Modeling and Evaluating Energy-Performance Efficiency of Parallel Processing on Multicore Based Power Aware Systems

Rong Ge
Marquette University
rong.ge@marquett.edu

Xizhou Feng
Virginia Tech
fengx@vt.edu

Kirk W. Cameron
Virginia Tech
kcameron@cs.vt.edu

Abstract

In energy efficient high end computing, a typical problem is to find an energy-performance efficient resource allocation for computing a given workload. An analytical solution to this problem includes two steps: first estimating the performances and energy costs for the workload running with various resource allocations, and second searching the allocation space to identify the optimal allocation according to an energy-performance efficiency measure. In this paper, we develop analytical models to approximate performance and energy cost for scientific workloads on multicore based power aware systems. The performance models extend Amdahl's law and power-aware speedup model to the context of multicore-based power aware computing. The power and energy models describe the power effects of resource allocation and workload characteristics. As a proof of concept, we show model parameter derivation and model validation using performance, power, and energy profiles collected on a prototype multicore based power aware cluster.

1 Introduction

Arguably speaking, we are currently entering an era of green computing in which the primary design constraints for high-end computer architectures and systems are shifting from highest peak performance to maximum performance-energy efficiency. This dramatic paradigm shift, driven by mixed forces from technical, economical, environmental, and political aspects, is evidenced by the ever-increasing number of green initiatives and green solutions from both industry and academia [1, 2, 9, 10, 23].

In high end computing, improvement of energy efficiency has been pursued from various levels and approaches, including system architecture innovation, code optimization, dynamic power management, and resource scheduling. While an ideal "green" solution would be a holistic approach that exploits power reduction and energy conservation at all levels, resource scheduling is a key factor affecting the extent to which energy-performance efficiency can be improved. This is because applications and workloads are different and unique; each workload requires a tailored resource allocation for its execution for the maximum energy performance efficiency.

In this paper, we focus our discussion on resource scheduling for parallel scientific workload on multicore-based power aware clusters. More specifically, we investigate how to identify an optimal system configuration for running a given parallel workload on state-of-art high-end systems. Here, a system configuration is denoted as a tuple (n, c, f), where n is the total number of allocated cores, c is the number of allocated cores per compute node, and f is the processor frequency. Intuitively, an analytical solution includes two steps: (1) determine the performances (i.e., execution times) and energy costs for computing a workload W with all possible configurations of varying (n, c, f); and (2) locate such a configuration that delivers highest energy-performance in the configuration space. While various heuristic or exhaustive search algorithms can address the second step, the first step requires accurate yet practical models to describe the relationship among performance, energy, configuration, and workload characteristics.

Speedup models are normally used to describe the performance gain of parallel processing for parallel workloads. Amdahl's law is a simple but insightful analytical model to predict the performance of a parallel workload with a given system size (i.e., the number of processors). In Amdahl's law, a central concept is a workload consists of a parallelizable portion and a serial portion; performance gain of parallel processing is limited by the serial portion. Recently, Amdahl's law has been extended to investigate the performance and energy scaling of

978-1-4244-3750-4/09/$25.00 ©2009 IEEE

multicore architectures [17, 21] and DVFS based power aware systems [6, 14]. Power aware speedup [14] is one of the derived performance models for power aware systems.

In this paper, we apply the rationale of Amdahl’s law and power aware speedup to develop analytical models for approximating the performance and energy cost of parallel scientific workload on a multicore based power aware architecture. The work presented in this paper includes the following contributions.

First, we extend the Amdahl's law and power aware speedup [14] for modeling the performance and energy of parallel workload on multicore-based power aware clusters.

Second, we present experimental results on the performance and energy profiles of two NPB benchmarks on a state-of-art multicore based power aware cluster and use these results in model parameterization and validation;

Third, by combing the analytical model and experimental results, we present several interesting observations about energy-performance efficient resource scheduling on high-end computing systems.

The rest of this paper is organized as follows. Section 2 presents our modeling methodology. Then as a proof of concepts and model validation, we show two case studies of how to apply our analytical models to real world workload. In these case studies, we also present power, energy, and performance profiles on a real system to support our model analysis. Section 4 overviews existing researches related to our work, followed by a concluding section to summarize the work presented in this paper.

