Statistical Power and Energy Modeling of multi-GPU kernels

Sayan Ghosh, Sunita Chandrasekaran and Barbara M. Chapman
sghosh@mail.uh.edu, sunita@cs.uh.edu, chapman@cs.uh.edu, Computer Science Dept., University of Houston

Introduction

In order to improve the energy efficiency of parallel applications, a better understanding of the energy behavior of various applications is mandatory. In this study we employ statistical methods to model power and energy consumption of some common optimized high performance kernels (DGEMM, FFT, PRNG and FD stencils) on a multi-GPU platform. We have also developed a library to capture the power consumption of our multi-GPU test node, to evaluate statistical predictions with actual empirical results.

Our results show, that using a limited set of GPU performance counters, it is possible to estimate power and energy in a reasonable manner. Average error between measured and predicted values of power and energy were found to be approximately 5% for power and 9% for energy, respectively.

It is difficult to measure the overall power of a node, unless one has specialized equipments. Statistical modeling techniques could be useful in approximating the power and energy consumption factor of applications with minimum instrumentation overheads and reasonable accuracy. Contrary to some of the existing studies in this area, our results indicate that it is possible to provide a power/energy estimation using a limited set of hardware counters and achieve acceptable precision, instead of considering a plethora of counters.

Linear Regression

Linear Regression is a technique in statistics to model the relationship between a dependent variable and a set of explanatory variables. This goal of the model is to predict the values of the dependent variable, according to the observed patterns from a data set.

The expected model for a dataset is represented as:

$$y = \beta 0 + \beta 1 \; X \; coefficient + e$$

where β0 is the y-intercept and β1 is the slope. Linear Regression tries to find best estimates of b0 and b1 by minimizing the error, denoted by e. In our case, the dependent variable being modeled is power/energy, and the independent variables are the counters or attributes that we have identified to have a strong correlation with power/energy. We are considering the following GPU counters, which we have collected from Nvidia CUDA Profiler:

- Instructions issued
- L2 cache read misses
- DRAM reads
- DRAM writes
- Warps launched per block per SM
- Threads launched per block per SM

To evaluate our predicted results, we have studied the residual errors; if the model is appropriate, the residuals should have a normal distribution, which we observe in our case. We used the R statistical package to run statistical tests on our data sets.

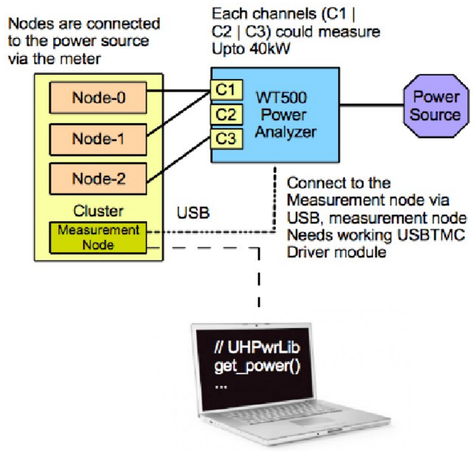
Testbed

The testbed node consists of an 2-way AMD Magny-Cours Processor - 6174 having a total of 24 physical cores. The node has 4 Tesla M2050 devices, each having 448 compute cores and 3 GB of GDDR5 memory. The processor cores are clocked at 2.2 Ghz, and the graphics unit have persistent mode on. The idle power of the node is approximately 320 Watts. CUDA 4.1 platform was used to compile GPU programs, and for host, Intel compiler 12.1 was used.

Power Measurement

We have used a Yokogawa WT500 3-phase power analyzer, here are some facts:

- Capable of measuring power for around 100 heterogeneous nodes
- Sampling frequency - 100 ms
- NIST calibrated, endorsed by SPEC
- Using UHPwrLib to extract power consumption from this device from user space



To make things easier for the user, we have written a thread-safe library called UHPwrLib, which could be used to extract power from the test node by issuing some function calls from user code (like Unix `gettimeofday()`).

