Los Alamos
NATIONAL LABORATORY

Los Alamos National Laboratory
RADIANT
www.lanl.gov/radiant

SUPERCOMPUTING
in SMALL SPACES

2004 International Supercomputer Conference
Most "Innovative Supercomputer Architecture" Award

Green Destiny and Its Evolving Parts:

Supercomputing for the Rest of Us

Wu Feng and Chung-Hsing Hsu

Research & Development in Advanced Network Technology (RADIANT)
Computer & Computational Sciences Division
Los Alamos National Laboratory

Outline

- Where is Supercomputing?
- Motivation: Efficiency, Reliability, Availability (ERA)
- A New Flavor of Supercomputing: Supercomputing in Small Spaces
    - Green Destiny: Origin and Architecture
- Benchmark Results for Green Destiny
- The Evolution of Green Destiny
    - Real-time, Constraint-based Dynamic Voltage Scaling
    - Initial Benchmark Results
- Conclusion

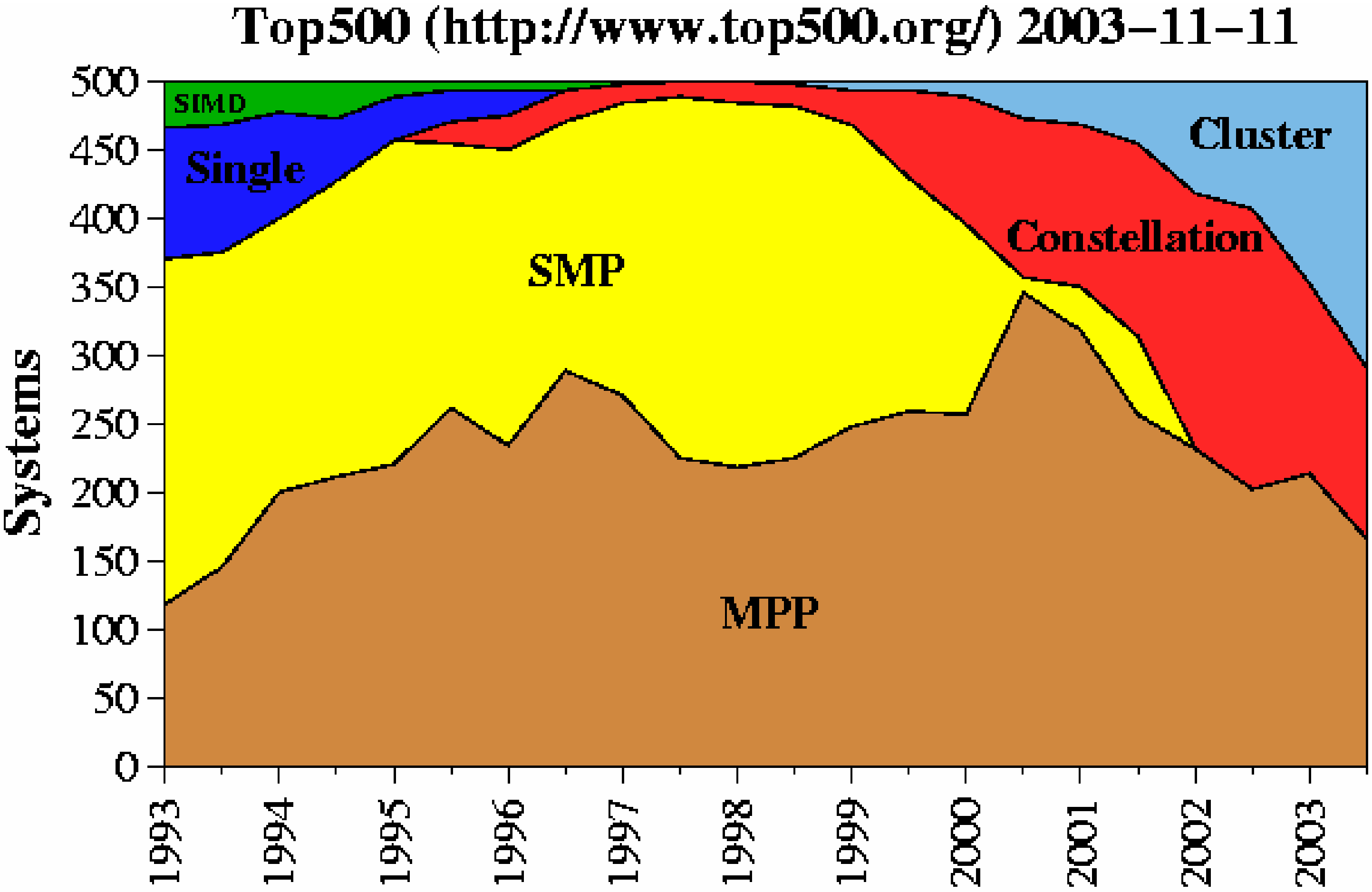
Where is Supercomputing?

(Pictures courtesy of Thomas Sterling, Caltech & NASA JPL)

Metrics for Evaluating Supercomputers

- *Performance (i.e., Speed)*
  - Metric: Floating-Operations Per Second (FLOPS)
  - Example: Japanese Earth Simulator, ASCI Thunder & Q.
- *Price/Performance → Cost Efficiency*
  - Metric: Cost / FLOPS
  - Examples: LANL Space Simulator, VT Apple G5 cluster.
- Performance & price/performance are important metrics, but …

Architectures from the Top 500 Supercomputer List

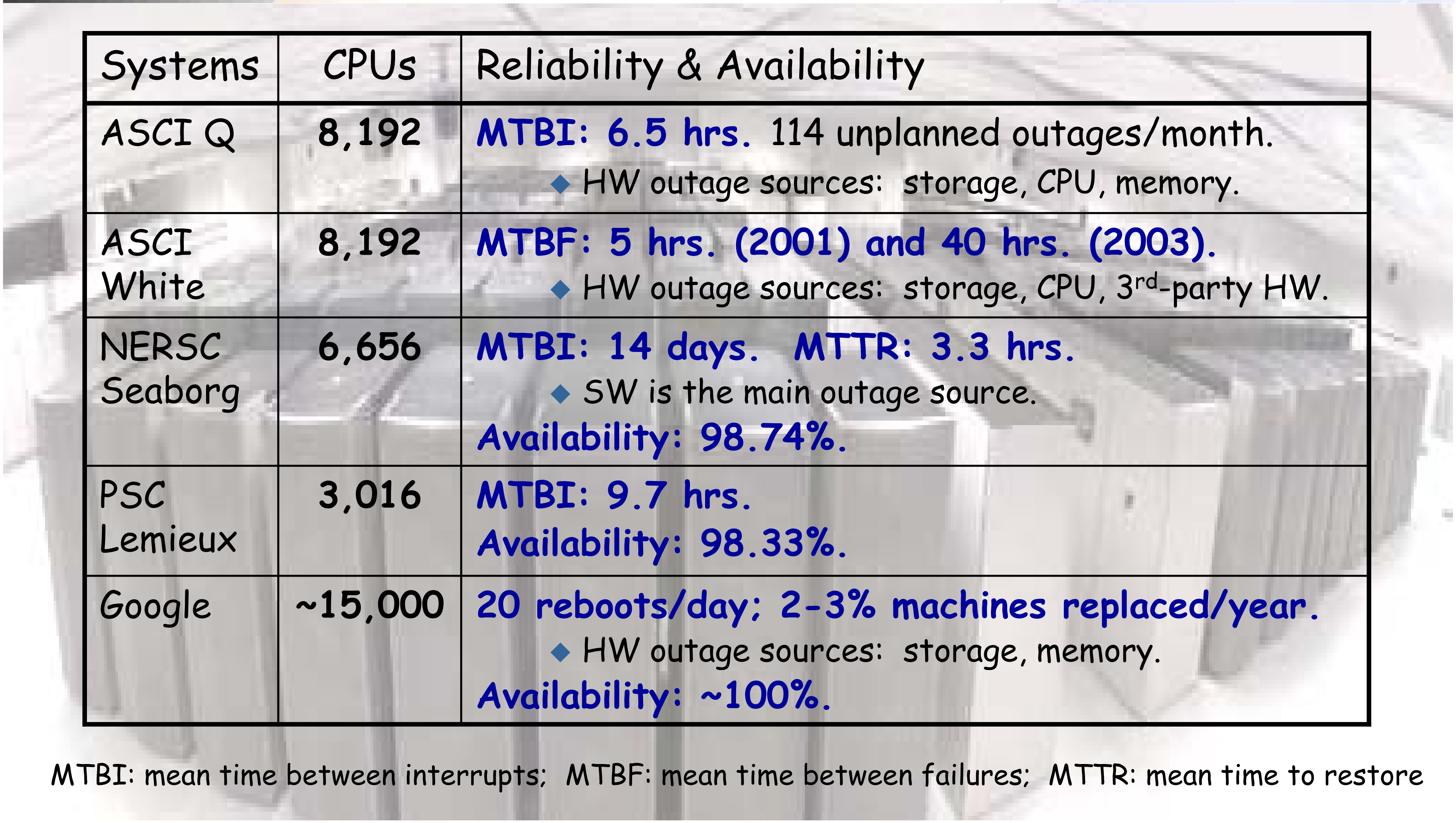


Wu Feng
feng@lanl.gov

http://www.lanl.gov/radiant
http://sss.lanl.gov

Chung-Hsing Hsu
chunghsu@lanl.gov

Reliability & Availability of Supercomputers



Systems	CPUs	Reliability & Availability
ASCI Q	8,192	MTBI: 6.5 hrs. 114 unplanned outages/month. ◆ HW outage sources: storage, CPU, memory.
ASCI White	8,192	MTBF: 5 hrs. (2001) and 40 hrs. (2003). ◆ HW outage sources: storage, CPU, 3rd-party HW.
NERSC Seaborg	6,656	MTBI: 14 days. MTTR: 3.3 hrs. ◆ SW is the main outage source. Availability: 98.74%.
PSC Lemieux	3,016	MTBI: 9.7 hrs. Availability: 98.33%.
Google	~15,000	20 reboots/day; 2-3% machines replaced/year. ◆ HW outage sources: storage, memory. Availability: ~100%.

