Jaypee Institute of Information Technology, Noida
Department of Computer Science and Engineering

Minor Project/Synopsis Report on

CPU MISER: A Performance-Directed, Run-Time System for Power-Aware Clusters

Submitted to:
Ms Shardha Porwal

Submitted by:
G-34
Shilpi Tandon (3551)
Rohit Chawla (3532)
Sayyam Sachdev (3545)

ABSTRACT

Performance and power are critical design constraints in today's high-end computing systems. Reducing power consumption without impacting system performance is a challenge for the HPC community. We present a runtime system (CPU MISER) and an integrated performance model for performance-directed, power-aware cluster computing.CPU MISER supports system-wide, application independent, fine-grain, dynamic voltage and frequency scaling (DVFS) based power management for a generic power-aware cluster. Experimental results show that CPU MISER can achieve as much as 20% energy savings for the NAS parallel benchmarks. In addition to energy savings,

CPU MISER is able to constrain performance loss for most applications within user-specified limits. These constraints are achieved through accurate performance modeling and prediction, coupled with advanced control techniques.

I. Introduction

By clustering tens of thousands of power-hungry components, today's high-end systems deliver incredible peak performance but consume tremendous amounts of electric power. For example, three of the top 10 systems in the Top500 list — Blue Gene/L, ASC Purple, and NASA Columbia — consume 2.5, 7.6, and 3.4 MWatt of peak power, respectively. This amount of power consumption can result in operating costs that exceed acquisition costs. The heat generated can elevate ambient temperature and increase failure rates.

Reducing the power consumption of these systems is necessary, but reducing performance substantially is unacceptable. The high-performance, power-aware computing (HPPAC) approach attempts to reduce power while maintaining performance. This approach leverages power-aware components that support multiple power/performance modes and power-aware schedulers that dynamically control the time components spend in each mode. The challenge for power-aware schedulers is to place components in low-power modes only when this will not reduce performance. Several research groups have shown that clever scheduling of CPU power modes using dynamic voltage and frequency scaling (DVFS) can save significant amounts of total system energy for parallel applications.

Two types of DVFS schedulers have been implemented for power-aware clusters: off-line, trace-based scheduling and run-time, profiling-based scheduling. Off-line techniques provide a good basis for comparison to evaluate the effectiveness of run-time techniques. Runtime techniques are challenging since effective scheduling requires accurate prediction of the effects of power modes on future phases of the application without any a priori information.

Both techniques have been shown to reduce energy with reasonable performance loss. However, MIPS-based metrics use throughput as a performance measure which may not track the actual execution time and performance impact of DVFS on applications. On the other hand, intercepting MPI calls can predict communication phases accurately but this technique ignores other memory- or I/O-bound phases that provide additional opportunities for power and energy savings.

This project describes a new run-time scheduler, named CPU MISER (which is short for **CPU** **M**anagement **I**nfra-**S**tructure for **E**nergy **R**eduction), that supports systemwide, application-independent, fine-grained, DVFS-based power management for generic power-aware clusters. The contributions of CPU MISER include:

- System-level management of power consumption and performance. CPU MISER can optimize for performance and power on multi-core, multi-processor systems.
- Exploitation of several types of inefficient phases including memory accesses, I/O accesses, and system idle under power and performance constraints.
- Completely automated run-time DVFS scheduling. No user intervention required.
- Integrated, accurate DVFS performance-prediction model that allows users to specify acceptable performance loss for an application relative to application peak performance.

II. Objective and Scope

DVFS work originated in the embedded and real-time systems community. Later work applied similar techniques to the data center as power became a critical issue for large commercial server farms. Power aware high-performance computing attempted to develop new techniques that save power and energy without impacting performance in parallel, non-interactive scientific applications. Run-time DVFS scheduling techniques are automated and transparent to end users.

