The Evolution of Power-Aware, High-Performance Computing: From the Datacenter to the Desktop

Wu Feng
feng@lanl.gov

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

Keynote Address
IEEE Int'l Parallel & Distributed Processing Symposium Workshop on High-Performance, Power-Aware Computing
4 April 2005
Outline

- **Motivation & Background**
  - Where is High-Performance Computing (HPC)?
  - The Need for Efficiency, Reliability, and Availability

- **Supercomputing in Small Spaces** ([http://sss.lanl.gov](http://sss.lanl.gov))
    - Architecture & Experimental Results
    - Architectural
      - MegaScale, Orion Multisystems, IBM Blue Gene/L
    - Software-Based
      - EnergyFit: Auto-adapting run-time system (β-adaptation algorithm)

- **Conclusion**

Wu FENG  
feng@lanl.gov

[http://www.lanl.gov/radiant](http://www.lanl.gov/radiant)  
[http://sss.lanl.gov](http://sss.lanl.gov)
Where is High-Performance Computing?
(Pictures: Thomas Sterling, Caltech & NASA JPL, and Wu Feng, LANL)

We have spent decades focusing on performance, performance, performance (and price/performance).
Where is High-Performance Computing?
Top 500 Supercomputer List

- Benchmark
  - LINPACK: Solves a (random) dense system of linear equations in double-precision (64 bits) arithmetic.
    - Introduced by Prof. Jack Dongarra, U. Tennessee

- Evaluation Metric
  - Performance (i.e., Speed)
    - Floating-Operations Per Second (FLOPS)

- Web Site
  - http://www.top500.org
Where is High-Performance Computing? Gordon Bell Awards at SC

- **Metrics for Evaluating Supercomputers (or HPC)**
  - *Performance (i.e., Speed)*
    - Metric: Floating-Operations Per Second (FLOPS)
    - Example: Japanese Earth Simulator, ASCI Thunder & Q.
  - *Price/Performance → Cost Efficiency*
    - Metric: Acquisition Cost / FLOPS
    - Examples: LANL Space Simulator, VT System X cluster.
    - (In general, Beowulf clusters.)

- Performance & price/performance are important metrics, but ...
Where is High-Performance Computing? (Unfortunate) Assumptions

- Humans are infallible.
  - No mistakes made during integration, installation, configuration, maintenance, repair, or upgrade.
- Software will eventually be bug free.
- Hardware MTBF is already very large (~100 years between failures) and will continue to increase.
- Acquisition cost is what matters; maintenance costs are irrelevant.

The above assumptions are even more problematic if one looks at current trends in HPC.
# Reliability & Availability of Leading-Edge Supercomputers

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

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

Source: Daniel A. Reed, UNC
Efficiency of Leading-Edge Supercomputers

“Performance” and “Price/Performance” Metrics ...
- Lower efficiency, reliability, and availability.
- Higher operational costs, e.g., admin, maintenance, etc.

Examples
- Computational Efficiency
  - Relative to Peak: Actual Performance/Peak Performance
  - Relative to Space: Performance/Sq. Ft.
  - Relative to Power: Performance/Watt
- 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.
Ubiquitous Need for Efficiency, Reliability, and Availability

- **Requirement:** Near-100% availability with efficient and reliable resource usage.
  - E-commerce, enterprise apps, online services, ISPs, data and HPC centers supporting R&D.

- **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/hour
    - Ebay (22 hours): $225,000/hour
    - Amazon.com: $180,000/hour
  - Social Effects: negative press, loss of customers who “click over” to competitor (e.g., Google vs. Ask Jeeves)

Source: David Patterson, UC-Berkeley
Where is High-Performance Computing?

(Pictures: Thomas Sterling, Caltech & NASA JPL and Wu Feng, LANL)

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

- Motivation & Background
  - Where is High-Performance Computing (HPC)?
  - The Need for Efficiency, Reliability, and Availability

- Supercomputing in Small Spaces (http://sss.lanl.gov)
    - Architecture & Experimental Results
    - Architectural
      - MegaScale, Orion Multisystems, IBM Blue Gene/L
    - Software-Based
      - EnergyFit: Auto-adapting run-time system
        (β-adaptation algorithm)

- Conclusion
Supercomputing in Small Spaces: Efficiency, Reliability, and Availability via Power-Aware HPC

- **Goal**
  - Improve efficiency, reliability, and availability (ERA) in large-scale computing systems.
    - Sacrifice a bit of raw performance.
    - Improve overall system throughput as the system will “always” be available, i.e., effectively no downtime, no HW failures, etc.
  - Reduce the total cost of ownership (TCO). Another talk ...

- **Crude Analogy**
  - Formula One Race Car: Wins raw performance but reliability is so poor that it requires frequent maintenance. Throughput low.
  - Honda S2000: Loses raw performance but high reliability results in high throughput (i.e., miles driven → answers/month).
How to Improve Efficiency, Reliability & Availability?

Observation

- High power density \( \alpha \) high temperature \( \alpha \) 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.

* The time to failure is a function of \( e^{-Ea/kT} \) where \( Ea \) = activation energy of the failure mechanism being accelerated, \( k \) = Boltzmann's constant, and \( T \) = absolute temperature

<table>
<thead>
<tr>
<th>Processor</th>
<th>Clock Freq.</th>
<th>Voltage</th>
<th>Peak Temp.**</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Pentium III-M</td>
<td>500 MHz</td>
<td>1.6 V</td>
<td>252° F (122° C)</td>
</tr>
<tr>
<td>Transmeta Crusoe TM5600</td>
<td>600 MHz</td>
<td>1.6 V</td>
<td>147° F (64° C)</td>
</tr>
</tbody>
</table>
Moore's Law for Power

Chip Maximum Power in watts/cm²

Year

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 efficient, reliable, and highly available but is still considered high performance?