Class	Kernel	Source
Dense Linear Algebra	DGEMM	CUBLAS
Spectral Methods	FFT	CUFFT
Pseudo-random Number Generation	Mersenne Twister	Mersenne Twister from CUDA SDK (modified for multiple GPUs)
Structured Grids	Finite Difference Isotropic stencil	Finite Difference stencils from CUDA SDK (modified for multiple GPUs)

Evaluation

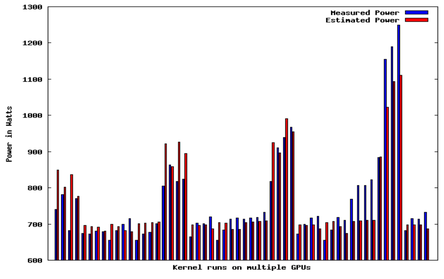
Apart from hardware counters, we noticed that some performance attributes like execution time and operation throughput having a positive impact on the model. Operations per unit time is a very useful metric to judge the performance of an application; it is a more suitable metric compared to FLOP/s, as different operations could consume a varied number of FLOPs. We notice that the estimation of power becomes more accurate, when we take into consideration the Operations per unit time (for instance, for DGEMM - number of multiplications per second, for PRNG, amount of random numbers generated per second, etc.).

Energy ~ (Instructions Issued + DRAM reads + L2 misses + DRAM writes + Elapsed Time)
Power ~ (Instructions Issued + DRAM reads + L2 misses + Threads launched per block per SM + Operations per unit time)

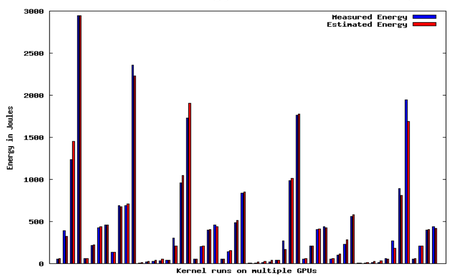
We compare the predicted results of power and energy, with that of actual measured results; the average error for power was found to be **5.32%** and that of energy was **8.73%** respectively.To assess the quality of the model, we would need to examine the distribution of the residual errors. Ideally, residuals should be randomly distributed.

Observations

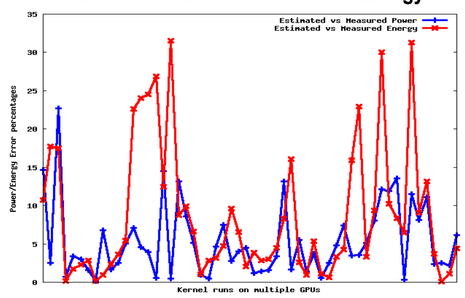
Measured Power Vs Estimated Power



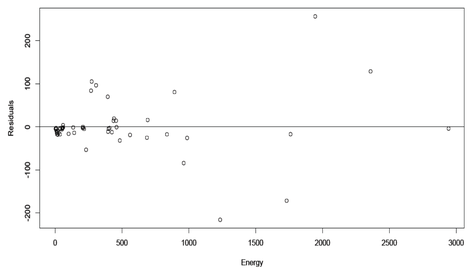
Measured Energy Vs Estimated Energy



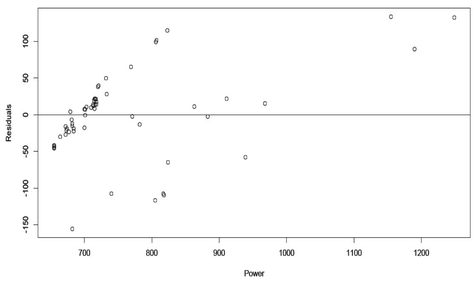
Estimation Errors – Power and Energy



Residual Error Distribution of Energy



Residual Error Distribution of Power



Conclusions

We have shown that it is possible to estimate power and energy with a small set of dependent variables, in the form of hardware counters and performance attributes. Our average errors for predicted values of power and energy are well within 10%, and we test the efficacy of our model by examining the residual error distribution, which is semi-normal, suggesting that removal of some results would not greatly affect the model. Since profiling an application with hardware counters, causes it to run for many iterations, using a small set could save considerable amount of time.

However, statistical tests rely a lot on input data, and although we sampled data from multiple runs of 4 different kernels, it is still not enough to accurately estimate a vast range of possibilities.

References

[1] Nagasaka, H. and Maruyama, N. and Nukada, A. and Endo, T. and Matsuoka, S., Proceedings of International Green Computing Conference (2010)
[2] Ma, X. and Dong, M. and Zhong, L. and Deng, Z., Proceeding of ACM SOSP Workshop on Power Aware Computing and Systems (HotPower) (2009)