2 Methodology and analytical models

2.1 System abstraction

To simplify our discussion, in this paper, we use an abstract parallel architecture and system shown in Figure 1. This architecture captures the major features of the state-of-art multicore-based power aware systems. In this architecture, a system is a cluster of compute nodes connected by an interconnection network; each compute node consists of a group of processor cores that have their own individual caches but share a common memory space; each core supports dynamic voltage and frequency scaling (DVFS) and can schedule its frequency individually without affecting another.

We model a system of this architecture with three parameters: N, the total number of available computing cores; C, the total number of cores per compute node; and F, the set of available processor frequencies. We make the following simplifications for such systems. First, we do not differentiate a node with a single multicore processor, a node with multiple unicore processors, or an SMP node with multiple multicore processors. Second, cores on the same node communicate through on-node shared memory and cores on different nodes communicate through the interconnection network. Finally, the power managements of processor cores are independent.

As mentioned earlier, to run a specific workload W on a system in the above architecture, we may choose a configuration (n, c, f) from the configuration space $N \times C \times F$. For simplicity, we limit our discussion to static configuration where each parameter is fixed during the entire workload execution.

2.2 Workload Modeling

Workload modeling is necessary for performance and power modeling. A workload model must be balanced between details for accuracy and abstraction for practical use. For the purpose to identifying the optimal energy-performance efficient configuration on a power aware system, the derived workload model needs to capture the power and performance effects of: parallel processing, frequency scaling, and process-core mapping.

We have proposed a power-aware speedup model for predicting the performance on power aware systems [14]. Power aware speedup model presents a baseline methodology to model the workload such that the first two requirements can be approximately met. By skipping the detailed derivation, in this paper we use a simplified form of the workload model used by the power-aware speedup model.

In this model, the total workload is normalized as 1 and broken into several fractions to reflect the performance effects of parallel processing and frequency scaling. First, the workload is decomposed to a parallel portion (*p*) and a serial portion (*1-p*). The parallel portion (*p*) of the workload can be concurrently computed using a number of cores to reduce its execution time, while serial portion (*1-p*) cannot. Second, the serial portion (*1-p*) is further broken into a processor frequency dependent or on-chip access portion $(1-p) * \alpha_s$ and a processor frequency independent or off-chip access portion $(1-p) * (1-\alpha_s)$. The on-chip access portion benefits from faster processor to speed up its execution while the off-chip access portion doesn’t. Similarly, the parallel portion (*p*) is further broken into a processor frequency dependent or on-chip access part $p * \alpha_p$ and a processor frequency independent or off-chip access part $p * (1-\alpha_p)$. Note here the ratio of on-chip access

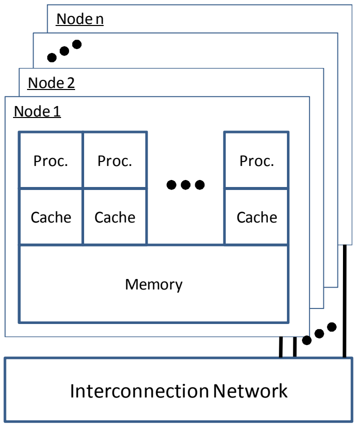


Fig. 1. Generic power aware cluster architecture

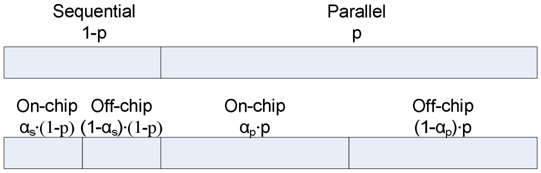


Fig. 2. A simplified view of the workload decomposition. This view does not show the parallel overhead.

to off-chip access is different between serial and parallel workloads. Figure 2 illustrates this workload decomposition approach.

Capturing the effects of process-core mapping is nontrivial. There are two major forces affecting performance: memory contention and communication cost. Using more cores on the same node reduces communication cost between the cores but increases memory contention. Since the focus of this paper is on energy-performance efficiency instead of memory contention and communication modeling, we use a single variable o to capture overhead of parallelization and memory contention. We note that the modeling of o is application specific and could be refined to guarantee sufficient accuracy for per workload cases.