Source: Fred Pollack, Intel. New Microprocessor Challenges in the Coming Generations of CMOS Technologies, MICRO32 and Transmeta
Moore's Law for Power

Chip Maximum Power in watts/cm²

Year

1985 1995 2001

Pentium 4 – 75 watts
Pentium III – 35 watts
Pentium II – 35 watts
Pentium Pro – 30 watts
Pentium – 14 watts

x86 – 5 watts
I386 – 1 watt
I486 – 2 watts

1 µm 0.7 µm 0.5 µm 0.35 µm 0.25 µm 0.18 µm 0.13 µm 0.1 µm 0.07 µm

Not too long to reach Nuclear Reactor

Source: Fred Pollack, Intel. New Microprocessor Challenges in the Coming Generations of CMOS Technologies, MICRO32 and Transmeta
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")**
  - ~ 25% of Intel PIII → ~ 7x less power consumption
  - Less power → lower “on-die” temp. → better reliability & availability
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?
Transmeta TM5x00 Comparison

<table>
<thead>
<tr>
<th></th>
<th>MEM</th>
<th>MEM</th>
<th>ALU</th>
<th>ALU</th>
<th>FPU</th>
<th>SSE</th>
<th>SSE</th>
<th>Br</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel P4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transmeta TM5x00</td>
<td>MEM</td>
<td></td>
<td>ALU</td>
<td></td>
<td>FPU</td>
<td></td>
<td></td>
<td>Br</td>
</tr>
</tbody>
</table>

- Previous-generation Transmeta TM5800 + HP-CMS
  - Performs better than an Intel PIII over iterative scientific codes on a clock-for-clock-cycle basis.
  - Performs only *twice* as slow as the fastest CPU (at the time) rather than three times as slow.
- Efficeon, the current-generation CPU from Transmeta, rectifies the above mismatch in functional units.
RLX ServerBlade™ 633 (circa 2000)

Modify the Transmeta CPU software to improve performance.

- Transmeta™ TM5600 633 MHz
- 128KB L1 cache, 512KB L2 cache
- LongRun, Northbridge, x86 compatible

Wu FENG
feng@lanl.gov

http://www.lanl.gov/radiant
http://sss.lanl.gov
RLX System™ 324 (circa 2000)

- 3U vertical space
  - 5.25” x 17.25” x 25.2”
- Two hot-pluggable 450W power supplies
  - Load balancing
  - Auto-sensing fault tolerance
- System midplane
  - Integration of system power, management, and network signals.
  - Elimination of internal system cables.
  - Enabling efficient hot-pluggable blades.
- Network cards
  - Hub-based management.
  - Two 24-port interfaces.

RLX System™ 300ex
- Interchangeable blades
  - Intel, Transmeta, or both.
- Switched-based management

Wu FENG
feng@lanl.gov

http://www.lanl.gov/radiant
http://sss.lanl.gov
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 April 2002)

- A 240-Node Beowulf Cluster in Five Sq. Ft.
- Each Node
  - 667-MHz Transmeta TM5600 CPU w/ Linux 2.4.x
    - Upgraded to 1-GHz Transmeta TM5800 CPUs
  - 640-MB RAM, 20-GB HD, 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 kW.
- Reliability & Availability
  - No unscheduled failures in 24 months.

Wu FENG
feng@lanl.gov

http://www.lanl.gov/radiant
http://sss.lanl.gov
Parallel Computing Platforms ("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

<table>
<thead>
<tr>
<th>Machine</th>
<th>Avalon Beowulf</th>
<th>ASCI Red</th>
<th>ASCI White</th>
<th>Green Destiny+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>1996</td>
<td>1996</td>
<td>2000</td>
<td>2002</td>
</tr>
<tr>
<td>Performance (Gflops)</td>
<td>18</td>
<td>600</td>
<td>2500</td>
<td>58</td>
</tr>
<tr>
<td>Area (ft²)</td>
<td>120</td>
<td>1600</td>
<td>9920</td>
<td>5</td>
</tr>
<tr>
<td>Power (kW)</td>
<td>18</td>
<td>1200</td>
<td>2000</td>
<td>5</td>
</tr>
<tr>
<td>DRAM (GB)</td>
<td>36</td>
<td>585</td>
<td>6200</td>
<td>150</td>
</tr>
<tr>
<td>Disk (TB)</td>
<td>0.4</td>
<td>2.0</td>
<td>160.0</td>
<td>4.8</td>
</tr>
<tr>
<td>DRAM density (MB/ft²)</td>
<td>300</td>
<td>366</td>
<td>625</td>
<td>30000</td>
</tr>
<tr>
<td>Disk density (GB/ft²)</td>
<td>3.3</td>
<td>1.3</td>
<td>16.1</td>
<td>960.0</td>
</tr>
<tr>
<td>Perf/Space (Mflops/ft²)</td>
<td>150</td>
<td>375</td>
<td>252</td>
<td>11600</td>
</tr>
<tr>
<td>Perf/Power (Mflops/watt)</td>
<td>1.0</td>
<td>0.5</td>
<td>1.3</td>
<td>11.6</td>
</tr>
</tbody>
</table>

