RC23555 (W0502-117) February 23, 2005
Computer Science

IBM Research Report

Power and Performance Optimization at the System Level

Valentina Salapura, Randy Bickford, Matthias Blumrich, Arthur A. Bright, Dong Chen, Paul Coteus, Alan Gara, Mark Giampapa, Michael Gschwind, Manish Gupta, Shawn Hall, Ruud A. Haring, Philip Heidelberger, Dirk Hoenicke, Gerard V. Kopcsay, Martin Ohmacht, Rick A. Rand, Todd Takken, Pavlos Vranas
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Power and Performance Optimization at the System Level

Valentina Salapura, Randy Bickford, Matthias Blumrich,
Arthur A Bright, Dong Chen, Paul Coteus, Alan Gara,
Mark Giampapa, Michael Gschwind, Manish Gupta, Shawn Hall,
Ruud A Haring, Philip Heidelberger, Dirk Hoenicke,
Gerard V Kopcsay, Martin Ohmacht, Rick A Rand, Todd Takken,
Pavlos Vranas
IBM T.J. Watson Research Center
Yorktown Heights, New York

ABSTRACT

The BlueGene/L supercomputer has been designed with a focus on power/performance efficiency to achieve high application performance under the thermal constraints of common data centers. To achieve this goal, emphasis was put on system solutions to engineer a power-efficient system. To exploit thread level parallelism, the BlueGene/L system can scale to 64 racks with a total of 65536 computer nodes consisting of a single compute ASIC integrating all system functions with two industry-standard PowerPC microprocessor cores in a chip multiprocessor configuration. Each PowerPC processor exploits data-level parallelism with a high-performance SIMD floating point unit.

To support good application scaling on such a massive system, special emphasis was put on efficient communication primitives by including five highly optimized communication networks. After an initial introduction of the BlueGene/L system architecture, we analyze power/performance efficiency for the BlueGene system using performance and power characteristics for the overall system performance (as exemplified by peak performance numbers).

To understand application scaling behavior, and its impact on performance and power/performance efficiency, we analyze the NAMD molecular dynamics package using the ApoA1 benchmark. We find that even for strong scaling problems, BlueGene/L systems can deliver superior performance scaling and deliver significant power/performance efficiency. Application benchmark power/performance scaling for the voltage-invariant $energy \times delay^2$ power/performance metric demonstrates that choosing a power-efficient 700MHz embedded PowerPC processor core and relying on application parallelism was the right decision to build a powerful, and power/performance efficient system.

Categories and Subject Descriptors

C.0 [**Computer Systems Organization**]: System Architectures; C.5.1 [**Computer System Implementation**]: Large and Medium Computers—*Super computers*; C.4 [**Performance of Systems**]: Performance Attributes—*application performance, power/performance efficiency*

General Terms

Algorithms, Performance, Design

Keywords

BlueGene/L, supercomputers, chip multiprocessors, power/performance efficient systems, application performance analysis, power/performance tradeoffs in systems, application scaling in multiprocessor systems

1. INTRODUCTION

In November 2004, a 16 rack configuration of the BlueGene/L system became the number #1 supercomputer in the world, at a sustained performance of 70.72 TeraFLOPS (LINPACK). In this paper, we analyze decisions and design choices which allow an air-cooled system based on the PowerPC industry standard architecture and a standard ASIC design flow to achieve this level of performance, surpassing a variety of highly specialized custom designed high performance systems.

The key decision in achieving the performance goals within the available design constraints was to optimize the system to exploit data parallelism, not single node performance. It was a stated goal of the BlueGene project to stay within the confines of traditional air-cooled data centers, which typically offer power and cooling limits of 400-1600 kW. With an approximate power consumption per rack of 25 kW, the completed BlueGene/L system with 64 racks will have a heat load of 1600 kW, fitting in the envelope of high-end data centers.

Many installations today use clusters of PCs attached to an Ethernet backbone to provide a large number of compute cycles - as evidenced by recent editions of the Top500 list maintained by the Universität Mannheim and Univ. of Tennessee at Knoxville. While these machines offer a high

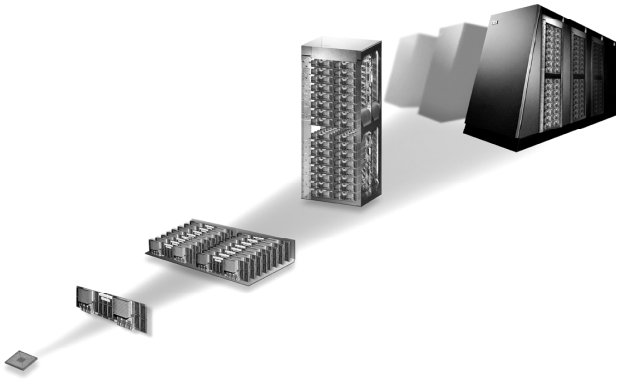

Figure 1: The BlueGene/L concept leverages parallelism and advanced packaging to deliver superior power/performance.