2.3 Performance Modeling

Based on the workload model described above, we estimate the parallel execution time of workload W with configuration (n, c, f). When only the system size is scaled, the execution time t^{n,c,f_0} with configure (n, c, f_0) is:

$$t^{n,c,f_0} = 1 - p + \frac{p}{n} + o_{n,c,f_0} \qquad (1)$$

Here o_{n,c,f_0} is the resulting parallel overhead. If parallel processing using n cores and frequency scaling from f_0 to f are both applied, the execution time $t^{n,c,f}$ becomes:

$$t^{n,c,f} = (1-p)\left(1 - a_s + a_s\frac{f_0}{f}\right) + \frac{p}{n}\left(1 - a_p + a_p\frac{f_0}{f}\right) + o_{n,c,f} \qquad (2)$$

If the parallel overhead o_{n,c,f_0} and $o_{n,c,f}$ are legible respectively, Equation (1) becomes the mathematical form of Amdahl's' law and Equation (2) becomes a simplified form of the power aware speed model, which can be viewed as generalized Amdahl's law with two scaling mechanisms.

Correspondingly, we can rewrite equation (1) and (2) into speedup forms, where the speedup S with configuration (n, c, f) is the ratio of sequential execution time at base frequency f_0 to parallel execution at a higher frequency f.

$$S^{n,c,f_0} = \frac{1}{1-p+\frac{p}{n}+o_{n,c,f_0}} \qquad (3)$$

$$S^{n,c,f} = \frac{1}{(1-p)\left(1-a_s+a_s\frac{f_0}{f}\right)+\frac{p}{n}\left(1-a_p+a_p\frac{f_0}{f}\right)+o_{n,c,f}} \qquad (4)$$

2.4 Power Modeling

The power consumption of systems with multicore based power aware architecture has rarely been modeled. In this work, we strive to derive a power model that captures the power effects of workload characteristics, parallel processing with multiple nodes and cores, and processor frequency scaling. To this end we take four steps. First, we break the power of a single compute node based on its physical components. For example, a node consists of processors/cores, memory modules, disks, and other parts such as

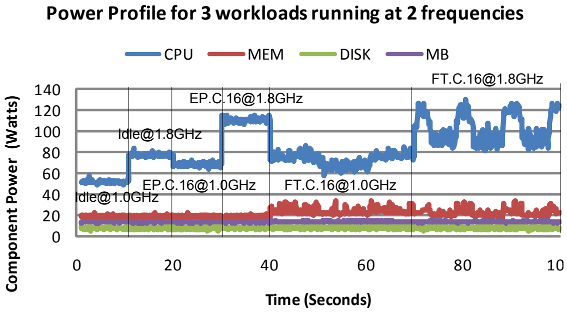


Fig. 3. A snapshot of the power profiles for three workloads running at two different frequencies. The workloads include system idle, NPB EP and FT benchmark with problem size C and running with 16 cores on 4 nodes. Power values correspond to component power of one node with processor frequency at 1.0 GHz and 1.8 GHz.

motherboard and power supply. Among them, processor, memory, and disk are the major computing components and their power consumptions vary with workloads. The power consumptions of other components do not vary noticeably with workload. For simplicity, in this work we treat processor and memory individually, and classify all other components into a single group denoted as *other*. We note that disk power can also be extracted when needed. Equation 5 reflects this break down of node power P_{node} into components.

$$P_{node} = P_{cpu} + P_{memory} + P_{other} \quad (5)$$

On a multicore system, a processor chip consists of many cores. Therefore, the processor power P_{cpu} is the sum of the core power P_{core} over the number of cores on the node.

$$P_{cpu} = \sum_{i=1,C} P_{core,i} \quad (6)$$

Second, we study how workload execution changes the power consumptions of cores and memory modules. To gain insight, we use a power profiling toolkit PowerPack [11] to reveal the variations of component power with workload on a dual-core dual processor DVFS capable cluster. Figure 3 shows the snapshot of the power profiles. We observe that power consumptions of processor and memory increase drastically when at load, and the power increments vary with workloads. This is because when at load, the components' activities increase, and so do their dynamic power consumption. To reflect the power effect of workload execution, we further break components' (processor and memory) power into idle power ($P_{core,i,idle}$ and $P_{memory,idle}$) and dynamic power ($P_{core,i,dynamic}$ and $P_{memory,dynamic}$). Here idle power quantifies the power when there is no user application running on the system, and dynamic power quantifies the additional power consumed by executing user workloads.

$$P_{core,i} = P_{core,i,idle} + P_{core,i,dynamic} \quad (7)$$

$$P_{memory} = P_{memory,idle} + P_{memory,dynamic} \quad (8)$$

At this step, we further investigate the correlations between components' dynamic power and workload characteristics. In particular, we would like to identify a small set of performance events that frequently occurs and whose power accounts for the majority of the components' power. We find that the constituent performance events of processor power on our prototype system include L1 cache access, L2 cache access, main memory access, retired operations, and retired floating-point operations; the memory power is reflected by a single performance event: memory access.