Source: Michael S. Warren, Los Alamos National Laboratory
## Parallel Computing Platforms Running the N-body Code

<table>
<thead>
<tr>
<th>Machine</th>
<th>Avalon Beowulf</th>
<th>ASCI Red</th>
<th>ASCI White</th>
<th>Green Destiny+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>1996</td>
<td>1996</td>
<td>2000</td>
<td>2002</td>
</tr>
<tr>
<td>Performance (Gflops)</td>
<td>18</td>
<td>600</td>
<td>2500</td>
<td>58</td>
</tr>
<tr>
<td>Area (ft²)</td>
<td>120</td>
<td>1600</td>
<td>9920</td>
<td>5</td>
</tr>
<tr>
<td>Power (kW)</td>
<td>18</td>
<td>1200</td>
<td>2000</td>
<td>5</td>
</tr>
<tr>
<td>DRAM (GB)</td>
<td>36</td>
<td>585</td>
<td>6200</td>
<td>150</td>
</tr>
<tr>
<td>Disk (TB)</td>
<td>0.4</td>
<td>2.0</td>
<td>160.0</td>
<td>4.8</td>
</tr>
<tr>
<td>DRAM density (MB/ft²)</td>
<td>300</td>
<td>366</td>
<td>625</td>
<td>30000</td>
</tr>
<tr>
<td>Disk density (GB/ft²)</td>
<td>3.3</td>
<td>1.3</td>
<td>16.1</td>
<td>960.0</td>
</tr>
<tr>
<td>Perf/Space (Mflops/ft²)</td>
<td>150</td>
<td>375</td>
<td>252</td>
<td>11600</td>
</tr>
<tr>
<td>Perf/Power (Mflops/watt)</td>
<td>1.0</td>
<td>0.5</td>
<td>1.3</td>
<td>11.6</td>
</tr>
</tbody>
</table>

Source: Michael S. Warren, Los Alamos National Laboratory
Efficiency, Reliability, and Availability for ...  

- **Green Destiny+**
  - **Computational Efficiency**
    - Relative to Space: Performance/Sq. Ft.
      - *Up to 80x better.*
    - Relative to Power: Performance/Watt
      - *Up to 25x better.*
  - **Reliability**
    - MTBF: Mean Time Between Failures
      - "Infinite"
  - **Availability**
    - Percentage of time that resources are available for HPC.
      - *Nearly 100%.*
Q&A Exchange with Pharmaceutical Companies

- Pharmaceutical: “Can you get the same type of results for bioinformatics applications?”
- Wu: “What is your primary application?”
- Pharmaceutical: “BLAST …”

J. Craig Venter in GenomeWeb on Oct. 16, 2002.

“… to build something that is replicable so any major medical center around the world can have a chance to do the same level of computing … interested in IT that doesn’t require massive air conditioning. The room at Celera cost $6M before you put the computer in. [Thus, I am] looking at these new green machines being considered at the DOE that have lower energy requirements” & therefore produce less heat.
mpiBLAST (http://mpiblast.lanl.gov)
Performance on Green Destiny

mpiBLAST
- An open-source parallelization of BLAST based on MPI and in-memory database segmentation.
- Downloaded over 10,000 times in two years.

BLAST Run Time for 300-kB Query against nt

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Runtime (s)</th>
<th>Speedup over 1 node</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>80774.93</td>
<td>1.00</td>
</tr>
<tr>
<td>4</td>
<td>8751.97</td>
<td>9.23</td>
</tr>
<tr>
<td>8</td>
<td>4547.83</td>
<td>17.76</td>
</tr>
<tr>
<td>16</td>
<td>2436.60</td>
<td>33.15</td>
</tr>
<tr>
<td>32</td>
<td>1349.92</td>
<td>59.84</td>
</tr>
<tr>
<td>64</td>
<td>850.75</td>
<td>94.95</td>
</tr>
<tr>
<td>128</td>
<td>473.79</td>
<td>170.49</td>
</tr>
</tbody>
</table>

The Bottom Line
- mpiBLAST reduces search time from 1346 minute (or 22.4 hours) to under 8 minutes.
Outline

- Motivation & Background
  - Where is High-Performance Computing (HPC)?
  - The Need for Efficiency, Reliability, and Availability
- Supercomputing in Small Spaces (http://sss.lanl.gov)
    - Architecture & Experimental Results
    - Architectural
      - MegaScale, Orion Multisystems, IBM Blue Gene/L
    - Software-Based
      - EnergyFit: Auto-adapting run-time system
        (β-adaptation algorithm)
- Conclusion
Inter-University Project: MegaScale
http://www.para.tutics.tut.ac.jp/megascale/
IBM Blue Gene/L

System
(64 cabinets, 64x32x32)

Cabinet
(32 Node boards, 8x8x16)

Node Card
(32 chips, 4x4x2)
16 Compute Cards

Compute Card
(2 chips, 2x1x1)

Chip
(2 processors)

2.8/5.6 GF/s
4 MB

5.6/11.2 GF/s
0.5 GB DDR

90/180 GF/s
8 GB DDR

2.9/5.7 TF/s
256 GB DDR

October 2003
BG/L half rack prototype
500 Mhz
512 nodes/1024 proc.
2 TFlop/s peak
1.4 Tflop/s sustained