MTBI: mean time between interrupts; MTBF: mean time between failures; MTTR: mean time to restore

Source: Daniel A. Reed, RENCI & UNC

Efficiency of Supercomputers

- “Performance” and “Price/Performance” Metrics …
  - Lower efficiency, reliability, and availability.
  - Higher operational costs, e.g., admin, maintenance, etc.
- Examples
  - Computational Efficiency
    - Relative to Space: Performance/Sq. Ft.
    - Relative to Power: Performance/Watt
    - Relative to Peak: Actual Perf/Peak Perf (see J. Dongarra)
  - Performance: 2000-fold increase (since the Cray C90).
    - Performance/Sq. Ft.: Only 65-fold increase.
    - Performance/Watt: Only 300-fold increase.
  - Massive construction and operational costs associated with powering and cooling.

Where is Supercomputing?

(Pictures courtesy of Thomas Sterling, Caltech & NASA JPL)

Efficiency, reliability, and availability will be *the* key issues of this decade.

Another Perspective: "Commodity-Use" HPC

- Requirement: Near-100% *availability* with *efficient* and *reliable* resource usage.
  - E-commerce, enterprise apps, online services, ISPs.
- Problems
  - Frequency of Service Outages
    - 65% of IT managers report that their websites were unavailable to customers over a 6-month period.
  - Cost of Service Outages
    - NYC stockbroker: **$ 6,500,000/hr**
    - Ebay (22 hours): **$ 225,000/hr**
    - Amazon.com: **$ 180,000/hr**
    - Social Effects: negative press, loss of customers who "click over" to competitor.

Source: David Patterson, UC-Berkeley

Another Perspective: "Commodity-Use" HPC

- Pharmaceutical, financial, actuarial, retail, aerospace, automotive, science and engineering, data centers.
- Sampling of Consumer Requirements of HPC Systems
  - Myself, LANL (high-performance network simulations)

    Traditional cluster fails weekly, oftentimes more frequently.

    [1] Low Power → Reliability, [2] Space, [3] Performance.
  - Peter Bradley, Pratt & Whitney (CFD, composite modeling)

    [1] Reliability, [2] Transparency, [3] Resource Management
  - Eric Schmidt, Google (instantaneous search)
    - Low power, NOT speed.
    - DRAM density, NOT speed.
    - Availability and reliability, NOT speed.

Outline

- Where is Supercomputing?
- Motivation: Efficiency, Reliability, Availability (ERA)
- A New Flavor of Supercomputing: Supercomputing in Small Spaces
  - Green Destiny: Origin and Architecture
- Benchmark Results for Green Destiny
- The Evolution of Green Destiny
  - Real-time, Constraint-based Dynamic Voltage Scaling
  - Initial Benchmark Results
- Conclusion

A New Flavor of Supercomputing

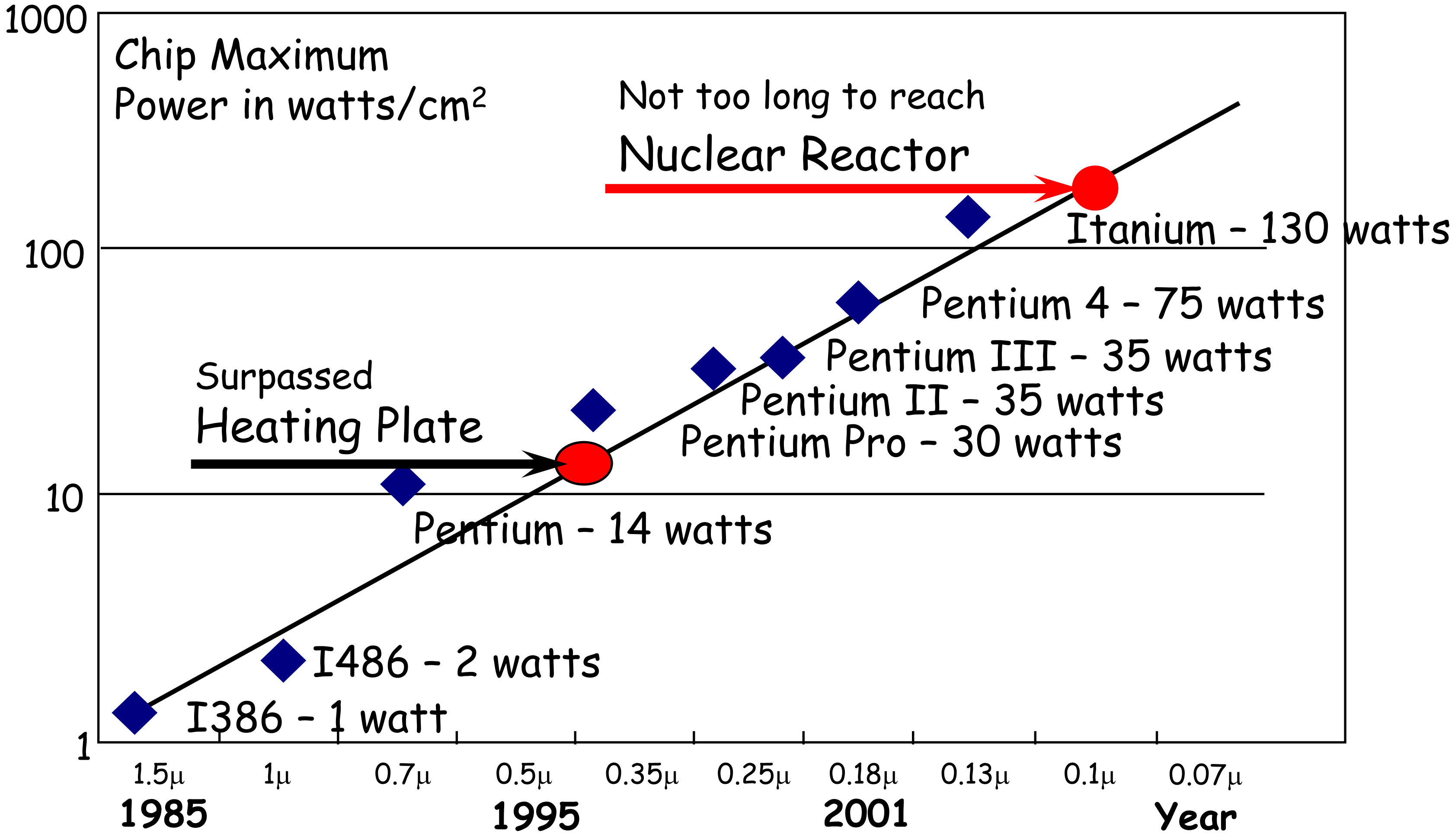
- Supercomputing in Small Spaces (http://sss.lanl.gov)
  - First instantiations: Bladed Beowulf
    - MetaBlade (24) and Green Destiny (240).
- Goal
  - Improve efficiency, reliability, and availability (ERA) in large-scale computing systems.
    - Sacrifice a little bit of raw performance.
    - Improve overall system throughput as the system will "always" be available, i.e., effectively no downtime, no hardware failures, etc.
  - Reduce the total cost of ownership (TCO). Another talk …
- Crude Analogy
  - Ferrari 550: Wins raw performance but reliability is poor so it spends its time in the shop. Throughput low.
  - Toyota Camry: Loses raw performance but high reliability results in high throughput (i.e., miles driven → answers/month).

How to Improve Efficiency, Reliability & Availability?

- Complementary Approaches
  - Via HW design & manufacturing (e.g., IBM, Transmeta)
  - Via a software reliability layer that assumes underlying hardware unreliability *a la* the Internet (e.g., Google).
  - Via systems design & integration (e.g., Green Destiny)
- Observation
  - High power α high temperature α low reliability.
  - Arrhenius' Equation

    (circa 1890s in chemistry → circa 1980s in computer & defense industries)
    - ☞ As temperature increases by 10° C …
      - The failure rate of a system doubles.
    - ☞ Twenty years of unpublished empirical data .

