Power-Aware Parallel Job Scheduling

Maja Etinski
maja.etinski@bsc.es

Julita Corbalan
julita.corbalan@bsc.es

Jesus Labarta
jesus.labarta@bsc.es

Mateo Valero
mateo.valero@bsc.es

Barcelona Supercomputing Center
Jordi Girona 31, 08034 Barcelona, Spain

Abstract

Recent increase in performance of High Performance Computing (HPC) centers has been followed by even higher increase in power consumption. Power draw of modern supercomputers is not only an economic problem but it has negative consequences on environment. Roughly speaking, CPU power presents 50% of total system power. Dynamic Voltage Frequency Scaling(DVFS) is a technique widely used to manage CPU power. The level of parallel job scheduling presents a good place for power management as the scheduler is aware of the whole system: current load, running jobs, waiting jobs and their wait times. This talk explains two power-aware parallel job scheduling policies that trade performance for energy trying to minimize the performance penalty. The first policy assigns job frequency based on predicted job performance while the other uses system utilization to decide when to run jobs at reduced frequency. In the end, a power budgeting policy will be described since power budgeting has become very important for reasons such as existing infrastructure limitations, reliability and/or carbon footprint. Interestingly, it shows that the DVFS technique can even improve overall job performance in case of a given power budget.

1. Introduction

In an HPC center power reduction techniques may be motivated by operating costs, system reliability and environmental concerns. One might want to decrease power dissipation accepting certain performance penalty in return. But there are other reasons for power management in HPC environments, for instance a power constraint might be imposed by existing power/cooling facilities. In such a situation the main goal is to maximize performance for a given power budget.

Processor power consumption presents significant portion of total system power. Though the portion is system and load dependent, roughly speaking it makes a half of the total system power when the systems is under load ([4]). DVFS (Dynamic Voltage Frequency Scaling) technique is commonly used to manage CPU power. A DVFS-enabled processor supports a set of frequency-voltage pairs i.e. gears. Running a processor at lower frequency/voltage results in lower power/energy consumption. Lower frequency normally increases application execution time thus frequency scaling should be applied carefully, especially in HPC systems as their main goal is still performance.

Since power consumption of HPC systems has been an issue over last decade, some power reduction approaches have been proposed. They can be divided into two main groups depending on the granularity they deal with. The first group targets power/energy consumption of HPC applications. Several runtime systems that apply DVFS in order to reduce energy consumed per an application have been implemented ([5, 8, 6]). These systems have been designed to exploit certain application characteristics like load imbalance of MPI applications or communication-intensive intervals. Therefore, they can be applied only to certain applications/jobs. The second group of works deals with system/CPU power management of whole workload instead of only one application. Powering down some nodes leads to energy saving but it has very significant effect on job performance ([7]). Linux governors use DVFS to reduce CPU power without taking into account entire system load. For instance *onDemand* applies frequency scaling based on single core utilization. In an HPC environment, DVFS should be used taking into account the entire system load.

User satisfaction in an HPC center does not depend only on job execution time but on its wait time as well. Job scheduler has a complete view of the whole HPC system:

it is aware of running jobs and current load; jobs in the wait queue and their wait times; available resources. Therefore, a job scheduler can estimate job performance loss due to frequency scaling. A job scheduler implements a job scheduling policy and in conjunction with resource selection policy it manages system resources at job level. Since power has appeared as an important resource, it is natural to enable job schedulers to manage power. Work presented in this talk deals with power-aware parallel job scheduling. The talk explains two energy saving policies ([1, 3]) and one power budgeting policy ([2]). All these policies have been integrated into the widespread EASY backfilling parallel job scheduling policy.

2 Energy Saving Policies

The utilization-driven energy saving policy assigns CPU frequency to each job at its start time depending on system utilization. In order to avoid negative effect on wait time of other jobs, frequency is scaled down only when system utilization is not very high. The scheduler uses three CPU frequencies for different system utilizations. One more parameter, the $WQthreshold$ threshold, is introduced to prevent scheduler from using reduced frequencies when there are more than $WQthreshold$ jobs in the wait queue.

The other energy saving policy, BSLD-driven policy, predicts job performance in BSLD metric if it would be run at frequency f. If the predicted performance is better than a given threshold $BSLDthreshold$, the job can be executed at frequency f. Such the lowest available frequency will be selected. This policy as well checks the wait queue length before applying any scaling.

The energy saving policies have user specified parameters to enable control over energy- performance trade-off. Five workload logs from production use have been used in policy evaluation. It has been observed that for highly loaded workloads it is difficult to obtain any energy savings without affecting performance significantly. Less loaded workloads are more suitable for energy- performance trade-off.

3 Power Budgeting Policy

In a power-constrained systems, the scheduler is not limited only by the available number of processors but by available power as well. Running jobs at reduced frequencies allows for more jobs to run simultaneously if there are enough processors.

The PB-guided policy assigns frequencies similarly to the BSLD-driven energy saving policy. The job is run at the lowest frequency at which its predicted performance is better than a given threshold. This time the threshold changes dynamically depending on current power draw. The closer to the power budget, the higher the threshold. The higher the threshold, the more aggressive frequency scaling is applied.

The policy achieves better job performance for all tested workload logs compared to the baseline case without DVFS and with the same power constrained. Better performance is achieved because of shorter wait times as more jobs can run at same time due to their lower power consumption at reduced frequencies.

4 Conclusions

After the evaluation of three power-aware policies, it can be concluded that the real potential of DVFS is in its application to power-constrained systems. In power-constrained systems DVFS leads to a reduction in average job wait time. Since overall job performance is determined by wait time and runtime, this decrease in average wait time due to frequency scaling results in better overall job performance.

References

[1] M. Etinski, J. Corbalan, J. Labarta, and M. Valero. Bsld threshold driven power management policy for hpc centers. In *Parallel and Distributed Processing Symposium, Workshops and PhD Forum 2010 Proceedings. IEEE International*, pages 1–8, Atlanta, GA, April 2010.

[2] M. Etinski, J. Corbalan, J. Labarta, and M. Valero. Optimizing job performance under a given power constraint in hpc centers. In *Green Computing Conference, 2010 International*, pages 257–267, Chicago, IL, August 2010.

[3] M. Etinski, J. Corbalan, J. Labarta, and M. Valero. Utilization driven power-aware parallel job scheduling. *Computer Science - Research and Development, Springer*, 25/2010:207–216, August 2010.

[4] R. Ge, X. Feng, S. Song, H.-C. Chang, D. Li, and K. W. Cameron. Powerpack: Energy profiling and analysis of high-performance systems and applications. *IEEE Trans. Parallel Distrib. Syst.*, 21:658–671, May 2010.

[5] C. hsing Hsu and W. chun Feng. A power-aware run-time system for high-performance computing. *sc*, 0:1, 2005.

[6] N. Kappiah, V. W. Freeh, and D. K. Lowenthal. Just in time dynamic voltage scaling: Exploiting inter-node slack to save energy in mpi programs. *sc*, 0:33, 2005.

[7] B. Lawson and E. Smirni. Power-aware resource allocation in high-end systems via online simulation. In *ICS '05: Proceedings of the 19th annual international conference on Supercomputing*, pages 229–238, New York, NY, USA, 2005. ACM.

[8] M. Y. Lim, V. W. Freeh, and D. K. Lowenthal. Adaptive, transparent frequency and voltage scaling of communication phases in mpi programs. *sc*, 0:14, 2006.