In the Proceedings of the IEEE International Conference on Cluster Computing (Cluster 2005).
Boston, MA, Sep. 2005.

# A Feasibility Analysis of Power Awareness in Commodity-Based High-Performance Clusters*

Chung-hsing Hsu and Wu-chun Feng
Computer & Computational Sciences Division
Los Alamos National Laboratory
{chunghsu, feng}@lanl.gov

## Abstract

*We present a feasibility study of a power-reduction scheme that reduces the thermal power of processors by lowering frequency and voltage in the context of high-performance computing. The study revolves around a 16-processor Opteron-based Beowulf cluster, configured as four nodes of quad-processors, and shows that one can easily reduce a significant amount of CPU and system power dissipation and its associated energy costs while still maintaining high performance. Specifically, our study shows that a 5% performance slowdown can be traded off for an average of 19% system energy savings and 24% system power reduction. These preliminary empirical results, via real measurements, are encouraging because hardware failures often occur when the cluster is running hot, i.e, when the workload is heavy, and the new power-reduction scheme can effectively reduce a cluster's power demands during these busy periods.*

## 1. Introduction

Power efficiency is critical for developing cost-effective, small-footprint clusters. When cluster nodes consume and dissipate more power, they must be spaced out and aggressively cooled; otherwise, undissipated power causes the temperature to increase rapidly enough that for every 10° C increase in temperature, the failure rate doubles [1]. Because the steep power demands of high-performance clusters mean that massive cooling facilities are needed to keep the clusters from failing, the total cost of ownership of these clusters can be quite high. For example, at IEEE Hot Interconnects 2004, Mark Seager of Lawrence Livermore National Laboratory (LLNL) noted the following:

1. For every watt of power consumed at LLNL, 0.7 watts of cooling is needed to dissipate the power.

2. The cooling bill for supercomputers at LLNL is $6M per year.

Consequently, ignoring the costs of acquisition, integration, upgrading, and maintenance, the annual cost to simply power and cool the cluster supercomputers at LLNL amounts to $14.6M per year!

The power efficiency of a cluster can be addressed by reducing the power consumption and dissipation of processors in the cluster. For example, at IEEE Cluster 2002, Feng et al. [2] presented a bladed Beowulf, dubbed *Green Destiny*, that is based on low-power Transmeta processors running high-performance code-morphing software. [1] Green Destiny has proven to be extraordinarily reliable even though it operates in a harsh 85° F warehouse environment. The recently announced Blue Gene/L [8] and Orion Multisystems workstations [10] arguably follow in the footsteps of Green Destiny.

For more traditional AMD- and Intel-based Beowulf clusters, the power reduction can be achieved by means of *dynamic voltage and frequency scaling* (DVFS), a mechanism that allows system software to increase or decrease processor frequency and voltage at run time. Despite a handful of empirical studies published earlier this year [3, 4, 5, 6], the power-efficiency benefits that the DVFS mechanism can provide in high-end Beowulf clusters are still not clear yet. Consequently, we present a feasibility study on a high-end Opteron-based Beowulf cluster.

* This work was supported by the DOE LDRD Exploratory Research Program through Los Alamos National Laboratory contract W-7405-ENG-36 and by Advanced Micro Devices, Inc. Available as LANL technical report: LA-UR 05-5649.

1 The high-performance code-morphing software improved floating-point performance as much as 100% over standard code-morphing software.

**Figure 1. The RLX ServerBlade.**

## 2. Background

In this section, we present two different approaches towards reducing power consumption, and hence, improving reliability in clusters: (i) the low-power approach of Green Destiny and (ii) the power-aware approach that uses dynamic voltage and frequency scaling (DVFS) to reduce power consumption.

### 2.1. Green Destiny

Green Destiny is the first large-scale instantiation of the "Supercomputing in Small Spaces" project [14] at Los Alamos National Laboratory. The origins of this work actually date back to September 2001 with a 24-node bladed Beowulf dubbed MetaBlade that also used Transmeta processors, albeit the previous generation TM5600.

Green Destiny has 240 nodes that fit into an industry-standard 19-inch rack (with a footprint of five square feet) and sip only three kilowatts of power when booted diskless. Each cluster node is an RLX ServerBlade 1000t, as shown in Figure 1, that contains a 933-MHz/1-GHz Transmeta TM5800 CPU, 128-MB DDR SDRAM, 512-MB SDRAM, 20-GB hard disk, and three 100-Mb/s Fast Ethernet network interfaces. Twenty-four such nodes are mounted into a chassis that fits in a 3U space. Putting ten such chassis together along with a standard tree-based network results in the Green Destiny cluster.

Each populated chassis consumes about 512 watts at load, i.e., 21 watts per cluster node. The low power of each cluster node can be attributed to the low power of the Transmeta processor which consumes no more than 7.5 watts.

The excellent power efficiency of Green Destiny made it extremely reliable. The cluster ran in a dusty 85° F warehouse environment at 7,400 feet above sea level without any unscheduled downtime over its two-year lifetime, and it did so without any special facilities, i.e., no air conditioning, no humidification control, no air filtration, and no ventilation. In contrast, our more traditional 100-processor Beowulf cluster that preceded Green Destiny failed on a weekly basis in the same hot environment. (This Beowulf cluster was actually a 128-processor cluster, but we were never able to get the entire cluster up and running reliably.)

