CTWatch QUARTERLY

ISSN 1555-9874 Volume 1 Number 3 August 2005

THE COMING ERA OF LOW POWER, HIGH-PERFORMANCE COMPUTING

TRENDS, PROMISES, AND CHALLENGES

Guest Editor: Satoshi Matsuoka, Tokyo Institute of Technology

FEATURE ARTICLES

NOTES AND COMMENTARY

AVAILABLE ON-LINE AT
http://www.ctwatch.org/quarterly/

Cyberinfrastructure Technology Watch
http://www.ctwatch.org/

Low Power Computing for Fleas, Mice, and Mammoth — Do They Speak the Same Language?

Satoshi Matsuoka
Tokyo Institute of Technology

Introduction

The main theme of this issue of *CTWatch Quarterly* is the new trend within the high performance computing (HPC) toward lower power requirements. Low power computing itself is not new — it has had a long history in embedded systems, where battery life is at a premium. In fact, the applicability of low power has widened its scope in both directions on the power consumption scale. Lower power consumption in the microwatts arena — so-called "ultra low power" (ULP) — is necessary to enable applications such as wireless remote sensing, where a device may have to run on a single small battery for months and need to be networked to collect data. In a more familiar context, most PCs have recently become Energy Star[1] compliant. In fact, a really dramatic shift in design emphasis occurred around 2003-2004, when the industry began to move from the pursuit of desktop performance alone to the pursuit of desktop performance/power in combination. Recent processors initially designed for energy efficient notebooks, such as Intel's Pentium-M, have started to find their way into desktop units. In fact, there is strong speculation that future mainstream PC processors will be successors of the Pentium-M style, a power efficient design.

But why do we want to save power in the HPC arena since the goal has always been to go faster at almost any cost? Certainly it is fair to say that performance/power has always been an engineering concern in designing HPC machines. For example, NEC claims to have achieved five times better performance/power efficiency in their SX-6 model over their previous generation SX-5.[2] Where HPC machines function as large servers in datacenters, reducing power would also result in substantial cost savings in their operations. And of course, there are important social and economic reasons for reducing the extremely high power consumption of many HPC installations.

However, the recent attention to low power in HPC systems is not merely driven by such "energy-conscious" requirements alone. There have been recent research results, especially spearheaded by those of the BlueGene/L[3] group, that seem to indicate that being low power may be *fundamental* to future system scalability, including future petascale systems, personalized terascale systems, and beyond. The purpose of the articles in this issue is to reveal such new trends and discuss the future of HPC from the perspective of low power computing.

In the remainder of this article, we will show how low power designs in the traditional arena of embedded computing, plus the very interesting ultra low power systems that are now receiving considerable attention, relate to low power HPC. In particular, we will discuss how technologies developed for low power embedded systems might be applicable to low power HPC and what the future holds for further research and development in this area that aims for greater performance in next generation HPC.

[1] The Energy Star Home Page, http://www.energystar.gov/

[2] Computers Division "Design of Eco Products SX-6", NEC Technical Journal, Vol. 57, No.1, 2004. http://www.nec.co.jp/techrep/ja/journal/g04/n01/t040105.pdf (in Japanese)

[3] IBM Journal of Research and Development, special double issue on Blue Gene, Vol.49, No.2/3, March/May, 2005

Is Saving Power Anything Special?

From an engineering point of view, it is obvious that one would want to save power to attain maximum efficiency in any largely-deployed infrastructure, as we mentioned earlier. But the metrics of tradeoffs in power vs. performance differ vastly depending on the application. In other technology areas, similar differences exist. For example, with automobiles, one metric is to shoot for maximum speed, as with a Formula One race car, where one gets only a little over one kilometer per liter in fuel efficiency. On the other hand, there are fuel-efficiency competitions where one attempts to maximize the distance that can be traveled with a liter of fuel; the current world record is 5134 km, which is nearly four orders of magnitude different than the race car example. Even though combustion technology is recognized as being fairly mature, we seldom observe the exponential growth that we see in the IT industry.

Still, the technological advancements in fuel efficiency improvement in the "standard" automotive industry is in the low percentage points, and even disruptive technologies such as fuel cell or battery-based EVs (Electrical Vehicles) will not improve the efficiency by an order of magnitude. With the IT industry, however, we all know that with Moore's Law performance has been increasing exponentially since the 1960s and is expected to continue until at least 2015. However, some of the problematic phenomena that drive up power consumption also follow this exponential curve. For example, static leakage current is directly related to the number of transistors, which gave rise to the exponential performance increase in the first place.

Pros and Cons of Low Power, Especially in HPC

While low power consumption may seem to be an obvious engineering ideal for computing systems, especially in HPC, achieving it requires designers to make various tradeoffs that have their own pros and cons.

Pros:

Higher density — With lower thermal density, an HPC architecture can be more densely packed. This is very important. As Table 1 shows, the absolute space occupancy is starting to limit the machine size, not just in terms of the physical real estate needed, but also, for example, in terms of cable length. In fact, if we were to build a Petascale machine now using Earth Simulator technology, not only would it require a 100MW scale electrical power plant, but it would occupy over 30,000 square meters of floor space (approx 330,000 square feet, or the size of a small football stadium). The weight of its cabling, amounting to approximately 400,000 kilometers or about 250,000 miles, would be a whopping 15,000 tons (several times more than the steel reinforcements that would be used in such a stadium)!

Reduced Cooling Infrastructure — Cooling requirements of large machines may add anywhere from 25 to 50 % to the consumed power of the machine. Moreover, most machines are designed to operate at their maximum performance level at some point, resulting in maximum thermal heat dissipation. The cost of the initial infrastructure, which would include maximum cooling capacity, could be millions of dollars of equipment and construction, not to mention substantial space consumption.

Improved error rate / MTBF — Higher cabinet temperatures will result in shorter mean time between failures (MTBF) for various parts of the machine, primarily disk storage, capacitors, and silicone. Some studies have shown that a ten degree increase in the operational temperature of a typical hard disk will reduce its lifetime to 1/10th of its typical rating.

All is not favorable, however. There are various drawbacks to low power designs, some of which are generic and some of which are more peculiar to HPC. Both types require substantial research and engineering.

Cons:

Increased system complexity — Low power design obviously adds complexity to the overall system both in hardware and software as well as overall management. We omit a more detailed discussion here for the sake of brevity.

Increased sporadic failures — In low power systems generally, the chance of sporadic failures may increase for several reasons, including reduced noise margins caused by lowering the supply voltage, timing issues, etc. Such failures will have to be compensated for by careful and somewhat conservative circuit design, sanity checking, redundancy, etc. Another possibility is to employ software checking and recovery more extensively, but such measures tend to be difficult to implement without some hardware support.

Increased failures as the number of components is scaled up — In some low power HPC architectures, the desire to exploit a "slow and parallel" strategy leads to designs with a higher number of nodes and thus a higher number of components in the system. For example, the largest BlueGene/L on the Top500 to date sports 65,536 CPU cores, an order of magnitude greater than any other machine on the Top 500. By comparison, the Earth Simulator has only 5120 cores. Certainly the number of cores is one particular metric and cannot account for overall machine stability. In fact, BlueGene/L has gone to great lengths to reduce the number of overall system components, and the results from early deployments have demonstrated that it is a quite reliable machine. Nonetheless, as we approach the petaflops range, the amount of component increase will be substantially more demanding.

Reliance on Extremely High Parallel Efficiency to Extract Performance — Since the performance of each processor in such low power designs will be slow, achieving good performance will require a much higher degree of parallel efficiency compared to conventional high-performance, high-power CPUs. Thus, unless the application is able to exhibit considerable parallel efficiency, we will not be able to attain proper performance from the system. If the inefficiency is due to the software or the underlying hardware, solutions may be available to resolve it. However, if the cause is fundamental to the algorithm, with unavoidable serialization capping limits on parallelism, then we will have to resort to somehow discovering the fundamental application algorithm. This is sometimes very difficult, however, especially for very large, legacy applications.

	Advanced Vectors (Earth Simulator => SX-8)	High Density Cluster (Itanium Mondecito Blade + Infiniband 4x)	Low PowerCPU? Super Highly Density (Blue Gene/L)
GFLOPS/CPU	16	8	2.8
CPU CORE/Chip	1	2	2
CPU Chip/Cabinet	8	72	1024
TFLOPS/Cabinet	0.128	1.152	5.7344
Memory BW/Chip (GB/s)	64	10.672	6.4
Memory BW/Cabinet (GB/s)	512	768.384	6553.6
Network BW/Chip (MB/s)	NA	625	1050
Network Bytes/s/Flop	0.125	0.0390625	0.1875
#Cabinets for 1PF (+30% Network)	10156	1128	174
Physical size relative to ES	13.22	1.47	0.23
Power/Cabinet (KW)	9	15	25
Total Power (30% cooling) (MW)	118.83	22.00	5.66
Power relative to ES (8MW)	14.85	2.75	0.71
Cost/Cabinet ($Million US)	1	1	1.5
Total Cost ($Billion US)	10.16	1.13	0.26
Cost relative to ES ($400 mil US)	25.39	2.82	0.65

Table 1. Modern HPC Machine Parameters

So, Where Do We Obtain the Power Savings?

With mainstream information technology, such as standard office application suites, speed requirements may have "matured." But the majority of application areas, in particular ones mentioned in this article, are still in need of significant (even exponential) improvements in both absolute performance and relative performance/power metrics over the next ten years, as we progress towards building a "true" cyberinfrastructure for science and engineering. Such is quite obviously the case for traditional HPC applications, where even a petaflop machine may not satisfy the needs of the most demanding applications. It is also evident in application areas that are taking a leap to next generation algorithms in order to increase scale, accuracy, etc. An example is large scale text processing/data mining where the proliferation of the web and the associated explosion of data call for more sophisticated search and mining algorithms to deal with "data deluge." Another example is the push to develop humanoid robots, where one is said to require more than five to six orders of magnitude processing power while retaining the human form factor.

The question is, can we achieve these goals? If so, will the techniques/technologies employed in respective domains, as well as their respective requirements, be different? If there are such differences, will this cause one power range to be more likely than the others? Or are there some uncharted territories of disruptive technologies with even more possibilities?

Major power saving techniques, in particular those being exploited by more traditional embedded systems, plus the recent breed of low power HPC systems could be categorized as follows:

- *Fundamental decrease in transistor power consumption and wire capacitance reduction* — Traditionally, one would save power "for free" with lower micron sizes, where the transistors become smaller and the wires become thinner. However, it is well known that this is becoming harder to exploit because of longer circuit delays, higher static leakage current, and other physical device characteristics that come into play. As an example, Intel's move to the .09 micron with the new version of their Pentium 4 processor (Prescott) resulted in higher power consumption than its previous generation (Northwood). Granted, there were substantial architectural changes. But the original idea seemed to have been that the move to .09 micron would more than compensate for the added power consumption due to increased logic complexity and higher transistor count. However, this proved not to be the case.

- *Voltage reduction (DVS: Dynamic Voltage Scaling)* — Closely related to the previous strategy is the idea of reducing voltage with each reduction in processor size. However, this too is reaching its limits, as the state machines (i.e. flip-flops constituting the various state elements and memory cells in the architecture) cannot get significantly below one Volt or so due to physical device limitations. Since DVS is one of the fundamental techniques that low power systems most frequently employ, especially for HPC applications, this is not good news. But there is still hope, as we will see later in the article.

- *Duty cycling / Power* — Another classical methodology is turning off the power when the device is not being used. Many of the ULP devices rely on this technique because they have duty cycles in seconds or even minutes and are effectively turned off most of the remaining time. Dynamic Voltage scaling as well as other techniques are employed extensively along with duty cycling to reduce the idle power as much as possible.

- *Architectural overhead elimination* — There are numerous features in modern-day processors and other peripherals that attempt to obtain relatively small increases in performance at significant hardware and thus power cost. By simplifying the architecture, as is done for embedded processors, one may obtain substantial gains in performance/power ratio while incurring only a small penalty.