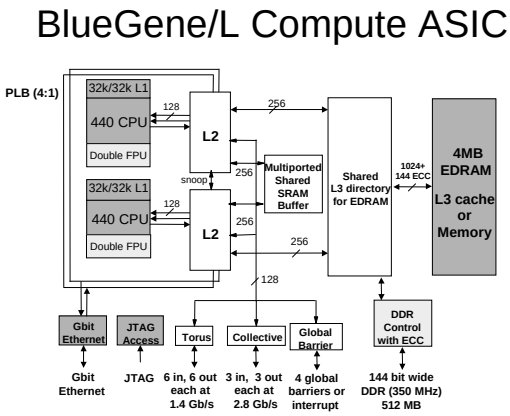

Figure 2: A single compute node ASIC integrates all node functions for the BlueGene compute and IO nodes.

peak FLOP rating which can be applied to highly parallel problems, this peak performance may not translate well into good sustained application performance due to limitations in the interconnect network.

To ensure good scaling, we have paid particular attention to interconnect networks, by providing five high performance networks in the BlueGene/L system. These have been highly optimized and integrated into the system architecture from the beginning of the design. To reduce communication overhead, the network interfaces are located on the same chip as the processing units and implemented with an optimized System-on-a-Chip design flow [3].

While BlueGene/L utilizes a novel system architecture based on a Chip Multiprocessor configuration (CMP), and unique power/performance tradeoffs during the design process, BlueGene/L also leverages the software infrastructure of the industry-standard PowerPC architecture. The aim was to differentiate where big gains were possible, and use standard components everywhere else. The standard components include industry standard networks from the Blue ASIC CoreConnect library, the PowerPC 440 processor core, embedded DRAM from the IBM 0.13μ CU-11 process technology, and IBM XL compiler technology.

Optimized components include the "Double Hummer" SIMD floating point architecture (based on a standard PowerPC floating point unit by replicating key functionality), collective and torus networks, and an optimized memory hierarchy with software-managed coherence between cores.

The remainder of the paper is structured as follows: section 2 gives an overview of the BlueGene/L architecture. Section 3 analyzes design constrains and describes power and performance efficiency of BlueGene/L systems. Section 4 analyzes power/performance application scaling, and section 5 describes the BlueGene/L software stack. We conclude in section 6.

2. SYSTEM OVERVIEW

To achieve high system performance, the BlueGene/L design opts for parallelism instead of high frequency. While frequency and voltage scaling, and/or more aggressively pipelined microprocessors achieve the highest single-node performance, the marginal cost of performance is extremely high, exceeding 2% increase in energy for 1% increase in performance [9]. Given that the heat dissipation of an air-cooled rack is limited, the most energy-efficient approach to reach maximum performance under power-constrained conditions is parallelism.

The BlueGene/L computer is a scalable system consisting of 65,536 nodes based on IBM CMOS CU-11 technology. Each node is built around a single System-on-Chip CMP compute node based on the PowerPC 440 processor core and 9 or 18 SDRAM-DDR memory chips. The BlueGene/L compute node contains a prefetching L2 cache and 4MB high bandwidth embedded DRAM used as on-chip L3-cache shared between the two processors on a chip. This high density/low component count SoC-based design approach is important to reach an optimum cost/performance point. As density decreases the system size and cost grows. Reliability suffers with decreasing density when the number of connectors and cables increases.

The PowerPC 440 microprocessor is a high-performance, out-of-order industry-standard PowerPC processor originally targeted at high-end embedded systems. The processor supports 2-way superscalar instruction execution with a seven stage pipelined microarchitecture. The processor core include highly associative first level instruction and data caches with a capacity of 32KB each. To processor includes standard embedded SoC interfaces based on the IBM CoreConnect specification.

While the BlueGene/L compute node uses the CoreConnect specification to attach to the PowerPC 440 core, as well as to other elements of the IBM BlueASIC library such as high-performance Ethernet interfaces, the core uses a multilevel cache hierarchy instead of a CoreConnect bus to interconnect system components.

System packaging is an integrated aspect of the BlueGene/L system design. In this design, a single rack consists

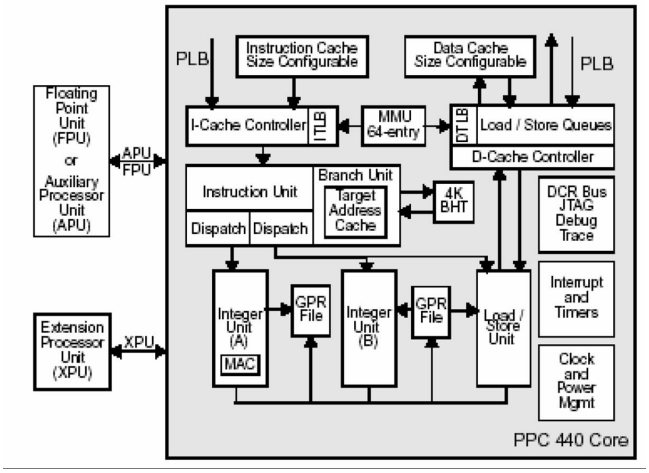

Figure 3: The PowerPC440.

of two midplanes. A midplane is populated with 512 processing nodes, with 5.6GF per compute node. In addition to compute nodes, a midplane is also populated with several I/O nodes. These I/O nodes are in fact implemented using the same ASIC SoC which implements the compute node, but configured to handle file I/O and host communication.