Since the debut of Green Destiny, similar solutions have appeared, e.g., Sun Microsystems' SunFire server, which uses AMD XP-M processors, HP's ProLiant BL server, which uses Intel Pentium M processors, Orion Multisystems' desktop cluster, which uses Transmeta Efficeon processors, and finally IBM's BlueGene/L, which uses IBM PowerPC 440 processors. All these processors are primarily used for mobile computing. Hence, a Green Destiny type of solution refers to a cluster design that achieves good power efficiency by using many low-power embedded or mobile processors (rather than fewer but more powerful server processors).

However, the Green Destiny type of solution has two major drawbacks. First, many cluster workloads do not scale as the number of cluster nodes increases. The inherently sequential part of workloads and the network bandwidth limitation prohibit performance scalability. Second, this type of solution is not entirely based on commodity technologies, and hence, may not be cost-effective. For example, BlueGene/L uses an extensively stripped-down version of the 700-MHz PowerPC 440 embedded processor while Green Destiny relies on a customized high-performance version of code-morphing software (CMS)[2] to achieve good performance, e.g., 12.6 Gflops on 24 processors. In contrast, the 16-processor MegaProto cluster [11] which uses the same type of processor as Green Destiny, achieves only 5.62 Gflops on Linpack because it does not have the customized high-performance version of CMS that Green Destiny had.

### 2.2. DVFS-Enabled Clusters

The idea of DVFS on commodity operating systems can be traced back as early as 1994 [15]. Processor power is reduced through lowering frequency and voltage because power is proportional to frequency and to the square of voltage [9]. However, commodity processors that actually supported DVFS did not appear until six years later in 2000 and did so only in the mobile computing market. It was not until 2003 that DVFS made its way into desktop processors, specifically the AMD Athlon64. By late 2004, DVFS gained support on server-class processors such as the AMD Opteron and Intel Xeon EM64T.

To date, we are only aware of two DVFS-enabled Beowulf clusters that have been built and evaluated [4, 5, 6]. The first cluster [6] uses sixteen notebook computers as cluster nodes because most notebook computers are powered by DVFS-enabled processors. (In this case, the mobile processor that is used in each node is a 600-1400MHz Intel Pentium M.) In contrast, the second cluster [4, 5] is

2 Each Transmeta processor has a software layer, called *code-morphing software*, that dynamically morphs x86 instructions into VLIW instructions. This provides x86 software with the impression that it is being run on native x86 hardware.

based on ten desktop motherboards, each of which has a 800-2000MHz AMD Athlon64 processor. For both clusters, each node has 1GB main memory.

Though the two DVFS-enabled clusters are commodity-based, we argue that they are not high-performance nor balanced. Performance-wise, the two clusters use slow 100-Mb/s Fast Ethernet networks. In contrast, none of the supercomputers on the Top500 list (http://www.top500.org) use Fast Ethernet, but nearly half of them use Gigabit Ethernet. More importantly, using fast processors but a slow network creates an *imbalanced* machine that allows the processor frequency to be lowered further than a balanced machine for the same level of performance impact, thus leading to noticeably more power and energy reduction than one would realize in a more balanced, high-performance cluster. Therefore, we use a 16-processor Opteron-based Beowulf cluster, interconnected by Gigabit Ethernet, to present a feasibility study on the power-efficient benefits that the DVFS mechanism can deliver.

## 3. CAFfeine: A DVFS-Enabled Opteron Cluster

This section presents technical details about our DVFS-enabled Opteron cluster dubbed *CAFfeine*. Specifically, we discuss both its commodity and high-performance attributes.

### 3.1. Commodity-Based CAFfeine

Currently, CAFfeine is configured as a 4-node, 16-processor cluster connected via a Gigabit-Ethernet (GbE) network. Each node is a Celestica 4U rack-mountable quad-Opteron server A8440, as shown in Figure 2. Inside each node, there are sixteen memory slots, four in front of each processor to enable low-latency access. The current configuration has two 512-MB registered ECC DDR-333 SDRAM memory modules for each Opteron processor. In terms of storage, four hot-swap SCSI drive bays are available, and CAFfeine currently has one 73GB hard disk for each node along with one DVD-ROM drive and one floppy drive. In addition to up to three hot-swap 500W power supplies with 2+1 redundancy, each A8440 has an ATI Rage XL VGA adapter, two 133MHz PCI-X slots, and two 66MHz PCI slots, all 64 bits, plus three rear-mounted Ethernet ports. Figure 3 shows a block diagram of the A8440.

Each processor in CAFfeine is an Opteron 846 processor that can run between 800MHz and 2000MHz in steps of 200MHz. The processor contains a 1MB same-speed on-chip L2 cache, and, unlike Intel processors, a memory controller is also built-in on the chip. Opterons in the A8440 are interconnected with coherent HyperTransport links running at 800MHz for fast I/O access. Each Opteron 846 is fabricated using 13-$\mu$m SOI process and consumes no more than 89W. With respect to valid frequency-voltage combinations needed in a DVFS-enabled processor, AMD has not yet published any documentation on this yet. We experimentally set the combinations as in Table 1. Hence, our DVFS-enabled CAFfeine cluster has seven different power-performance combinations.

**Figure 2. Celestica A8440.**

| $f$ (GHz) | 2.0 | 1.8 | 1.6 | 1.4 | 1.2 | 1.0 | 0.8 |
|---|---|---|---|---|---|---|---|
| $V$ (V) | 1.5 | 1.4 | 1.3 | 1.2 | 1.1 | 1.0 | 0.9 |

**Table 1. Valid Frequency-Voltage Combinations in CAFfeine.**

With respect to software, CAFfeine runs on the Linux 2.6.7 operating system. All the benchmark codes mentioned in this paper are compiled using GNU compilers 3.3.3. For DVFS, the change between different frequency-voltage combinations is performed through the `cpufreq` interface provided by the `powernow-k8` kernel module distributed along with Linux 2.6.7 kernel.