- *Exploiting Parallelism (Slow and Parallel)* — Because increases in processor frequency will also incur voltage increases, if we can attain perfect parallel speedup, then reducing the clockspeed in exchange for parallelism (slow and parallel) will generate greater power savings. This is the principle now being employed in various recent multi-core CPU designs; the technical details are covered in the BlueGene/L article ("Lilliputians of Supercomputing Have Arrived") in this issue of *CTWatch Quarterly*.

- *Algorithmic changes* — On the software side, one may save power by fundamentally changing the algorithm to consume less computing steps and/or reducing reliance on power-hungry features of the processors and instead using more efficient portions. While the former is obvious and always exploitable, the latter may not be so obvious and not always exploitable, depending on the underlying hardware. For example, in the latter one may attempt to utilize the on-die temporary memory to reduce the off-chip bus traffic as much as possible. But its effectiveness depends on whether the processor's external bus driver power, relative to the power consumption of the internal processing, would be significant or not.

- *Other new techniques* — there are other technologies in development, which we will not be able to cover here due to lack of space.

How do these techniques apply to different types of systems in order to optimize different kinds of metrics? To clarify the differences, we have divided the power ranges by three orders of magnitudes, namely in the Microwatts, Milliwatts, and Watts and beyond. The table below shows the resulting power ranges and their principal application domains, metrics, technical characteristics, example systems, etc. One can observe here that there is significant divergence in the respective properties of the systems.

Average Power Consumption	Microwatt - Milliwatt	Milliwatt-Watt	Over One Watt
Application Domain	Ubiquitous Sensor Networks	Standard Embedded Devices	PCs/Workstations, Servers, HPC
Important Metrics	Longivity: device powering months~years with a single battery, environmental harvesting of power	Long battery life of dedicated, real-time applications	Maintaining high performance, high thermal density
Technical Characteristics	Programming of Long Duty Cycle applications in Tiny CPU/Storage Environment Ultra Low Power Wireless Autonomous Configuration amongst a group of nodes Fault Tolerance via Massive Redundancy	Various "Classical" low power techniques Adjusting CPU speed voltage in periodic real time processings Dynamic reconfiguration via Software/Hardware co-design	CPU power consumption dominant => "Slow and Parallel" Fine-Grained software control significant – measurement, prediction, planning, (DVS) control Need for low power high performance networking High reliability and scalability
Example Systems	Mote, TinyOS (UC Berkeley)	Various Embedded OSes	NotePC/Blade Server BlueGene/L Green Destiny

Table 2. Low Power System Power Range Categorizations and their Properties.

Examining this table, one might argue that systems in the Microwatt and the Milliwatt ranges do have some similarities, four of which are outlined below:

1. Devices in both categories are typically driven by batteries and/or some independent (solar or energy-harvesting) generators, without AC electrical wirings, and as such longevity of battery life is of utmost concern.

2. Their application space is dedicated, or at least fairly restricted per each device; they are not meant to be general-purpose computing devices to be used for every possible application. Also the applications tend to be real-time in nature, primarily sensing, device control, and multimedia. These application characteristics have two consequences. First, when combined, these properties allow duty cycling to be performed extensively. For example, the Berkeley Mote envisions applications where a single battery will last for months, with sensing and networking duty cycling in phases of tens of seconds to minutes. Second, they will sometimes allow dedicated and/or reconfigurable hardware to be employed for the performance/energy demanding portion of the application, such as multimedia encoding/decoding. In some cases this will bring about orders of magnitude performance/power ratio improvement.

3. Their physical locations tend to be spaced apart, as seen in mobile devices. Coupled with being very low power, thermal density is not the primary concern (although in some modern embedded multimedia devices it may be, but it is not the primary or driving motivation for achieving low power).

4. Although in modern applications they are often networked, these devices do not work together in a tightly-coupled fashion to execute a single application, and as a result network bytes/flop is not as demanding as is with HPC systems.

In the HPC arena, by contrast, the important point to note is that low power is now being considered the essential means to achieve the traditional goals of high performance. This may at first seem oxymoronic, since lower power usually means lower performance in embedded devices and ULP devices, and great efforts are made to "recover" lost performance as much as possible. However, BlueGene/L and other low power, HPC machines that utilize low power technologies have demonstrated that, by exploiting the "slow and parallel" characteristics, we may achieve higher performance.

Still, their properties produce a different opportunity space for low power than embedded or ULP devices confront:

- HPC machines are typically powered by AC; so the motivation is not only lower energy utilization, which is what is needed to extend battery life, but also peak power requirements, as these requirements will mostly determine the necessary capacity of the electrical infrastructure as well as maximum cooling capacity.

- HPC machines are more general purpose, and as such the application space is rather broad. Such applications usually demand continuous computing or are I/O intensive, or both. Also these applications are not necessarily real time, but will usually be optimized to minimize their execution time. This will restrict the use of duty cycling, as any idle compute time will be subject to elimination via some optimization.

- The generality of the application space will make dedicated, hardwired hardware acceleration effective in only a limited set of applications. There are some instances of successful HPC accelerators such as the GRAPE system, but its effectiveness is restricted to a handful of (albeit important) applications.

- Density is one of the driving factors for achieving low power, since some of the large machines are at the limit of practical deployability with respect to their physical size. Simply reducing their volume, however, will result in significant thermal complications, primarily critical "hot spots." Thus, power control that will guarantee that such hot spots will not occur is an absolute must for stable operation of the entire system.

- Many HPC applications are tightly coupled and make extensive use of networking capabilities. Network bytes/flop is an important metric, and the difficulty is to meet the low power requirements in high-bandwidth networking.

Are the Low-power HPC Systems Too Divergent to Traditional, Embedded Low Power Systems?

Given the observations above, could we go as far to say that low power HPC systems are so divergent from traditional embedded systems that there are no research results or engineering techniques they can share? As a matter of a fact, there are commonalities that permit such sharing, and we are starting to see some "convergence" between the low power realization techniques in HPC and those with other power ranges. Here are some of the examples:

- Although it is difficult to duty cycle HPC applications, there are still opportunities to fine tune the usage and exploit the potentially "idle" occasions in the overall processing. One example of this approach, dubbed *power aware computing*, would be to adjust the processor DVS features in a fine-grained fashion so that one can achieve minimum energy consumption for a particular phase in a computation. Another possibility is to exploit the load imbalance in irregular parallel applications, where one may slow down processors so they all synchronize at the same time. Details of the techniques are covered in Dr. Feng's article, "The Importance of Being Low Power in High-Performance Computing," in this *CTWatch Quarterly* issue.

- There are direct uses of low power embedded processors, augmented with HPC features such as vector processing, low power high performance networking, etc. Examples are BlueGene/L, Green Destiny,[4] and MegaProto.[5] Fundamental power savings are realized with lower voltage, smaller number of transistors, intricate DVS features, etc. In fact, BlueGene/L has demonstrated that the use of low power processors is one of the most promising methodologies. There are still issues, however, since the power/performance ratio of embedded processors applied to HPC are not overwhelmingly advantageous, especially with the development of the power efficient processors that will be arriving in 2006-2007, where similar implementation techniques are being used. Moreover, although one Petaflop would be quite feasible with today's technologies, to reach the next plateau of performance, i.e. ten Petaflops and beyond, we will need a ten-fold increase in power/performance efficiency. In light of the limits in voltage reduction and other constraints, the question of where to harvest such efficiency is a significant research issue.

- Dedicated vector co-processing accelerators have always been used in some MPPs; in the form of GPUs, they are already in use in PCs and will be more aggressively employed in next generation gaming machines such as Microsoft's Xbox 360 and Sony's PlayStation 3. Such co-processing accelerators offer much more general purpose programming opportunities than previous generations of GPUs have had, aiding to considerably boost the Flops/power ratio. For example, the Xenon GPU in the Xbox 360 has 48 parallel units of 4-way parallel SIMD vector processors + scalar processors, achieving 216 Gflops at several tens of watts, or about four to seven Gflops/Watt. Also, some embedded processors are starting to employ reconfigurable FPGA devices to dynamically configure hardware per each application. One example is Sony's new flash-based "Network Walkman" NW-E507, where MP3 decode circuitry is programmed on-the-fly in its internal FPGA to achieve 50 hours of playback in a device as small as 47 grams. The use of reconfigurable devices and modern-day, massively-parallel vector co-processors is still not at the stage of massive use within the HPC arena due to cost and technical immaturity but it will be a promising approach for the future.

[4] W. Feng, "Making a Case for Efficient Supercomputing," *ACM Queue*, 1(7):54-64, October 2003.

[5] Hiroshi Nakashima, Hiroshi Nakamura, Mitsuhisa Sato, Taisuke Boku, Satoshi Matsuoka, et. al. (2 more authors) "MegaProto: 1 TFlops/10 kW Rack Is Feasible Even with Only Commodity Technology", *Proc. IEEE/ACM Supercomputing 2005*, the IEEE Computer Society Press, Nov. 2005 (to appear).

The Future of Low Power HPC is "Overdesign" and "Portability"

We have examined the relationships between the various areas of low power computing, focusing especially on the similarities and differences. Overall, the attention given to low power in HPC is still not well recognized by the community, despite the success of BlueGene/L. In particular, controlling power requires a sophisticated application of self-system control. This type of control is being practiced as a norm in other disciplines but is quite crude in computers, especially large HPC machines.

For example, modern fighter aircraft are deliberately made to be somewhat aerodynamically unstable in order to improve their maneuverability; in order to recover and maintain operational stability, they use massive, dynamic, computer-assisted real-time control. Modern automobiles embed massive amounts of self-control for engines and handling without which the car would easily break down or at least suffer from poorer performance. Compared to these technology domains, power/performance controls in modern-day HPC machines are meager at best. They may contain some simple feedback loops that, for example, upturn the cooling fans when the internal chassis temperature climbs higher, or that apply some crude, spontaneous automated control of voltage/frequency without concern for application characteristics. There are other promising avenues of research, as the other articles in this issue show, but further investigation is required to identify the limits of such control methodologies, as well as discover better ways to conserve power.

One promising conceptual design principle that this author envisions is to "overdesign" the system, i.e. engineer it so that, without software self-control, the the system will break down (say thermally), hit other power limits, or become very power/performance inefficient. Most of the machines we design now follow quite conservative engineering disciplines so that no matter how much we hammer them they will not break. Altenatively, we may design for maximum efficiency out of the theoretical peak achievable. Now that we are quickly approaching the one billion transistor mark in our CPUs (and quickly going onto ten billion) there are many transistors to consume power if exploited directly or used for alternative purposes. Moreover, we will have better understanding of how we may monitor and control power, depending on the system/application states (including multiple applications within the system). With multiple failovers in place, we could "overdesign" the system so that it will operate at maximum performance/power ratio (which may be somewhat below the maximum computational efficiency), but driving the efficiency above this will "break" the system. In order to achieve such a subtle balance, there will be various hardware and software sensors to monitor performance/power metrics and perform regulatory feedback into the system, enabling dynamic fine tuning of both software (such as scheduling) and hardware (such as DVS).

Such a design principle may allow substantial improvement in the various metrics that motivate the pursuit of low power in HPC in the first place. For example, one may put an extensive set of thermal sensors in a machine that is densely packed to intricately control the power/performance so as to maintain thermal consistency throughout the system. In such a machine, it would be impossible to achieve theoretical maximum performance, since doing so would break the system and, as a result, some failover mechanism would have to kick in to throttle the system. Overall, its performance per volume may be substantially greater than a conservative machine for various reasons, including that the machine would be running more units in parallel at the best performance/energy tradeoff point.

Many technical challenges would have to be conquered for such a system to become a reality, however. For example, most current motherboards, including sever-grade, high-end versions, lack the sensors required to perform such intricate monitoring of thermal and power consumptions. In many cases, the only available sensors may be a few thermistors, with no power sensors present except voltage meters on power lines. Although the state of the art in analysis of performance/power tradeoffs are advancing (as seen in Dr. Feng's article mentioned previously), most of the results are still early, with no real broad-based community efforts, such as standardization, to enable, facilitate, or promote usage of the technology. In fact, because of the significant effect such low power systems will have on the software infrastructure, including the compilers, run-time systems, libraries, performance monitors, etc., it is currently impractical to expect any portability across different types of machines. Here, theoretical modeling of such machines, leading to eventual standardization, will be necessary for realistic deployment to occur.