BlueGene/L addresses communication requirements to achieve good application performance scaling by providing 5 dedicated communication networks: the torus network, collective network, barrier network, Ethernet and IEEE1149.1 (JTAG). The networks are described in more detail in [4].

An important element of the BlueGene/L system concept is support for multiple concurrent users. This "multi-user" mode is accomplished through logical partitioning (LPAR) of the machine which allows each user to have a dedicated set of nodes for the user application including dedicated network resources. This partitioning is accomplished through the use of the link chips.

All of BlueGene/L's networks pass through the BlueGene/L link chip, as the network links cross midplane boundaries. The link chip is used to redrive signals to preserve the high speed signal characteristics over the cabling across midplanes. The link chip can also redirect signal between its different ports. This redirection function enables partitioning of a single BlueGene/L system into multiple, logically separate systems.

The BlueGene/L compute node is a 2-processor CMP based on the PowerPC 440 core with a SIMD floating point unit achieving a peak performance of 2.8 GFLOPS per core. The PowerPC architecture Embedded Processor Option (EPO) allows for user-defined extensions to the ISA. Additionally, the Auxiliary Processor Unit (APU) interface on the PowerPC 440G5 Core allows coprocessors to support new instructions – referred to as APU instructions – without requiring modifications to the CPU core. While APU instructions typically do not become part of the architecture proper, they can still be utilized by assemblers and compilers that target the specific implementation.

A SIMD approach was advantageous because it allows simultaneous loading and multiple parallel executions while also reducing the size of the code footprint, and the required bandwidth for power-intensive instruction fetching and issuing. The Double Hummer SIMD architecture goes beyond the advantages of adding another pipeline in a SIMD ap-

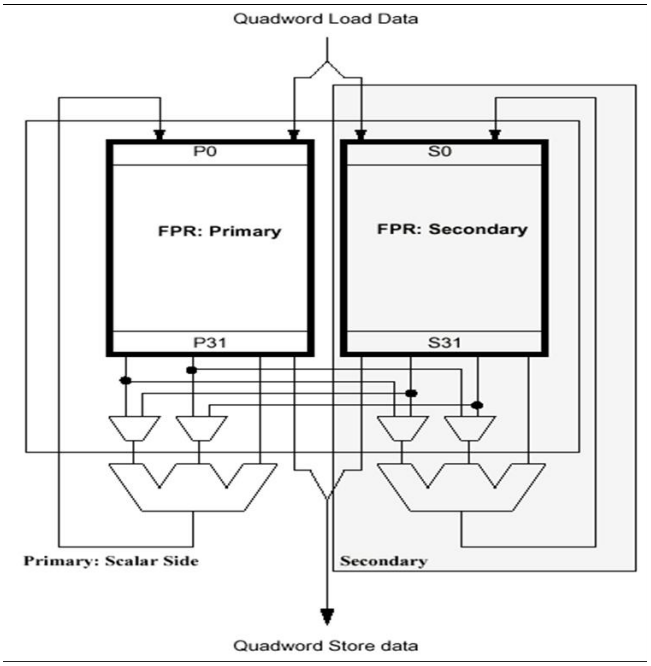

Figure 4: The Double Hummer dual-floating point SIMD architecture extends the traditional PowerPC architecture to deliver 4 floating point operations per cycle with a single instruction.