The `cpufreq` interface allows system software to set a desired CPU clock frequency by writing the frequency value (in terms of megahertz) to a particular `/sys` file. However, not all the CPU frequencies are directly supported by a DVFS-enabled processor due to hardware constraints. The information about which CPU frequency is supported is stored in the `powernow-k8` kernel module in a tabular form similar to Table 1, as is the selection scheme when the desired frequency is not supported.

### 3.2. High-Performance CAFfeine

CAFfeine is also high performance with respect to the two existing DVFS-enabled clusters mentioned in Sec-

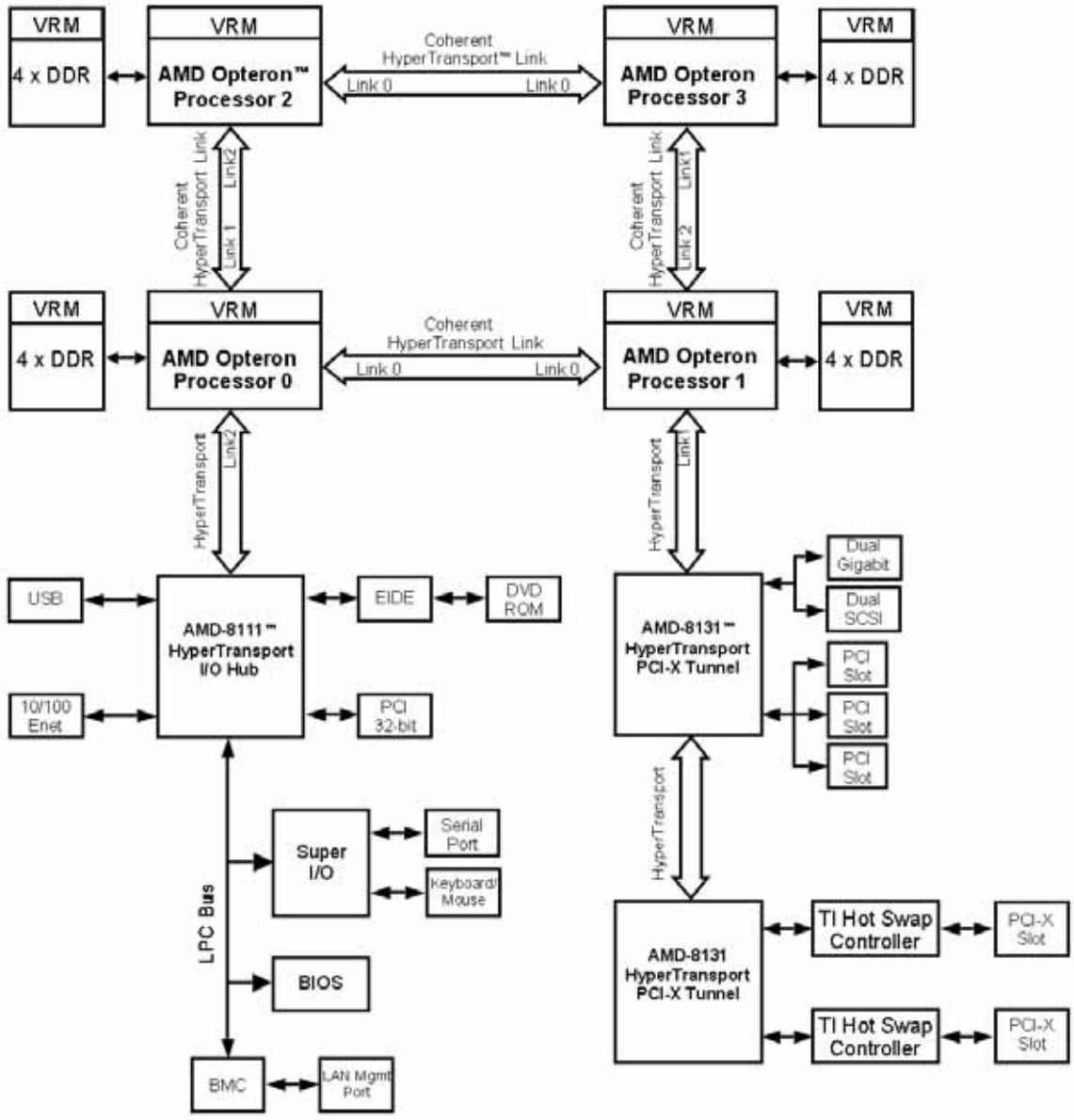



**Figure 3. The Block Diagram of Celestica A8440.**

tion 2. Figure 4 shows a performance comparison of running NAS MPI benchmarks on the Opteron-based CAFfeine versus the Athlon64-based cluster [4, 5]. The comparison is made using NAS MPI benchmarks in terms of the total execution time on two different workload classes B and C. (Class C represents a larger workload as well as a larger memory footprint.) The execution times on the Athlon64-based cluster was derived from the figures in [4, 5]. From this figure, we clearly see that CAFfeine performs better than this Athlon64-based cluster for the same cluster size. Since both clusters have processors of the same speed, i.e., 2GHz, we attribute the higher performance of CAFfeine to a larger L2 cache (1MB versus 512KB), faster network (Gigabit Ethernet versus Fast Ethernet), and SMP-based architecture.

For the Pentium M-based cluster [6], since there is not enough timing information in [6], we are unable to compare CAFfeine with this cluster directly. Nevertheless, we expect CAFfeine to deliver higher performance because CAFfeine has faster processors installed (2GHz versus 1.4GHz).

