Evaluating DVFS and Concurrency Throttling on IBM's Power8 Architecture

Wei Wang
University of Delaware
Email: weiwang@udel.edu

Edgar A. León
Lawrence Livermore National Laboratory
Email: leon@llnl.gov

***Abstract*—Two of the world's next-generation of supercomputers at the U.S. national laboratories will be based on IBM's Power architecture. Early insights of how this architecture impacts applications are beneficial to attain the best performance. In this work, we investigate the effects of DVFS and thread concurrency throttling on the performance of applications for an IBM OpenPower, Power8 system. We apply these techniques dynamically on a per region basis to three HPC codes: miniFE, LULESH, and Graph500. Our empirical results offer the following insights. First, concurrency throttling provides significant performance improvements. Second, 4-way simultaneous multi-threading (SMT) performs as well as 8-way SMT with potential gains in power-efficiency. And, third, applying informed frequency and concurrency throttling combinations on memory-bound regions results in greater performance and energy efficiency.**

I. Introduction

The goal of exascale computing continues to drive innovations in all aspects of high performance computing (HPC). IBM's new Power8 processor is designed for big data, analytics, and cloud environments [1]. Previous evaluations of this processor [2] show its potential for scientific applications. However, more explorations and insights are needed to best leverage this system.

IBM's Power8 architecture features 8-way simultaneous multi-threading (SMT-8) and per-core DVFS control with 69 distinct frequencies. In this work, we characterize application performance using different DVFS and concurrency settings. A key aspect of our analysis is the application of these features at the granularity of loops rather than the entire application.

Our contributions are as follows. We evaluated how DVFS affects application performance on the Power8 architecture and how to improve energy consumption without performance degradation; we also evaluated concurrency throttling showing significant performance improvements; and, finally, SMT-4 performs as well as SMT-8 for several applications.

II. Methodology

We measure application performance under various thread concurrencies and DVFS frequencies. We also compared application performance using SMT-8 and SMT-4 configurations with the same number of threads. We tested three OpenMP codes: Graph500, LULESH 2.0, and miniFE. The granularity of our measurements is per OpenMP parallel loop. Such fine-grain analysis provides more insights than just measuring the entire application. By analyzing the results from running applications with different frequency and concurrency settings, we empirically identify memory-bound regions that could benefit from dynamic runtime frequency change. We invoke energy control API calls before and after such code regions to achieve the best performance and energy consumption.

Prepared by LLNL under Contract DE-AC52-07NA27344. LLNL-ABS-676202.

III. Results

The experiments were performed on a TYAN GN70-BP010 Power8 system. It has 4 processor cores and supports up to 32 hardware threads on the SMT-8 configuration.

A. OpenMP Thread Concurrency Throttling

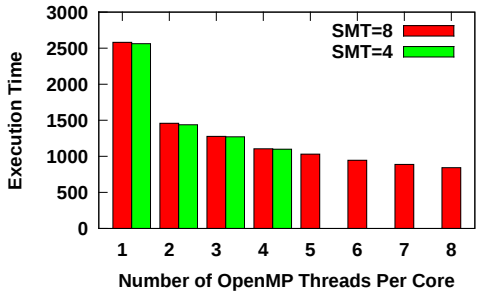
Figures 1a and 1b show the execution times when changing the number of OpenMP threads per core for Graph500 and LULESH. Applications do not always achieve the best performance with 8 threads per core. For example, LULESH (and miniFE; not shown due to space constraints) performs best with 4 threads per core. Graph500, on the other hand, is able to leverage all 8 threads per core. Comparing SMT-8 and SMT-4, SMT-4 performans similarly to SMT-8. SMT-4 can potentially lead to better energy efficiency for applications that are memory-intense like LULESH and miniFE.

B. DVFS

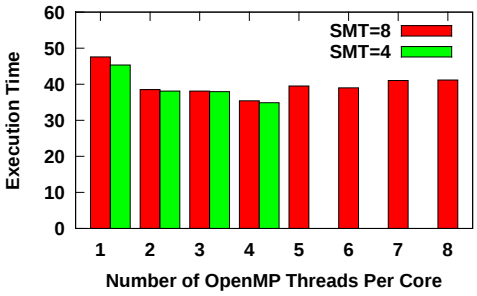
Figures 2 and 3 show the execution times when changing the DVFS frequency for Graph500 and miniFE. A lower frequency leads to longer execution time for both benchmarks. However, applications slow down quite differently when reducing the frequency. Memory bound applications like miniFE provide energy saving opportunities for DVFS. Running at 2.061GHz, Graph500 slows down by 1.6× while miniFE only slows down by 1.2×.