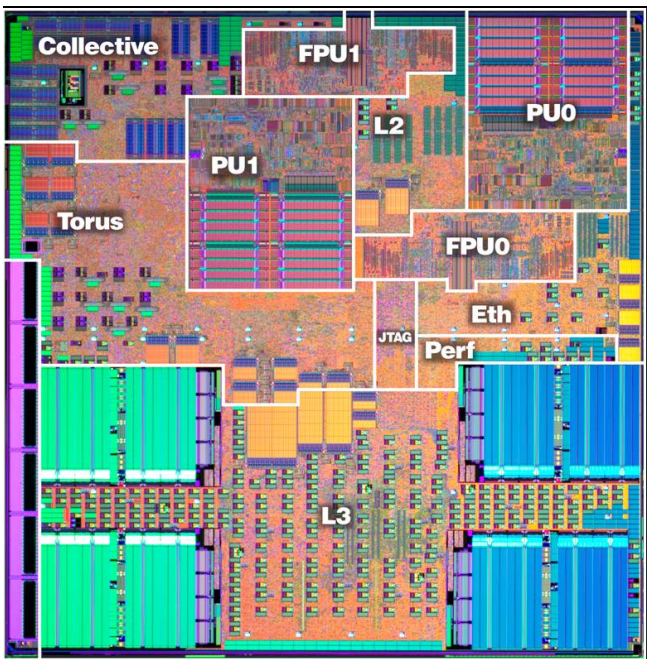

Figure 5: The BlueGene/L compute node chip integrates two PowerPC 440 processors with a SIMD FP2 unit, dense EDRAM L3 on-chip cache and I/O capabilities to drive several high-performance communication networks.

proach. Figure 4 shows the design of the FP2 core. The Double Hummer uses two copies of the architecturally defined PowerPC floating-point register file. Both register files (primary and secondary) are independently addressable; in addition, they can be jointly accessed by SIMD instructions.

The primary register file is used in the execution of the pre-existing PowerPC floating-point instructions as well as the new SIMD instructions, while the secondary register file is reserved for use by the new instructions. Along with the two register files, there are also primary and secondary pairs of datapaths, each consisting of a computational datapath and a load/store datapath.

To reduce the cost of maintaining cache coherence between the nodes on a system, BlueGene/L uses the MPI message passing programming model between the 65536 nodes, and software-managed coherence between the two cores on a compute node. While hardware-managed cache coherence is normally a key ingredient to ensure correct multiprocessor operation, carefully tuned applications, such as those targeting high end supercomputers, can usually be tuned to not required hardware cache coherence.

The final aspect of low power design in BlueGene/L was the System-on-a-Chip design approach. By leveraging SoC integration to reduce component count, many high-power off-chip I/O signals driven across the signal pins and PCBs are eliminated. As described previously in [3], the BlueGene ASICs were built with an optimized ASIC design flow incorporating guided placement and bitstacking, but no custom circuit work.

3. POWER/PERFORMANCE EFFICIENCY

Power/performance efficiency was a prime design constraint to arrive at a high computing density system that would fit in the form factor of air-cooled racks in a standard machine room. Here, we analyze the power/performance efficiency of the final BlueGene/L system and compare design choices in the design of systems, and how they influence power efficiency. While peak numbers are an eye-catching metric, delivered power/performance on applications is the relevant metric. Thus, our analysis is based on a detailed analysis of workloads to understand how well the BlueGene/L system delivered on its promise of power/performance efficiency.

Power is a critical parameter as the densities that we are aiming for are more than a factor of 10 beyond where we could go with nodes based on traditional uni-processors. In addition, there are serious cost and reliability issues associated with high power density designs.

Several metrics have been proposed for characterizing energy efficiency. The most common of these metrics is MIPS / Watt. This metric corresponds to energy per operation, i.e., it does not assume that there is any benefit in speeding up computation, or cost for reducing its speed. Gonzalez and Horowitz argue for the use of energy-delay product as a metric, which corresponds to paying 1% energy for an increase of 1% in performance [7, 6]. Martin *et al.* propose the $energy \times delay^2$ product as an efficiency metric for VLSI computation [9, 10, 11]. This metric is considered to be superior to other metrics such as energy or energy-delay because it is reflects a "better" design point regardless of voltage. In contrast, under the energy-delay metric, design optimality changes under the assumption of scaling to a different voltage. From another perspective, when voltage scaling is an option, the highest cost which can be justified is 2% energy for 1% performance – if the cost becomes higher than this, voltage scaling is always more profitable.[1]

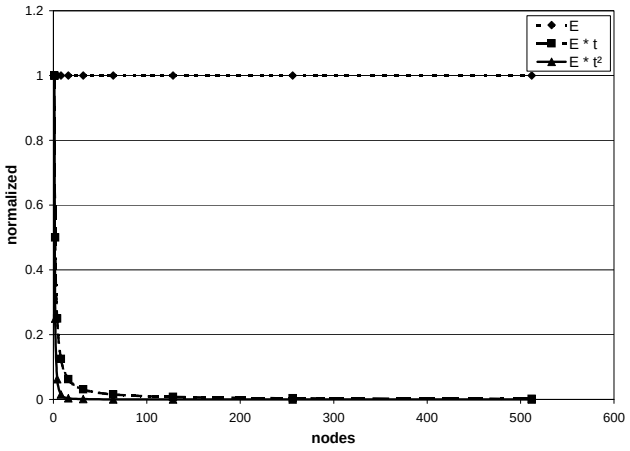

Figure 6: PEAK power/performance scaling across a range of BlueGene/L partition sizes