Moore's Law for Power

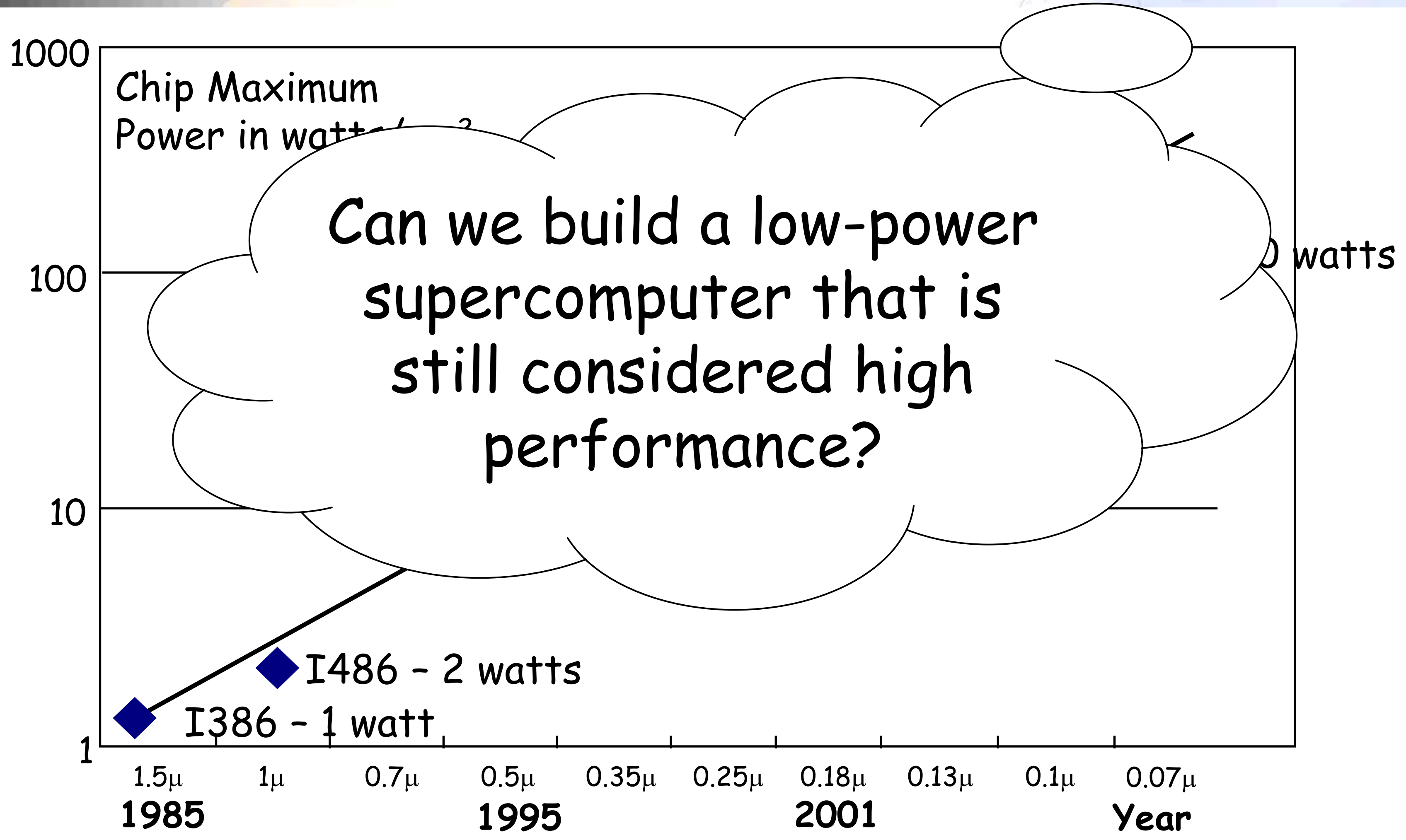


Not too long to reach
Nuclear Reactor

Surpassed
Heating Plate

Source: Fred Pollack, Intel. New Microprocessor Challenges in the Coming Generations of CMOS Technologies, MICRO32 and Transmeta

Moore's Law for Power



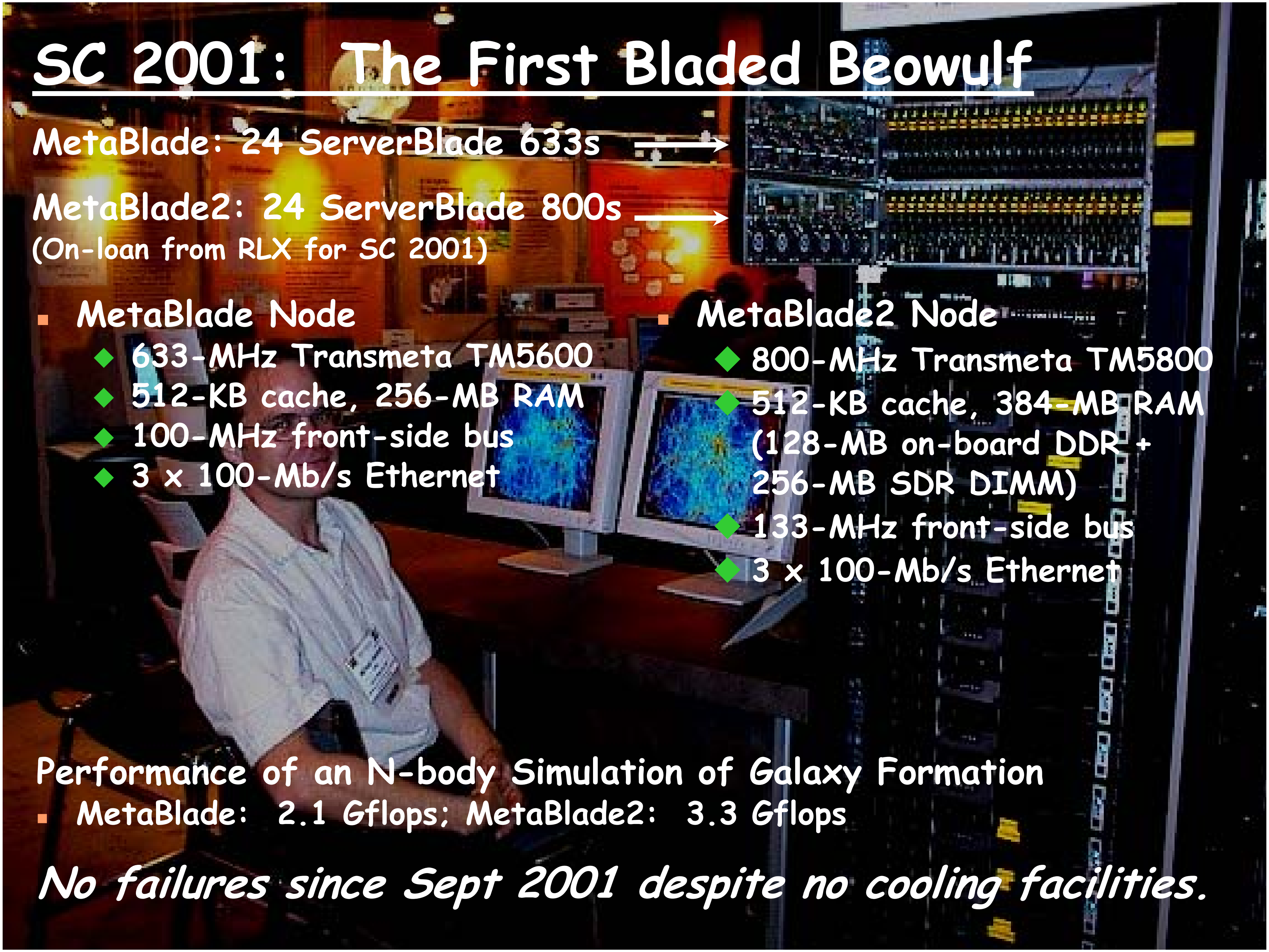
Can we build a low-power supercomputer that is still considered high performance?

Source: Fred Pollack, Intel. New Microprocessor Challenges in the Coming Generations of CMOS Technologies, MICRO32 and Transmeta

MetaBlade: The Origin of Green Destiny

- <u>Project Conception: Sept. 28, 2001.</u>
  - On a winding drive home through Los Alamos Canyon … the need for reliable compute cycles.
    - Leverage RLX web-hosting servers with Transmeta CPUs.
- <u>Project Implementation: Oct. 9, 2001.</u>
  - Received the "bare" hardware components.
  - Two man-hours later …
    - Completed construction of a 24-CPU RLX System 324 (dubbed *MetaBlade*) and installation of system software.
  - One man-hour later …
    - Successfully executing a 10-million N-body simulation of a galaxy formation
- <u>Public Demonstration: Nov. 12, 2001 at SC 2001.</u>

SC 2001: The First Bladed Beowulf



MetaBlade: 24 ServerBlade 633s →

MetaBlade2: 24 ServerBlade 800s →
(On-loan from RLX for SC 2001)

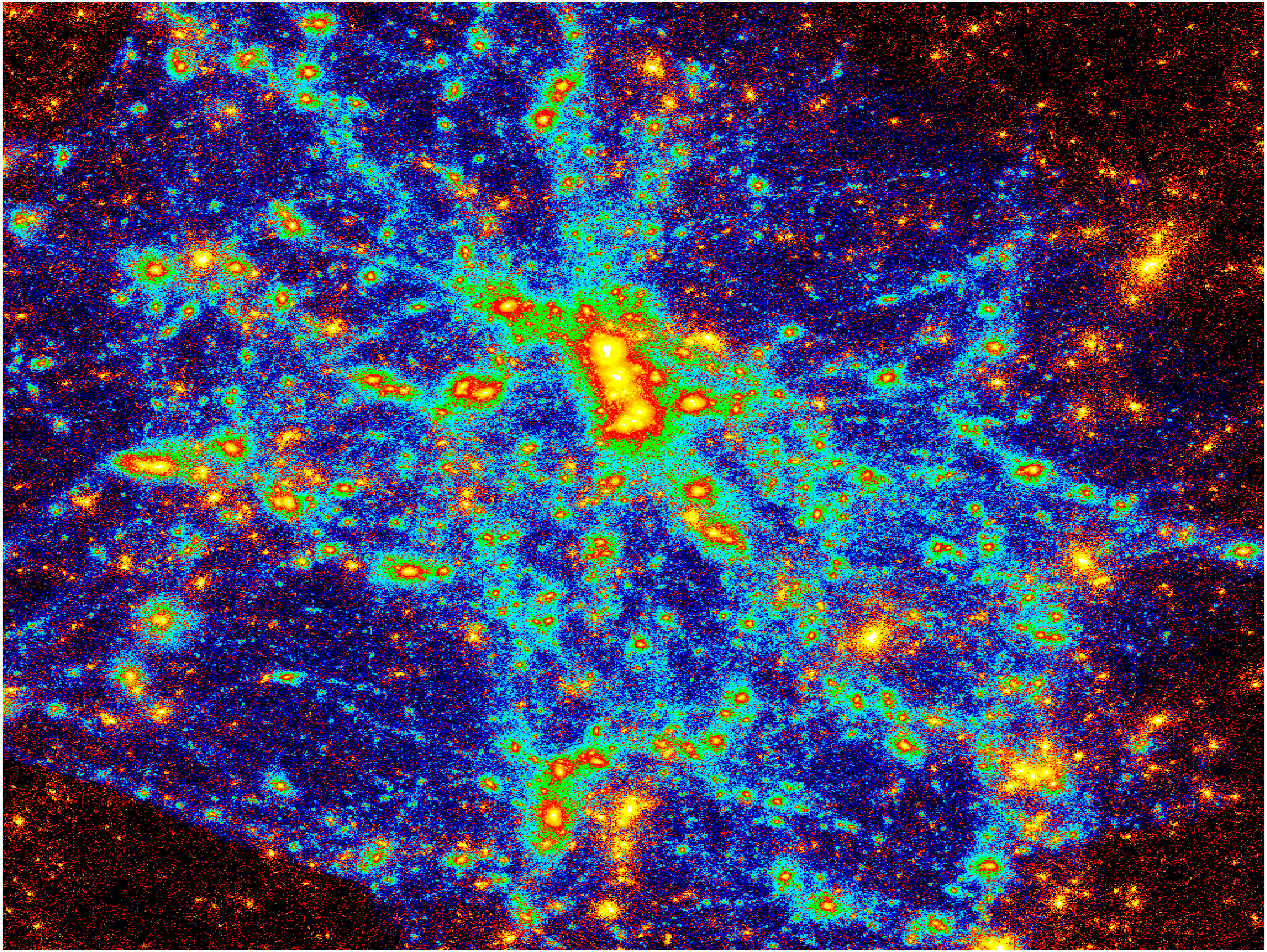
- MetaBlade Node
  - 633-MHz Transmeta TM5600
  - 512-KB cache, 256-MB RAM
  - 100-MHz front-side bus
  - 3 x 100-Mb/s Ethernet