Third, we include the power impacts of parallel processing in our power model. We assume $P_{node} = 1$ when the workload executes on one core at the baseline processor frequency f_0. To estimate the power consumption using c cores, we must consider the changes of processor and memory dynamic power when the workload execution shifts from sequential mode to parallel mode. We introduce two additional variables β: the ratio of processor dynamic power $P_{core,dynamic}$ to P_{node}, and γ: the ratio of memory dynamic power $P_{memory,dynamic}$ to P_{node}, where P_{node} is the node power using one core at frequency f_0. These two parameters reflect the dynamic power of additional $(c-1)$ processors and memory modules when the workload execution shifts from sequential mode to parallel mode.

Thus, when using c cores per node at frequency f_0, we can write the node power consumption as:

$$P_{node}^{c,f_0} = 1 + (c-1)(\beta + \gamma) \quad (9)$$

Fourth, we take into account the power effect of processor frequency scaling. When a core is sped up to frequency f from f_0, both the core idle power and dynamic power increase, as observed from figure 3. We introduce parameter ρ_s to describe the scaling factor of core idle power $P_{core,idle}$ and ρ_d to describe the scaling factor of core dynamic power $P_{core,dynamic}$. Suppose the ratio of $P_{core,ilde}$ to P_{node} when using 1 core per node at frequency f_0 is μ, then the node power with all the c cores running at frequency f becomes:

$$P_{node}^{n,c,f} = 1 + (c-1)(\beta + \gamma) + c\mu(\rho_s - 1) + c\beta(\rho_d - 1) \quad (10)$$

Note $\rho_s = \rho_d = 1$ for frequency f_0. On the right hand side the equation, the items are the node power when one of cores is executing a workload at baseline frequency; the power due to parallel computing using extra $(c-1)$ cores, the additional core idle power due to processor frequency scaling, and the additional core dynamic power due to processor frequency scaling.

2.5 Energy Modeling

Once we have the performance and power models, the derivation of energy models become straightforward. Here we first start with the definition of energy as shown below:

$$e^{n,c,f} = \int_0^{t^{n,c,f}} \sum_{i=1}^{n/c} p_{node,i}^{n,c,f}(\tau) d\,\tau \quad (11)$$

Where $e^{n,c,f}$ is the total energy consumed to exeucte workload using n cores on n/c nodes, each core running at frequency f. $p_{node,i}^{n,c,f}$ is the power consumption of node i, and $t^{n,c,f}$ is the total execution time. The total energy consumption is the integral of the sum of the power on all requested compute nodes over the entire execution time.

We further assume the entire workload execution is broken into the five regions as described by the power aware speedup model: I: sequential frequency dependent execution; II: sequential frequency indepenent execution; III: parallel frequency depenent execution; IV: parallel frequency independent region; and V: communication regions. According to the workload modeling and performance modeling, we denote the execution times for these five regions as:

$$t_I = a_s(1-p)\frac{f_0}{f} \quad (12)$$

$$t_{II} = (1-a_s)(1-p) \quad (13)$$

$$t_{III} = a_p\frac{p}{n}\frac{f_0}{f} \quad (14)$$

$$t_{IV} = (1-a_{p)}\frac{p}{n} \quad (15)$$

$$t_V = o_{n,c,f} \quad (16)$$

Consequently, we can rewrite Equation (11) as

$$e^{n,c,f} = \sum_{j=I..V} p_j t_j \quad (17)$$

$$p_j = \sum_{i=1}^{n/c} p_{node,i}^{n,c,f} \quad where\ t_{j-1} \le t \le t_j \quad (18)$$

For each of the above five regions, its power consumption p_j can be determined based on the equations presented in the power modeling section and the actual execution scenario. For regions I and II, only one core can compute the sequential workload. For frequency independent regions II, IV, and V, we may schedule core frequency to save power and energy. Further, when not all cores are used in computation, we could schedule the idle cores to lowest frequency to save power and energy.