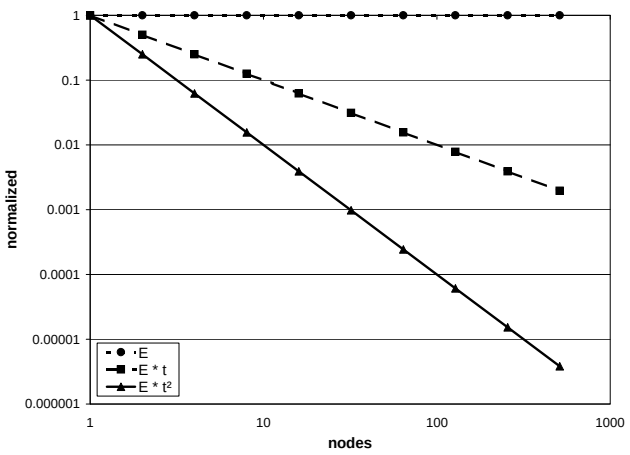

Figure 7: PEAK power/performance scaling across a range of BlueGene/L partition sizes using log scale

Based on the observation that for many large-scale scientific problems, multi-processor scaling gives much better return on hardware resources than scaling a single processor, it is advantageous to address such problem classes with a system-level approach. Large scale scientific problems typically offer multi-processor efficiency by exploiting thread-level parallelism in the 60+% percent range, far exceeding the improvements to be achieved by a microprocessor-centric optimization approach.

Based on the Top 500 submissions for BlueGene/L, LINPACK shows about 75% efficiency (Rmax/Rpeak of DD2

[1]The energy, energy-delay and $energy \times delay^2$ metrics are closely related to the MIPS^n / W ratings, where $energy = 1 / (MIPS/W)$, $energy \times delay = 1 / (MIPS^2/W)$ and $energy \times delay^2 = 1 / (MIPS^3 / W)$.

hardware). While LINPACK efficiency may seem overly optimistic for actual applications, we will show below that applications also exhibit significant scaling efficiency.

Figure 6 shows the normalized energy, energy-delay and $energy \times delay^2$ metrics for achievable peak FLOPS for a range of BlueGene/L configurations. The scaling of this peak metric is closely tracked by the reported LINPACK benchmark results.

In this and the following charts, each curve has been self-normalized to allow all three metrics to be represented in a single figure. In keeping with the interpretation of this number, a smaller energy-delay product is better, representing either less energy at the same performance, or more performance at the same energy, or both.

As can be expected from a peak benchmark, the energy per operation (curve labeled E) remains constant across all configuration, as performance per processor and power per processor remain unaffected by the increasing number of nodes. Introducing parallelism reduces execution time, without *ideally* increasing the power consumption per node, thus keeping energy consumption for a problem constant with dropping execution time. This is reflected by the improvement of of the peak performance energy-delay curve (labeled E × t).

The third metric E × t^2 puts more emphasis on performance than the E and E × t metrics, thus favoring speedup via parallelism at a constant energy budget even more. The $energy \times delay^2$ curve (labeled $E \times t^2$) reflects a constant energy-delay metric under the assumption of voltage scaling – i.e., this metric remains constant as a system is voltage scaled to higher or lower performance. This metric is useful in considering tradeoffs between higher-frequency, higher voltage design points, and more power efficient lower frequency lower power cores. Thus, according to this metric a 100 node system offers a nominal four orders of magnitude better power/performance efficiency than voltage scaling a single core. Evidently, voltage scaling cannot cover such a range, but relative figures on the curve offer insights into tradeoffs in system design. For example, the peak performance of a 128 node system could also be obtained by a voltage scaled 100 node system at a loss of power/performance efficiency of $et2_{128}/et2_{100}$, where $et2_i$ indicates the value of the E × t^2 metric for a system with i nodes.

While we have discussed the use of these metrics for peak performance, these observations will be most useful when applied to actual benchmark performance and power data to evaluate the power/performance efficiency of a massively parallel system such as BlueGene/L.

4. APPLICATION RESULTS

While many large problems can be arbitrarily parallelized to allow the problem to match the size of the system on which computation is performed – such as LINPACK – many applications are fixed size problems requiring constant amount of computing independently of the size of the system. This is referred to as strong scaling. Strong scaling problems give a more conservative performance evaluation, as they characterize what can be gained from a parallel system on many real problems. Application performance results are also more realistic in that they include multiprocessor overhead, such as communication overhead, synchronization, program sections which cannot be parallelized, etc.