- MetaBlade2 Node
  - 800-MHz Transmeta TM5800
  - 512-KB cache, 384-MB RAM (128-MB on-board DDR + 256-MB SDR DIMM)
  - 133-MHz front-side bus
  - 3 x 100-Mb/s Ethernet

Performance of an N-body Simulation of Galaxy Formation

- MetaBlade: 2.1 Gflops; MetaBlade2: 3.3 Gflops

No failures since Sept 2001 despite no cooling facilities.

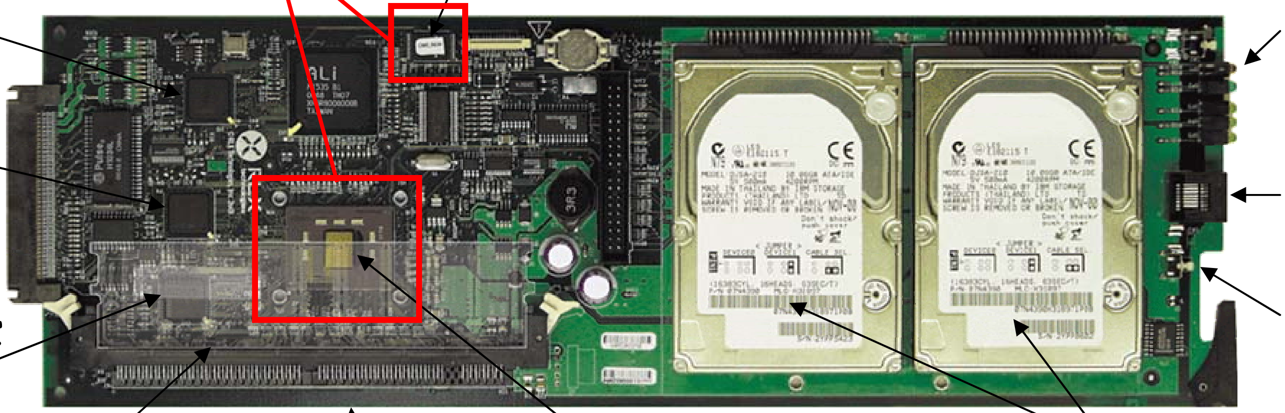


Scaling *MetaBlade* …

- Interest in MetaBlade and MetaBlade2 ?
  - Continual crowds over the three days of SC 2001.
- Inspiration
  - Build a full 42U rack of MetaBlade clusters.
    - Scale up performance/space to 3500 Mflop/sq. ft.
  - Problem: In 2001, performance per node on MetaBlade was nearly *three* times worse than the fastest processor at the time.
  - Can we improve performance while maintaining low power? Yes via Transmeta's code-morphing software, which is part of the Transmeta CPU.
    - What is code-morphing software?

Green Destiny Architecture RLX ServerBlade™ 633 (circa 2000)

Modify the Transmeta CPU software to improve performance.



Code Morphing Software (CMS), 1 MB

Public NIC
33 MHz PCI

Private NIC
33 MHz PCI

Management NIC
33 MHz PCI

Status LEDs

Serial RJ-45
debug port

Reset Switch

128MB, 256MB, 512MB
DIMM SDRAM
PC-133

512KB
Flash ROM

Transmeta™
TM5600 633 MHz

ATA 66
0 or 1 or 2 - 2.5" HDD
10 or 30 GB each

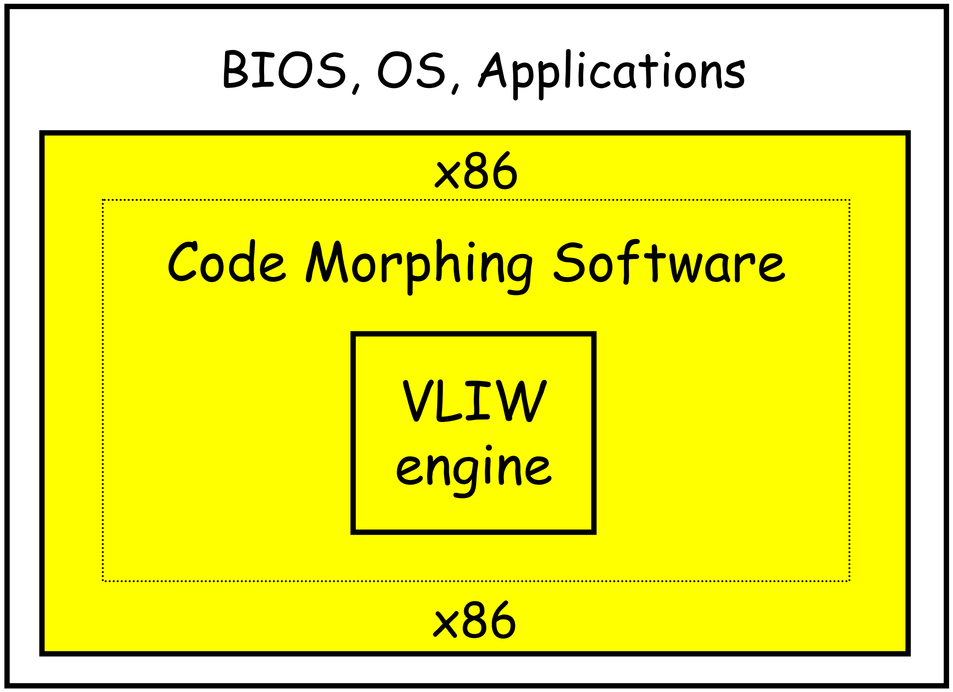
Crusoe™

RLX ServerBlade™ 1000t
$999 (as of Dec. 2003)

128KB L1 cache, 512KB L2 cache
LongRun, Northbridge, x86 compatible

Transmeta TM5600 CPU: VLIW + CMS

- VLIW Engine
  - Up to four-way issue
    - In-order execution only.
  - Two integer units
  - Floating-point unit
  - Memory unit
  - Branch unit



- VLIW Transistor Count ("Anti-Moore's Law")
  - ~$\frac{1}{4}$ of Intel PIII → ~ 6x-7x less power consumption
  - Less power → lower "on-die" temp. → better reliability & availability

Green Destiny Architecture Transmeta TM5x00 CMS

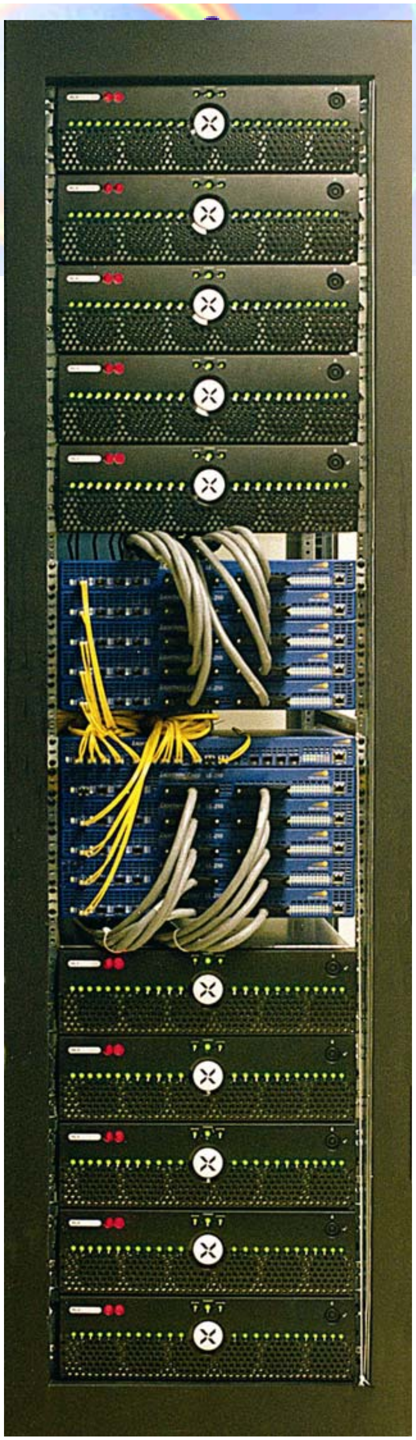
- Code-Morphing Software (CMS)
  - Provides compatibility by dynamically “morphing” x86 instructions into simple VLIW instructions.
  - Learns and improves with time, i.e., iterative execution.