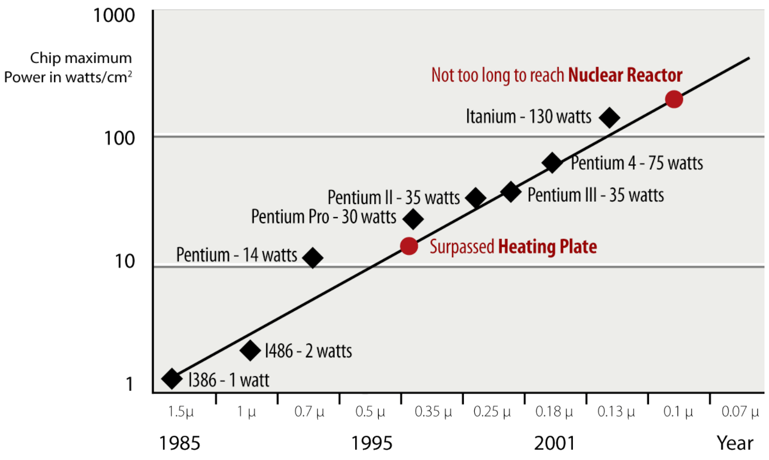
The Importance of Being Low Power in High-Performance Computing

Wu-chun Feng
Los Alamos National Laboratory

Introduction

Why should the high-performance computing community even care about (low) power consumption? The reasons are at least two-fold: (1) efficiency, particularly with respect to cost, and (2) reliability.

For decades, we have focused on performance, performance, and occasionally, price/performance, as evidenced by the Top500 Supercomputer List[1] as well as the Gordon Bell Awards for Performance and Price/Performance at SC.[2] So, to achieve better performance per compute node, microprocessor vendors have not only doubled the number of transistors (and speed) every 18-24 months, but they have also doubled the power densities, as shown in Figure 1. Consequently, keeping a large-scale high-performance computing (HPC) system functioning properly requires continual cooling in a large machine room, or even a new building, thus resulting in substantial operational costs. For instance, given that the cooling bill alone at Lawrence Livermore National Laboratory (LLNL) is $6M/year and given that for every watt (W) of power consumed by an HPC system at LLNL, 0.7 W of cooling is needed to dissipate the power; the annual cost to both power and cool HPC systems at LLNL amounts to a total of $14.6M per year, and this does not even include the costs of acquisition, integration, upgrading, and maintenance.[3] Furthermore, when nodes consume and dissipate more power, they must be spaced out and aggressively cooled; otherwise, such power causes the temperature of a system to increase rapidly enough that for every 10° C increase in temperature, the failure rate doubles, as per Arrhenius' equation as applied to microelectronics.[4]



Soure: Intel

Figure 1. Moore's Law for Power Consumption

Our own informal empirical data from late 2000 to early 2002 indirectly supports Arrenhius' equation. In the winter, when the temperature inside our warehouse-based work environment at Los Alomas National Laboratory (LANL) hovered around 21-23° C, our 128-CPU Beowulf cluster — Little Blue Penguin (LBP) — failed approximately once per week. In contrast, the LBP cluster failed roughly twice per week in the summer when the temperature in the warehouse reached 30-32° C. Such failures led to expensive operational and maintenance costs relative to technical staff working to fix the failures and the cost of replacement parts. Furthermore, there is the lost productivity of technical staff due to the failures.

[1] http://www.top500.org/

[2] http://www.sc-conference.org

[3] M. Seager, "What Are The Future Trends in High-Performance Interconnects for Parallel Computers?" *IEEE Symp. on High-Performance Interconnects Panel*, August 2004.

[4] W. Feng, "Making a Case for Efficient Supercomputing," *ACM Queue*, 1(7):54-64, October 2003.

Perhaps more disconcerting is how our warehouse environment affected the results of the Linpack benchmark when running on a dense Beowulf cluster back in 2002: The cluster produced an answer outside the residual (i.e., a silent error) after only ten minutes of execution. Yet when the same cluster was placed in an 18-19° C machine-cooled room, it produced the correct answer. This experience loosely corroborated a prediction made by Graham, et al — "In the near future, soft errors will occur not just in memory but also in logic circuits."[5]

Power (and its affect on reliability) is even more of an issue for larger-scale HPC systems, such as those shown in Table 1. Despite having exotic cooling facilities in place, the reliability of these large-scale HPC systems is measured in hours,[6] and in all cases, the leading source of outage is hardware, with the cause often being attributed to excessive heat. Consequently, as noted by Eric Schmidt, CEO of Google, what matters most to Google "is not speed but power — low power, because data centers can consume as much electricity as a city."[7] That is, though speed is important, power consumption (and hence, reliability) is more so. By analogy, what Google, and arguably application scientists in HPC, desires is the fuel-efficient, highly reliable, low-maintenance Toyota Camry of supercomputing, not the Formula One race car of supercomputing with its energy inefficiency, unreliability, and exorbitant operational and maintenance costs. In addition, extrapolating today's failure rates to an HPC system with 100,000 processors suggests that such a system would "spend most of its time checkpointing and restarting. Worse yet, since many failures are heat related, the [failure] rates are likely to increase as processors consume more power."[5]

System	CPUs	Reliability
ASCI Q	8,192	MTBI: 6.5 hours. Leading outage sources: storage, CPU, memory.
ASCI White	8,192	MTBF: 5.0 hours ('01) and 40 hours ('03). Leading outage sources: storage, CPU, 3rd-party HW.
PSC Lemieux	3,016	MTBI: 9.7 hours.

MTBI: mean time between interrupts = wall clock hours / # downtime periods

MTBF: mean time between failures (measured)

Table 1. Reliability of Leading-Edge HPC Systems

Low-Power HPC: The Past

Based on the above evidence, I would argue that although performance and price/performance are important, we need to focus more attention on efficiency and reliability in the coming decades. And as contended above, this translates into a substantial reduction in the power consumption of HPC systems via low-power (or power-aware) approaches. Our Green Destiny cluster was arguably one of the first such systems,[4 8 9] designed in late 2001 and debuting in early 2002 as the first major instantiation of the *Supercomputing in Small Spaces* project.[10]

Green Destiny, as shown in Figure 2a, was a 240-CPU Linux-based cluster with a footprint of only five square feet and a power appetite of as little as 3.2 kW (i.e., two hairdryers). Performance-wise, it produced 101 Gflops on the Linpack benchmark, which was as fast as a 256-CPU SGI Origin 2000 at the time.[11] Despite its competitive performance then,[12] many still felt that Green Destiny sacrificed too much performance *to achieve low power consumption, and consequently, high efficiency and unprecedented reliability, i.e., no unscheduled downtime*

[5] S. Graham, M. Snir, and C. Patterson, eds., Getting Up to Speed: *The Future of Supercomputing*, National Research Council, Committee on the Future of Supercomputing, National Academies Press, 2005.

[6] D. Reed, "High-End Computing: The Challenge of Scale," *Director's Colloquium*, Los Alamos National Laboratory, May 2004.

[7] J. Markoff and S. Lohr, "Intel's Huge Bet Turns Iffy," The New York Times, September 29, 2002.

[8] W. Feng, M. Warren, and E. Weigle, "The Bladed Beowulf: A Cost-Effective Alternative to Traditional Beowulfs," *4th IEEE International Conference on Cluster Computing (IEEE Cluster)*, Chicago, IL, September 2002.

[9] G. Johnson, "At Los Alamos, Two Visions of Supercomputing," *The New York Times*, June 25, 2002.

[10] http://sss.lanl.gov; At SC2001 in November, we demonstrated a small-scale 24-node prototype dubbed MetaBlade, running a simulation of a 10-million-body galaxy formation.

[11] http://www.top500.org/list/2001/11

[12] The original performance of Green Destiny on the Linpack benchmark was indeed "low performance" at about 68 Gflops. However, given that the Transmeta CPU was a hardware-software hybrid, we were able to optimize its floating-point performance (in system software) by 50%, resulting in a Linpack rating of 101 Gflops.

in its 24-month lifetime while running at 7,400 feet above sea level in a dusty 85° F warehouse without any cooling, air filtration, or air humidification.

The above tradeoff is captured (in part) in Table 2, where we present the raw configuration and execution numbers of four HPC systems as well their efficiency numbers with respect to memory density, storage density, and computational efficiency relative to space and power consumption.[13] As one would expect from a Formula One race car for supercomputing, the ASCI White supercomputer leads all the raw performance categories (shown in red). On the other hand, given that Green Destiny was specifically designed with low power and high efficiency in mind, it handily "wins" all the efficiency categories: Memory density, storage density, and computational efficiency relative to space and power are all two orders of magnitude better (or nearly so) than the other HPC systems, as shown in red in Table 2.

Metric / HPC System	Avalon Beowulf	ASCI Red	ASCI White	Green Destiny	
Year	1996	1996	2000	2002	
# CPUs	140	9298	8192	240	
Performance (Gflops)	18	600	2500	58	
Space (ft^2)	120	1600	9920	5	
Power (kW)	18	1200	2000	5	
DRAM (GB)	36	585	6200	150	(270 max)
Disk (TB)	0.4	2.0	160.0	4.8	(38.4 max)
DRAM Density (MB/ft^2)	300	366	625	30000	(54000 max)
Disk Density (GB/ft^2)	3.3	1.3	16.1	960.0	(7680 max)
Perf/Space ($Mflops/ft^2$)	150	375	252	11600	
Perf/Power (Mflops/W)	1.0	0.5	1.3	11.6	

Table 2. Comparison of HPC Systems on an n-body Astrophysics Code for Galaxy Formation

Low-Power HPC (and Power-Aware HPC): The Present

The preceding work has now bifurcated into two different directions but both are still oriented towards reducing power consumption: (1) a low-power, architectural approach and (2) a power-aware, software-based approach.

Low-Power, Architectural Approach

In the arena of low-power architectures for HPC, there exist three related but distinct approaches. The first, and most natural, evolution of Green Destiny is the MegaScale Computing project whose goals are more ambitious than Green Destiny's were. The MegaScale Computing project[14] is a multi-institutional project that is looking towards building future computing systems with over a million processing elements in total. Like the Supercomputing in Small Spaces project, the MegaScale Computing project aims to simultaneously achieve high performance and low power consumption via high-density packaging and adopting low-power CPUs, but with the loftier design goals of one Tflop/rack, 10 kW/rack, and 100 Mflops/W. Similar to Green Destiny, their first prototype of an ultra low-power

[13] We note up-front that the comparison is an "apples-to-oranges" one given that the HPC systems are from different eras and have different architectures. The choice of HPC systems was motivated by the fact that we had complete configuration information of the systems and complete and unencumbered access to the systems to tune our n-body code.

[14] http://www.para.tutics.tut.ac.jp/megascale/r_mproto.html

MegaScale system, called MegaProto, also leverages Transmeta CPUs, which deliver very low power but reasonable HPC performance, resulting in extraordinary performance-power ratios.[15] A picture of their MegaProto prototype that was demonstrated at SC2004 is shown in Figure 2b; it is a 16-CPU low-power cluster with dual Gigabit Ethernet for data communication and Fast Ethernet for management and control — all in a compact 1U chassis that consumes only 330 W. (As a point of reference, a traditional dual-CPU compute node consumes 250 W of power. Thus, for 16 CPUs, the aggregate power consumption would run on the order of 2000 W and would then need an additional 1400 W of power to cool the system for a total of 3400 W, or over ten times more power consumption.)

The second and more modest architectural approach to low power is a commercial evolution of Green Destiny, as embodied by Orion Multisystems.[16] The company has two offerings: the DT-12 (i.e., DeskTop-12 nodes) and DS-96 (i.e., DeskSide-96 nodes), as shown in Figure 2c. Their offerings are intended to fill the widening performance gap between PCs and supercomputers, as shown in Figure 3, whereas the ultimate goal of the MegaScale Computing project is to create the capability of constructing a supercomputer with one-million processing elements.