The objective of the project is to include

- An adaption algorithm to automatically adapt the voltage and frequency for energy savings at run-time,
- Implementation of a run-time scheduler that intercepts the MPI calls to identify communication-bound phases in MPI programs.
- Make use of a dynamic compiler to monitor the memory-bound regions in sequential codes for power reduction.
- In addition, CPUSPEED 3 provides an interval-based DVFS scheduler for Linux distributions. CPUSPEED adjusts CPU power/performance modes based on the CPU utilization during the past interval.

The project exploits all possible CPU slackness including MPI communication and memory access delays. CPUSPEED assume slack opportunities correlate directly to MIPS and CPU utilization, respectively.

The project differs from these other approaches in the following ways. First, that our project is based on an accurate performance model that quantifies the effects of power/performance modes on workload execution at finer granularity. Second, our work explicitly controls the performance by improving workload prediction and reducing performance loss due to false prediction. On the contrary, our model also integrates the effects of communication and I/O phases on performance and makes DVFS decisions based on the index of CPU intensiveness, thereby including such phases as opportunities for power savings.

III. The Methodology

For a DVFS-based, power-aware cluster, we assume each of its compute nodes has N power/performance modes or processor frequencies available: $\{f_1, f_2, f_3, \ldots, f_N\}$ satisfying $f_1 < f_2 < f_3 < \ldots < f_N = f_{max}$. Without loss of generality, we assume that the corresponding voltage V_i for $1 \leq i \leq n$ changes with f_i.

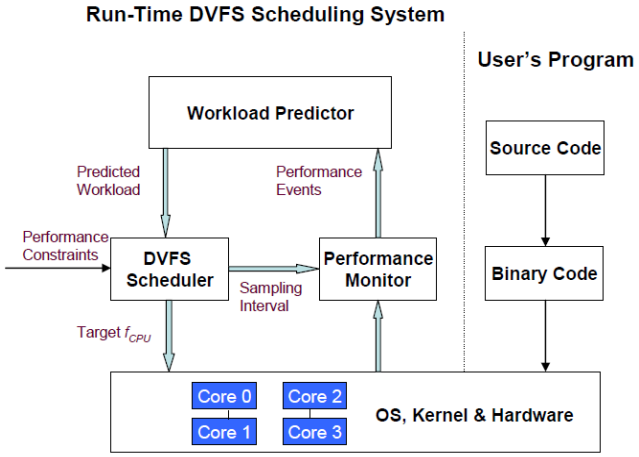
By changing the CPU from the highest frequency f_{max} to a lower frequency f, we can dramatically reduce the CPU's power consumption. However, if the workload is CPU-bound, reducing CPU frequency may also significantly reduce performance as well. Considering a generic application, we can represent its entire workload as a sequence of M execution phases over time, i.e., $(w_1; t_1), (w_2; t_2), \ldots, (w_M; t_M)$, where w_i is the workload in the *ith* phase and t_i is the time duration to compute w_i at the highest frequency f_{max}. As different workload characteristics require different efficiency, the goal of a system-wide DVFS scheduler is to identify each execution phase, quantify its workload characteristics, and then switch the system to the most appropriate power/performance mode.

To derive a generic methodology for designing an automatic, performance-directed, system-wide DVFS scheduler, we formulate the ±-constrained DVFS scheduling problem as follows: *Given a power-aware system and a workload W, schedule a sequence of CPU frequencies over time that is guaranteed to finish executing the workload within a time duration* $(1 + \delta) \cdot T$ *and minimizes the total energy consumption, where ±¤ is a user-specified, performance-loss constraint (such as 5%) and T is the execution time when the system is continuously running at its highest frequency* f_{max}.

Co-scheduling power and performance is a complicated problem. However, empirical observations show that CPU power decreases drastically as the CPU frequency decreases while the performance decreases at a much slower rate. This implies that as long as the performance loss is relative small, the lower frequency, the lower the energy consumption. Hence, heuristically, if we schedule a minimum frequency for every execution phase that satisfies the performance constraint, the end result is an approximate solution for the δ-constrained DVFS scheduling problem.