- High-Performance Code-Morphing Software (HP-CMS)
  - Results (circa 2001)
    - *Optimized to improve floating-pt. performance by 50%.*
    - *1-GHz Transmeta performs as well as a 1.2-GHz PIII-M.*
  - How?

Green Destiny Architecture Low-Power Network Switches

- WWP LE-410: 16 ports of Gigabit Ethernet
- WWP LE-210: 24 ports of Fast Ethernet via RJ-21s
- (Avg.) Power Dissipation / Port: A few watts.

“Green Destiny” Bladed Beowulf
(circa 2002)



- A 240-Node Beowulf in One Cubic Meter
- Each Node
  - 667-MHz Transmeta TM5600 CPU w/ Linux 2.4.x
    - Upgraded to 1-GHz Transmeta TM5800 CPUs
  - 640-MB RAM, 20-GB hard disk, 100-Mb/s Ethernet (up to 3 interfaces)
- Total
  - 160 Gflops peak (240 Gflops with upgrade)
  - 150 GB of RAM (expandable to 276 GB)
  - 4.8 TB of storage (expandable to 38.4 TB)
  - Power Consumption: Only 3.2 – 5.2 kW.
- Linpack: 101 Gflops in March 2003.
- Reliability & Availability
  - *No unscheduled failures in 24 months.*

Outline

- Where is Supercomputing?
- Motivation: Efficiency, Reliability, Availability (ERA)
- A New Flavor of Supercomputing: Supercomputing in Small Spaces
  - Green Destiny: Origin and Architecture
- Benchmark Results for Green Destiny
- The Evolution of Green Destiny
  - Real-time, Constraint-based Dynamic Voltage Scaling
  - Initial Benchmark Results
- Conclusion

Gravitational Microkernel on Transmeta CPUs

(Data courtesy of Michael S. Warren, T-6 at Los Alamos National Laboratory.)

- Gravitational Microkernel Benchmark (circa June 2002)

Processor	Math sqrt	Karp sqrt
500-MHz Intel PIII	87.6	137.5
533-MHz Compaq Alpha EV56	76.2	178.5
633-MHz Transmeta TM5600	115.0	144.6
800-MHz Transmeta TM5800	174.1	296.6
375-MHz IBM Power3	298.5	379.1
1200-MHz AMD Athlon MP	350.7	452.5

Units are in Mflops.

Bottom Line: CPU performance was competitive. Memory bandwidth was not (i.e., 300-350 MB/s with STREAMS).

Treecode Benchmark for n-Body Galaxy Formation

(Data courtesy of Michael S. Warren, T-6 at Los Alamos National Laboratory.)

Year	Site	Machine	CPUs	Gflops	Mflops/CPU
2003	LANL	ASCI QB	3600	2793	775.8
2003	LANL	Space Simulator	288	179.7	623.9
2002	NERSC	IBM SP-3	256	57.70	225.0
2000	LANL	SGI O2K	64	13.10	205.0
2002	LANL	Green Destiny	212	38.90	183.5
2001	SC'01	MetaBlade2	24	3.30	138.0
1998	LANL	Avalon	128	16.16	126.0
1996	LANL	Loki	16	1.28	80.0
1996	SC '96	Loki+Hyglac	32	2.19	68.4
1996	Sandia	ASCI Red	6800	464.90	68.4
1995	JPL	Cray T3D	256	7.94	31.0
1995	LANL	TMC CM-5	512	14.06	27.5

Treecode Benchmark for n-Body Galaxy Formation

(Data courtesy of Michael S. Warren, T-6 at Los Alamos National Laboratory.)

Year	Site	Machine	CPUs	Gflops	Mflops/CPU
2003	LANL	ASCI QB	3600	2793	775.8
2003	LANL	Space Simulator	288	179.7	623.9
2002	NERSC	IBM SP-3	256	57.70	225.0
2000	LANL	SGI O2K	64	13.10	205.0
2002	LANL	Green Destiny	212	38.90	183.5
2001	SC'01	Meta[illegible]2	24	3.30	138.0
1998	LANL	[illegible]	[illegible]	16.16	126.0
1996	[illegible]	[illegible]	[illegible]	[illegible]	80.0
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]4
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]4
1995	[illegible]	[illegible]	[illegible]	[illegible]	31.0
1995	LANL	TMC CM[illegible]	[illegible]	14.06	27.5

Upgraded "Green Destiny" (Dec. 2002)

58 Gflops → **274 Mflops/CPU**

(Balance: 1 Mflop – 1 MB/s – 1 Mb/s with network-latency hiding.)

Performance Metrics for …

- Efficiency, Reliability, and Availability (ERA)
  - Total Cost of Ownership
  - Computational Efficiency
    - Relative to Space: Performance/Sq. Ft.
    - Relative to Power: Performance/Watt
  - Reliability
    - MTBF: Mean Time Between Failures
  - Availability
    - Percentage of time that resources are available for HPC.

Parallel Computing Platforms (An "Apples-to-Oranges" Comparison)

- Avalon (1996)
  - 140-CPU *Traditional Beowulf Cluster*
- ASCI Red (1996)
  - 9632-CPU *MPP*
- ASCI White (2000)
  - 512-Node (8192-CPU) *Cluster of SMPs*
- Green Destiny (2002)
  - 240-CPU *Bladed Beowulf Cluster*

Parallel Computing Platforms Running the N-body Code

Machine	Avalon Beowulf	ASCI Red	ASCI White	Green Destiny
Year	1996	1996	2000	2002
Performance (Gflops)	18	600	2500	39
Area (ft^2)	120	1600	9920	6
Power (kW)	18	1200	2000	5
DRAM (GB)	36	585	6200	150
Disk (TB)	0.4	2.0	160.0	4.8
DRAM density (MB/ft^2)	300	366	625	25000
Disk density (GB/ft^2)	3.3	1.3	16.1	800.0
Perf/Space (Mflops/ft^2)	150	375	252	6500
Perf/Power (Mflops/watt)	1.0	0.5	1.3	7.5

Parallel Computing Platforms Running the N-body Code

Machine	Avalon Beowulf	ASCI Red	ASCI White	Green Destiny
Year	1996	1996	2000	2002
Performance (Gflops)	18	600	2500	39
Area (ft^2)	120	1600	9920	6
Power (kW)	18	1200	2000	5
DRAM (GB)	36	585	6200	150
Disk (TB)	0.4	2.0	160.0	4.8
DRAM density (MB/ft^2)	300	366	625	25000
Disk density (GB/ft^2)	3.3	1.3	16.1	800.0
Perf/Space (Mflops/ft^2)	150	375	252	6500
Perf/Power (Mflops/watt)	1.0	0.5	1.3	7.5

Parallel Computing Platforms Running the N-body Code

Machine	Avalon Beowulf	ASCI Red	ASCI White	Green Destiny+
Year	1996	1996	2000	2002
Performance (Gflops)	18	600	2500	*58*
Area (ft^2)	120	1600	9920	6
Power (kW)	18	1200	2000	5
DRAM (GB)	36	585	6200	150
Disk (TB)	0.4	2.0	160.0	4.8
DRAM density (MB/ft^2)	300	366	625	25000
Disk density (GB/ft^2)	3.3	1.3	16.1	800.0
Perf/Space ($Mflops/ft^2$)	150	375	252	9667
Perf/Power (Mflops/watt)	1.0	0.5	1.3	11.6

Green Destiny vs. Earth Simulator: LINPACK

Machine	Green Destiny+	Earth Simulator
Year	2002	2002
LINPACK Performance (Gflops)	101	35,860
Area (ft^2)	6	17,222 * 2
Power (kW)	5	7,000
Cost efficiency ($/Mflop)	3.35	11.15
Space efficiency (Mflops/ft^2)	16,833	1,041
Power efficiency (Mflops/watt)	20.20	5.13

Disclaimer: This is not a fair comparison. Why?

(1) Price and the use of area and power do *not* scale linearly.

(2) Goals of the two machines are different.

Summary of ERA Performance Metrics for …

- Green Destiny
  - Computational Efficiency
    - ☞ Relative to Space: Performance/Sq. Ft.
      Up to 60x better.
    - ☞ Relative to Power: Performance/Watt
      Up to 20x better.
  - Reliability
    - ☞ MTBF: Mean Time Between Failures
      "Infinite"
  - Availability
    - ☞ Percentage of time that resources are available for HPC.
      Nearly 100%.