© 2004 IBM Corporation

Wu FENG
feng@lanl.gov

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

© 2004 IBM Corporation
The Road from **Green Destiny** to Orion Multisystems

- **Trends in High-Performance Computing**
  - Rise of cluster-based high-performance computers.
    - Price/performance advantage of using “commodity PCs” as cluster nodes (Beowulf: 1993-1994.)
    - Different flavors: “homebrew” vs. “custom”
The Road from **Green Destiny** to Orion Multisystems

- **Trends in High-Performance Computing**
  - Rise of cluster-based high-performance computers.
    - Price/performance advantage of using “commodity PCs” as cluster nodes (Beowulf: 1993-1994.)
    - Different flavors: “homebrew” vs. “custom”
  - Maturity of open-source cluster software.
    - Emergence of Linux and MPI as parallel programming APIs.
  - Rapid decline of the traditional workstation.
    - Replacement of workstation with a PC.
    - 1000-fold (and increasing) performance gap with respect to the supercomputer.
    - Still a desperate need for HPC in workstation form.
Evolution of Workstations: Performance Trends

- PC performance caught up with workstations
  - PC OSes: NT and Linux

- A large gap has opened between PCs and supercomputers
  - 3 Gflops vs. 3 Tflops

Source: Orion Multisystems, Inc.

Widening Gap of Opportunity for Orion Cluster Workstation

1981-1996
Heyday of the Workstation

Advent of Clustering

Supercomputers
Workstations
PCs


feng@lanl.gov
http://sss.lanl.gov
Need: A Cluster Workstation

Specifications
- Desktop or deskside box with cluster inside
- A cluster product - not an assembly
- Scalable computation, graphics, and storage
- Meets power limits of office or laboratory

Reality of (Homebrew) Clusters
- Ad-hoc, custom-built collections of boxes
- Hard for an individual to get exclusive access (or even share access)
- Power-, space-, and cooling-intensive
- IT support required

Source: Orion Multisystems, Inc.
Why a Cluster Workstation?

- **Personal Resource**
  - No scheduling conflicts or long queues.
  - Application debugging with scalability at your desktop
  - Redundancy possibilities (eliminate downtime)

- **Improvement of Datacenter Efficiency**
  - Off-load “repeat offender” jobs
  - Enable developers to debug their own code on their own system
  - Manage expectations
  - Reduce job turnaround time

Source: Orion Multisystems, Inc.
Cluster Technology

Low-Power Systems Design

Linux

But in the form factor of a workstation ... a cluster workstation
ORION DT-12 DESKTOP CLUSTER WORKSTATION

Imagine a 36 Gflop cluster on your desk!

- **LINPACK Performance**
  - 13.80 Gflops

- **Footprint**
  - 3 sq. ft. (24" x 18")
  - 1 cu. ft. (24" x 4" x 18")

- **Power Consumption**
  - 170 watts at load

- **How does this compare with a traditional desktop?**

  12 Nodes
  in a single computer

  36 Gflops
  peak processing power

  24 GBytes
  memory capacity

  1 TByte
  internal storage

DESIGNED FOR THE INDIVIDUAL
The Orion DT-12 cluster workstation is a fully integrated, completely self-contained, personal workstation based on the best of today's cluster technologies. Designed to be an affordable individual resource it is capable of 36 Gflops peak performance (18 Gflops sustained) with models starting at under $10k.

The Orion DT-12 cluster workstation provides supercomputer performance for the engineering, scientific, financial and creative professionals who need to solve computationally complex problems without waiting in the queue of the back-room cluster.

FASTER SOFTWARE DEVELOPMENT
The Orion DT-12 cluster workstation is the perfect platform for developers writing (and deploying) cluster software packages. It comes with cluster software development tools pre-installed, including libraries and a parallel compiler that allows you to spread one multiple-file compile to all the nodes in the system. Also included is a suite of system monitoring and management software.

NO ASSEMBLY REQUIRED
Orion workstations are designed from the ground up as a single computer. The entire system boots with the push of a button and has the ergonomics and ease of use of a personal computer. The modular design allows for flexible configurations and scalability by stacking up to 4 systems as one 48 node cluster.

PRESEVER SOFTWARE INVESTMENTS
Orion workstations are built around industry standards for clustering: x86 processors, Ethernet, the Linux operating system and standard parallel programming libraries, including MPI, PVM and SGE. Existing Linux cluster applications run without modification.

PERFORMANCE AND FEATURES
The Orion DT-12 is a cluster of 12 x86-compatible nodes linked by a switched Gigabit Ethernet fabric. The cluster operates as a single computer with a single on-off switch and a single system image rapid boot sequence, which allows the entire system to boot in less than 90 seconds.

The Orion DT-12 cluster workstation is highly efficient, consuming a maximum of 220 Watts of power under peak load—about the same as an average desktop PC. It operates quietly, plugs into a standard 110V 15A wall socket and fits unobtrusively on a desk or lab bench.
What's Inside?
Orion Multisystems' Workstation Architecture

Source: Orion Multisystems, Inc.
ORION DS-96 DESKSIDE CLUSTER WORKSTATION

Imagine a 300 Gflop cluster... under your desk.

96 Nodes
in a single computer

300 Gflops
peak processing power

192 GBytes
memory capacity

9.6 TBytes
internal storage

INCREASE YOUR PRODUCTIVITY
The Orion DS-96 cluster workstation is the highest performance general-purpose computing platform that can be plugged into a standard wall outlet and operated in an office or laboratory environment.