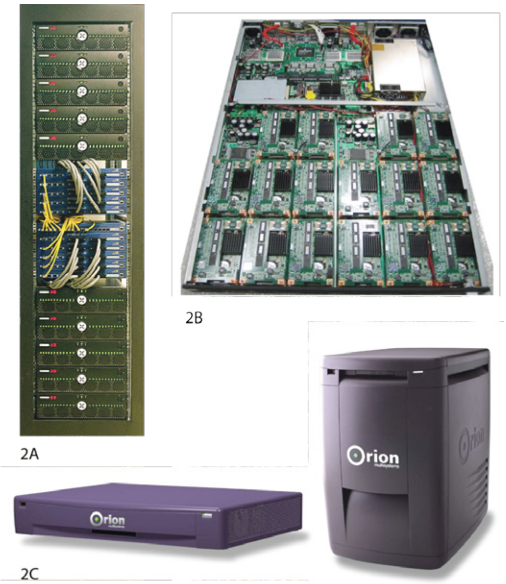


Figure 2a. Green Destiny
Figure 2b. MegaProto: An Ultra Low-Power Prototype of the Megascale Computing Project
Figure 2c. Orion Multisystems DT-12 and DS-96

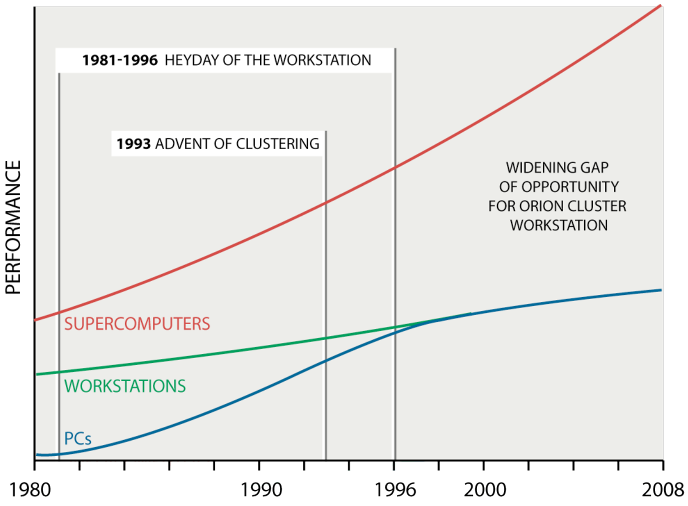


Figure 3. The Widening Performance Gap Between PCs and Supercomputers

Orion Multisystems identified three technology trends that make their offerings ideally positioned as the cluster workstation of the future: (1) the rise of cluster-based high-performance computers, (2) the maturity of open-source cluster software, and (3) the rapid decline of the traditional workstation. By placing a cluster workstation at the hands of an applications scientist, it can be more naturally used as a dedicated personal resource — application debugging with scalability at the desktop, redundancy possibilities whenever the datacenter HPC resource is down and unavailable, and no more scheduling conflicts or long queues for access to a datacenter HPC resource. And perhaps most importantly, by leveraging low-power components, both the DT-12 and DS-96 can be plugged into a standard electrical wall outlet

[15] 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," *IEEE Workshop on High-Performance, Power-Aware Computing (in conjunction with the IEEE Parallel & Distributed Processing Symposium)*, Denver, CO, April 2005.

[16] http://www.orionmulti.com

in any office, as the former only consumes as much power as an overhead light with two 75-W light bulbs and the latter consumes as much as a typical hairdryer, i.e., 1.5 kW.

Finally, the most prominent architectural approach to low-power supercomputing is IBM BlueGene/L, which debuted nine months ago on the Top500 Supercomputer List[1] as the fastest supercomputer in the world, relative to the Linpack benchmark. For an overview of the IBM BlueGene/L architecture and system software, see respective notes.[17] [18] Initial performance evaluations of IBM BlueGene/L can also be found in notes.[19] [20] [21] In short, IBM Blue Gene/L is a very large-scale, low-power (for its size) supercomputer. Its 65,536 CPUs, which are PowerPC 440s, are organized into 64 racks of 1024 CPUs per rack, where each rack of 1024 CPUs consumes only 28.14 kW, resulting in an aggregate power consumption of 1.8 MW.

Given that the only program that has been run across the aforementioned systems is the Linpack benchmark, Table 3 presents the same evaluation metrics as in Table 2 but for the Linpack benchmark.[22] And as in Table 2, Table 3 highlights the leader for a given metric in red.[23] One of the most striking aspects of this table is that IBM Blue Gene/L does *not* use the most amount of space or power despite having the most number of CPUs. Its resulting performance-space and performance-power ratios are consequently astounding, at least relative to Linpack. As an additional reference point, the Japanese Earth Simulator, which has been argued to be the most powerful supercomputer in the world relative to executing *real applications*, reaches 35,860 Gflops for Linpack while occupying 17,222 ft^2 and consuming 7,000 kW. This translates to performance-space and performance-power ratios of 2,082 Mflops/ft^2 and 5.13 Mflops/W, respectively.

Metric \ HPC System	ASCI Red	ASCI White	Green Destiny	MegaProto	Orion DS-96	IBM Blue Gene/L
Year	1996	2000	2002	2004	2005	2005
Performance (Gflops)	2379	7226	101	5.62	110	136800
Space (ft^2)	1600	9920	5	3.52	2.95	2500
Power (kW)	1200	2000	5	0.33	1.58	1800
DRAM (GB)	585	6200	150	4	96	32768
Disk (TB)	2.0	160.0	4.8	n/a	7.68	n/a
DRAM Density (MB/ft^2)	366	625	30000	1136	32542	13107
Disk Density (GB/ft^2)	1	16	960	n/a	2603	n/a
Perf/Space (Mflops/ft^2)	1487	728	20202	1597	37228	54720
Perf/Power (Mflops/W)	2	4	20	17	70	76

Table 3. Comparison of HPC Systems on the LINPACK Benchmark

Despite the performance of HPC systems such as Green Destiny, MegaProto, Orion Multisystems DT-12 and DS-96, and IBM Blue Gene/L, many HPC researchers gripe about the raw performance per compute node, which then requires additional compute nodes to compensate for the lower per-node performance. This, of course, is in contrast to using fewer but more powerful and more power-hungry server processors, e.g., Power5 in ASC Purple, which is slated to require 7.5 MW to power and cool its 12,000+ CPU system. The full system is expected to generate more than 16,000,000 BTU/h in heat, thus requiring new air-handling

[17] IBM and Lawrence Livermore National Laboratory, "An Overview of the BlueGene/L Supercomputer," *IEEE/ACM SC2002: High-Performance Networking & Computing Conference*, Baltimore, MD, November 2002.

[18] G. Almasi, R. Bellofatto, J. Brunheroto, C. Cascaval, J. G. Castanos, L. Ceze, P. Crumley, C. C. Erway, J. Gagliano, D. Lieber, X. Martorell, J. Moreira, A. Sanomiya, and K. Strauss, "An Overview of the Blue Gene/L System Software Organization," *Euro-Par 2003 Conference*, Klagenfurt, Austria, August 2003.

[19] V. Bulatov, W. Cai, J. Fier, M. Hiratani, G. Hommes, T. Pierce, M. Tang, M. Rhee, K. Yates, and T. Arsenlis, "Scalable Line Dynamics in ParaDiS," *IEEE/ACM SC2004: High-Performance Computing, Networking, and Storage Conference*, Pittsburgh, PA, November 2004.

[20] K. Davis, A. Hoisie, G. Johnson, D. Kerbyson, M. Lang, S. Pakin, and F. Petrini, "A Performance and Scalability Analysis of the BlueGene/L Architecture," *IEEE/ACM SC2004: High-Performance Computing, Networking, and Storage Conference*, Pittsburgh, PA, November 2004.

[21] G. Almasi, S. Chatterjee, A. Gara, J. Gunnels, M. Gupta, A. Henning, J. Moreira, and B. Walkup, " Unlocking the Performance of the BlueGene/L Supercomputer," *IEEE/ACM SC2004: High-Performance Computing, Networking, and Storage Conference*, Pittsburgh, PA, November 2004.

[22] We note that in addition to the differences in machine architectures and eras (which makes direct comparisons difficult) that power and space consumption do not scale linearly. So, the presented data should only be taken as ballpark figures.

[23] None of the power numbers include the wattage needed for cooling. This means that for ASCI Red, ASCI White, and IBM Blue Gene/L that the power numbers would increase by a factor of 1.7 to 2.0 times. Furthermore, none of the space numbers include the extra floor(s) needed to cool the HPC systems.

designs and specifications. Furthermore, all the above solutions do *not* rely entirely on commodity technologies, and hence, may not be cost-effective. For instance, Blue Gene/L is a stripped-down version of the 700-MHz PowerPC 400 embedded CPU while Green Destiny relies on a customized high-performance version of Transmeta's code-morphing software (CMS)[24] that improves floating-point performance between 50% and 100%, e.g., 12.6 Gflops on 24 CPUs. In contrast, the 16-processor MegaProto cluster is a custom hardware solution that uses the same processor that Green Destiny did but *without* the high-performance code-morphing software (HP-CMS). Consequently, its 16 CPUs only achieve 5.62 Gflops on Linpack. To address the criticisms with respect to non-commodity parts and low performance, the next section proposes an alternative approach for reducing power consumption, one that is largely architecture-independent and based on high-end commodity hardware.

Power-Aware, Software-Based Approach

Because many systems researchers argue that the low-power architectural approach sacrifices too much performance for low power consumption and high reliability, an alternative approach in HPC has recently emerged — one that is more architecture-independent than the low-power, architectural approach and one that takes the "middle ground" relative to the tradeoff between performance and low power consumption. This alternative approach is a power-aware, software-based one, as described in the cited feasibility studies[25] [26] [27] [28] [29] and autonomic systems.[30] [31] [32] [33] The basic idea is to start with a high-performance, high-power CPU that supports a mechanism called *dynamic voltage and frequency scaling* (e.g., an AMD Opteron with support for PowerNow!) and then to create a power-aware algorithm (i.e., policy) that conserves power by scaling down the CPU supply voltage and frequency at appropriate times, as power draw is directly proportional to the CPU frequency and the square of the CPU supply voltage.

Ideally, the appropriate time to scale down the CPU voltage and frequency is whenever there is an off-chip access that the CPU is blocking-on, e.g., memory access, as the CPU has no reason to "sit and spin its wheels" at the maximum voltage and frequency while waiting for the off-chip accesses to complete. In practice, however, knowing *when* to scale the voltage and frequency and *what* to scale them to are difficult tasksfor the following reasons. First, off-chip memory accesses are done in hardware, thus power-aware software would have no way of knowing that the CPU is waiting on a memory access. Second, changing the voltage and frequency settings must be done judiciously, because at the system level, it currently takes on the order of *milliseconds* (i.e., millions of clock cycles) for the voltage and frequency to transition and stabilize at their new settings.

The current and most ubiquitous approach for power-awareness is based primarily on CPU utilization and is meant to extend the battery life in a laptop computer. When the CPU utilization drops below some threshold, the CPU voltage and frequency are lowered to conserve energy; when the CPU utilization exceeds some threshold, the CPU voltage and frequency are raised to improve performance. While this simple approach is both application and input independent as well as transparent to the end user, it is only effective for interactive use, e.g., laptop usage of Microsoft Office, and depends critically upon the choice of the threshold values.[34] For scientific applications, the approach is ineffective as such applications do not have an abundance of CPU idle time that can be taken advantage of.[32] Therefore, there exists a need for a power-aware algorithm that works effectively on scientific applications.

[24] 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.

[25] X. Feng, R. Ge, and K. Cameron, "Power and Energy Profiling of Scientific Applications on Distributed Systems," *19th IEEE International Parallel & Distributed Processing Symposium*, Denver, CO, April 2005.