Outline

- Where is Supercomputing?
- Motivation: Efficiency, Reliability, Availability (ERA)
- A New Flavor of Supercomputing: Supercomputing in Small Spaces
  - ◆ Green Destiny: Origin and Architecture
- Benchmark Results for Green Destiny
- The Evolution of Green Destiny
  - ◆ Real-time, Constraint-based Dynamic Voltage Scaling
  - ◆ Initial Benchmark Results
- Conclusion

The Evolution of Green Destiny

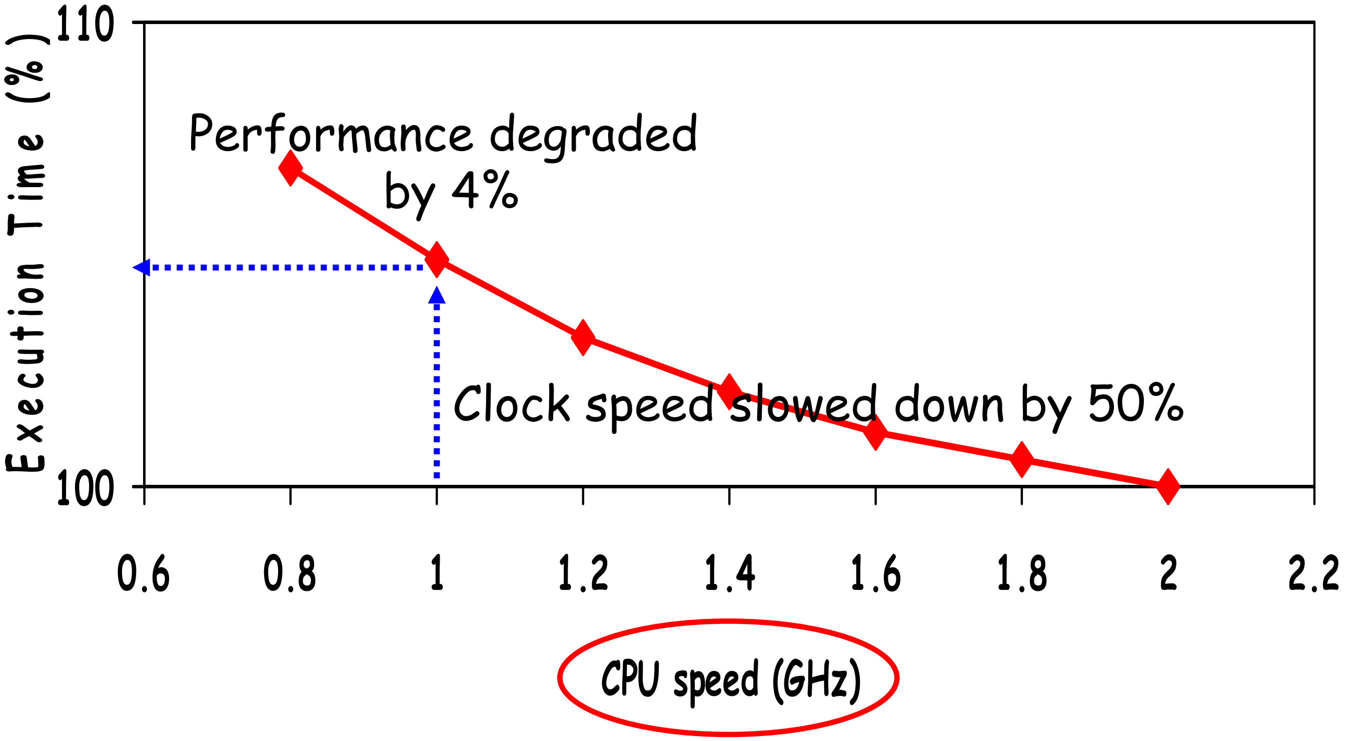
- Problems with Green Destiny (even with HP-CMS)
  - An architectural approach that ties us to a specific vendor, i.e., RLX, who is headed in a different direction.
  - Raw performance of a compute node.
    - Up to two times worse than the fastest CPU at the time of construction (2002). Now, upwards of four times worse (2004).
- Solution
  - Transform our architectural approach into a software-based one that works across a wide range of processors.
  - Start with higher-performing commodity components to achieve performance goals but use the above software-based technique to reduce power consumption dramatically.
- But How?
  - Dynamic voltage scaling + efficient scheduling algorithm.

Dynamic Voltage Scaling (DVS)

- DVS Technique
  - Trades CPU performance for power reduction by allowing the CPU supply voltage and/or frequency to be adjusted at run-time.
- Why is DVS important?
  - Recall: Moore's Law for Power.
  - CPU power consumption is directly proportional to the *square of the supply voltage* and to *frequency*.
- DVS Algorithm
  - Determines *when* to adjust the current frequency-voltage setting and *what* the new frequency-voltage setting should be.

Motivation for Real-Time Constraint-Based DVS

- Key Observation
  - The execution time of many programs are insensitive to the CPU speed change. e.g., NAS IS benchmark.



Performance degraded by 4%

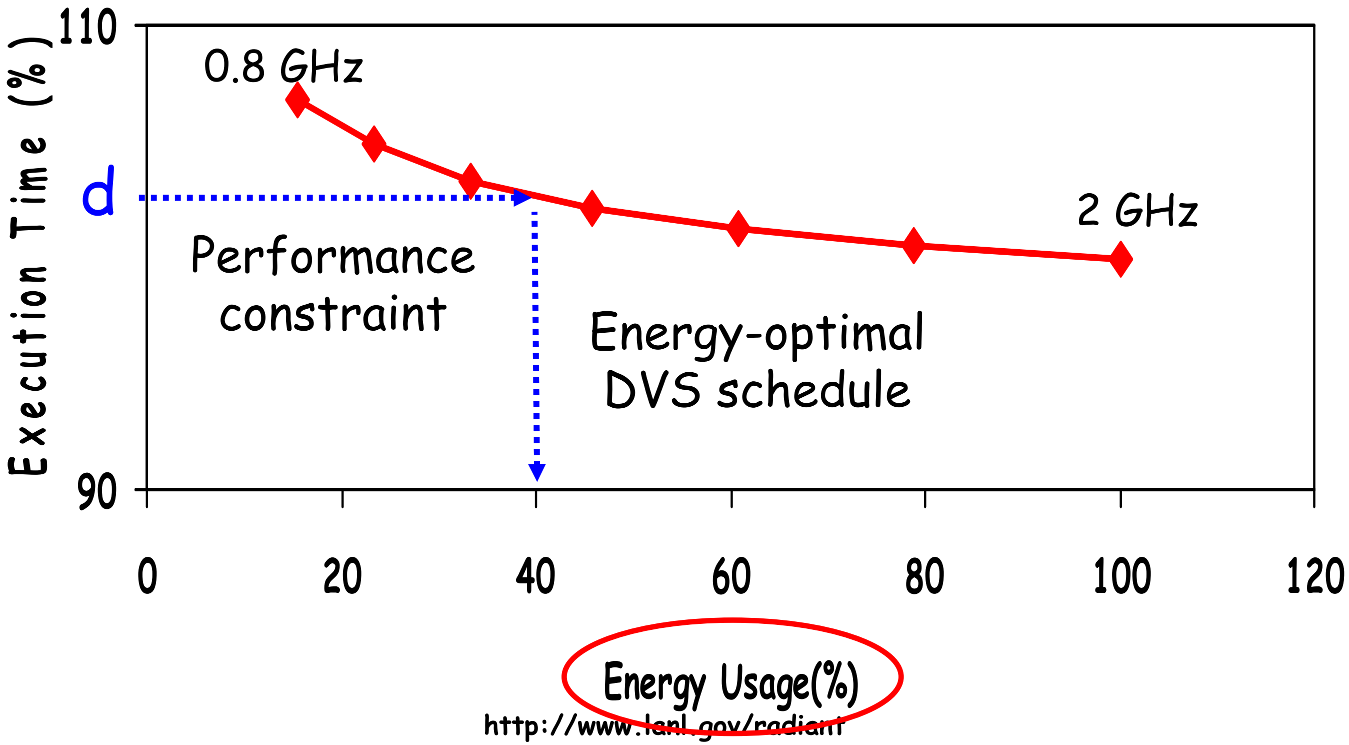
Clock speed slowed down by 50%

Wu Feng
feng@lanl.gov

Chung-Hsing Hsu
chunghsu@lanl.gov

Approach to Real-Time Constraint-Based DVS

- Key Idea
  - Applying DVS to these programs will result in significant power and energy savings at a minimal performance impact.



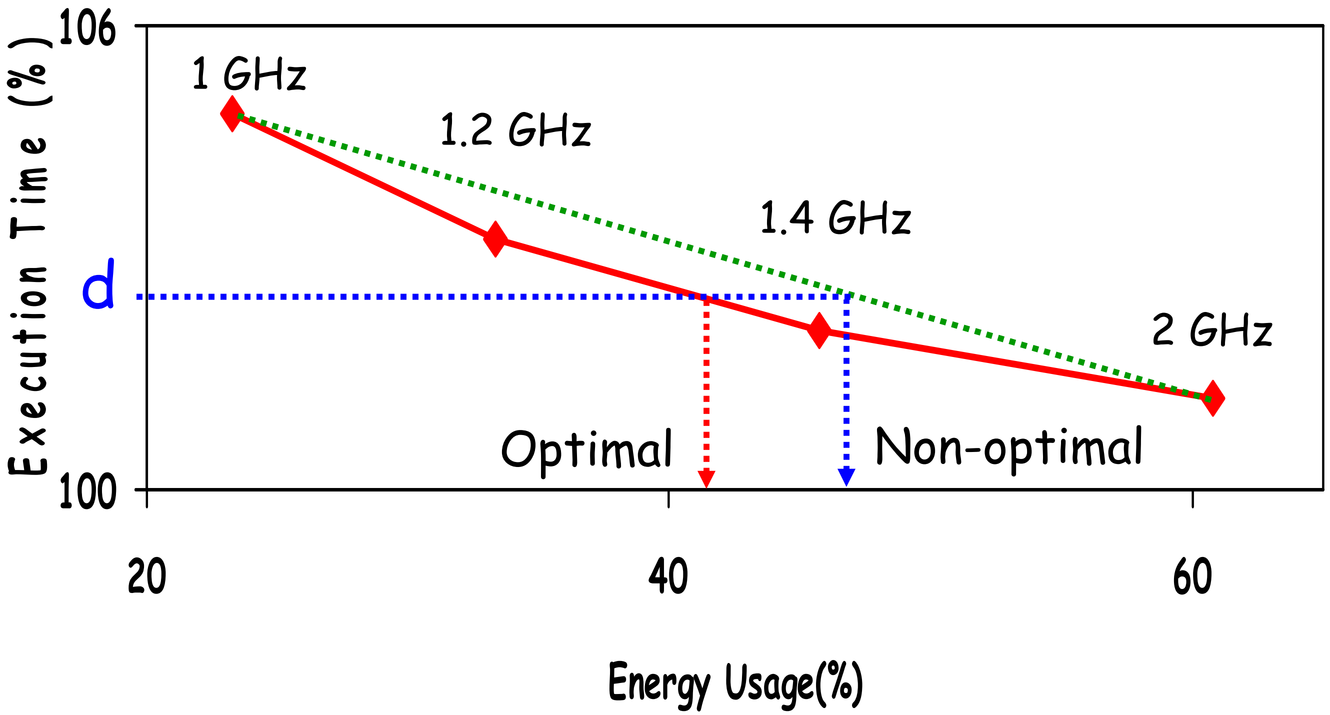
Problem Formulation for Real-Time Constraint-Based DVS