3 Case studies

In this section, we show how to apply the above analytical models for energy-performance efficiency analysis. To support and validate the analysis, we also provide energy-performance profiles on an experimental power aware cluster called DORI. DORI consists of one head node and eight compute nodes. Each node is a two-way dual AMD Opteron 265 SMP system with 6GB SDRAM memory modules. The dual-core Opteron processor is DVFS-enabled suppoting five processor frequenceis: 1.0GHz, 1.2GHz, 1.4GHz, 1.6GHz, and 1.8GHz.

We profile performance and power consumption using an enhanced version of the PowerPack toolkit. PowerPack combines direct power management and software processing to profile power, and energy, and performance at component level. It also automatically correlates power profile with source code through code instrumentations and timestamp mappings.

3.1 Energy-performance analysis of EP benchmark

NAS EP (Embarrasingly Parallel) benchamrk [5] represents a class of scientific applications that is ideally suited for parallel processing. EP workload can be summarized as follows: (1) the parallelizable fraction of the workload is close to 1, i.e., $p = 1$; (2) there is very little parallel overhead, i.e., $o = 0$; (3) the on-chip memory access ratio is constant across a wide range configurations. Thus, give a configuration (n, c, f), we have the following performance equations:

$$t^{n,c,f} = \frac{1}{n}(1 - a_p + a_p\frac{f_0}{f}) \quad (19)$$

$$S^{n,c,f} = \frac{n}{1-a_p+a_p\frac{f_0}{f}} \quad (20)$$

These two equations indicate the execution time of EP decreases linearly while the speedup increases linearly with the number of computing cores. By assuming static configuration during the execution, we have the node power:

$$p_{node}^{n,c,f} = 1 + (c-1)(\beta+\gamma) + c\mu(\rho_s - 1) + c\beta(\rho_d - 1) \quad (21)$$

The power consumption per node with *c* cores for EP is constant relative to the total number of computing cores. Considering that the execution time of EP decreases lineary with the number of cores, its total energy consumption would be independent of the number of computing nodes. This energy is:

TABLE I. Comparisons of measured and projected energy and speedup for EP.C benchmark running at 16 cores on 4 nodes with varying processor frequencies.

f (GHz)	Meas. Speedup	Pred. Speedup	Meas. Energy	Pred energy
1.0	16.00	16.00	0.27	0.27
1.2	19.00	19.20	0.22	0.24
1.4	22.13	22.40	0.22	0.22
1.6	25.35	25.60	0.21	0.20
1.8	28.63	28.80	0.21	0.19

TABLE II. Comparisons of measured and projected energy and speedup for EP.B running at 1.8 GHz using varying total core counts (n) and core counts per node (c)

n	Measured Perf.		Predicted Perf.		Measured Energy		Predicted Energy	
	c=1	c=4	c=1	c=4	c=1	c=4	c=1	c=4
1	1.8		1.8		0.66		0.66	
2	3.6		3.6		0.66		0.66	
4	7.2	7.2	7.2	7.2	0.66	0.20	0.66	0.19
8	14.1	14.3	14.4	14.4	0.66	0.20	0.66	0.19
16		28.7		28.8		0.20		0.19

$$e^{n,c,f} = \frac{n}{c} t^{n,c,f} p^{n,c,f} = \frac{1}{c} (1 - a_p + a_p \frac{f_0}{f})\, p_{node}^{n,c,f} \quad (22)$$

The constant total energy consumption implies using more cores and nodes for executing EP results in better performance with same energy consumption.

All parameters shown in (19)-(22) can be derived through experimental measurements with linear regression. We have measured a set of performance and power data points with selected configurations, and calculated the parameter values based on the parameter definitions and performance model. To validate the models and parameterization, we compared the model predicted performance and energy for EP benchmark against the actual measurements with various system configuration in (n, c, f). Table I and II show that the model predictions match the measurements very well.

Using the derived parameters, we have projected the total energy across the nodes and performance scaling of EP, as plotted in Figure 4. From this figure, we observe the following properties of energy and performance of EP benchmark on multicore-based power aware system.