[26] V. Freeh, D. Lowenthal, F. Pan, and N. Kappiah, "Using Multiple Energy Gears in MPI Programs on a Power-Scalable Cluster," *ACM Symposium on Principles and Practices of Parallel Programming (PPoPP'05)*, June 2005.

[27] V. Freeh, D. Lowenthal, R. Springer, F. Pan, and N. Kappiah, "Exploring the Energy-Time Tradeoff in MPI Programs on a Power-Scalable Cluster," *19th IEEE International Parallel & Distributed Processing Symposium*, Denver, CO, April 2005.

[28] R. Ge, X. Feng, and K. Cameron, "Improvement of Power-Performance Efficiency for High-End Computing," *1st IEEE Workshop on High-Performance, Power-Aware Computing (in conjunction with the 19th IEEE International Parallel & Distributed Processing Symposium)*, Denver, CO, April 2005.

[29] C. Hsu and U. Kremer, "The Design, Implementation, and Evaluation of a Compiler Algorithm for CPU Energy Reduction," *ACM Conference on Programming Languages Design and Implementation (PLDI'03)*, June 2003.

[30] W. Feng and C. Hsu, "The Origin and Evolution of Green Destiny," *IEEE Cool Chips VII: An International Symposium on Low-Power and High-Speed Chips*, Yokohama, Japan, April 2004.

[31] W. Feng and C. Hsu, "Green Destiny and Its Evolving Parts," Innovative Supercomputer Architecture Award, *19th International Supercomputer Conference*, Heidelberg, Germany, June 2004.

[32] C. Hsu and W. Feng, "Effective Dynamic Voltage Scaling Through CPU-Boundedness Detection," *4th ACM Workshop on Power-Aware Computer Systems*, Portland, OR, December 2004.

[33] C. Hsu and W. Feng, "A Power-Aware Run-Time System for High-Performance Computing," *ACM/IEEE SC2005: The International Conference on High-Performance Computing, Networking, and Storage*, Seattle, WA, November 2005.

[34] D. Grunwald, P. Levis, K. Farkas, C. Morrey, and M. Neufeld, "Policies for Dynamic Clock Scheduling," *4th Symposium on Operating System Design and Implementation (OSDI'00)*, Oct. 2000.

We propose such a power-aware algorithm called β-adaptation, which works on any commodity platform that supports dynamic voltage and frequency scaling (DVFS), [33] e.g., AMD Opteron with PowerNow! Implementing the algorithm in the run-time system results in a power-aware runtime system that transparently and automatically adapts CPU voltage and frequency in order to reduce power and energy consumption while minimizing impact on performance. For example, Figure 4 shows that our power-aware run-time system running NAS-MPI Class C on a four-node, 16-CPU Opteron-based cluster saves nearly an average of 20% CPU energy while impacting performance by only 3% on average. (Note: For the MG benchmark, our β-adaptation algorithm not only reduces energy consumption by 14% but it also *improves* performance slightly.)

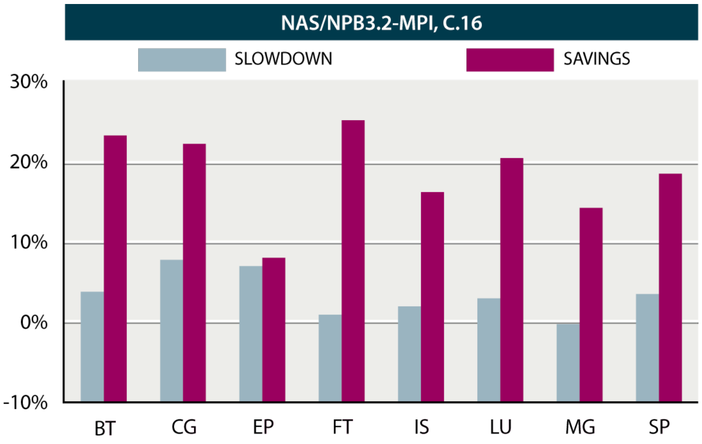


Figure 4. NAS-MPI benchmarks for Class C Workload on a Four-Node, 16-CPU Opteron-based Cluster — http://www.nas.nasa.gov/Software/MPB

Low-Power HPC (and Power-Aware HPC): The Future

Implicit in the preceding discussion is the distinction between capability and capacity computing. According to Graham et al,[5] capability computing applies maximum processing power to solve a large problem in a short period of time — with the main figure of merit being "time to solution." Another important facet to capability computing is the ability to solve problems of a magnitude that have never been solved before. Examples of such systems are the DOE ASCI-class supercomputers such as ASCI White and the recently demonstrated ASC Purple supercomputer — the Formula One race cars of supercomputing.

In contrast, capacity systems are typically cheaper and less performance-capable than capability systems on a per-node basis as well as relative to the entire system. Capacity systems allow scientists to explore design alternatives that are often needed to prepare for larger-scale runs on capability systems. In addition, capacity systems typically solve a multitude of smaller problems simultaneously. Systems such as Green Destiny, MegaProto, Orion Multisystems DS-96, and arguably Blue Gene/L fit into this category.

Because low-power HPC generally sacrifices a measurable amount of performance (e.g., 3.6-GHz Intel Xeon CPU versus 1.4-GHz Transmeta Efficeon CPU) to achieve substantially lower power consumption per node (e.g., 151 W versus 7 W), and hence, better efficiency and reliability, low-power HPC will be confined to capacity computing for the foreseeable future. See citations[35] [36] for the latest results in low-power HPC.

[35] C. Hsu, W. Feng, and J. Archuleta, "Towards Efficient Supercomputing: A Quest for the Right Metric," *1st IEEE Workshop on High-Performance, Power-Aware Computing (in conjunction with the 19th International Parallel & Distributed Processing Symposium)*, Denver, CO, April 2005.

[36] H. Nakashima, M. Sato, T. Boku, S. Matuoka, D. Takahashi, and Y. Hotta, "MegaProto: 1 Tflops/ 10kW Rack Is Feasible Even with Only Commodity Technology," *ACM/IEEE SC2005: The International Conference on High-Performance Computing, Networking, and Storage*, Seattle, WA, November 2005.

But what about capability computing? HPC vendors now realize that in building capability systems, power consumption is becoming a primary design constraint because of the exorbitant operational costs associated with such systems due to their inefficiency and because of its effect of reliability, as noted in Table 1. Excessive power consumption is becoming such a dominant issue that ASC Purple requires new air-handling designs and specifications because of the 7.5-MW required to power the system and the cooling equipment. This 7.5-MW appetite equates to powering 7,500 typical homes.

With low-power HPC unable to support the requirements of capability computing and too much power being consumed by traditional capability systems, what the HPC community should expect to see over the next decade is the emergence of power-aware solutions for capability computing. These solutions will ultimately reduce operational costs and improve reliability and availability, particularly in capacity systems, while minimizing impact on overall performance. We are already seeing indications of this trend at SC2005 where the following three technical papers will be presented on power-aware HPC:

1. R. Ge, X. Feng, and K. Cameron, "Performance-Constrained, Distributed DVS Scheduling for Scientific Applications on Power-Aware Clusters." *Describes a software framework for implementing and evaluating dynamic voltage and frequency scaling, where performance-directed scheduling is of particular interest.*

2. C. Hsu and W. Feng, "A Power-Aware Run-Time System for High-Performance Computing." *Presents a power-aware run-time system on a high-end commodity cluster that automatically and transparently adapts its voltage and frequency settings to achieve about 20% energy savings on average with minimal impact on performance.*

3. N. Kappiah, V. Freeh, and D. Lowenthal, "Just-in-Time Dynamic Voltage Scaling: Exploiting Inter-Node Slack to Save Energy in MPI Programs." *Saves energy by taking advantage of the slack time that exists when the computational load is not perfectly balanced across a HPC system.*

As noted earlier, a power-aware approach makes use of commodity processors (e.g., AMD Opteron[33]) with dynamic voltage and frequency scaling (e.g., PowerNow![33]) to ensure high-end capability performance while reducing power consumption. For the capability supercomputer called ASC Purple, using our power-aware run-time system would reduce the power envelope by 1.3 MW on average, thus reducing its electrical bill by $1.37M/year, when assuming a rate of $0.12/kWh. Furthermore, such a dramatic reduction in power consumption would lengthen the life of system components in the supercomputer, and hence, improve overall reliability of the supercomputer as well as those presented in Table 1.

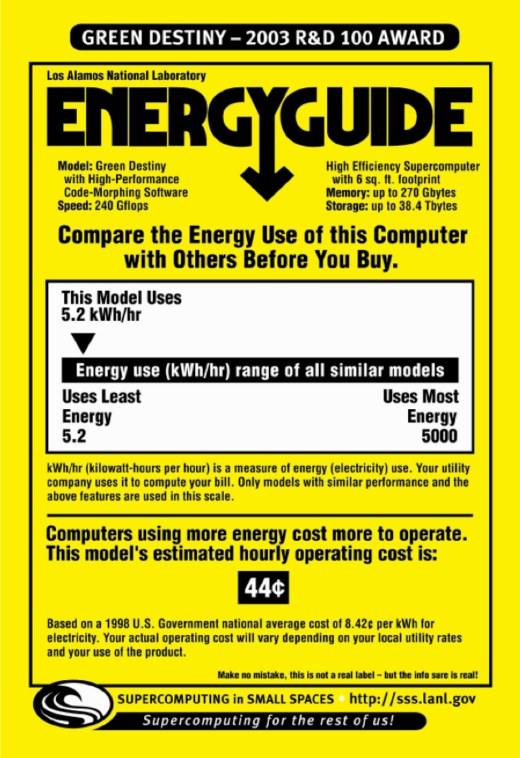
Conclusion

Power consumption has become an increasingly important issue in HPC. Ignoring power consumption as a design constraint results in a HPC system with high operational costs and diminished reliability, which translates into lost productivity. Examples of such (capability) systems include ASCI White, ASC Q, and the recently unveiled ASC Purple.

Specifically, due to the exorbitant power consumption of ASC Purple, the facility that houses ASC Purple requires new air-handling designs and specifications to deal with ASC Purple's gargantuan 7.5-MW appetite. With an average utility rate of $0.12/kWh, the electrical bill alone for this system would run nearly $8M/year. If we scale this architecture up to a petaflop machine, it would need approximately 75 MW to power up and cool down the machine. The power bill for such a system would then be on the order of $80M/year, assuming energy costs stay at $0.12/kWh. In addition, the expected mean time between failures for systems of this size is forecasted to be on the order of hours rather than days; further scaling of such capability supercomputers would result in HPC systems that would have several failures per hour by 2010.[5]

For the above reasons, this article presented a case for low-power (and power-aware) HPC in order to significantly improve reliability and efficiency, particularly with respect to operational costs. However, the main issue with low-power HPC is that it sacrifices too much raw performance in order to achieve its goals. Perhaps what the HPC community needs is an EnergyGuide sticker for HPC systems, like the one shown in Figure 5 for Green Destiny. Or more seriously, perhaps we should remember that our attitude towards energy contributed to the massive rolling blackouts in the summers of 2000, 2001, and 2003 and cost the U.S. billions of dollars and disrupted millions of lives, as noted this month by President George W. Bush when signing the 10-year, $12.3-billion Energy Policy Act of 2005.

As a compromise, there exists an emerging body of research in power-aware HPC. The basic idea is to start with a high-performance, high-power CPU that supports a mechanism called dynamic voltage and frequency scaling and then to create a power-aware algorithm that conserves power by scaling down the CPU supply voltage and frequency at appropriate times, as power draw is directly proportional to the CPU frequency and the square of the CPU supply voltage. Because the CPU consumes the largest percentage of power in a HPC node, this technique has been shown to be highly effective in reducing the overall power and energy consumption in an HPC system.



GREEN DESTINY – 2003 R&D 100 AWARD

Los Alamos National Laboratory

ENERGYGUIDE

Model: Green Destiny
with High-Performance
Code-Morphing Software
Speed: 240 Gflops

High Efficiency Supercomputer
with 6 sq. ft. footprint
Memory: up to 270 Gbytes
Storage: up to 38.4 Tbytes

Compare the Energy Use of this Computer
with Others Before You Buy.