PRESERVE SOFTWARE INVESTMENTS
Orion workstations are built around industry standards for clustering: x86 processors, the Linux operating system and standard parallel programming libraries, including MPI, PVM and SGE. Your existing Linux cluster software applications can run without modification.

NO ASSEMBLY REQUIRED
Orion workstations are designed from the ground up as a single computer. The entire system boots with the push of a button and has the ergonomics and ease of use of a personal computer. Modular solid state design allows for flexible configurations and scalability.

PERFORMANCE AND FEATURES
The Orion DS-96 cluster workstation is a fully integrated, completely self-contained personal workstation based on the best of today’s cluster technologies and commodity components. Designed to be an individual or departmental resource, it is capable of 300 Gflops peak performance (150 Gflops sustained). The DS-96 is also highly efficient, consuming a maximum of 1500 Watts of power under peak load. It operates quietly, plugs into a standard 110V/15A wall socket, and fits unobtrusively beneath a desk or lab bench.

The DS-96 is a cluster of 96 x86-compatible nodes linked by an integrated Gigabit Ethernet fabric. The cluster operates as a single computer, with a single on/off switch, and a single-system-image rapid boot sequence which allows the entire system to boot in less than 2 minutes. The DS-96 comes with standard Linux and drivers pre-installed, including an optimized MPI message-passing library. Also included is a suite of cluster software development tools, system monitoring and system management software.

Road to Tflop?
◆ 10 DS-96s → ~ 1 Tflop LINPACK

LINPACK Performance
◆ 110.8 Gflops

Footprint
◆ 3 sq. ft. (17" x 25")
◆ 6 cu. ft. (17" x 25" x 25")

Power Consumption
◆ 1580 watts at load

Recall....
GD: 101 Gflops

Wu FENG
feng@lanl.gov

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

Los Alamos
National Laboratory
## Parallel Computing Platforms Running LINPACK

<table>
<thead>
<tr>
<th>Machine</th>
<th>ASCI Red</th>
<th>ASCI White</th>
<th>Green Destiny+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>1996</td>
<td>2000</td>
<td>2002</td>
</tr>
<tr>
<td>Performance (Gflops)</td>
<td>2379</td>
<td>7226</td>
<td>101.0</td>
</tr>
<tr>
<td>Area (ft²)</td>
<td>1600</td>
<td>9920</td>
<td>5</td>
</tr>
<tr>
<td>Power (kW)</td>
<td>1200</td>
<td>2000</td>
<td>5</td>
</tr>
<tr>
<td>DRAM (GB)</td>
<td>585</td>
<td>6200</td>
<td>150</td>
</tr>
<tr>
<td>Disk (TB)</td>
<td>2.0</td>
<td>160.0</td>
<td>4.8</td>
</tr>
<tr>
<td>DRAM density (MB/ft²)</td>
<td>366</td>
<td>625</td>
<td>30000</td>
</tr>
<tr>
<td>Disk density (GB/ft²)</td>
<td>1</td>
<td>16</td>
<td>960</td>
</tr>
<tr>
<td>Perf/Space (Mflops/ft²)</td>
<td>1487</td>
<td>728</td>
<td>20202</td>
</tr>
<tr>
<td>Perf/Power (Mflops/watt)</td>
<td>2</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>Machine</td>
<td>ASCI Red</td>
<td>ASCI White</td>
<td>Green Destiny+</td>
</tr>
<tr>
<td>------------------</td>
<td>----------</td>
<td>------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Year</td>
<td>1996</td>
<td>2000</td>
<td>2002</td>
</tr>
<tr>
<td>Performance (Gflops)</td>
<td>2379</td>
<td>7226</td>
<td>101.0</td>
</tr>
<tr>
<td>Area (ft²)</td>
<td>1600</td>
<td>9920</td>
<td>5</td>
</tr>
<tr>
<td>Power (kW)</td>
<td>1200</td>
<td>2000</td>
<td>5</td>
</tr>
<tr>
<td>DRAM (GB)</td>
<td>585</td>
<td>6200</td>
<td>150</td>
</tr>
<tr>
<td>Disk (TB)</td>
<td>2.0</td>
<td>160.0</td>
<td>4.8</td>
</tr>
<tr>
<td>DRAM density (MB/ft²)</td>
<td>366</td>
<td>625</td>
<td>30000</td>
</tr>
<tr>
<td>Disk density (GB/ft²)</td>
<td>1</td>
<td>16</td>
<td>960</td>
</tr>
<tr>
<td>Perf/Space (Mflops/ft²)</td>
<td>1487</td>
<td>728</td>
<td>20202</td>
</tr>
<tr>
<td>Perf/Power (Mflops/watt)</td>
<td>2</td>
<td>4</td>
<td>20</td>
</tr>
</tbody>
</table>
Outline

- Motivation & Background
  - Where is High-Performance Computing (HPC)?
  - The Need for Efficiency, Reliability, and Availability

- Supercomputing in Small Spaces (http://sss.lanl.gov)
    - Architecture & Experimental Results
    - Architectural
      - MegaScale, Orion Multisystems, IBM Blue Gene/L
    - Software-Based
      - EnergyFit: Auto-adapting run-time system
        (β-adaptation algorithm)

- Conclusion
Power-Aware HPC Today: The Start of a New Movement

- **Traditional View of Power Awareness**
  - Extend battery life in laptops, sensors, and embedded systems (such as PDAs, handhelds, and mobile phones)

- **Controversial View of Power Awareness (2001-2002)**
  - *Potentially* sacrifice a bit of performance to enhance efficiency, reliability, and availability in HPC systems
  - *Gripe:* HPC unwilling to “sacrifice” performance

- **The Start of a New Movement (2004-2005)**
DVS Mechanism
- 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 Scheduling Algorithm
- Determines when to adjust the current frequency-voltage setting and what the new frequency-voltage setting should be.

