Reliable Performance Auto-tuning in presence of DVFS[‡]

Md Rakib Hasan[*], Eric Van Hensbergen[†], Wade Walker[†]
Louisiana State University[*], ARM Inc.[†]

Abstract

In the era where exascale systems are imminent, maintaining a power budget is one of the most critical problem to overcome. Along with much research on balancing performance and power, Dynamic Voltage and Frequency Scaling (DVFS) is being used extensively to save idle-time CPU power consumption.

The drawback is that the inherent random behavior of DVFS makes walltime unreliable to be used as a performance metric which causes random performance from libraries (e.g. ATLAS) that rely on machine-specific auto-tuning of several characteristics to achieve the best performance.

In this poster:

1. We show that a sub-optimal selection (not the worst case) of kernel and block size during auto-tuning can cause ATLAS to lose 40%of DGEMM performance and
2. We present a more reliable performance metric in presence of DVFS that can achieve the same performance as no-DVFS.

Introduction & Motivation

Auto-tuning:

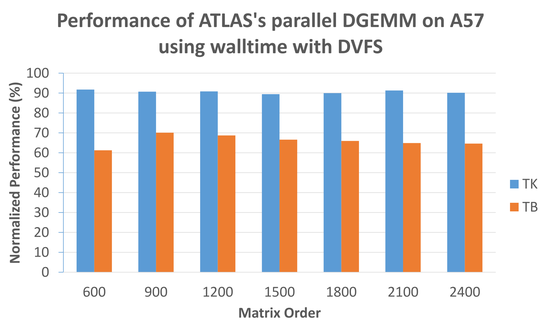
- ❑ Helps libraries (e.g. ATLAS) to build machine-specific optimized BLAS.
- ❑ Helps iterative compilers (e.g. iFKO) to find the best optimizations for the machine to apply to the compiling code.
- ❑ Can provide up to 10x speedup over generic optimizations.

Dynamic Voltage and Frequency Scaling (DVFS):

- ❑ Efficient technique to save idle-time power consumption.
- ❑ OS collaborates with the hardware based on CPU usage and temperature.
- ❑ Walltime is not usable anymore to compare performance.

Two instances of bad auto-tuning of ATLAS:

1. A sub-optimal (the second best) kernel is selected.
2. A sub-optimal block size (inefficient use of cache) is selected.



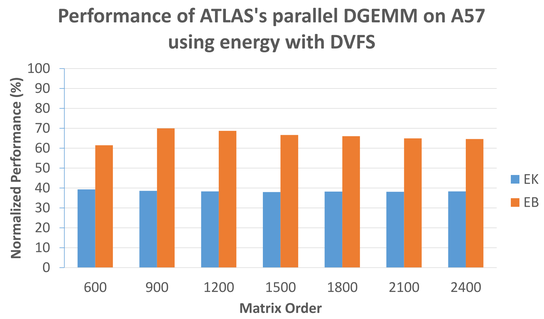
- ❑ About 9% loss for sub-optimal kernel selection.
- ❑ About 40% loss for an inefficient block size selection.

[‡]This material is based upon work supported in part by the Department of Energy and Lawrence Livermore National Security, LLC ("LLNS") under contract number DE-AC52-07NA27344 as part of the Fast Forward 2 ("FF2") program.

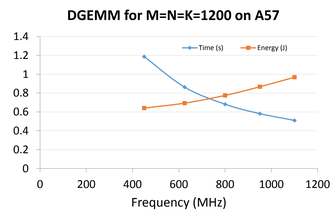
Using Energy Consumption

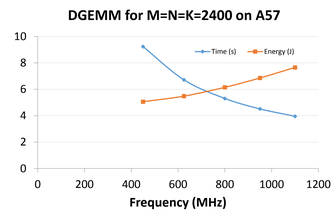
Using total energy consumption as the performance metric:

- ❑ Slower kernel will run for longer, yielding higher energy consumption.



- ❑ Performance loss is about 60% for sub-optimal kernel selection.
  - Need to analyze the relationship between energy consumption and frequency.