This Model Uses
5.2 kWh/hr

▼

Energy use (kWh/hr) range of all similar models

Uses Least Energy 5.2	Uses Most Energy 5000

kWh/hr (kilowatt-hours per hour) is a measure of energy (electricity) use. Your utility company uses it to compute your bill. Only models with similar performance and the above features are used in this scale.

Computers using more energy cost more to operate.
This model's estimated hourly operating cost is:

44¢

Based on a 1998 U.S. Government national average cost of 8.42¢ per kWh for electricity. Your actual operating cost will vary depending on your local utility rates and your use of the product.

Make no mistake, this is not a real label – but the info sure is real!

SUPERCOMPUTING in SMALL SPACES • http://sss.lanl.gov
Supercomputing for the rest of us!

Figure 5. EnergyGuide Sticker for Green Destiny

In the longer term, e.g., by 2020 when the failure rate is expected to reach several failures *per minute*,[5] we will need the continued proactive approach towards power consumption espoused here in order to stave off the aforementioned forecast as well as reactive fault detection and fault handling in order to give the user the illusion of a fault-free machine.

This work was supported by the DOE LDRD Exploratory Research Program through Los Alamos National Laboratory contract W-7405-ENG-36.

Lilliputians of Supercomputing Have Arrived!

Jose Castanos
George Chiu
Paul Coteus
Alan Gara
Manish Gupta
Jose Moreira
IBM T.J. Watson Research Center

Introduction

In *Gulliver's Travels* (1726) by Jonathan Swift, Lemuel Gulliver traveled to various nations. One nation he traveled to, called Lilliput, was a country that consisted of weak pygmies. Another nation, called Brobdingnag, was that of mighty giants. When we build a supercomputer with thousands to more than hundreds of thousands of chips, is it better to choose a few mighty and powerful Brobdingnagian processors, or is it better to start from many Lilliputian processors to achieve the same computational capability? To answer this question, let us trace the evolution of computers.

The first general purpose computer, ENIAC (Electronic Numerical Integrator And Calculator), was publicly disclosed in 1946. It took 200 microseconds to perform a single addition and it was built with 19,000 vacuum tubes. The machine was enormous, 30 m long, 2.4 m high and 0.9 m wide. Vacuum tubes had a limited lifetime and had to be replaced often. The system consumed 200 kW. ENIAC cost the US Ordnance Department $486,804.22.

In December 1947, John Bardeen, Walter Brattain, and William Shockley at Bell Laboratories invented a new switching technology called the transistor. This device consumed less power, occupied less space, and was made more reliable than those of vacuum tubes. Impressed by these attributes, IBM built its first transistor based computer called Model 604 in 1953. By early 1960, transistor technology became ubiquitous. Further drive towards lower power, less space, higher reliability, and lower cost resulted in the invention of integrated circuits in 1959 by Jack Kilby of Texas Instruments. Kilby made his first integrated circuit in germanium. Robert Noyce at Fairchild used a planar process to make connections of components within a silicon integrated circuit in early 1959, which became the foundation of all subsequent generations of computers. In 1966, IBM shipped the System/360 all-purpose mainframe computer made of integrated circuits.

Within the transistor circuit families, the most powerful transistor technology was the bipolar junction transistor (BJT) rather than the CMOS (Complementary Metal Oxide Semiconductor) transistor. However, compared to CMOS transistors, the bipolar ones, using the fastest ECL (emitter coupled logic) circuit, cost more to build, had a lower level of integration, and consumed more power. As a result, the semiconductor industry moved en masse to CMOS in early 1990s. From then on, the CMOS technology became the entrenched technology, and supercomputers were built with the fastest CMOS circuits. This picture lasted until about 2002 where CMOS power and power density rose dramatically to the point that they exceeded those of the corresponding bipolar numbers in the 1990's. Unfortunately, there was no lower power technology lying in wait to diffuse the crisis. Thus, we find ourselves again at a crossroad to build the next generation supercomputer. According to the "traditional" view, the way to build the fastest and largest supercomputer is to use the fastest microprocessor chips as the building block. The fastest microprocessor is in turn built upon the fastest CMOS switching technology that is available to the architect at the time the chip is designed. This line of thought is sound provided that there are no other constraints to build supercomputers. However, in the real world there are many constraints (heat, component size, etc.) that make this reasoning unsound.

In the mean time, portable devices such as PDAs, cellphones, and laptop computers, developed since the 1990's, all require low power CMOS technology to maximize the battery recharge interval. In 1999, IBM foresaw the looming power crisis and asked the question whether we could architect supercomputers using low power, low frequency, and inexpensive (Lilliputian) embedded processors to achieve a better *collective* performance than using high power, high frequency (Brobdingnagian) processors. While this approach has been successfully utilized for special purpose machines such as the QCDOC supercomputer, this counter-intuitive proposal was a significant departure from the traditional approach to supercomputer designs. However, the drive toward lower power and lower cost remained a constant theme throughout.

We chose an embedded processor optimized for low power and low frequency design, rather than performance. Such a processor has a performance/power advantage compared to a high performance and high power processor. A simple relation is

$$performance/rack \; = \; performance/watt \;\; \text{x} \;\; watt/rack.$$

The last term in this expression, *watt/rack*, is determined by thermal cooling capabilities of a given rack volume. Therefore, it imposes the same limit (of the order of 25 kilowatts) for using either high-frequency, high-power chips or using low-frequency, low-power chips. To maximize *performance/rack*, it is the *performance/watt* term that must be compared among different CMOS technologies. This clearly illustrates one of the areas in which electrical power is critical to achieving rack density. We have found that in terms of *performance/watt*, the low frequency, lower power embedded IBM PowerPC 440 core consistently outperforms high frequency, high power microprocessors by a factor of about ten regardless of the manufacturers of the systems. This is one of the main reasons we chose the low power design point for our Blue Gene/L supercomputer. Figure 1 illustrates the power efficiency of some recent supercomputers. The data is based on total peak Gflops (giga floating-point operations per second) divided by total system power in watts, when that data is available. If the data is not available, we approximate it using Gflops/chip power (an overestimate of the true system Gflops/power number).

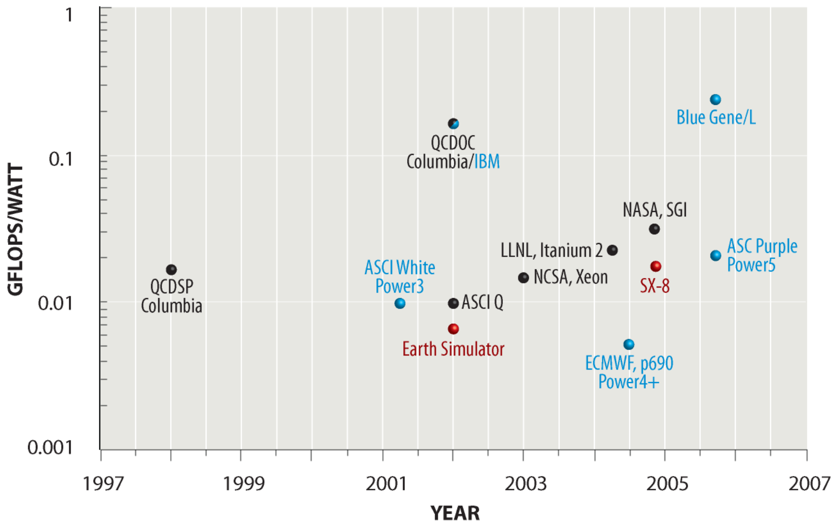


Figure 1. Power efficiencies of recent supercomputers.
(Blue = IBM Machines, black = other U.S. machines, red = Japanese machine)

This chart presents empirical evidence of the fact that *in the presence of a common power envelope, the collective peak performance per unit volume is superior with low- power CMOS technology*. We now explain the theoretical basis of the superior collective performance of low power systems. Any performance metric such as flops , MIPS (millions instructions per sec), or SPEC benchmarks is linearly proportional to the chip clock frequency. On the other hand, the power consumption of the i^{th} transistor is given by the expression:

$$\begin{aligned} P_i &= \text{switching power of transistor i + leakage power of transistor i} \\ &= \tfrac{1}{2} C_{Li} V^2 f_i + \text{leakage power of transistor i,} \end{aligned}$$

where C_{Li} is the load capacitance of the i^{th} transistor, $V = V_{DD}$ is the supply voltage, and f_i is the switching frequency of the i^{th} transistor. Note that not every transistor participates in switching on every clock cycle f. Although the leakage power is increasingly important for 90nm, 65nm and 45nm technologies, we ignore the leakage power of the Blue Gene/L chips which, built in 130 nm technology, contributes less than 2% of the system power. The switching power consumed in a chip is the sum of the power of all switching nodes. It can be expressed as:

$$P_{chip} = \Sigma \text{ switching power of transistor i } = \tfrac{1}{2} C_{sw} V^2 f,$$

where the average switching chip capacitance is given by

$$C_{sw} = (\Sigma\, C_{Li}\, f_i)/f.$$

It is difficult to predict C_{sw} accurately because we seldom know the switching frequencies f_i of every transistor in every cycle, and furthermore f_i is different for each application. To simplify the discussion, we use an averaged value of C_{sw} obtained either from direct measurement or from power modeling tools. For high power, high frequency CMOS chips, the clock frequency f is roughly proportional to the supply voltage V, thus the power consumed per chip P_{chip} is proportional to $V^2 f$ or f^3. Therefore, in the cubed-frequency regime, the power grows by a factor of eight, if the frequency is doubled. If we use eight moderate frequency chips, each of them half the frequency of the original high frequency chip, we burn the same amount of power, yet we have a fourfold increase in flops/watt. This then is the basis of our Blue Gene/L design philosophy. One might ask if we can do this indefinitely. If 100,000 processors at some frequency is good, are not 800,000 processors at ½ the frequency even better? The answer is complex, because we must consider also the mechanical component sizes, power to communicate between processors, the failure rate of those processors, the cost of packaging those processors, etc. Blue Gene/L is a complex balance of these factors and many more. Moreover, as we lower the frequency, the power consumed per chip drops from cubic frequency dependence to quadratic dependence and finally to linear dependence. In the linear regime, both power and performance are proportional to frequency; there is no advantage of reducing frequency at that point.

Blue Gene/L Architecture

The Blue Gene/L supercomputer project is aimed to push the envelope of high performance computing (HPC) to unprecedented levels of scale and performance. Blue Gene/L is the first supercomputer in the Blue Gene family. It consists of 65,536 high-performance compute nodes (131,072 processors), each of which is an embedded 32-bit PowerPC dual processor, and has 33 Terabytes of main memory. Furthermore, it has 1024 I/O nodes, using the same chip that is used for compute nodes. A three-dimensional torus network and a sparse combining network are used to interconnect all nodes. The Blue Gene/L networks were designed with extreme scaling in mind. Therefore, we chose networks that scale efficiently in terms of both performance and packaging. The networks support very small messages (as small as 32 bytes) and include hardware support for collective operations (broadcast, reduction, scan, etc.), which will dominate some applications at the scaling limit. The compute nodes are designed to achieve a 183.5 Teraflops/s peak performance in the co-processor mode, and 367 Teraflops/s in the virtual node mode.[1]

The system on chip approach used in the Blue Gene/L project integrates two processors, cache (Level 2 and Level 3), internode networks (torus, tree, and global barrier networks), JTAG and Gigabit Ethernet links on the same die. By using the embedded DRAM, we have enlarged the on-chip Level 3 cache to four MB, four to eight times larger than competitive cache's made of SRAM and greatly enhancing the amount of realized performance of the processor. By integrating the inter-node networks, we can take advantage of the same generation technology, i.e., these networks scale with chip frequency. Furthermore, the off-chip drivers and receivers can be optimized to consume less power than those of industry standard networks. Figure 2 is a photograph of multi-rows of the Blue Gene/L system. The first two rows have their black covers on, whereas the remaining rows are uncovered.