CAFfeine also performs well when compared to Green Destiny within the same power budget. The power consumed by a chassis in Green Destiny and by a CAFfeine node is roughly the same, i.e., around 500 watts. How-

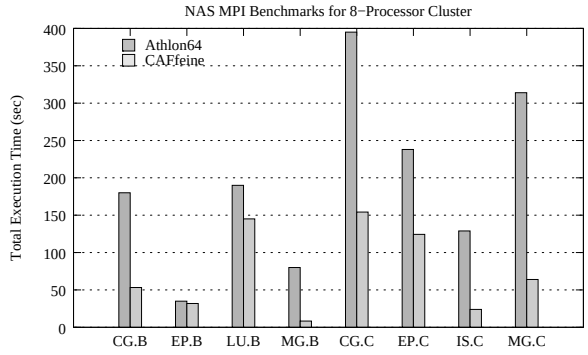



**Figure 4. Performance Comparison to Existing DVFS-Enabled Clusters.**

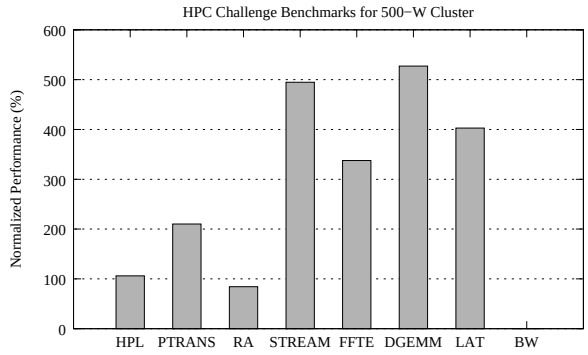



**Figure 5. Performance Comparison to Green Destiny.**

ever, in terms of performance, the CAFfeine node runs much faster in almost every aspect, e.g., from floating-point performance to memory bandwidth. Figure 5 shows a performance comparison for all seven measurements from the HPC Challenge (HPCC) benchmark suite [7]. All the performance measurements are normalized with respect to Green Destiny. Note that the BW number is not shown because the communication methods are different. The CAFfeine node performs intra-node communication via HyperTransport links whereas the Green Destiny chassis has to do inter-node communication via a much slower 100-Mb/s network. (But in terms of raw bandwidth, the CAFfeine nodes perform 157 times better than Green Destiny.)

## 4. Feasibility Analysis on Power Awareness

Here we present a feasibility analysis on the power awareness of CAFfeine via DVFS. But before we do so, we briefly describe our measurement infrastructure.

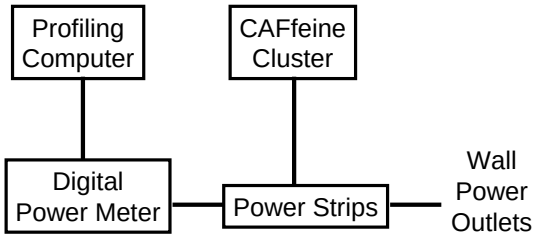



**Figure 6. The Measurement Infrastructure.**

In this study, the reported execution time is referred to as the wall clock time of program execution. The power and energy numbers are with respect to the entire cluster. The reported power numbers refer to the average system wattage, while the energy numbers refer to the as the total system energy consumption.

To measure the execution time of a program, we use the Linux `time` command. To get the power- and energy-consumption numbers, we use an industry-strength power meter, a Yokogawa WT210/WT230 series. This power meter is connected to power strips that pass electrical energy from the wall power outlets to power up CAFfeine, as shown in Figure 6. The power meter periodically samples the instantaneous system wattage at a rate of 20 $\mu$s per sample. The total energy consumption is then calculated as the integration of these wattages over time. The average power consumption is the total energy consumption divided by the execution time.

### 4.1. HPL Benchmark

To begin with, we conduct an in-depth study on the potential DVFS-induced power reduction at the cluster-node level. We choose the High-Performance Linpack (HPL) benchmark code [13] to stress-test the processor. HPL is an open-source implementation of the Linpack benchmark that solves a random, dense linear system of equations in double-precision arithmetic in parallel. HPL has long been argued to have exceptional temporal locality that makes its performance number (in terms of Gflops) much higher than what can be observed in real-life scientific applications. Nevertheless, HPL's high CPU utilization makes it a good CPU stress-test for a CAFfeine SMP node.

How does the power consumption of HPL compare to that of other benchmarks? Figure 7 shows the power consumption of running the NAS MPI benchmarks and HPL on a quad-Opteron CAFfeine node. The workloads for the NAS MPI benchmarks are carefully chosen to be class B so that they only require in-core executions. Thus, all benchmarks in Figure 7 only stress the CPUs and their respective memory subsystems. We can see from the figure that

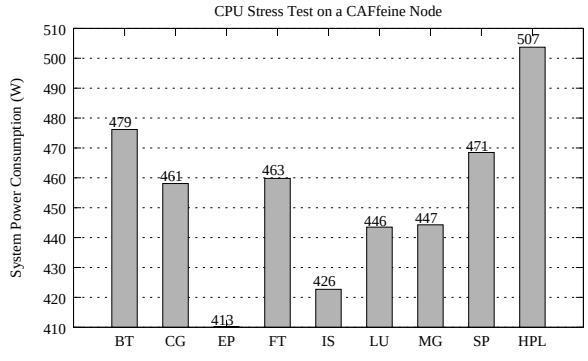



**Figure 7. Power Consumption of NAS+HPL MPI Benchmarks.**

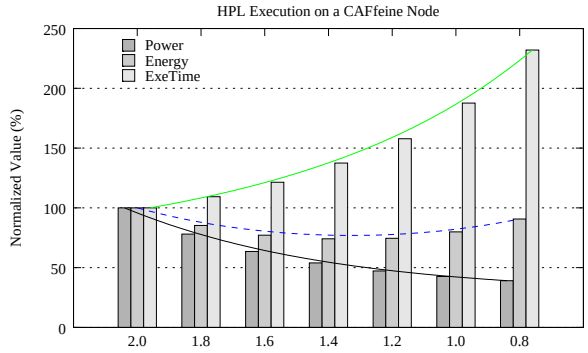



**Figure 8. Power and Performance Trend of HPL Execution.**

the wattage of HPL stands out.