First, when c (the number of used cores per node) and f (the processor frequency of the cores) are fixed, the total energy consumption is constant relative to the number of nodes. This means for embarrassingly parallel workloads, we can always add more symmetric computing nodes to speed up EP execution without consuming additional energy.

Second, using more cores per node can result in significant energy savings. For instance, a configuration using four cores per node only consumes 30% energy as a configuration using one core per node. This is explained by the fact that doubling the core counts will double the performance, but not double the node power since core power only accounts for a small fraction of the node power.

Third, increasing processor frequency from 1.0 GHz to 1.8 GHz not only leads to larger speedups but also results in energy savings. For instance, with 4 cores, the normalized energy consumptions are 0.2 and 0.3 if running at 1.8 GHz and 1.0 GHz respectively. Increasing frequency from 1.0 GHz to 1.8 GHz could lead to 33% energy saving.

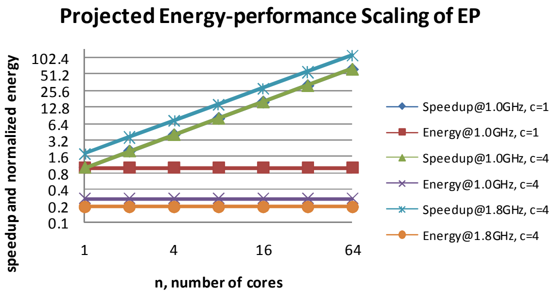


Fig. 4. The projected energy and performance of EP on the DORI Cluster

3.2 Energy-performance analysis of FT benchmark

To analyze FT benchmark, we can use the same procedures used in EP benchmark. However, the multiple execution phases exhibited in FT benchmark makes it more challenging for accurate prediction. Parallel FT workload consists of alternating computation phases and communication intensive phases, as shown in Figure 3. The computation phase is similar as EP, while the communication phases are essentially parallel overhead and more difficult to predict. The communication cost is a function of the total number of cores and core counts per node, and processor frequency. In particular, it is impacted by both inter-node data movement and intra-node data movement.

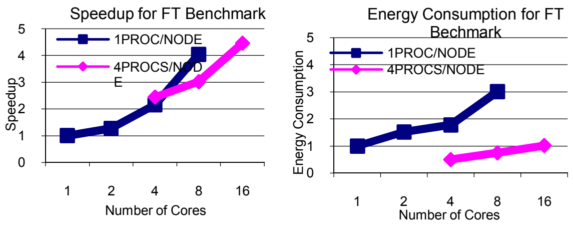
For power modeling, we can follow the same procedure used in EP to determine the power scaling parameters for FT for both computation and communication phases. Due to space constraints, we skip the tedious model analysis and focus our discussion on experimental results.

Figure 5 shows the measured energy and performance under different n and c with fixed frequency on DORI cluster. This figure demonstrates the following efficiency trends:

First, as the total number of cores n increases, the average node power is almost constant, while the performance gains of using more nodes diminishes due to the parallel overhead. As a result, running FT with more nodes will require significant additional energy. In our analysis, we find that the parallel overhead ranges from 0.22 to 0.12 when using 2~8 nodes (assume the total sequential execution time is 1).

Second, allocating more cores on a same node achieves higher energy efficiency, compared to allocating one core per node. This is because using more cores on a same node multiples the dynamic power of CPU cores, and this dynamic power only accounts for a small faction of the total power. Thus, the more cores involved in computing FT, the smaller share of the total power overhead for each active core.

Third, with a fixed n, allocating more computing cores on a single node might result in worse performance than allocating one core per node, as shown by data points with $n = 4$ in Figure 5a. While using more cores per node reduces inter-node



(a) (b)

Fig. 5. The measured energy and performance scaling of FT with core counts. The frequency is fixed at 1.8 GHz. The values are normalized against the one at (1, 1, 1.0 GHz).

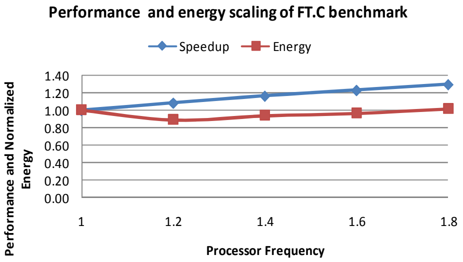


Fig. 6. The measured energy and performance scaling with frequency of FT Benchmark. The core counts are fixed at 16 cores on 4 nodes. All values are normalized against the one at (16, 4, 1.0 GHz).

communications, it might cause memory contention that degrades performance.