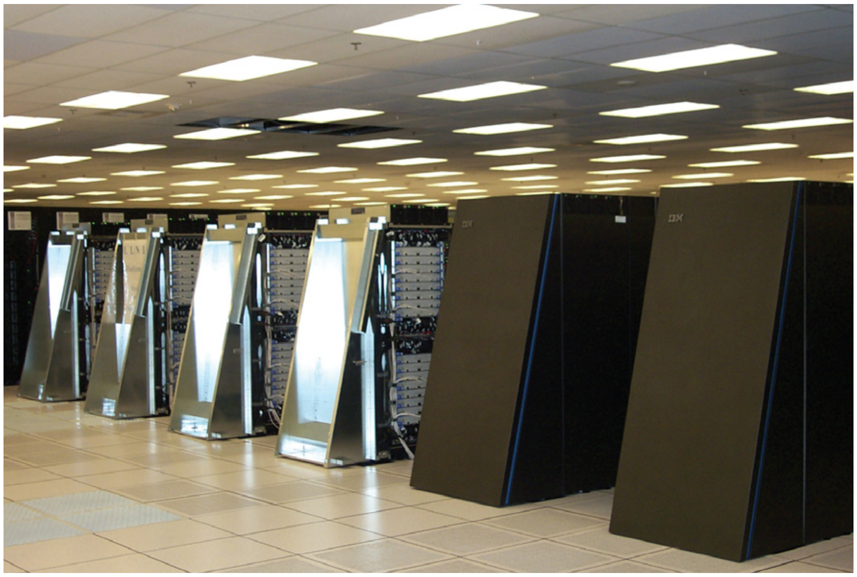


Figure 2. The Blue Gene/L system installed at the Lawrence Livermore National Laboratory.

One of the key objectives in the Blue Gene/L design was to achieve cost/performance on a par with the COTS (Commodity Off The Shelf) approach, while at the same time incorporating a processor and network design so powerful that it can revolutionize supercomputer systems.

Using many low power, power-efficient chips to replace fewer, more powerful ones succeeds only if the application users can realize more performance by scaling up to a higher number of processors. This indeed is one of the most challenging aspects of the Blue Gene/L system design and must be addressed through scalable networks along with software that will efficiently leverage these networks.

[1] IBM Journal of Research and Development, special double issue on Blue Gene, Vol.49, No.2/3, March/May, 2005.

System Software

The system software for Blue Gene/L was designed with two key goals, *familiarity* and *scalability*. We wanted to make sure that high performance computing users could migrate their parallel application codes with relative ease to the Blue Gene/L platform. Secondly, we wanted the operating environment to allow parallel applications to scale to the unprecedented levels of 64K nodes (128K processors). It is important to note that this requires scaling not only in terms of performance but also in reliability. A simple *mean-time-between-failure* calculation shows that if the software on a compute node fails about once a month, under the assumption that failures over all nodes are independent, a node failure would be expected once every 40 seconds! Clearly, this shows the need for compute node software to be highly reliable.

We have developed a programming environment based on familiar programming languages (Fortran, C, and C++) and the *single program multiple data* (SPMD) programming model, with message passing supported via the *message passing interface* (MPI) library. This has allowed the porting of several large scientific applications to Blue Gene/L with a modest effort (often within a day).

We have relied on *simplicity* and a *hierarchical organization* to achieve scalability of software in terms of both performance and reliability. Two major design simplifications that we have imposed are:

- *Strictly space sharing*: only one parallel job can run at a time on a Blue Gene/L partition; we go one step further and support only one thread of execution per processor. This allows us to use efficient, user-space communication without protection problems (the Blue Gene/L partitions are electrically isolated). Furthermore, having a dedicated processor behind every application-level thread leads to more deterministic execution and higher scalability.

- *No demand paging support*: the virtual memory available on a node is limited to the physical memory size. This restriction, besides simplifying the compute node kernel, leads to a performance benefit that there are no page faults or translation lookaside buffer misses during program execution, leading to higher and more deterministic performance.

The software for Blue Gene/L is organized in the form of a three-tier hierarchy. A light-weight kernel, together with the runtime library for supporting user applications, constitutes the programming environment on the compute node. Each I/O node, which can be viewed as a parent of a set of compute nodes (referred to as a *processing set* or *pset*), runs Linux and supports a more complete range of operating system services, including file I/O and sockets, to the applications via offloading from the compute nodes. The Linux kernel on I/O nodes also provides support for job launch. Finally, the control system services run on a service node, which is connected to the Blue Gene/L computational core via a control network.

Results

In October 2004, an 8-rack Blue Gene/L system, which occupied less than 200 square feet of floorspace, and consumed about 200 KW in power, passed the Earth Simulator (which occupies an area of about 70,000 square feet and consumes about seven MW of power) in LINPACK performance. In the recent, June 2005 TOP500 list,[2] a 32 rack Blue Gene/L system, which has been delivered to Lawrence Livermore National Laboratory, occupies the #1 spot with a LINPACK performance of 136.8 Teraflop/s. Blue Gene/L systems account for five of the top ten entries in the June 2005 TOP500 list.

More importantly, several scientific applications have been successfully ported and scaled on the Blue Gene/L system. The applications reported in our studies[3] [4] have achieved, on Blue Gene/L, their highest ever performance. Those results also represent the first proof point that MPI applications can effectively scale to over ten thousand processors.

Conclusions

In this paper, we described the main thrust of the Blue Gene/L supercomputer made of Lillputian low power, low frequency processors. By exploiting the superior performance/watt metric, we can package ten times more processors in a rack, thus it became the number one rated supercomputer since November 2004. In June 2005, five of the top ten supercomputers in the 25th TOP500 list were based on Blue Gene/L architecture. Blue Gene/L is currently producing unprecedented simulation in classical and quantum molecular dynamics, climate, quantum chromodynamics, and the list is growing. The future is likely to be even more power constrained due to the slowing of the power-performance scaling of the underlying transistor technologies. This will likely drive systems to aggressively search for opportunities to build even more power efficient systems, likely driving to more Blue Gene/L-like parallelism. In the future, the Lilliputians are likely to be active in nearly every area of computing.

Acknowledgement

The Blue Gene/L project has been supported and partially funded by the Lawrence Livermore National Laboratory on behalf of the United States Department of Energy under Lawrence Livermore National Laboratory Subcontract No. B517522.

[2] TOP500 Supercomputer Sites, http://www.top500.org/

[3] G. Almasi et al. Scaling physics and material science applications on a massively parallel Blue Gene/L system. *In Proceedings of International Conference on Supercomputing*, Cambridge, MA, June 2005.

[4] G. Almasi et al. Early Experience with Scientific Applications on the BlueGene/L Supercomputer. *In Proceedings of Euro-Par 2005*, Lisboa, Portugal, August-September 2005.

CONFERENCE REPORT

Cyberinfrastructure and the Social Sciences

Fran Berman
San Diego Supercomputer Center

Ruzena Bajcsy
University of California, Berkeley

At first blush, the technical issues involved with designing, implementing, and deploying Cyberinfrastructure seem to present the greatest challenges. Integrating diverse resources to deliver aggregate performance, engineering the system to provide both usability and reliability, developing and building adequate user environments to monitor and debug complex applications enabled by Cyberinfrastructure, ensuring the security of Cyberinfrastructure resources, etc. are all immensely difficult technical challenges and all are more or less still works-in-progress.

After ten years of experience since the I-Way Grid experiment at SC'95, and many more years of experience with team-oriented distributed projects and experiences such as the Grand Challenge program from the 1980s, NSF's large-scale ITR projects, TeraGrid, etc. it is clear that some of the most challenging problems in designing, developing, deploying, and using Cyberinfrastructure arise from the social dynamics of making large-scale, coordinated projects and infrastructure work. From an increasingly substantive experience base with such projects, it is clear that the *cultural, organizational, and policy dynamics,* as well as the *social impact* of Cyberinfrastructure will be critical to its success.

The expansion of the focus on social scientists as **end users** of Cyberinfrastructure to critical **designers** and **process builders** of Cyberinfrastructure motivated the organization of the NSF SBE-CISE Workshop on Cyberinfrastructure and the Social Sciences[1] in March at Airlie House in Warrenton, Virginia. Targeted to a broad spectrum of decision makers and innovative thinkers in the Social Sciences and Computer Sciences, and organized by a multi-disciplinary team of SBE and CISE researchers including a Political Scientist (Henry Brady, UC Berkeley), an Economist (John Haltiwanger, University of Maryland), and two Computer Scientists (Ruzena Bajcsy from UC Berkeley and Fran Berman from SDSC and UC San Diego), the workshop strived to provide substantive, useful and usable feedback to NSF on programs and activities for which the SBE and CISE communities could partner together to build, deploy, and use Cyberinfrastructure. The Airlie workshop focused on two goals:

1. To develop a Final Report that lays out a forward path of Cyberinfrastructure research, experimentation, and infrastructure for the SBE and CISE community and provide a framework for projects and efforts in this integrated area.

2. To provide a venue for community building within the SBE and CISE communities, and in particular, a venue for a multi-disciplinary synergistic community that leverages the perspectives and research of both SBE and CISE constituencies.

Workshop Framework

The Workshop combined distinguished plenary talks from Dr. Arden Bement, Director of the NSF, Dr. Dan Atkins, Chair of the Blue Ribbon Panel on Cyberinfrastructure, and Dr. Nikolaos Kastrinos, Office of Directorate General Research of the European Union Commission, as well as a set of intensive break-outs and report-outs. The group interactions were designed to focus discussions so that the participants could start developing a roadmap for how both social scientists and computer scientists can better impact the design, organization, processes, policies, and impacts of Cyberinfrastructure, as well as how they can

[1] http://www.sdsc.edu/sbe/

improve the relevancy and usefulness of Cyberinfrastructure for the social, behavioral, and economic sciences. Breakout sections focused on the following areas:

Cyberinfrastructure-mediated Interaction
co-chaired by Computer Scientist Ruzena Bajcsy and Psychologist Philip Rubin
How is Cyberinfrastructure-enabled interaction changing relationships? This session focused on the issues involved in developing Cyberinfrastructure-enabled communication mechanisms and their effect on the conduct of science, interpersonal relationships and social networks, and the mediation of cultural and national boundaries.

Cyberinfrastructure Tools for the Social Sciences
co-chaired by Political Scientist Henry Brady and Computer Scientist Allan Snavely
What tools are needed to facilitate social science? This session focused on developing a set of needs and requirements for social scientists as well as a focus on the characteristics and distribution of effective tools.

The Economics of Cyberinfrastructure
co-chaired by Economist Jeff Mackie-Mason and Computer Scientist Rich Wolski
The session focused on two framing questions: What can economics contribute to CS research about efficiently building and operating Cyberinfrastructure?; and How can computer science help identify ways in which to effectively use Cyberinfrastructure to answer economic questions?

The Organization of Cyberinfrastructure and Cyberinfrastructure-enabled Organizations
co-chaired by Computer Scientist Fran Berman and Public Policy Professor Jane Fountain
How will Cyberinfrastructure transform organizations and how can effective organizational approaches transform Cyberinfrastructure? The session focused on the models, frameworks, incentive structures and other mechanisms for advancing the organizational use and structure of Cyberinfrastructure.

Malevolent Uses of Cyberinfrastructure
co-chaired by Statistician Stephen Fienberg and Engineer Shankar Sastry
How can we protect Cyberinfrastructure from intended or unintended malevolent use? This session examined the broad spectrum of security, policy, privacy, confidentiality and other issues critical to ensuring the safe and secure use and development of Cyberinfrastructure.

The Impact of Cyberinfrastructure on Jobs and Income
co-chaired by Economist John Haltiwanger and Computer Scientist Stephen Wright
How has and will Cyberinfrastructure change the workplace? This session focused on the impact of Cyberinfrastructure in the workplace and the implication for firms, markets, and competitiveness.

The almost 100 participants (including roughly 20 participants from the National Science Foundation) felt that the Airlie Workshop constituted the beginning for a critical and important community of Cyberinfrastructure designers, builders and users. The workshop Final Report provides a comprehensive summary of the issues and discussions at the workshop. Arden Bement commented "This SBE-CISE workshop broke new ground by enabling these communities to explore key issues and opportunities for collaboration in designing, developing and delivering better information infrastructure. The final report leverages the immense expertise of NSF communities to develop useful and usable Cyberinfrastructure to support breakthrough science and engineering research and education for the 21st century."