By plotting the system wattage of HPL execution for each frequency-voltage combination in CAFfeine, as shown in Figure 8, we find that DVFS-induced system power savings can be as high as 61%! The power savings are achieved at the expense of performance degradation, in this case, a slowdown of 2.32-fold.

More importantly, system power reduction does not always lead to system energy reduction. For the HPL example, Figure 8 shows that the most *power-efficient* execution occurs when the CPUs run at 0.8GHz, whereas the most *energy-efficient* execution occurs when the CPUs run at 1.4GHz. Why doesn't the most energy-efficient execution also occur at 0.8GHz? Because at that frequency, HPL takes *significantly* longer to run.[3] As a result, Figure 8 shows that the resulting curve for system energy consumption is U-shaped. In fact, not only does HPL possess this U-shaped curve for system energy consumption; but all the NAS MPI benchmarks do as well, as we will see in Section 5.

Why is the above observation important? It turns out that many DVFS utilization schemes are based on an assumption that energy consumption will be reduced whenever the CPU frequency is reduced. Our HPL example has shown that this assumption is invalid for some programs on certain hardware platforms. As a result, the misuse of these schemes may produce unsatisfactory results.

It may be argued that running HPL at 1.4GHz, though energy efficient, sacrifices too much performance. Even running at 1.8GHz, which results in 15% performance degradation, may not be considered conducive towards high performance. Hence, we seek to answer the question of how much energy savings would we be able to achieve at say 5% performance loss. To do so, we use the following approach. We solve a linear-programming problem for a particular performance-slowdown requirement $D$: Given a set of tuples $\{(P_i, T_i)\}$ for each frequency-voltage combination $i$, $1 \leq i \leq n$, where $P_i$ and $T_i$ denote the system wattage and the total execution time of the target application, respectively, find an optimal solution vector $(r_1^*, r_2^*, \cdots, r_n^*)$ for the minimization problem

$$E^*(D) = \min \left\{ \sum_i r_i \cdot (P_i \cdot T_i) : \begin{array}{l} \sum_i r_i \cdot T_i \leq D \\ \sum_i r_i = 1 \\ r_i \geq 0 \end{array} \right\} \quad (1)$$

with respect to a given deadline $D$ (in seconds). By varying $D$, we can derive the energy-performance curve (i.e., $E^* - D$ curve) for the target application.

The energy-performance curve for HPL is shown in Figure 9 at two different granularities with respect to performance slowdown. For a 5% performance-slowdown requirement, CAFfeine can reduce system energy by 8% (12% for system wattage). The figure also shows that as the performance requirement is relaxed, the rate of system energy savings drops.

Finally, the energy savings derived from Equation (1) is realizable. Basically, we calculate the desired frequency $f^*$ as follows.

$$f^* = \sum_i r_i \cdot f_i \quad (2)$$

For HPL execution at 5% performance slowdown, the desired frequency $f^*$ is about 1.9GHz. Since CAFfeine (or more specifically, the AMD Opteron processor) does not support this frequency directly, we have to emulate the frequency using 1.8GHz and 2.0GHz, e.g., run iteratively at 1.8GHz for one second and then 2.0GHz for the following second. The reason this scheme works is because $E^*(T_i) = P_i \cdot T_i$ holds. In short, the $D - E^*$ curve is a piecewise-linear combination of $(T_i, P_i \cdot T_i)$ when the condition holds, and the condition holds when the total execu-