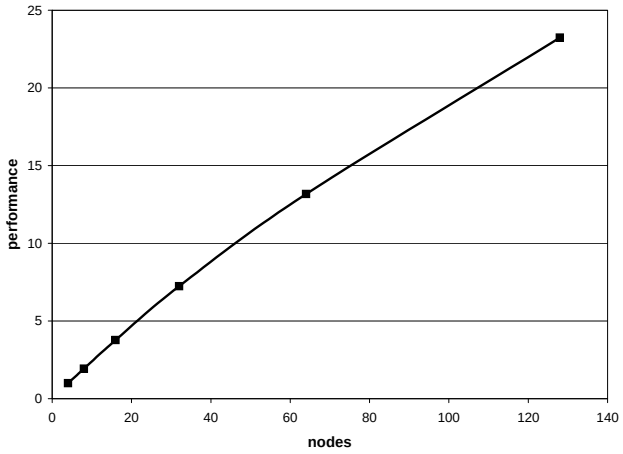

Figure 8: NAMD performance scaling across a range of BlueGene/L partition sizes

As a representative of typical life-science applications which are an important application area for the BlueGene/L project, we use NAMD, a molecular dynamic simulation system [8], for the remainder of this discussion as an exemplary case study. The NAMD code is available as open source. To compare NAMD performance across a wide range of parallel systems, the NAMD distribution includes a benchmark problem – also referred to as ApoA1, for apoprotein A1 – which serves as the basis of the results reported here. This benchmark for NAMD models one high density lipoprotein particle (apoprotein A1) found in the bloodstream. The setup consists of the apoprotein A1 molecule solvated in water and has a fixed problem size of 92224 atoms of lipid, protein and water calculated in 500 steps.

Figure 8 shows the scaling of the NAMD molecular dynamic code [8] on the BlueGene/L system and plots the normalized performance (steps per time unit) against the number of nodes. System performance scales well across a range of configurations with the increased number of nodes. Detailed analysis with a number of installed system shows extremely advantageous scaling behavior of the BlueGene/L system. The increase in number of processors translates directly into increased system performance with only a small impact of multiprocessor overhead. Another characteristic of good scaling is the absence of any sudden performance degradation as the number of nodes is increased.

Figure 9 analyzes energy and energy-delay metrics for a range of BlueGene/L partition sizes. Similar to the previous peak power/performance analysis, we show all three curves self-normalized. Based on the scaling behavior of NAMD shown in figure 8, the overall energy consumption shows an increase as the problem scales to a bigger system. Compared to figure 6, this shows the cost of multiprocessor overhead due to non-parallel program sections, communication and synchronization overhead, and so forth, compared to the ideal scaling of peak performance with node count.

At the same time, the energy-delay metric shows a significant improvement based on the overall performance gain and shows an order of magnitude improvement when scal-

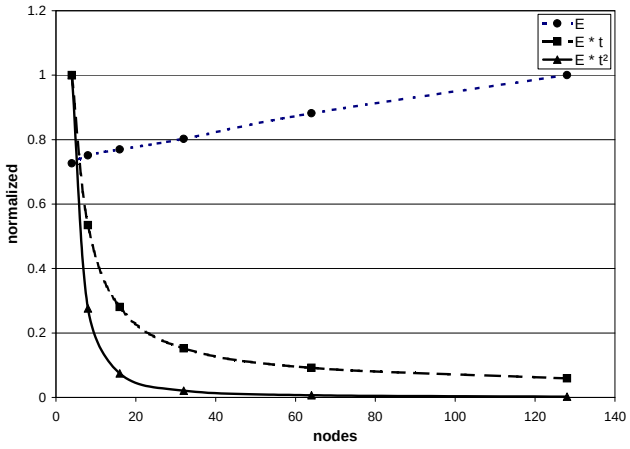

Figure 9: NAMD application power/performance scaling across a range of BlueGene/L partition sizes

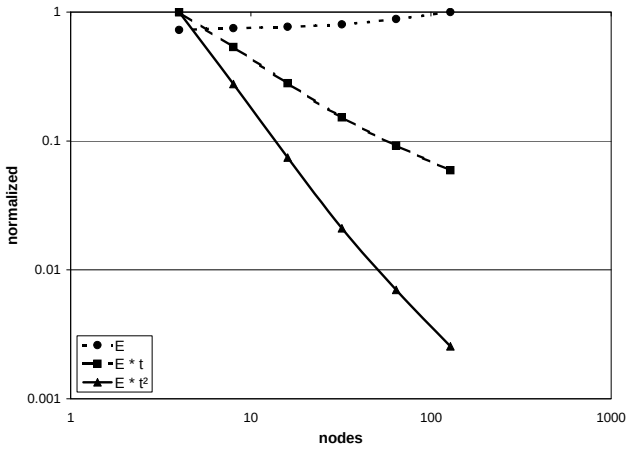

Figure 10: NAMD application power/performance scaling across a range of BlueGene/L partition sizes using log scale

ing from a 4 node system to a 128 node configuration (see figure 10).

The power/performance efficiency advantage is even more pronounced for the *energy* $\times$ *delay*^{2} metric showing up to three orders of magnitude efficiency gain of a 128 node BlueGene/L system over the base 4 node configuration.

The energy-delay product for the NAMD application improves by 2 orders of magnitude as the BlueGene/L partition size is increased, representing significant application performance scaling with only a moderate increase in energy. Unsurprisingly, with its greater emphasis on application performance, the *energy* $\times$ *delay*^{2} product shows even higher improvement as the system size is increased.