C. Combining DVFS and Concurrency Throttling

miniFE contains a dominant loop that is memory-bound. The loop (Loop1 in Figures 4 and 5) is insensitive to reducing the frequency and the number of threads. The same two figures show that loops in miniFE behave differently when the frequency and the number of threads are reduced. Given this observation, we executed miniFE with 16 threads and mixed frequencies. Just before entering Loop1, the DVFS



(a) Graph500 (best with 8 threads per core)



(b) LULESH (best with 4 threads per core)

Fig. 1. Comparing the application performance with varying number of threads per core and SMT settings

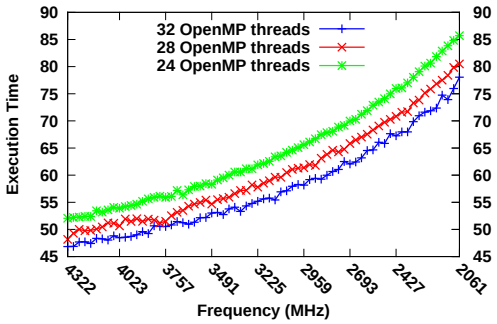


Fig. 2. Graph500: performance slowdown of more than 1.6X

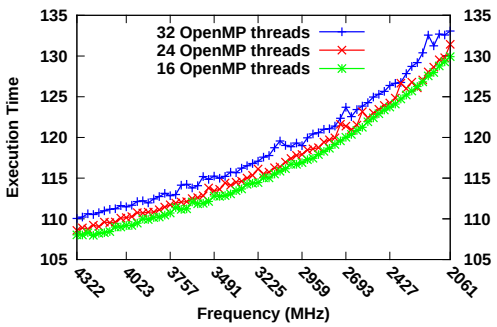


Fig. 3. miniFE: performance slowdown of only 1.2X

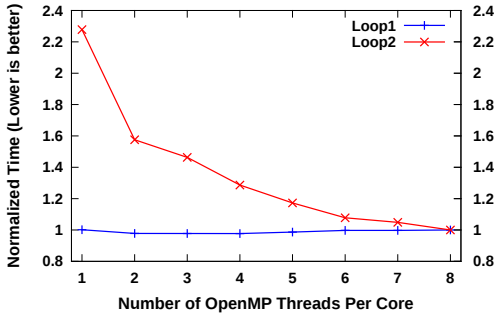


Fig. 4. miniFE: Concurrency throttling affects the two loops differently

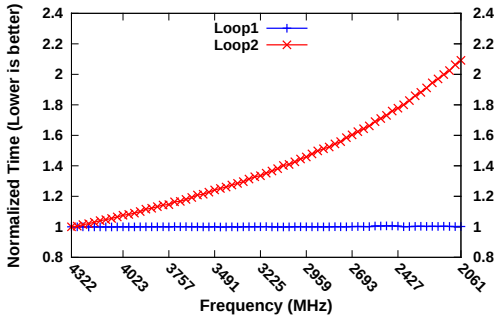


Fig. 5. miniFE: DVFS affects the two loops differently

frequency is set to the minimum possible. The frequency is reset to the maximum immediately after Loop1. We achieve slight performance improvement by combining concurrency throttling and DVFS.

IV. Conclusions

We evaluated application performance on IBM's Power8 architecture leveraging DVFS and thread concurrency throttling. Although this architecture provides eight hardware threads per core (CPUs), the optimal number of software threads may be significantly lower particularly for memory intense regions. To leverage the best performance on this highly-threaded architecture, concurrency throttling is necessary. In addition, we found that SMT-4 performs as well as SMT-8 for applications that do not require all eight hardware threads. Our fine-granularity analysis shows that memory-bound regions are fairly insensitive to frequency changes resulting in energy savings. Finally, combining DVFS and concurrency throttling on the Power8 architecture can lead to improved performance and energy consumption.

References

[1] J. Friedrich, H. Le, W. Starke, J. Stuechli, B. Sinharoy, E. Fluhr, D. Dreps, V. Zyuban, G. Still, C. Gonzalez, D. Hogenmiller, F. Malgioglio, R. Nett, R. Puri, P. Restle, D. Shan, Z. Deniz, D. Wendel, M. Ziegler, and D. Victor, "The POWER8™ processor: Designed for big data, analytics, and cloud environments," in *IEEE International Conference on IC Design Technology*, 2014.

[2] A. Adinetz, P. Baumeister, H. Böttiger, T. Hater, T. Maurer, D. Pleiter, W. Schenck, and S. Schifano, "Performance Evaluation of Scientific Applications on POWER8," in *International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems*, 2014.