3 Remember that energy consumption is the product of power consumption and execution time.

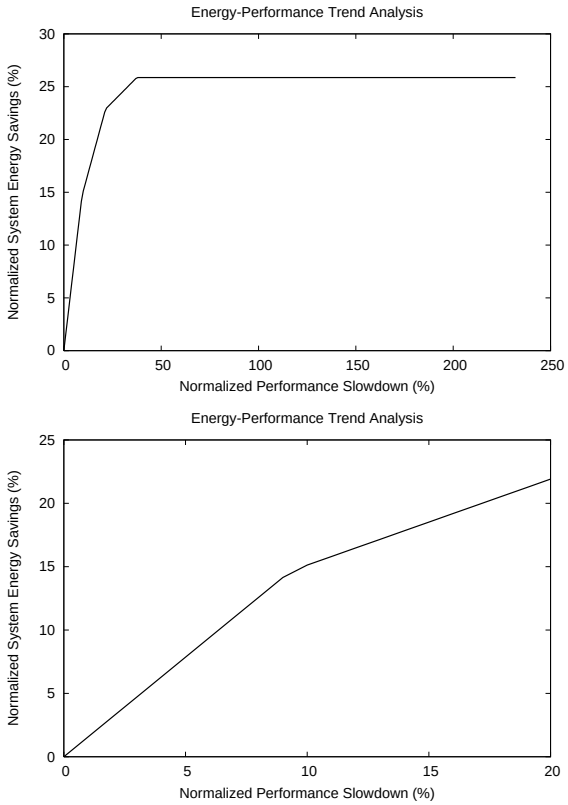



**Figure 9. Energy-Performance Tradeoffs of HPL Execution.**

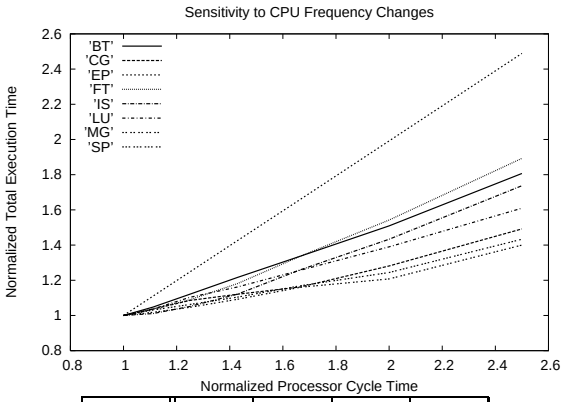
tion time is a linear function of CPU cycle time, which HPL execution is.

### 4.2. NAS MPI Benchmarks

As we mentioned earlier, HPL is an atypical scientific application in that it possesses exceptional memory locality. For a more typical scientific application that is oftentimes bottlenecked by memory and network performance, CAFfeine can achieve even higher system energy savings within the same performance constraint. To support our claim, we run the entire NAS MPI benchmark suite on CAFfeine.

The NAS MPI benchmarks [12] consist of eight benchmark codes. Together they mimic the computation and data movement characteristics of large-scale computational fluid dynamics (CFD) applications. These benchmarks cover a wide range of sensitivity to CPU speed changes. For example, the MG benchmark measures memory bandwidth, in contrast to HPL, which measures floating-point performance of CPUs.

Figure 10 plots the normalized execution time with respect to normalized processor cycle time for all NAS MPI




| | EP | FT | BT | IS |
|---|---|---|---|---|
| slope | 0.99 | 0.60 | 0.54 | 0.51 |

| | LU | CG | SP | MG |
|---|---|---|---|---|
| slope | 0.40 | 0.31 | 0.28 | 0.26 |

**Figure 10. Sensitivity Analysis on Entire CAFfeine Using Workload Class C.**

benchmarks in version 3.2 using workload class C. Without knowing what specific problem each benchmark solves, the figure tells us that EP is the most sensitive to CPU frequency changes. In contrast, MG and SP are the least sensitive.

Using the same analysis as we did for HPL, we can derive the range of DVFS-induced system energy savings for NAS MPI benchmarks. Table 2 shows the potential savings for the entire CAFfeine cluster. At a 5% performance loss, one can save an average of 19% system energy for CAFfeine. In other words, CAFfeine is capable of reducing a significant amount of system power and energy while still maintaining high performance.

These preliminary empirical results, via real measurements, are encouraging because hardware failures often occur when the cluster is running hot, i.e, when the workload is heavy, and CAFfeine can effectively reduce the occurrences of such overheating-induced failures in a high-performance commodity-based cluster during these busy periods.

## 5. An Analysis of Opportunities for Power Awareness

For completeness, we present in Figure 11 the performance and power trend of each NAS MPI benchmark running on the entire CAFfeine cluster.