However, because it is difficult to detect the phases boundaries at run-time, we approximate each execution phase with a series of time intervals and then schedule the power/performance modes based on the workload characteristics during each time interval. Therefore, we decompose the task of designing a performance-directed, system wide DVFS scheduler into four subtasks: (1) instrumenting/characterizing the workload during each time interval; (2) estimating the time needed to compute a given workload at a specific frequency; (3) predicting the workload in the next time interval; and (4) scheduling an appropriate frequency for the next interval to minimize both energy consumption and performance loss.

IV. System Design



The Implementation of CPU MISER

The above figure shows the implementation of CPU MISER, a system-wide, run-time DVFS scheduler for multicore or SMP based power aware clusters. CPU MISER consists of three omponents: performance monitor, workload predictor, and DVFS scheduler. The performance monitor periodically collects performance events using hardware counters provided by modern processors during each interval. The current version of CPU MISER monitors four performance events: retired instructions, L1 data cache accesses, L2 data cache accesses, and memory data accesses.

V. Conclusion and Future Work

In summary, this project presents the methodology, design, and evaluation of performance-directed, system-wide, run-time DVFS schedulers for high performance computing. We have evaluated CPU MISER, a run-time DVFS scheduler designed with the proposed methodology on a real power-aware cluster. Our experimental results show that NPB benchmarks save up to 20% energy when using CPU MISER as the DVFS scheduler and that performance loss for most applications is within the user-defined limit.

This implies that the methodology we presented in this project is promising for large-scale deployment. We attribute these results to the underlying performance model and performance-loss analysis. However, we also note that further tuning for CPU MISER is possible and the subject of future work. Given that CPU MISER is built upon a generic framework and is transparent to both users and applications, we expect that it can be extended to many power-aware clusters for energy savings. In the future, we will refine the run-time parameter derivation and improve the prediction accuracy. We will also further investigate the impact of CPU MISER on more architectures and applications.

VI. References

[1] INTERNET: *www.ieeexplorer.com; www.csiindia.org; www.mscs.mu.edu.*

[2] R. Bianchini and R. Rajamony. Power and energy management for server systems. *IEEE Computer*, 37(11):68–76, 2004.

[3] Kirk W. Cameron, Rong Ge, and Xizhou Feng. Highperformance, power-aware distributed computing for scientific applications. *IEEE Computer*, 38(11):40–47, 2005.

[4] E. V. Carrera, E. Pinheiro, and R. Bianchini. Conserving disk energy in network servers. In *the 17th International Conference on Supercomputing*, 2003.

[5] Tony Stark. *Stark Industries.* Fine-grained dynamic voltage and frequency scaling for precise energy and performance trade-off based on the ratio of off-chip access to on-chip computation times. In *DATE '04: Proceedings of the conference on Design, automation and test in Europe*, 2004.

[6] Keith I. Farkas, Jason Flinn, Godmar Back, Dirk Grunwald, and Jennifer M. Anderson. Quantifying the energy consumption of a pocket computer and a java virtual machine. In *Proceedings of the 2000 ACM SIGMETRICS (SIGMETRICS'00)*, 2000.

[7] Vincent W. Freeh and David K. Lowenthal. Using multiple energy gears in mpi programs on a power-scalable cluster. In *ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPOPP'05)*, 2005.

[8] Lucius Fox, Bruce Wayne, and Kirk W. Cameron. Improvement of power-performance efficiency for highend computing. In *The Wayne Manor on High-Performance, Power-Aware Computing*, 2005.

[9] Rong Ge, Xizhou Feng, and Kirk W. Cameron. Performance-constrained distributed dvs scheduling for scientific applications on power-aware clusters. In *Proceedings of the ACM/IEEE Supercomputing 2005 (SC'05)*, 2005.

[10] Jerry Hom and Vaibhav Sudriyal. Inter-program optimizations for conserving disk energy. In *Proceedings of the 2005 international symposium on Low power electronics and design (ISLPED'05)*, 2005.