"... and leakage current varies as the cube of frequency ..."
The execution time of many programs is insensitive to CPU speed change (because the processor-memory performance gap, i.e., the memory wall, routinely limits performance of scientific codes).

![Graph showing the relationship between CPU speed (GHz) and execution time (Performance degraded by 4%)](image)
Applying DVS to these programs (i.e., embracing the memory wall) will result in significant power and energy savings at a minimal performance impact.
Problem Formulation: LP-Based Energy-Optimal DVS Schedule

- **Definitions**
  - A DVS system exports \( n \) \( \{ (f_i, P_i) \} \) settings.
  - \( T_i \): total execution time of a program running at setting \( i \)
- **Given a program with deadline \( D \), find a DVS schedule \( (t_1^*, \ldots, t_n^*) \) such that**
  - If the program is executed for \( t_i \) seconds at setting \( i \), the total energy usage \( E \) is minimized, the deadline \( D \) is met, and the required work is completed.

\[
\min E = \sum_i P_i \cdot t_i
\]

subject to

\[
\begin{align*}
\sum_i t_i &\leq D \\
\sum_i t_i/T_i &\leq 1 \\
t_i &\geq 0
\end{align*}
\]
Related Work in Power-Aware (Embedded) Computing

From an ad-hoc “power” perspective

- \( P \propto V^2 f \)
  1. Simplify to \( P \propto f^3 \) [assumes \( V \propto f \)]
  2. Discretize \( V \). Use continuous mapping function, e.g., \( f = g(V) \), to get discrete \( f \). Solve as ILP (offline) problem.

- Simulation-based research with simplified power model
  1. Does not account for leakage power.
  2. Assumes zero-time switching overhead between \((f, V)\) settings.
  3. Assumes zero-time to construct a DVS schedule.
  4. Does not assume realistic CPU support.

- Recent examples based on more realistic power model
  1. Compile-time (static) DVS using profiling information.
     ACM SIGPLAN PLDI, June 2003.
  2. Run-time (dynamic) DVS via an auxiliary HW circuit.

Discretize \( V \) and \( f \), e.g., AMD frequency-voltage table.

Realistic power model.

Automatic DVS adaptation at run time with low overhead.
Related Work in Power-Aware (Embedded) Computing

From a “performance modeling” perspective ...

- Traditional Performance Model
  \[ T(f) = \frac{1}{f} \times W \]
  where \( T(f) \) (in seconds) is the execution time of a task running at \( f \) and \( W \) (in cycles) is the amount of CPU work to be done.

- Problems?
  - \( W \) needs to be known a priori. Difficult to predict.
  - \( W \) is not always constant across frequencies.
  - It predicts that the execution time will double if the CPU speed is cut in half. (Not so for memory & I/O-bound.)
Related Work in Power-Aware (Embedded) Computing

- Re-Formulated Performance Model
  Two-Coefficient Performance Model
  \[ T(f) = W_{CPU} / f + T_{MEM} \]
  where \( W_{CPU} / f \) models on-chip workload (in cycles)
  \( T_{MEM} \) models off-chip accesses (invariant to CPU)

- Problems?
  - This breakdown of the total execution time is inexact when the target processor supports out-of-order execution because on-chip execution may overlap with off-chip accesses.
  - \( W_{CPU} \) and \( T_{MEM} \) must be known a priori and are oftentimes determined by the hardware platform, program source code, and data input.
Problem Formulation Based on Single-Coefficient $\beta$ Perf. Model

- Our Formulation: Single-Coefficient $\beta$ Performance Model
  - Define the relative performance slowdown $\delta$ as
    $$ \frac{T(f)}{T(f_{MAX})} - 1 $$
  - Re-formulate previous two-coefficient model as a single-coefficient model:
    $$ \frac{T(f)}{T(f_{MAX})} = \beta \cdot \frac{f_{MAX}}{f} + (1 - \beta) $$
    with
    $$ \beta = \frac{W_{cpu}}{W_{cpu} + T_{mem} \cdot f_{MAX}} $$
  - The coefficient $\beta$ is computed at run-time using a regression method on the past MIPS rates reported from the built-in PMU.
    $$ \beta = \frac{\sum_i (\frac{f_{MAX}}{f_i} - 1)(\frac{mips(f_{MAX})}{mips(f_i)} - 1)}{\sum_i (\frac{f_{MAX}}{f_i} - 1)^2} $$

Wu FENG
feng@lanl.gov

http://www.lanl.gov/radiant
http://sss.lanl.gov
How to Determine $f$?