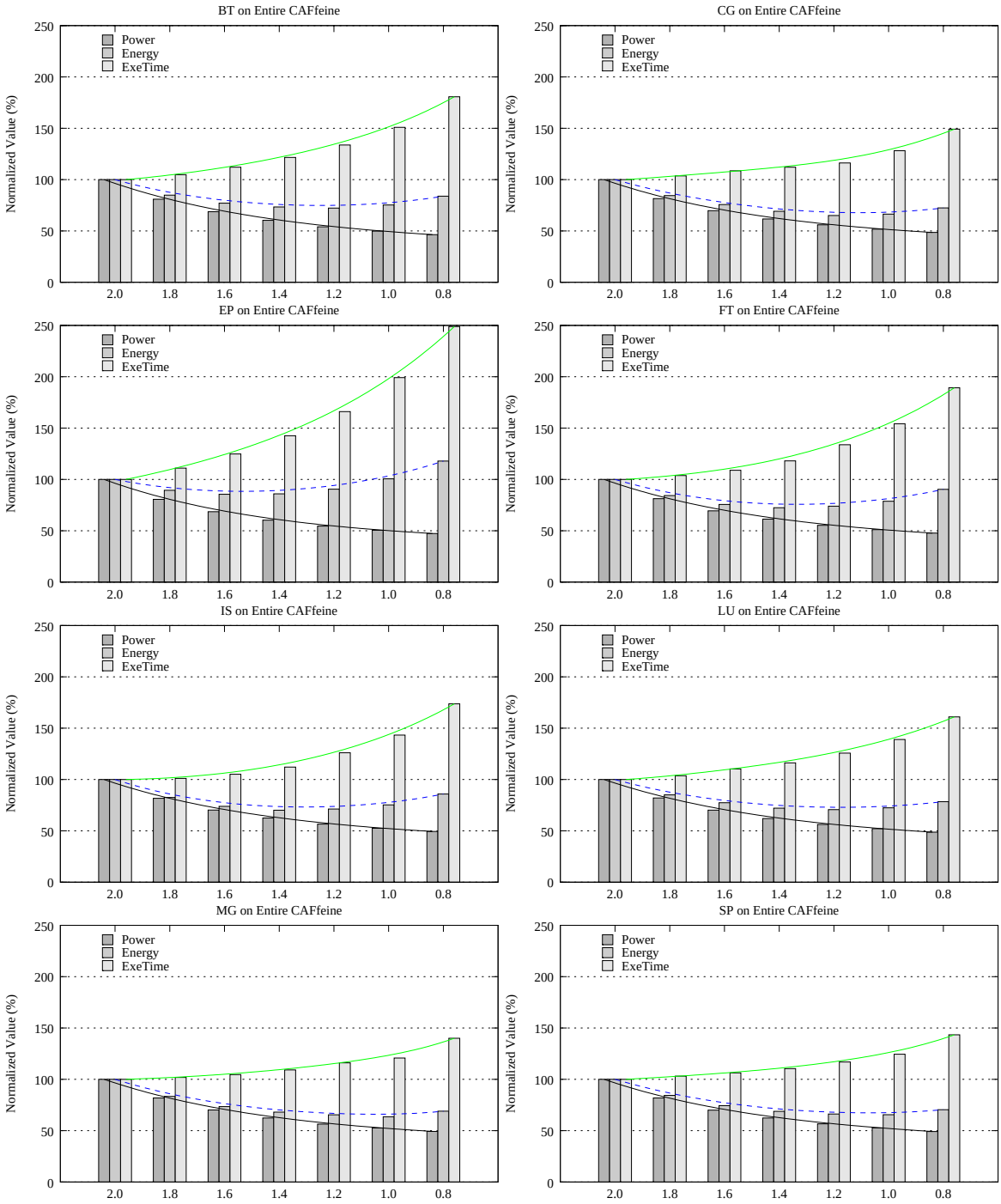



**Figure 11. Power and Performance Trend of NAS MPI Benchmarks on Workload C.**

| Class C Workload on Entire CAFfeine | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $D$ | BT | CG | EP | FT | IS | LU | MG | SP | Average |
| 5% | 15% | 18% | 5% | 18% | 26% | 17% | 27% | 22% | 19% |
| 10% | 21% | 27% | 10% | 25% | 29% | 22% | 32% | 31% | 25% |
| $\infty$ | 28% | 35% | 14% | 28% | 30% | 30% | 36% | 34% | 29% |

**Table 2. DVFS-Induced System Energy Reduction for NAS MPI Benchmarks.**

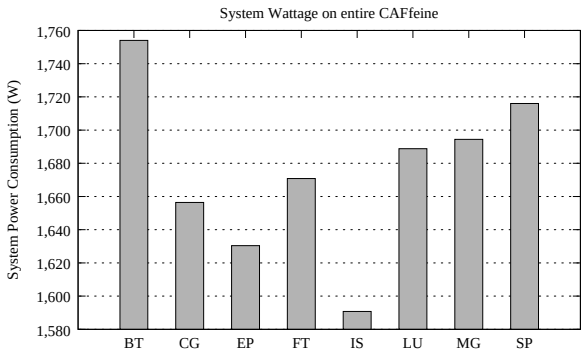



**Figure 12. Power Consumption of the NAS MPI Benchmarks on the Entire CAFfeine Cluster.**

### 5.1. Application Characteristics

First, as discussed in the previous section, system power reduction does not necessarily result in system energy savings. The U-shape curve for system energy usage is exhibited in every NAS MPI benchmark. However, the lowest point in this U-shape curve varies from benchmark to benchmark. Similar to HPL execution, system wattage savings tend to flatten out as the CPU frequency decreases, whereas the total execution time starts to climb up at the same time. Hence, Figure 11 further supports the thesis that running at the lowest frequency on CAFfeine is not a good idea in reducing system energy costs in a high-performance cluster.

Second, an application that is sensitive to CPU frequency changes does not necessarily dissipate much heat. For example, both EP and HPL (in Figure 8) are sensitive to CPU frequency changes. However, Figure 7 shows that one consumes the most power (and thus dissipates much heat) whereas the other consumes the least power among all tested benchmarks.

Similarly, a memory-intensive application does not always consume less power. Both MG and SP measure memory bandwidth and they all have the least sensitivity to CPU frequency changes. Yet they consume higher power, according to Figure 12, than EP. Obviously, the above profiles translate into opportunities for a DVFS-enabled cluster to significantly reduce energy consumption (and hence, costs) without needing to sacrifice high performance.

### 5.2. CPU Power Reduction

So far we have been using system wattage to evaluate power awareness of CAFfeine. It might be interesting to know how much wattage is reduced on the processor chips as DVFS can only change processor frequency and voltage and logically only affects processor wattage. To do so in a non-intrusive way, we use the following approach. We first collect the system wattage of NAS MPI benchmarks on a CAFfeine node for each frequency-voltage combination $\{(P_i, f_i, V_i)\}$. Then we use regression method to fit the measurement data into the following first-order power model [9]

$$P(f, V) = c_1 \cdot V^2 \cdot f + c_0 \quad (3)$$

in order to compute the two constants $c_1$ and $c_0$. Since DVFS only changes the first term, $c_1 \cdot V^2 \cdot f$, we can then estimate processor wattage.

On average, processor wattage consumes about 77 watts and accounts for 69% of total system wattage. This percentage is quite large. Thus, we conclude that DVFS can effectively reduce power and energy consumption on this type of hardware platform.

### 5.3. DVFS-Induced Phase-Oriented Scheduling

There have been attempts (e.g., [4, 6]) to exploit execution phase characteristics for DVFS-induced system power reduction and energy savings. For example, the FT benchmark on the Athlon64-based cluster has a power profile (as shown by the bottom curve in Figure 13) that consists of regular spike-and-valley power usage pattern corresponding to the interleaved computation and communication phases of the FT code. Since the reduction of CPU frequency has little performance effect on communication phases where network is the performance bottleneck, researchers propose to execute the communication phases of FT at a non-peak frequency and execute the computation phases at the peak frequency. This is called phase-oriented DVFS scheduling.