The report is expected to become a key document in the development of NSF's Cyberinfrastructure plan and serves to outline a broad agenda of participatory research, infrastructure, and educational opportunities for both Social Scientists and Computer Scientists.

Workshop Findings

Participants in the workshop explored the concepts of Social Science-enabled Cyberinfrastructure and Cyberinfrastructure-enabled Social Science. Participants also identified key challenges in the social impacts and implications of Cyberinfrastructure. The Final Report[2] Executive Summary describes the following conclusions drawn from workshop discussions:

> *1. "**Cyberinfrastructure can make it possible for the SBE sciences to make a giant step-forward** — Cyberinfrastructure can help the social and behavioral sciences by enabling the development of more realistic models of complex social phenomena, the production and analysis of larger datasets (such as surveys, censuses, textual corpora, videotapes, cognitive neuroimaging records, and administrative data) that more completely record human behavior, the integration and coordination of disparate datasets to enable deeper investigation, and the collection of better data through experiments and simulations on the Internet.*
>
> *… Cyberinfrastructure provides the ability to do these things at unprecedented scale and intensity … just at a time when social and behavioral scientists face the possibility of becoming overwhelmed by the massive amount of data available and the challenges of comprehending and safeguarding it.*
>
> *2. "**SBE scientists can help CISE researchers design a functional and effective Cyberinfrastructure which achieves its full potential** — Cyberinfrastructure requires unprecedented organization, coordination, and integration and will have immense impact on the social dynamics, technological resources, and communication and interaction paradigms for both science and society. … SBE leaders are needed to help guide the design, development, and deployment of a functional Cyberinfrastructure …*
>
> *3. "**Together, SBE and CISE researchers can assess the impacts of Cyberinfrastructure on society and find ways to maximize the benefits of Cyberinfrastructure** … It is already an accepted part of the mission of the SBE sciences to assess societal impact, but it is particularly important to assess the impacts of Cyberinfrastructure for engineering and the sciences. Social and behavioral scientists can be especially helpful in understanding changes in social interactions, changes in jobs and income, the impact of policy, and new conceptions of privacy and trust in the networked world. …"*

The Final Report Executive Summary continues

> *"… true collaborative research is needed between SBE and CISE researchers. In order to achieve this, both intellectual and material interfaces must be shared. For example, it is not sufficient for SBE researchers to be told about Cyberinfrastructure possibilities if they do not*

[2] http://vis.sdsc.edu/sbe/reports/SBE-CISE-FINAL.pdf

> *possess the technical expertise to understand their ramifications. Many SBE researchers lack the technical know-how to participate without significant support from Cyberinfrastructure experts. Similarly, CISE researchers often lack sufficient domain-specific knowledge to appreciate the complexity of the technical problems that truly need to be solved by SBE researchers. The level of knowledge required by both sides will require true collaboration between the two research communities to make a joint research initiative successful. SBE researchers must become familiar with emerging Cyberinfrastructure technologies and CISE researchers must learn about the social sciences."*

One of the concrete outcomes of a successful integrative workshop is the number of collaborations generated out of issues exposed within workshop discussions and collaborations begun at the workshop. Based on this criteria, the NSF SBE-CISE Workshop on Cyberinfrastructure and the Social Sciences was a resounding success. More information on the workshop can be found at http://www.sdsc.edu/sbe/ .

OPINION EDITORIAL

On Perfect Storms, Competitiveness, and the "Gretzky Rule"

Fran Berman
Director, San Diego Supercomputer Center
HPC Endowed Chair, UC San Diego

Recent articles in community publications have focused on the critical need for capable high performance computing (HPC) resources for the open academic community. Compelling reports from the National Research Council, PITAC, the National Science Board, and others point to our current diminished ability to provide adequate computational and data management support for U.S. researchers, and the impact of insufficient technology capacity and capability on the loss of U.S. competitiveness and leadership.

As stated compellingly and increasingly, adequate capability and capacity in HPC is necessary, but it is not sufficient for leadership and competitiveness in science and engineering. Beyond the gear, concrete and strategic goals are critical to achieve competitiveness in science and engineering.

What do we want to accomplish as a nation in science and engineering? Competitiveness for many is reduced to an HPC "arms race" — who has the top spots on the Top500 list? For others, competitiveness amounts to U.S. dominance in the science and engineering world, represented by the number of awards, prizes, and other recognitions for U.S. researchers. For still others, competitiveness is represented by what researchers and educators see as the diversion of a looming "perfect storm" — decreasing funding for science and engineering in the U.S., increasing outsourcing of people and ideas to Europe, Asia and elsewhere, and decreasing students graduating in the sciences and engineering.

For any definition of competitiveness, the means to the end is a serious application of the Gretzky Rule: "Skate to where the puck will be." It is clear that we need concrete goals and a plan, timetable, and resources to achieve them. But what should our goals be? Which goals should have priority over others? How should we accomplish our goals? More funding is an easy answer, and indeed, nothing substantive can be done without resources. But leadership, concrete goals, and a strategic plan for achieving these goals ranks just as highly to ensure that funding is well spent and our efforts are successful.

So how can we apply the Gretzky rule to the going definitions of competitiveness?

The Gretzky Rule and Competitiveness in the HPC "Arms Race"

These days, competitiveness in high performance computing is commonly measured by ranking on the Top500 list.[1] This approach is inadequate to really measure architectural innovation, robustness, or even performance on applications that do not resemble the Linpack benchmarks; however, it is an easy measure and it has been effective in making the case for competitiveness beyond the scientific community. The current top spot on the list is occupied by Livermore's Blue Gene, however the emergence several years ago of the Japanese Earth Simulator (now at spot 4) provided a "wake up call" (Dongarra called it "computenik" in the *New York Times*) to the U.S.

The Earth Simulator provides a textbook application of the Gretzky Rule: Japan committed roughly 5 years and 500 million dollars to planning and executing the Earth Simulator, which stayed at the top spot on the Top500 list between June 2002 and June 2004 inclusive. Careful

[1] http://www.top500.org/

planning, investment, and commitment enabled the Earth Simulator to create an impact which is still being felt in the U.S. and Europe.

So what did we learn about competitiveness from the Earth Simulator? A concrete goal achieved by strategic planning, commitment, and resources over an appropriate timeframe made this a reality.

The Gretzky Rule and Competitiveness in Science and Engineering Research

For most academics, competitiveness is measured by quality of results and track record through publications, and the most highly valued research and researchers are candidates for community prizes — the Fields Medal (mathematics), the Turing Award (computer science), the Pulitzer Prize (literature), and of course, the Nobel Prize (various disciplines). The ultimate goal of competitiveness is leadership, and to achieve the kind of leadership recognized by community prizes, researchers must devote many years in an environment that supports creativity, innovation, deep thinking, and does not penalize the many false starts, wrong turns, and other building blocks that lead to our best and most important results.

To create an environment in which U.S. scientists and engineers are competitive involves developing an environment where the best, the brightest, and the most creative can work, and over the long periods of time that are required for fundamental advances. For many of today's scientists and engineers, infrastructure and professional support is decreasing in the university environment, and there is increasing difficulty in getting funded by federal agencies (currently the "hit rate" for computer science and engineering proposals at the NSF is 20% or less, i.e. only one in every five proposals is funded). In addition, increasing risk aversion in the funding environment penalizes against bold, long-term, or unusual approaches.

Optimizing for competitiveness in science and engineering research mandates a different approach than the HPC "arms race" to the provision of high performance computational and data management infrastructure as well. Rather than optimizing for Top500 ranking, enabling HPC platforms for the researchers who need them must optimize for the support of real science and engineering applications. Data-intensive HPC applications, latency tolerant grid-friendly applications, latency-intolerant "traditional HPC" applications, etc. require a diverse set of capable and high-capacity HPC architectures to best support the diverse needs of the broad academic community. One size (architecture or site) does not fit all here. At the same time, we can't currently afford 100's or perhaps even 10's of these facilities — economies of scale must be applied to optimize for adequate capacities and capabilities, as well as the costs of support, maintenance, and user service and training required to best leverage national-scale HPC resources for the broad community.

So how can we become more competitive in U.S. science and engineering research? Our research and education portfolio would benefit from the same approach we use to balance our personal investments. We should be investing in a strategic balance of short-term, long-term, high-risk, and low-risk endeavors. We should acknowledge that infrastructure enables new discovery but also incurs cost. If the "puck" is leadership through a greater U.S. percentage of top prizes and high-impact results, we need to focus our resources on developing an environment where this can happen, and begin skating in that direction.

Competitiveness in Sustaining a Science and Engineering Workforce — A Perfect Storm Looming

The outsourcing of research, education, service, and innovation is an increasing focus for discussion in the public and private sector. According to Science Resource Statistics[2], as of 2003, 22% of professional scientists and engineers did not have a B.A. or B.S. and only 9% held Ph.D.s and professional degrees. The number of doctorates awarded have been decreasing in science and engineering since 1998,[3] and despite the fact that our kids are increasingly technology-savvy, as a society, our understanding of science and engineering is seriously limited. The National Science Board's 2004 Science and Engineering Indicators report states "Many people do not seem to have a firm understanding of basic scientific facts and concepts."[4]

For many of us in academia, the increasing competitiveness of our colleagues in Europe and Asia through committed funding programs and resources, the drop in support in the U.S. for research, education, and information infrastructure, and the increased outsourcing of technology innovation and service outside of the U.S. are creating a "perfect storm" that will batter U.S. leadership and competitiveness not just now, but over the next generation. Investment in maintaining and sustaining a competitive U.S. workforce in science, engineering, and technology is a long-term investment. It will require planning, commitment, and resources for our educational system, expansion of our training environments, and evolution of our cultural perceptions to recognize the critical role science and engineering play in driving key societal challenges such as better health, improved safety, a sustainable environment, etc.

If we have a concrete idea of where we want the puck to be, it's much easier to skate there. Setting strategic priorities and concrete goals, commitment to providing the leadership, perseverance, and resources to meet those goals, and responsibly estimating the costs and the timeframes required to reach them are key to competitiveness and leadership. This is not rocket science, but without a more thoughtful and strategic approach, advances and new discoveries in rocket science and other disciplines will be much more difficult to achieve.

Acknowledgement

From "On Perfect Storms, Competitiveness, and the 'Gretzky Rule'" — HPCWire, Volume 14, Issue 30, July 29, 2005. Reprinted by permission of Tabor Communications, Inc.

[2] http://www.nsf.gov/sbe/srs/infbrief/nsf04333/start.htm

[3] http:www.nsf.gov/statistics

[4] http://www.nsf.gov/statistics/seind04/c7/c7i.htm#c7il1

PUBLISHERS

Fran Berman, Director of SDSC

Thom Dunning, Director of NCSA

EDITOR-IN-CHIEF

Jack Dongarra, UTK/ORNL

MANAGING EDITOR

Terry Moore, UTK

CENTER SUPPORT

Greg Lund, SDSC

Karen Green, NCSA

PRODUCTION EDITOR

Scott Wells, UTK

GRAPHIC DESIGNER

David Rogers, UTK

CTWatch QUARTERLY

Volume 1 Number 3 August 2005

THE COMING ERA OF LOW POWER, HIGH-PERFORMANCE COMPUTING

TRENDS, PROMISES, AND CHALLENGES

GUEST EDITOR: SATOSHI MATSUOKA, TOKYO INSTITUTE OF TECHNOLOGY

AVAILABLE ON-LINE:
www.ctwatch.org/quarterly/

CTWatch
http://www.ctwatch.org/

E-MAIL:
quarterly@ctwatch.org

ICL UT
http://icl.cs.utk.edu/

NCSA
http://www.ncsa.uiuc.edu/

SDSC
http://www.sdsc.edu/

CTWatch Quarterly is a publication of the CyberInfrastruture Partnership (CIP).