Put another way, according to this metric, parallel NAMD execution on a 128 node BlueGene/L system is nearly three orders of magnitude more efficient than voltage and frequency scaling of the microprocessor in a 4 node configuration. While ranging across 25x performance differential is not possible with voltage scaling, the curve shows possible tradeoff points. The impact of choosing a higher performance microprocessor in a node can be determined from $E \times t^2$ curve. Exploiting the metric invariance under voltage scaling, we can choose a configuration with n nodes and voltage scale it until it reaches the performance of a configuration with m nodes under the idealized assumption that all system components can be voltage scaled. While the curve does not tell us *how much* we would have to scale, we can determine the outcome in terms of loss in power/performance efficiency for compute-intensive problems with the ratio $et2_n/et2_m$, where $et2_i$ indicates the value of the $E \times t^2$ metric for a system with i nodes. Evidently, a BlueGene/L system consists of computation, communication, memory, storage, and I/O components which will all have distinct power and performance characteristics in response to voltage scaling. Thus, programs which derive their performance and power characteristics from these other subsystems need to be analyzed in accordance with the scaling rules for those domains.

While voltage scaling may not allow to span the performance differential between any two configurations of n and m nodes, other tools are at the microarchitect's disposal. However, many of these techniques offer even worse % energy for % performance tradeoffs than voltage scaling.

A study on efficiency of pursuing a microprocessor-centric approach to achieve performance post-dates the BlueGene effort – reported by Bose *et al.* [2] - and confirms this decision. Bose *et al.* study the efficiency of microarchitecture changes and in particular increasing pipeline depth to achieve higher clock frequency. The results point to a very limited potential for power/performance efficiency improvement with modestly deep pipelines, and significant power/performance efficiency degradation beyond that point.

While concentrating on microprocessor performance is not the central optimization point, microprocessor performance should not be neglected. A variety of processor design choices can be made which allow to generate higher-performing code. Examples of such optimizations are making available more registers to hide memory latency, and to exploit data parallelism when available.

To this effect, the BlueGene/L system implements the "Double Hummer" dual floating point unit, a SIMD architecture offering four parallel double precision operations per issued instructions (each SIMD instruction can issue a dual merged multiply-add operation). By tuning code to achieve better blocking factors by exploiting the ability to store 64 double precision values in architected floating point registers, and exploiting parallelism, efficiency can be improved with only a modest cost in power and area. Again, parallelism (in the form of data parallelism) offers high leverage for power performance optimization (as can be seen from the small area dedicated to the FP units in the floorplan).

5. BLUEGENE SYSTEM SOFTWARE

As can be expected from a system the scale of BlueGene/L, the software stack poses a set of interesting challenges. The basic programming model for BlueGene/L is the MPI message passing interface between nodes. Two configuration are possible for this model, as each node contains two processors, allowing for running modes such as having each processor handle its own communication ("virtual node

mode") and a mode where one processor is dedicated to communication and one to computation ("communication coprocessor mode").

Each compute node has a minimalist kernel that can handle all functions necessary for high performance real time execution. The kernel supports single user single program operation, running at most two threads simultaneously. The kernel provides an interface to the hardware for interrupts, timers, and error handling which are executed with supervisor privilege. To allow for fast communication and synchronization during execution of an application, access to the memory-mapped torus network interface is mapped into the user address space. Thus, MPI messages are passed to other nodes without incurring the cost of a context switch from user to supervisor mode.

The BlueGene/L computer node kernel does not implement a paging system to support virtual memory, reflecting the large number of nodes and threads provided in the system. Given the fact that nodes do not have private disk or other secondary storage devices, paging would be required over the I/O networks which would be prohibitive in a system of this size.

Instead, all threads use the same address space, mapping PowerPC effective addresses (virtual addresses) directly to real (physical) addresses. The TLB is statically allocated at system startup and implements a flat 256MB effective to real address translation. Similarly, software threads map directly to hardware threads.

To allow multiple users to use the BlueGene/L system concurrently, partitioning of the system is implemented by reprogramming the link chips. Within each BlueGene/L partition, the operating system supports single user running single program application. This approach protects machine resources – such as memory or communication channels – from accidental corruption so they can be used reliably for error detection, debugging, and performance monitoring [1].

Ensuring reliable operation is another system function. In a large system of such as BlueGene/L with up to 65536 nodes operating on computations for extended time periods, node failures are to be expected. Even with low MTTF rates, the compounding effect of large system image and long run times will make node failures a reality to be dealt with. Instead of expensive hardware recovery mechanism, reliability is a system function achieved by the system software layer through the implementation of a checkpointing system.

External access to a BlueGene/L system occurs via the I/O nodes which provide an offload engine for I/O and interface traffic. I/O nodes do not participate in the the torus network, and in the MPI communication protocol. Instead, they run a standard Linux operating system kernel with appropriate service extensions to communicate with the compute nodes.