- Key Challenge
  - Find a performance-constrained, energy-optimal DVS schedule on a realistic processor in real time.
- Previous Related Work Targeted at Embedded Systems …
  - $P \alpha V^2 f$
    1. $P \alpha f^3$ [assumes $V \alpha f$]
    2. Discretize *V*. Use continuous mapping function, e.g., *f = g(V),* to get discrete *f*, e.g., 512 MHz, 894 MHz. Solve as ILP (offline) problem.
    3. Discretize *V* and *f*, e.g., AMD frequency-voltage table.
  - Simulation vs. Real Implementation
    - ☞ Problem with Simulation: Simplified Power Model
      - Does not account for leakage power.
      - Assumes zero-time switching overhead between *(f, V)* settings.
      - Assumes zero-time to construct a DVS schedule.
      - Does not assume realistic CPU support.

Creating an Energy-Optimal DVS Schedule

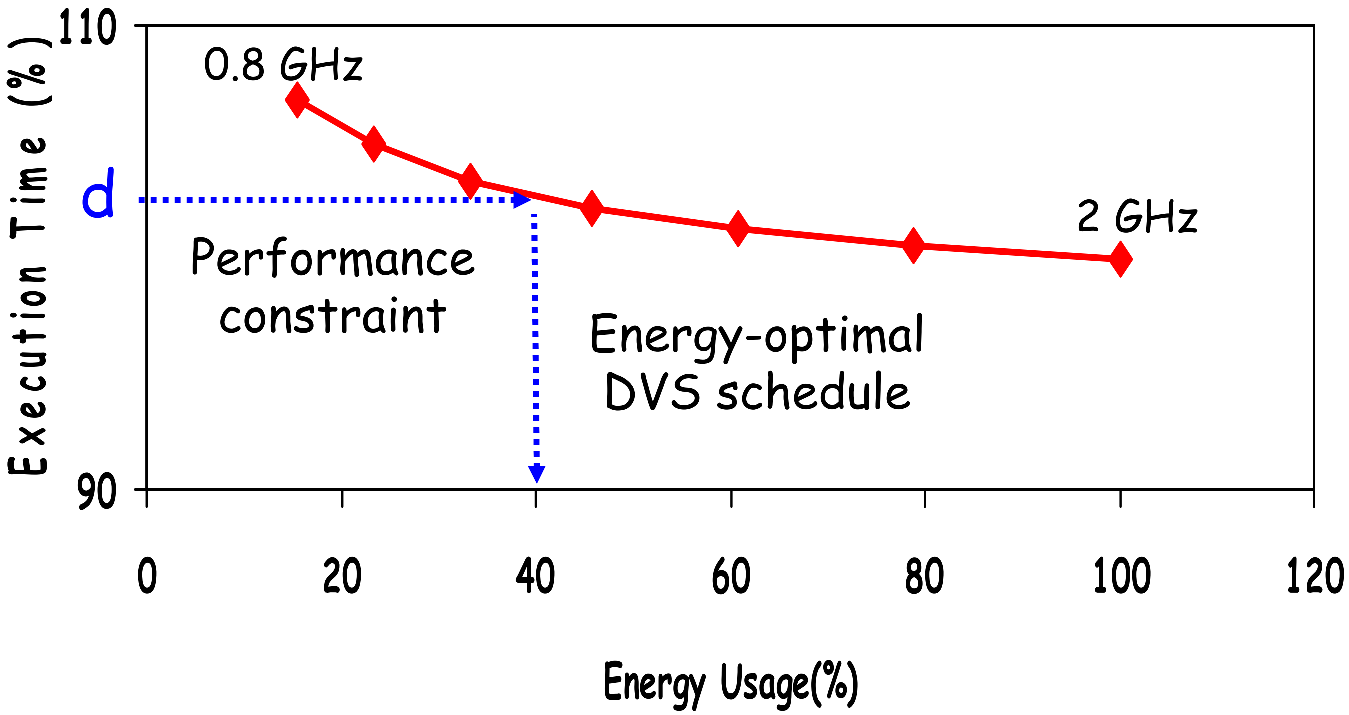
- Solve the following constraint-based problem:

$$E = \min\{\sum_i r_i \cdot E_i : \ \sum_i r_i \cdot T_i \le d, \ \sum_i r_i = 1, \ r_i \ge 0\}$$



Theorem for Real-Time Constraint-Based DVS

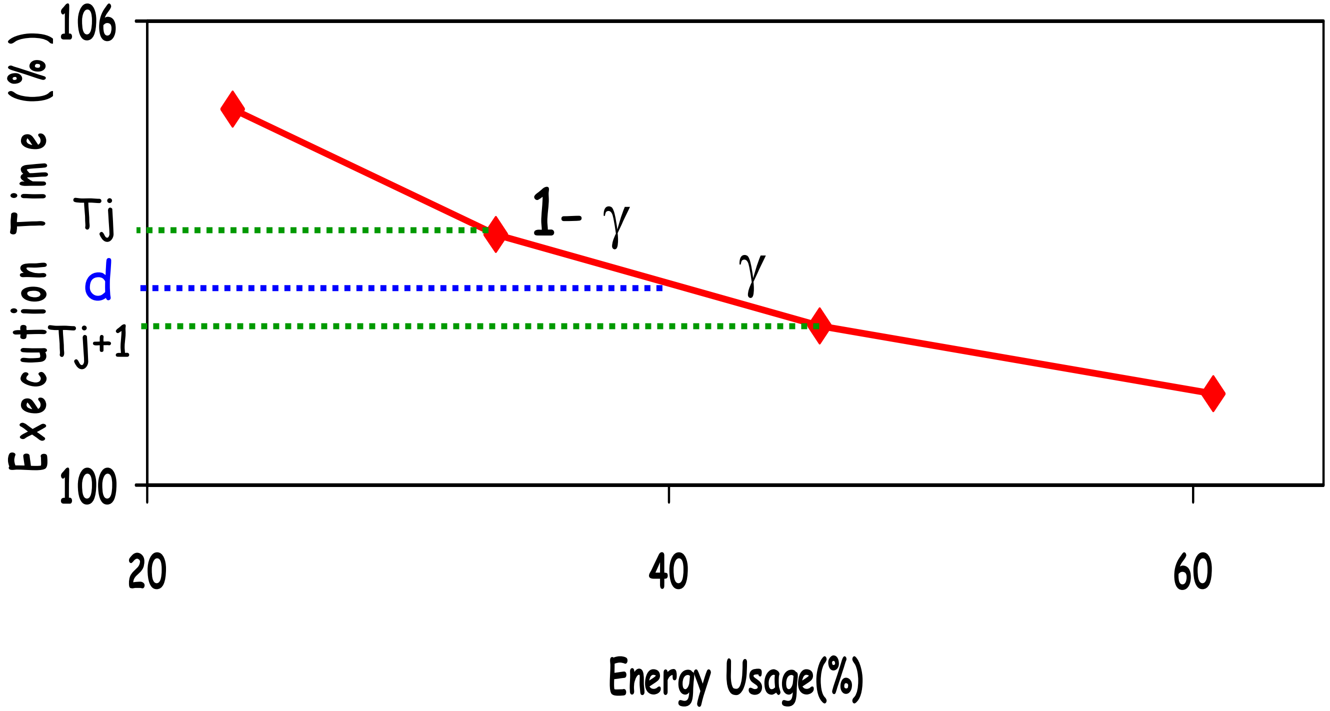
- If the execution-time vs. energy curve is convex, then the *energy-optimal DVS schedule* can be constructed in constant time.