However, when the network becomes faster, such as the 1-Gb/s performance in CAFfeine, such strategies have a di-

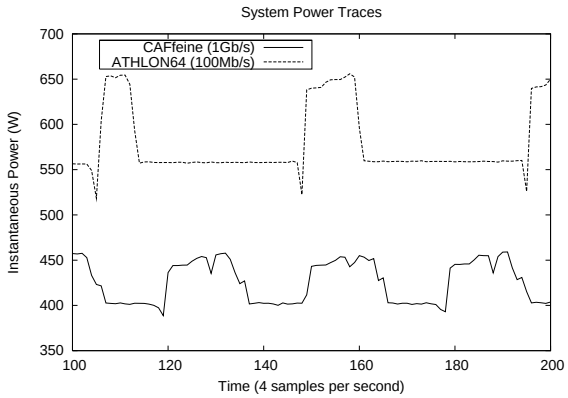



**Figure 13. Power Profile of NAS MPI Benchmark FT (Normalized to 4-Processor Setting).**

minishing energy-saving effect due to shorter communication phases (e.g., the top curve in Figure 13). Hence, lowering the CPU frequency whenever the program execution enters a communication phase may generate a negative performance effect as each DVFS call introduces additional performance overhead, currently on the order of milliseconds if via the `cpufreq` interface. (This performance overhead includes the `/sys` file access, table-entry search, various assertion checks, and the real transition time.) A similar argument holds for MPI collective operations as well.

## 6. Conclusion

Steep power demands and their subsequent energy costs and thermal-related reliability is a serious design issue for building commodity-based high-performance clusters. We address the challenge in this paper by presenting a DVFS-enabled Opteron cluster dubbed CAFfeine and a feasibility study on its potential power awareness. While CAFfeine has better performance than two existing DVFS-enabled Beowulf clusters, one based on 2GHz Athlon64 and the other based on 1.4GHz Pentium M, we show that one can still reduce a significant amount of CPU and system power dissipation and the associated energy costs (an average of 19%) while still maintaining high performance (at most 5% performance slowdown).

## 7. Acknowledgements

First and foremost, we would like to thank Douglas O'Flaherty of AMD for his tremendous support of our research efforts. His contributions to the project were invaluable. We are also indebted to Paul Devriendt and Mark Langsdorf for providing technical details about PowerNow! on Opteron. Next, we acknowledge Western Scientific for building our CAFfeine cluster and for providing technical support. Finally, we wish to recognize Jeremy S. Archuleta for his tireless efforts in building, configuring, and administering all the computing platforms that were used in this paper.

## References

[1] W. Feng. Making a case for efficient supercomputing. *ACM Queue*, 1(7):54–64, Oct. 2003.

[2] W. Feng, M. Warren, and E. Weigle. The bladed Beowulf: A cost-effective alternative to traditional Beowulfs. *Proc. IEEE Int'l Conf. Cluster Computing (CLUSTER 2002)*, Sept. 2002.

[3] X. Feng, R. Ge, and K. Cameron. Power and energy profiling of scientific applications on distributed systems. *Proc. IEEE Int'l Parallel & Distributed Processing Symp. (IPDPS 2005)*, Apr. 2005.

[4] V. Freeh, D. Lowenthal, F. Pan, and N. Kappiah. Using multiple energy gears in MPI programs on a power-scalable cluster. *Proc. ACM SIGPLAN Symp. Principles and Practices of Parallel Programming (PPoPP'05)*, June 2005.

[5] V. Freeh, D. Lowenthal, R. Springer, F. Pan, and N. Kappiah. Exploring the energy-time tradeoff in MPI programs on a power-scalable cluster. *Proc. IEEE Int'l Parallel & Distributed Processing Symp. (IPDPS 2005)*, Apr. 2005.

[6] R. Ge, X. Feng, and K. Cameron. Improvement of power-performance efficiency for high-end computing. *Proc. 1st Workshop on High-Performance, Power-Aware Computing (HP-PAC 2005)*, Apr. 2005.

[7] HPC Challenge Benchmark. http://icl.cs.utk.edu/hpcc/.

[8] IBM Research Blue Gene Project. http://www.research.ibm.com/bluegene.

[9] T. Mudge. Power: A first class design constraint for future architectures. *IEEE Computer*, 34(4):52–58, Apr. 2001.

[10] Orion Multisystems. http://www.orionmulti.com/.

[11] H. Nakashima, H. Nakamura, M. Sato, T. Boku, S. Matsuoka, D. Takahashi, and Y. Hotta. MegaProto: A low-power and compact cluster for high-performance computing. *Proc. 1st Workshop on High-Performance, Power-Aware Computing (HP-PAC 2005)*, Apr. 2005.

[12] NAS Parallel Benchmarks. http://www.nas.nasa.gov/software/NPB/.

[13] A. Petitet, R. Whaley, J. Dongarra, and A. Cleary. HPL - a portable implementation of the high-performance Linpack benchmark for distributed-memory computers.

[14] Supercomputing in Small Spaces Project. http://sss.lanl.gov.

[15] M. Weiser, B. Welch, A. Demers, and S. Shenker. Scheduling for reduced CPU energy. *Proc. 1st Symp. Operating Systems Design and Implementation (OSDI'94)*, Nov. 1994.