- Solve the following optimization problem:
  - $\min \{ P(f): \frac{T(f)}{T(f_{\text{max}})} \leq 1 + \delta \}$
  - $= \min \{ P(f): \beta \frac{f_{\text{max}}}{f} + (1 - \beta) \leq 1 + \delta \}$
  - $= \min \{ P(f): f \geq f_{\text{max}} / (1 + \delta / \beta) \}$

- If the power function $P(f)$ is an increasing function, then we can describe the desired frequency $f^*$ in a closed form:
  - $f^* = \max (f_{\text{min}}, f_{\text{max}} / (1 + \delta / \beta))$
**β-Adaptation DVS Scheduling Algorithm**

- **Input:** Relative slowdown $\delta$ and performance model $T(f)$.
- **Output:** Constraint-based DVS schedule.
- **For every $I$ seconds do**
  1. Compute coefficient $\beta$
  2. Compute the desired frequency $f^*$
     - If $f^*$ is not a supported frequency, then
       1. Identify $f_j$ and $f_{j+1}$.
       2. Compute the ratio $r$.
       3. Run $r \cdot I$ seconds at frequency $f_j$.
       4. Run $(1 - r) \cdot I$ seconds at frequency $f_{j+1}$.
       5. Update $\text{mips}(f_j)$ and $\text{mips}(f_{j+1})$.
     - Else run at $f^*$.

\[
\begin{align*}
  f^* &= \begin{cases} 
    f_{\min} & \text{if } \beta \leq \delta \\
    f_{\max}/(1 + \delta/\beta) & \text{otherwise}
  \end{cases} \\
  r &= \frac{(1 + \delta/\beta)/f_{\max} - 1/f_{j+1}}{1/f_j - 1/f_{j+1}}
\end{align*}
\]
Experimental Set-Up

- Profiling Computer
- Wall Power Outlet
- Digital Power Meter
- Power Strip
- AC Adapter
- Tested Computer
Experimental Specifics

- Tested Computer Platforms with PowerNow! Enabled
  - Mobile AMD Athlon XP (with five frequency-voltage settings) - same processor used in the Sun Blade System.
  - 64-bit AMD Athlon 64
  - 64-bit AMD Opteron → CAFfeine Power-Aware Cluster

- Digital Power Meter
  - Yokogawa WT210: Continuously samples every 20 $\mu$s.

- Benchmarks Used
  - Uniprocessor: SPEC.
  - Multiprocessor: mpiBLAST, NAS, and LINPACK.
Current DVS Scheduling Algorithms

- **2step** (i.e., SpeedStep):
  - Using a dual-speed CPU, monitor CPU utilization periodically.
  - If `utilization` > pre-defined upper threshold, set CPU to fastest.
  - If `utilization` < pre-defined lower threshold, set CPU to slowest.

- **nqPID**: A refinement of the 2step algorithm.
  - Recognize the similarity of DVS scheduling and a classical control-systems problem → Modify a PID controller (Proportional-Integral-Derivative) to suit the DVS scheduling problem.

- **freq**: Reclaims the slack time between the actual processing time and the worst-case execution time.
  - Track the amount of remaining CPU work $W_{left}$ and the amount of remaining time before the deadline $T_{left}$.
  - The desired CPU frequency $f_{new}$ at each interval is simply $f_{new} = W_{left} / T_{left}$.
  - The algorithm assumes that the total amount of work in CPU cycles is known a priori, which, in practice, is often unpredictable and not always a constant across frequencies.
Current DVS Scheduling Algorithms

- **mips**: A DVS strategy guided by an externally specified performance metric. Specifically, the new frequency $f_{\text{new}}$ at each interval is computed by

\[ f_{\text{new}} = f_{\text{prev}} \cdot \frac{\text{MIPS}_{\text{target}}}{\text{MIPS}_{\text{observed}}} \]

where $f_{\text{prev}}$ is the frequency for the previous interval, $\text{MIPS}_{\text{target}}$ is the externally specified performance requirement, and $\text{MIPS}_{\text{observed}}$ is the real MIPS rate observed in the previous interval.
## SPEC Performance Results

<table>
<thead>
<tr>
<th>program</th>
<th>$\beta$</th>
<th>2step</th>
<th>nqPID</th>
<th>freq</th>
<th>mips</th>
<th>beta</th>
</tr>
</thead>
<tbody>
<tr>
<td>swim</td>
<td>0.02</td>
<td>1.00/1.00</td>
<td>1.04/0.70</td>
<td>1.00/0.96</td>
<td>1.00/1.00</td>
<td>1.04/0.61</td>
</tr>
<tr>
<td>tomcatv</td>
<td>0.24</td>
<td>1.00/1.00</td>
<td>1.03/0.69</td>
<td>1.00/0.97</td>
<td>1.03/0.83</td>
<td>1.00/0.85</td>
</tr>
<tr>
<td>su2cor</td>
<td>0.27</td>
<td>0.99/0.99</td>
<td>1.05/0.70</td>
<td>1.00/0.95</td>
<td>1.01/0.96</td>
<td>1.03/0.85</td>
</tr>
<tr>
<td>compress</td>
<td>0.37</td>
<td>1.02/1.02</td>
<td>1.13/0.75</td>
<td>1.02/0.97</td>
<td>1.05/0.92</td>
<td>1.01/0.95</td>
</tr>
<tr>
<td>mgrid</td>
<td>0.51</td>
<td>1.00/1.00</td>
<td>1.18/0.77</td>
<td>1.01/0.97</td>
<td>1.00/1.00</td>
<td>1.03/0.89</td>
</tr>
<tr>
<td>vortex</td>
<td>0.65</td>
<td>1.01/1.00</td>
<td>1.25/0.81</td>
<td>1.01/0.97</td>
<td>1.07/0.94</td>
<td>1.05/0.90</td>
</tr>
<tr>
<td>turb3d</td>
<td>0.79</td>
<td>1.00/1.00</td>
<td>1.29/0.83</td>
<td>1.03/0.97</td>
<td>1.01/1.00</td>
<td>1.05/0.94</td>
</tr>
<tr>
<td>go</td>
<td>1.00</td>
<td>1.00/1.00</td>
<td>1.37/0.88</td>
<td>1.02/0.99</td>
<td>0.99/0.99</td>
<td>1.06/0.96</td>
</tr>
</tbody>
</table>

### relative time / relative energy
with respect to total execution time and system energy usage

- $\beta$ indicates performance sensitivity to changes in CPU speed (with $\beta = 1$ being the most sensitive).
SPEC Insights ...