In Figure 6, we show how performance and energy scale with processor frequency when running on 16 cores. This figure indicates a higher processor frequency is prefered in terms of overall energy-performance efficiency. This observation is a little beyond our expectation since it is different from our previous findings [15]. We argue that it is the different system architectures and communication subsystems used in those two experiments that accounts for such differences.

4 Related work

In this paper, we discsuss approaches to modeling the performance and energy of parallel sceientific workload on muticore based power aware systems. There are two areas closely related to this work.

The first related area is parallel speedup and scalability modeling. Since the introduction of Amdahl's law [3], this research area has been extensively investigated during the past two decades. For example, fixed-time speedup was proposed by Gustafson et al [16] to reflect the intention of solving larger problems on larger systems; memory-bound speedup was proposed by Sun and Ni [24] to reflect the performance impact of available memory space. As a further extension, Grama et al introduced isoefficiency metric to study how to scale workload to maintain parallel computing efficiency [4]. While these models are previously focused on multicomputer based parallel systems, today's research revisits Amdhal's law for how to organize the cores on a state-of-art multicore and many-core chip. Recently, Hill et al applied Amdahl's law to compare three multicore chip designs and observed that the performance benefits of dynamic multicore chip design [17].

While research in this related area provides fruitful and extensive insights for modeling parallel performance, they provide little information about power and energy-performance efficiency, a critical concern in today's high performance computing and data center operations.

Modeling, profiling, and optimzing the energy-performance efficiency of parallel processing in HPC is the other area closely related to this work. Several groups have studied the power and energy profiling for parallel scientific workload on conventional clusters and power aware clusters [11, 15, 20]. Findings of those work exposed the problems and needs of optimizing energy-performance efficiency in parallel processing. There are various efforts in developing analytical models as a guide of energy-efficient high end computing [6, 7, 14]. One of such models is the power aware speedup model proposed by Ge and Cameron [14], which is intended to provide a general form of parallel speedup model that supports the emerging power aware architecture. Recently, Cho et al proposed a corollary to Amdahl's law for energy on parallel systems, and used such corollary to study the interaction between parallelism and energy consumption [6]. In addition to efforts in profiling and modeling, there are efforts in optimizing energy-performance efficiency using low power high performnace computing [13, 22, 25], DVFS-based power management [12, 15, 18], and accelerator-based high performance computing [8, 19].

Though built upon many of the previous works,we justify that our work presented in this paper, is significantly different from existing research. First, the models presented in this paper are comprehensive including those of performance, system power, and total energy consumption. Second, they are well-balanced between mathematical assessment and practical usage. These models are founded on experimental observations of parallel processing on real systems. Not only all model parameters can be experimentally determined and validated, these models can also be directly applied to analyze real workload on real systems, either for predication or for validation. Second, the models presented here target the merging multicore based power aware systems. To our best knowledge, this is a new topic that have not been exploited in previous work.

5 Conclusion and summary

In summary, we have developed a set of analytical models for investigating the energy-performance efficiency of parallel workload on emerging multicore based power aware systems. These models extend Amdahl's law and power aware speedup to take energy into consideration. The models include parameters to describe the interactions between workload, perfomance, power, and energy. They can be put into practical use besides studying the performance and energy bounds of prallel processing. In this paper, we also presented many first-hand experimental results on a state-of-art power aware cluster. These results illustrate several trends of energy-performance scaling on multicore based power aware systems.

In the future, we plan to polish the presented models, automate the model parameter derivation procedures, and integrate these models with existing resource planning and scheduling packages. With the advent of petascale systems, we envision scheduling the most appropriate configuration for computing tasks will become one of the important ways for maintaining and optimzing energy performance efficiency of large scale parallel processing.

