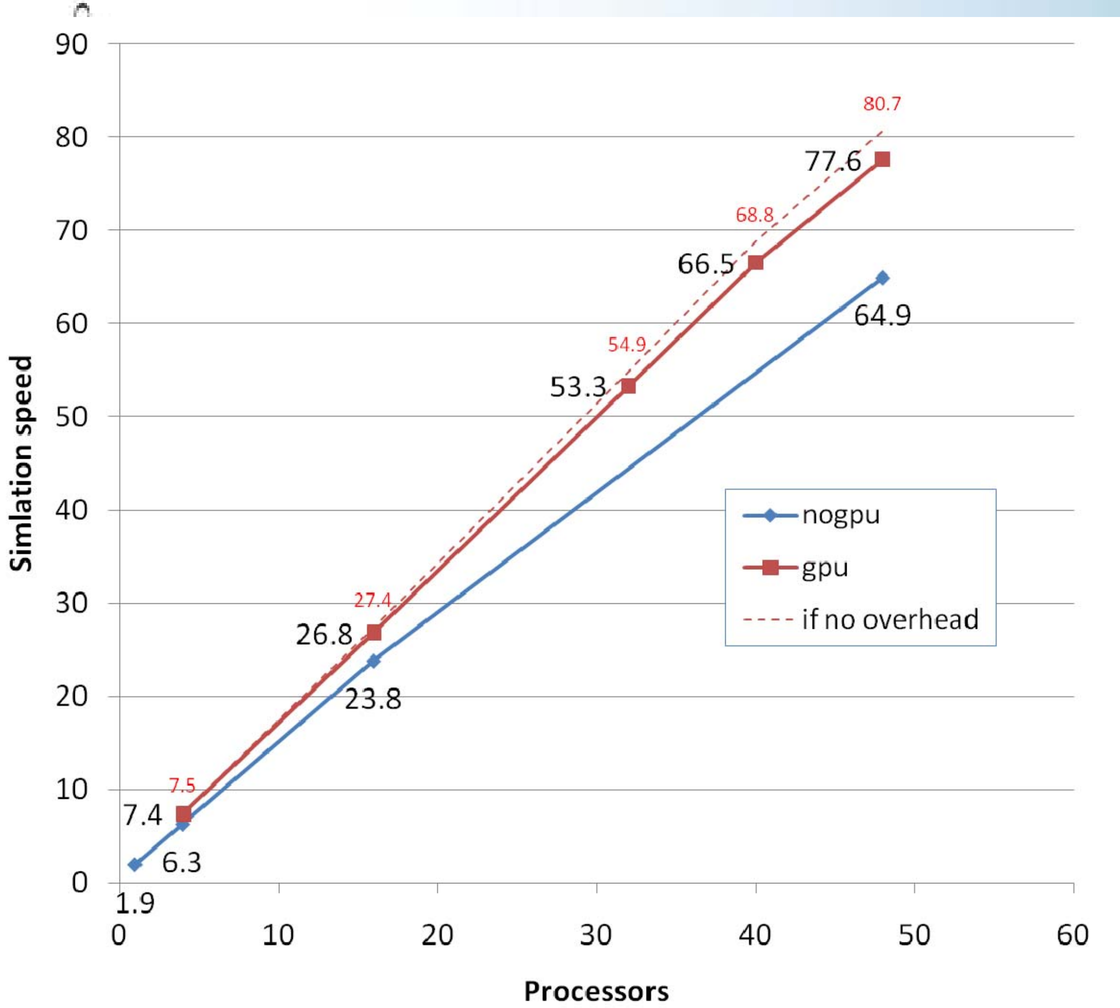


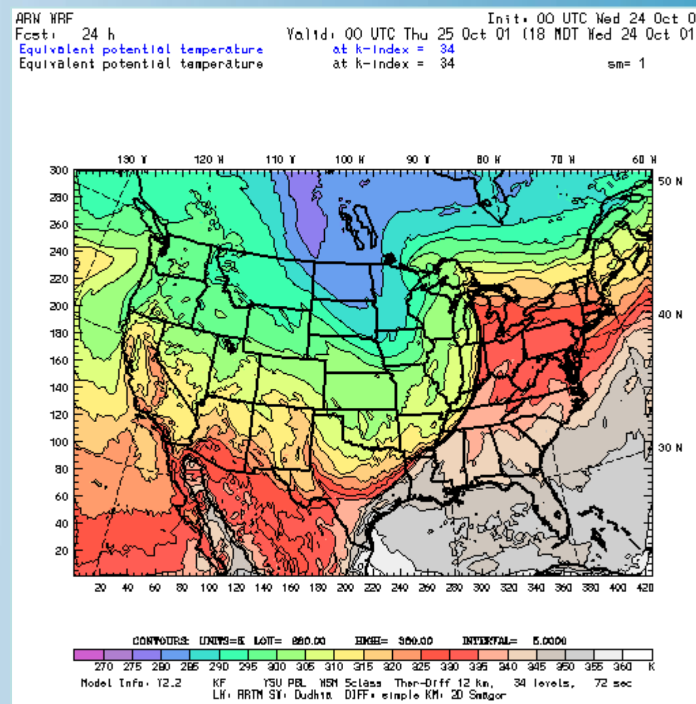
50 MFLOPS/W 38 MFLOPS/W 20 MFLOPS/W

Computational & Information Systems Laboratory

NCAR

Here Come the Accelerators:
WRF 12 km CONUS Benchmark





qp.ncsa.uiuc.edu

16 Dual dual-core 2.4 GHz Opteron nodes, each with Four NVIDIA 5600 GTX GPUs

NCAR

Courtesy of John Michalakes

Credit: Wen-mei Hwu, John Stone, and Jeremy Enos

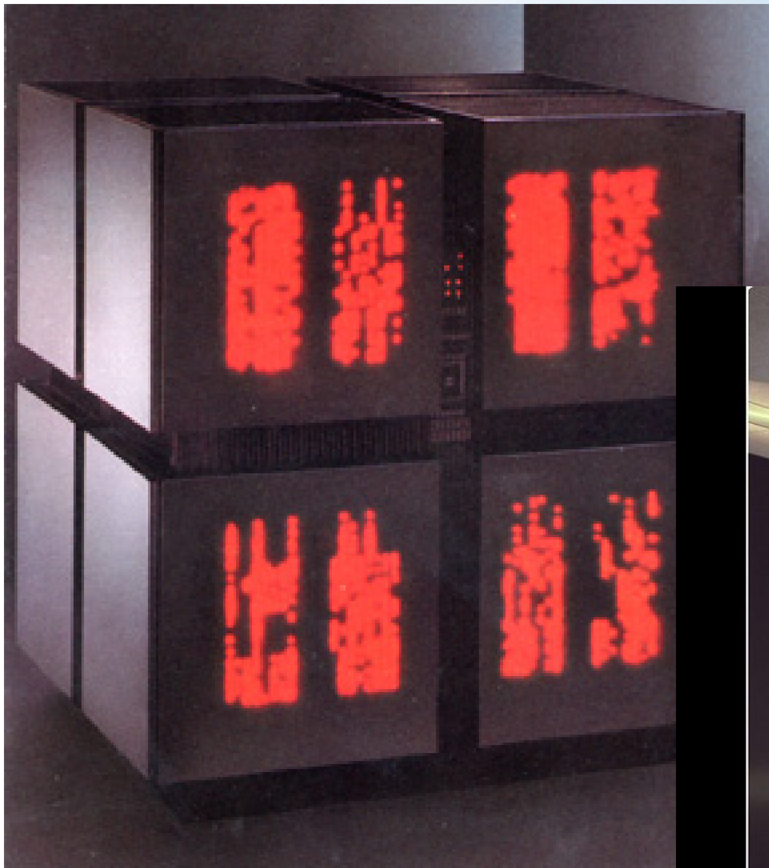
So where are we?

- **These GPU results are interesting and encouraging, but not yet compelling.**

What we need to facilitate migration to accelerators…

- Got CI? => accelerate, but…
- Need robust hardware
  - Error trapping, IEEE compliance
  - Performance counters
  - Circuitry support for synchronization
- Need a programming model for these things
  - CUDA? Brook+?
  - Pragmas? Language extensions?
    - Begin/end define region
    - Data management: local allocation, data transfer support
- Need Robust Compilers
  - Automate computer intensity/profitability analysis.
  - Provide feedback about it to user.

This Harkens back to the First Era of Massively Parallel Computing (1986-1994)



TMC CM-2

TMC CM-5

The Difference: This Time, the Accelerators are Commodity Hardware

- First 1 TFLOPS GPU is out (February, 2008)
- 11 million PS3 units shipped in 2007
- Attract teens to supercomputing?
- Leverage new sources of talent and new t

Maybe this sounds crazy…

Computational & Information Systems Laboratory