- **β-Adaptation Algorithm (a.k.a. EnergyFit)**
  - Delivers low-overhead adaptation of $f$ and $V$ *and* simultaneously provides tight control over performance loss by effectively exploiting sub-linear performance slowdown.

- **nqPID Algorithm**
  - Provides more power and energy reduction but at the cost of loose control over performance loss.

- **mips Algorithm**
  - Provides tight control over performance loss but does not save as much power or energy.

- **2step and freq Algorithms**
  - CPU utilization clearly does *not* provide enough information.
### SPEC Performance Results vs. ACM SIGPLAN PLDI '03

Source: C. Hsu

<table>
<thead>
<tr>
<th>program</th>
<th>$\beta$</th>
<th>Hsu (training)</th>
<th>beta adaptation</th>
</tr>
</thead>
<tbody>
<tr>
<td>swim</td>
<td>0.02</td>
<td>1.01 / 0.75</td>
<td>1.04 / 0.61</td>
</tr>
<tr>
<td>tomcatv</td>
<td>0.14</td>
<td>1.03 / 0.70</td>
<td>1.00 / 0.85</td>
</tr>
<tr>
<td>hydro2d</td>
<td>0.19</td>
<td>1.03 / 0.75</td>
<td>1.02 / 0.84</td>
</tr>
<tr>
<td>su2cor</td>
<td>0.27</td>
<td>1.01 / 0.88</td>
<td>1.03 / 0.85</td>
</tr>
<tr>
<td>applu</td>
<td>0.34</td>
<td>1.03 / 0.87</td>
<td>1.04 / 0.85</td>
</tr>
<tr>
<td>apsi</td>
<td>0.37</td>
<td>1.03 / 0.85</td>
<td>1.05 / 0.83</td>
</tr>
<tr>
<td>mgrid</td>
<td>0.51</td>
<td>1.01 / 1.00</td>
<td>1.03 / 0.89</td>
</tr>
<tr>
<td>wave5</td>
<td>0.52</td>
<td>1.00 / 1.00</td>
<td>1.04 / 0.87</td>
</tr>
<tr>
<td>turb3d</td>
<td>0.79</td>
<td>1.04 / 0.95</td>
<td>1.05 / 0.94</td>
</tr>
<tr>
<td>fpppp</td>
<td>1.00</td>
<td>1.00 / 1.00</td>
<td>1.06 / 0.95</td>
</tr>
</tbody>
</table>

Beta adaptation

Source: C. Hsu
**CAFfeine: 10GigE**

**Power-Aware Supercomputer**

- **Network**
  - Fujitsu XG800 12-port 10GigE Switch
    - Flow-Through Latency: < 1 µs!

- **Compute Node**
  - Celestica AMD Quartet A8440
    - CPU: Four AMD Opterons w/ PowerNow!
    - Memory: 4-GB DDR333 SDRAM
    - Storage: 80-GB, 7200-rpm HD
    - Interfaces: Two independent PCI-X buses
    - Network Adapter: Chelsio Communications T110

- **Performance**
  - Up to 60% power reduction with only 1-6% performance impact on SPEC benchmarks.
  - Up to a three-fold improvement in performance-power ratio.

"Getting jazzed with less juice!"

Chelsio
AMD
Fujitsu

"Innovative Supercomputer Architectures" Award at the 2004 Int'l Supercomputer Conference, Heidelberg, Germany.
Summary of The Evolution of **Green Destiny**

- **Architectural**
  - MegaScale Project (a.k.a. Green Destiny II initially)
  - Orion Multisystems
    - Desktop DT-12 and Deskside DS-96

- **Software-Based**
  - $\beta$-Adaptation DVS Algorithm
    - Laptop Cluster: AMD Athlon XP (uniprocessor)
    - Server Cluster: AMD Athlon-64 (multiprocessor / data ctr)
    - HPC Cluster: AMD Opteron (multiprocessor / data ctr)
Selected Publications

http://sss.lanl.gov (... about three years out of date ...)

Sampling of Media Overexposure

Adding to the Media Hype …
Conclusion

- Efficiency, reliability, and availability will be *the key issues* of this decade.
- Approach: Reduce power consumption via HW or SW.

Cheesy Sound Bite for the DS-96 Personal Deskside Cluster (PDC):

“... the horsepower of a 134-CPU Cray T3E 1200 on LINPACK but in the power envelope of a hairdryer ...”
Acknowledgements

- **Green Destiny**
  - *Mike Warren and Eric Weigle*
  - Funded by DOE Los Alamos Computer Science Institute and the Information Architecture project (IA-Linux) at Los Alamos National Laboratory

- **EnergyFit**
  - *Chung-Hsing Hsu*
  - Funded by DOE Laboratory Directed Research & Development

- **Encouragement & Support**
  - *Gordon Bell, Mark Gardner, Chris Hipp, Linus Torvalds*
Wu-chun (Wu) Feng
feng@lanl.gov