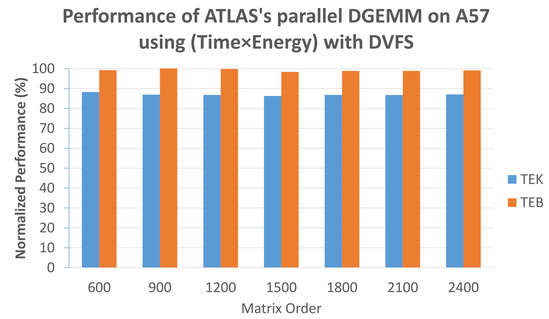




- ❑ Even at the lowest frequency, the overall energy consumption is low.
  - Causing any kernel or block size running at a lower frequency treated as faster than others running at a slightly higher frequency.
  - Need to penalize the metric using the increased walltime.

Using Time-Energy product as the performance metric:

- ❑ Penalizing the energy consumption with walltime improves the selection.
- ❑ For this research, we used a product of walltime and total energy consumption.

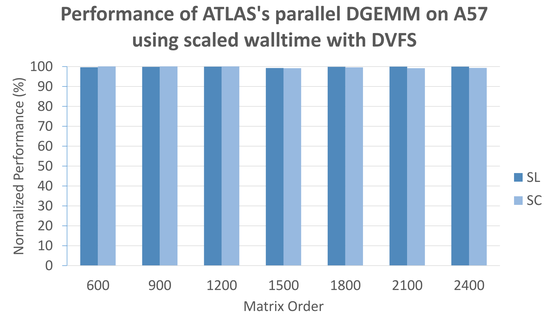


- ❑ It can fix the problem with block size selection.
- ❑ But still suffers from bad decision for the best kernel selection.

Using Power Consumption

Using scaled walltime (from average power) as the performance metric:

- ❑ Power consumption is consistent at a certain frequency.
- ❑ Our idea is to use the average power consumption to estimate the average frequency, then use the computed frequency to scale the wall time.
- ❑ Scaled walltime, $T_s = \frac{T \times f_a}{f_b}$
- ❑ Average frequency, $f_a = F(P)\ where\ P = average\ power = \frac{E}{T}$
- ❑ $F(P)$ can be empirically tuned or manually fed using system specification.
- ❑ f_b can be set to any frequency so that T_s represents the estimated walltime at f_b



- ❑ SC represents the ATLAS installation where $F(P)$ was empirically tuned and fitted to be non-linear.
- ❑ SL represents the ATLAS installation where $F(P)$ was formed to be linear using the lowest and highest frequency and corresponding power.
- ❑ Both cases were successfully able to enable proper auto-tuning and the performance loss is less than 1%.

Conclusions and Future Work

- ❑ **Conclusions:**
  1. Auto-tuning libraries or compilers can lose performance due to DVFS.
  2. We presented a reliable performance metric that enables proper auto-tuning without any loss of performance in presence of DVFS.
- ❑ **Future Work:**
  1. Explore alternative power measurement technologies (e.g. RAPL) that doesn't require built-in power monitoring systems.
  2. Explore the power-frequency relationship on systems with higher frequency and significantly higher power requirements.
  3. Explore the effect of memory DVFS along with CPU DVFS during auto-tuning and their impact in performance.

References

1. Q. Deng, D. Meisner, A. Bhattacharjee, T. Wenisch, and R. Bianchini. Coscale: Coordinating cpu and memory system dvfs in server systems. In Microarchitecture (MICRO), 2012 45th Annual IEEE/ACM International Symposium on, pages 143–154, Dec 2012.
2. A. Tiwari, A. Gamst, M. Laurenzano, M. Schulz, and L. Carrington. Modeling the impact of reduced memory bandwidth on hpc applications. In F. Silva, I. Dutra, and V. Santos Costa, editors, Euro-Par 2014 Parallel Processing, volume 8632 of Lecture Notes in Computer Science, pages 63–74. Springer International Publishing, 2014.
3. R. C. Whaley and J. Dongarra. Automatically Tuned Linear Algebra Software. In Ninth SIAM Conference on Parallel Processing for Scientific Computing, 1999. CD-ROM Proceedings.
4. R. C. Whaley and A. Petitet. Atlas homepage. http://math-atlas.sourceforge.net/, 2011.
5. R. C. Whaley and D. B. Whalley. Tuning high performance kernels through empirical compilation. In The 2005 International Conference on Parallel Processing, pages 89–98, Oslo, Norway, June 2005.