CISL

NCAR

Why?

- Why is it that we understand that we need a heroic-scale supercomputing effort to provide stewardship of our nuke stockpile, but we can't imagine the need for a similar program to assure stewardship of our planet?

Computational & Information Systems Laboratory

CISL

NCAR


Thanks!
Any Questions?

The Interdisciplinary and Interagency Team Working on Climate Scalability

- **Contributors:**

  D. Bailey (NCAR)
  F. Bryan (NCAR)
  T. Craig (NCAR)
  A. St. Cyr (NCAR)
  J. Dennis (NCAR)
  J. Edwards (IBM)
  B. Fox-Kemper (MIT,CU)
  E. Hunke (LANL)
  B. Kadlec (CU)
  D. Ivanova (LLNL)
  E. Jedlicka (ANL)
  E. Jessup (CU)
  R. Jacob (ANL)
  P. Jones (LANL)
  S. Peacock (NCAR)
  K. Lindsay (NCAR)
  W. Lipscomb (LANL)
  R. Loy (ANL)
  J. Michalakes (NCAR)
  A. Mirin (LLNL)
  M. Maltrud (LANL)
  J. McClean (LLNL)
  R. Nair (NCAR)
  M. Norman (NCSU)
  T. Qian (NCAR)
  M. Taylor (SNL)
  H. Tufo (NCAR)
  M. Vertenstein (NCAR)
  P. Worley (ORNL)
  M. Zhang (SUNYSB)

- **Funding:**
  - DOE-BER CCPP Program Grant
    - DE-FC03-97ER62402
    - DE-PS02-07ER07-06
    - DE-FC02-07ER64340
    - B&R KP1206000
  - DOE-ASCR
    - B&R KJ0101030
  - NSF Cooperative Grant NSF01
  - NSF PetaApps Award

- **Computer Time:**
  - Blue Gene/L time:
    - NSF MRI Grant
      - NCAR
      - University of Colorado
      - IBM (SUR) program
    - BGW Consortium Days
      - IBM research (Watson)
    - LLNL
    - Stony Brook & BNL
  - CRAY XT3/4 time:
    - ORNL
    - Sandia