A host computer is required for compiling, diagnostics, and result analysis The host computer is also responsible for file system input/output and program loading, which is accomplished via message passing. The choice of host will depend on the class of applications and their bandwidth and performance requirements.

Since the processor at the core of the BlueGene/L system is the industry standard PowerPC architecture, the familiar compiler and tool infrastructure available for the PowerPC family can be used to program the BlueGene/L system. The XL compiler family has also been extended to support generating code which exploits the high performance dual floating point SIMD unit available in each core.

System bringup and testing is performed with the BGL ADE (BlueGene/L Advanced Diagnostic Environment), an operating system which was designed expressly with the purpose to exploit and access all of BlueGene/L's capabilities [5]. The BGL ADE system can deconfigure portions of a chip so as not to trigger hardware components which are suspected of being defective, and allow isolated testing of all system components.

6. CONCLUSIONS

The BlueGene/L system leverages parallelism to achieve high performance under power-constrained conditions. In BlueGene/L systems, we exploit data- *and* thread-level parallelism with a massively parallel system using a data-parallel floating point unit as its compute engine.

We have given an overview of the BlueGene/L architecture, and analyzed performance and power/performance characteristics of the BlueGene/L system under a variety of conditions. LINPACK efficiency tracks peak performance at approximately 75% efficency across a wide range of configurations, as demonstrated by submitted LINPACK benchmark results.

To derive actual application performance we have analyzed the scaling of the NAMD molecular dynamic package on a BlueGene/L system. We have also analyzed performance and power/performance characteristics using energy and energy-delay metrics. For the voltage-scaling invariant energy $\times$ delay^2 metric, we show that exploiting thread-level application scaling with lower power cores offers significantly better power/performance characteristics than using higher frequency cores with high power consumption.

7. ACKNOWLEDGMENTS

This work has benefited from the cooperation of many individuals in IBM Research (Yorktown Heights, NY), IBM Engineering & Technology Services (Rochester, MN), and IBM Microelectronics (Burlington, VT and Raleigh, NC).

The Blue Gene/L project has been supported and partially funded by the Lawrence Livermore National Laboratories on behalf of the United States Department of Energy, under Lawrence Livermore National Laboratories Subcontract No. B517552.

NAMD was developed by the Theoretical and Computational Biophysics Group in the Beckman Institute for Advanced Science and Technology at the University of Illinois at Urbana-Champaign.

8. REFERENCES

[1] G. Almasi, C. Cascaval, J. Castanos, D. Lieber, and J. Moreira. Developing system software for Blue Gene. Technical report, IBM TJ Watson Research Center, 2001.

[2] P. Bose, D. Brooks, P. Emma, M. Gschwind, V. Srinivasan, P. Strenski, and V. Zyuban. Integrated analysis of power and performance for pipelined microprocessors. IBM Research Report RC22913, IBM

TJ Watson Research Center, Yorktown Heights, NY, April 2003.

[3] A. Bright, M. Ellavsky, A. Gara, R. Haring, G. Kopcsay, R. Lembach, J. Marcella, M. Ohmacht, and V. Salapura. Creating the BlueGene/L supercomputer from low power SoC ASICs. In *Internationcal Solid State Circuits Conference*. IEEE, February 2005.

[4] A. Gara et al. An overview of the BlueGene/L system architecture. *IBM Journal of Research and Development*, 49(2), 2005.

[5] M. Giampapa, R. Bellofatto, M. Blumrich, D. Chen, A. Gara, P. Heidelberger, D. Hoenicke, G. Kopcsay, B. Nathanson, B. Steinmacher-Burow, M. Ohmacht, V. Salapura, and P. Vranas. BlueGene/L advanced diagnostics environment. *IBM Journal of Research and Development*, 49(2), 2005.

[6] R. Gonzalez, B. Gordon, and M. Horowitz. Supply and threshold voltage scaling for low power CMOS. *IEEE Journal of Solid State Circuits*, 32(8):1210–1216, August 1997.

[7] R. Gonzalez and M. Horowitz. Energy dissipation in general purpose microprocessors. *IEEE Journal of Solid State Circuits*, 31(9):1277–1284, September 1996.

[8] L. Kalé et al. NAMD2: Greater scalability for parallel molecular dynamics. *Journal of Computational Physics*, 151:283–312, 1999.

[9] A. Martin. Towards an energy complexity of computation. *Information Processing Letters*, 77:181–187, 2001.

[10] A. Martin, M. Nyström, and P. Pénzes. ET^2: a metric for time and energy efficiency of computation. In R. Melhem and R. Graybill, editors, *Power-Aware Computing*. Kluwer Academic Publishers, 2001.

[11] P. Pénzes and A. Martin. Energy-delay efficiency of VLSI computations. In *Proc. of the 12th Grand Lakes Symposium on VLSI*, pages 104–107, April 2002.