Reference

1. *The Green Grid*. 2008, website: http://www.thegreengrid.org/.
2. U. Agency, *Report to Congress on Server and Data Center Energy Efficiency: Public Law 109-431*. Public Law. **109**: p. 431.
3. G.M. Amdahl. *Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities*. in AFIPS Spring Joint Computer Conference. 1967. Reston, VA.
4. Y.G. Ananth, G. Anshul, and K. Vipin, *Isoefficiency: Measuring the Scalability of Parallel Algorithms and Architectures.* IEEE Parallel and Distributed Technology: System and Technology, 1993. **1**(3): p. 12-21.
5. D. Bailey, T. Harris, W. Saphir, R.v.d. Wijngaart, et al., *The NAS parallel benchmarks 2.0*, in *Technical report, NASA Ames Research Center Technical Report #NAS95020*. 1995.
6. S. Cho and R. Melhem, *Corollaries to Amdahl's Law for Energy*. Computer Architecture Letters, 2008. **7**(1): p. 25-28.
7. Y. Ding, K. Malkowski, P. Raghavan, and M. Kandemir. *Towards energy efficient scaling of scientific codes*. in IEEE International Symposium on Parallel and Distributed Processing. 2008.
8. Z. Fan, F. Qiu, A. Kaufman, and S. Yoakum-Stover. *GPU Cluster for High Performance Computing*. 2004: IEEE Computer Society Washington, DC, USA.
9. W. Feng and K. Cameron, *The Green500 List: Encouraging Sustainable Supercomputing.* Computer, 2007. **40**(12): p. 50-55.
10. W. Feng, X. Feng, and R. Ge, *Green Supercomputing Comes of Age.* IEEE IT Professional, 2008. **10**(1): p. 17-23.
11. X. Feng, R. Ge, and K.W. Cameron. *Power and Energy Profiling of Scientific Applications on Distributed Systems (IPDPS 05)*. in 19th IEEE International Parallel and Distributed Processing Symposium. 2005. Denver, CO.
12. V. Freeh and D. Lowenthal. *Using multiple energy gears in MPI programs on a power-scalable cluster*. in 10th ACM Symposium on Principles and Practice of Parallel Programming (PPoPP). 2005: ACM New York, NY, USA.
13. A. Gara, M. Blumrich, D. Chen, G. Chiu, et al., *Overview of the Blue Gene/L system architecture.* IBM Journal of Research and Development, 2005. **49**(2): p. 195-212.
14. R. Ge and K.W. Cameron, *Power-Aware Speedup*, in *IEEE International Parallel & Distributed Processing Symposium (IPDPS) 2007*. 2007: Long Beach, CA.
15. R. Ge, X. Feng, and K. Cameron. *Performance-constrained Distributed DVS Scheduling for Scientific Applications on Power-aware Clusters*. in Proceedings of the ACM/IEEE Supercomputing 2005 (SC'05). 2005.
16. J.L. Gustafson, *Reevaluating Amdahl's law.* Communications of the ACM, 1988. **31**(5): p. 532-533.
17. M. Hill and M. Marty, *Amdahl's Law in the Multicore Era.* Computer, 2008. **41**(7): p. 33-38.
18. C. Hsu and W. Feng. *A power-aware run-time system for high-performance computing*. in Proceedings of the ACM/IEEE Supercomputing 2005 (SC'05). 2005.
19. J. Kahle, M. Day, H. Hofstee, C. Johns, et al., *Introduction to the Cell multiprocessor.* IBM Journal of Research and Development, 2005. **49**(4/5): p. 589.
20. S. Kamil, J. Shalf, and E. Strohmaier. *Power efficiency in high performance computing*. in IEEE International Symposium on Parallel and Distributed Processing. 2008.
21. G. Loh. *The Cost of Uncore in Throughput-Oriented Many-Core Processors*. in Workshop on Architectures and Languages for Throughput Applications in conjuction with ISCA-35. 2008. Beijing, China.
22. H. Nakashima, H. Nakamura, M. Sato, T. Boku, et al. *MegaProto: 1 TFlops/10kW Rack Is Feasible Even with Only Commodity Technology*. in Proceedings of the ACM/IEEE SC 2005 Conference. 2005.
23. SPEC, *The SPEC Power Benchmark*. 2008, website: http://www.spec.org/power_ssj2008/.
24. X.-H. Sun and L. Ni, *Scalable problems and memory-bounded speedup.* Journal of Parallel and Distributed Computing, 1993. **19**: p. 27-37.
25. M.S. Warren, E.H. Weigle, and W.-C. Feng. *High-Density Computing: A 240-Processor Beowulf in One Cubic Meter*. in IEEE/ACM SC2002 Conference. 2002. Baltimore, Maryland.