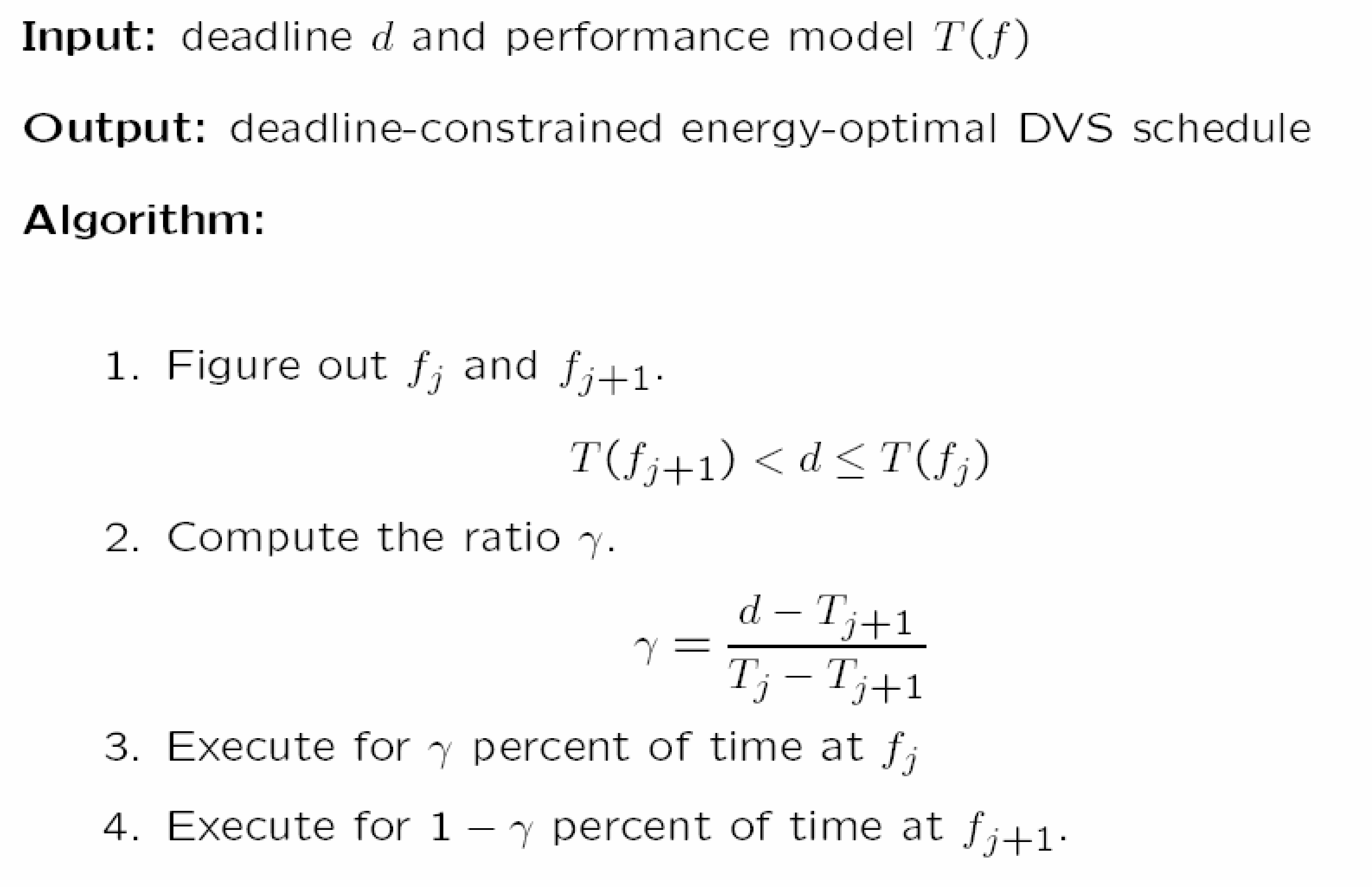
Emulating Frequencies for an Energy-Optimal DVS Schedule

$$E = \gamma \cdot E_j + (1-\gamma) \cdot E_{j+1} \text{ where}$$

$$\gamma = \frac{d - T_{j+1}}{T_j - T_{j+1}} \quad \text{and} \quad T_{j+1} < d \le T_j$$



DVS Scheduling Algorithm



Input: deadline d and performance model $T(f)$

Output: deadline-constrained energy-optimal DVS schedule

Algorithm:

1. Figure out f_j and f_{j+1}.

$$T(f_{j+1}) < d \leq T(f_j)$$

2. Compute the ratio γ.

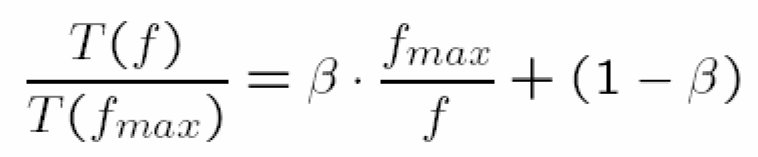
$$\gamma = \frac{d - T_{j+1}}{T_j - T_{j+1}}$$

3. Execute for γ percent of time at f_j

4. Execute for $1 - \gamma$ percent of time at f_{j+1}.

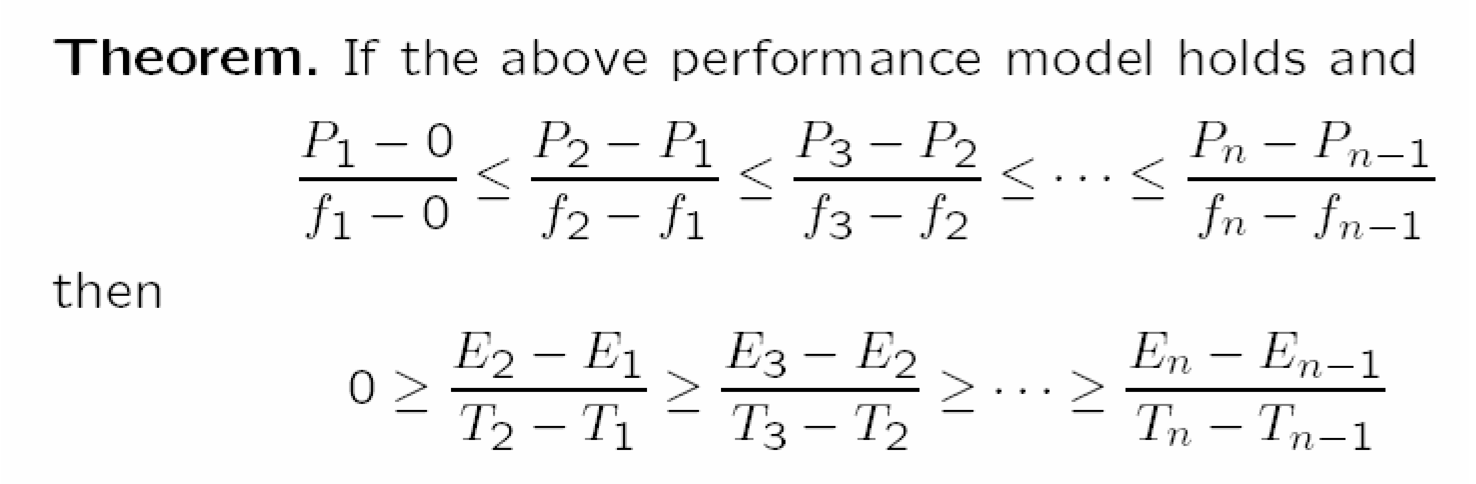
DVS Scheduling Algorithm

- Many programs can be modeled as follows:



$$\frac{T(f)}{T(f_{max})} = \beta \cdot \frac{f_{max}}{f} + (1 - \beta)$$

- To guarantee the execution-time vs. energy curve is convex, the following theorem is useful:



Theorem. If the above performance model holds and

$$\frac{P_1 - 0}{f_1 - 0} \leq \frac{P_2 - P_1}{f_2 - f_1} \leq \frac{P_3 - P_2}{f_3 - f_2} \leq \cdots \leq \frac{P_n - P_{n-1}}{f_n - f_{n-1}}$$

then

$$0 \geq \frac{E_2 - E_1}{T_2 - T_1} \geq \frac{E_3 - E_2}{T_3 - T_2} \geq \cdots \geq \frac{E_n - E_{n-1}}{T_n - T_{n-1}}$$

Initial Experimental Results

- Tested on a mobile AMD Athlon XP system with 5 settings
- Measured through Yokogawa WT210 digital power meter
- $\beta \in [0, 1]$ indicates performance sensitivity to changes in CPU speed, with 1 being most sensitive.

program	β	T_{rel}/E_{rel}
swim	0.02	1.02/0.46
tomcatv	0.24	1.01/0.80
su2cor	0.27	1.02/0.81
compress	0.37	1.05/0.80
mgrid	0.51	1.04/0.84
vortex	0.65	1.06/0.85
turb3d	0.79	1.04/0.92
go	1.00	1.05/0.93

Conclusion

- Efficiency, reliability, and availability will be *the* key issues of this decade.

- Performance Metrics for Green Destiny (circa 2002)
  - Performance: 2x to 2.5x worse than fastest AMD/Intel.
  - Price/Performance: 2x to 2.5x worse.
  - Overall Efficiency (Total Price-Performance Ratio)
    - 1.5x to 2.0x better. See ACM Queue, Oct. 2003.
  - Power Efficiency (Perf/Power): 10x to 20x better.
  - Space Efficiency (Perf/Space): 20x to 60x better.
  - Reliability: "Infinite"
  - Availability: Nearly 100%.

Conclusion

- Problem with Green Destiny
  - Architectural solution that sacrifices too much performance.
- Evolution of Green Destiny: Software-Based Solution
  - Real-time, constraint-based dynamic voltage scaling.
  - Performance on AMD XP-M
    - Power reduction of as much as 56% with only a 2% loss in performance.
  - Promising initial results on AMD Athlon-64 and Opteron.
- Future Directions
  - Calculation of β at run-time and at finer granularities.
  - Refinement of the DVS scheduling algorithm.
  - Profiling on multiprocessor platforms and benchmarks.

Acknowledgments

- Contributions to Green Destiny
  - Mike Warren, Eric Weigle, Mark Gardner, Adam Engelhart, Gus Hurwitz
- Encouragement & Support
  - Gordon Bell, Chris Hipp, and Linus Torvalds
- Funding Agencies
  - Los Alamos Computer Science Institute
  - IA-Linux at Los Alamos National Laboratory

SUPERCOMPUTING
in SMALL SPACES

Supercomputing
For the Rest of Us …

http://sss.lanl.gov

Wu Feng

Research and Development in Advanced Network Technology

RADIANT

http://www.lanl.gov/radiant