Power-aware Computing on GPGPUs

INTRODUCTION

Recently, GPGPU accelerated computing systems have drawn the attention of researchers. Because GPGPUs have abundant cores and arithmetic computational units, they are inherently suited for massively parallel and computation intensive workloads. The most recently released supercomputer, Tianhe-2A, equipped with more than 16,000 NVIDIA GPGPU Tesla M2090, has reached up to 3.20 PFlops on the top spot on the TOP500 list in June 2013. Coming along with the exciting computational capability, the power consumption of supercomputers has become a serious issue. For example, the average power consumption of the TOP500 supercomputing centers was 13.2 Mw in 2006 and 3.2 Mw in 2010, translating to a multi-million-dollar electric bill. Designers must employ aggressive power-management techniques to keep ballooning power cost under control. A key challenge to effective runtime power management is estimating the real-time power consumption. Although the power estimation for the processor, memory, disks, and fans has been introduced, the power estimation technique of GPGPUs is relatively less developed.

In this work, we explore the use of the NVML library to measure runtime power consumption of several fundamental BLAS (Basic Linear Algebra) operations and LAPACK algorithms. We use implementations from the MAGMA library (http://icl.cs.utk.edu/magma/), which includes support for power management. To measure the real-time power consumption, we use the latest NVIDIA C2075 Fermi GPU, which has been optimized to save power and extend the GPU’s lifetime.

RESULTS

We analyzed the real-time power consumption of two fundamental linear algebra algorithms – the LU factorization (magma_dgetrf) for solving dense linear systems of equations and the upper Hessenberg reduction (magma_dgehrd) for solving the general eigenvalue problem. Results show that MAGMA implementations of these algorithms achieve astounding energy efficiency. Depending on the hardware and software configuration, we have demonstrated that MAGMA uses as little as 150% of the energy of traditional multi-core CPUs. Shown are the performance charts for the two algorithms along with energy consumption traces. The MAGMA LU factorization is a complex bound algorithm (expressed in terms of GEMMs) and the MAGMA Hessenberg reduction is memory bound (expressed in terms of GEMVs and GEMM), correspondingly 70% and 80% of the power consumption. The real-time power consumption for the two basic BLAS kernels (Gemm and GEMV) used in these algorithms is also shown.

MODEL

To measure the real-time power consumption of a GPGPU, we use NVIDIA’s NVML Library Management Library (NVML) which provides access to power usage readings for devices, in real-time. This is the power draw for the entire board, including GPU memory, etc. A device is a single GPU and refers to the hardware which is a software approach. We use NVML to read power or temperature measurements on a single thread and run MAGMA LU factorization or the MAGMA Hessenberg reduction on another and measure power in real-time. We are developing an analytical model to estimate the power draw for arbitrary linear algebra operations on the GPGPU. We use our model to estimate the power consumption and validate the accuracy of our model using the real-time power consumption data from our power measurement experiments.

HPC @ 1/10TH THE COST & 1/20TH THE ENERGY

LU Factorization in Double Precision (DP)

Hessenberg Factorization in DP

CONCLUSION & FUTURE WORK

The current work has shown that measuring performance leads to reduced execution time which results in proportional reduction in the energy consumption. A more detailed analysis is needed, though, and libraries like LAPACK and ScaLAPACK should be crucial in enabling it. Other parameters influencing energy saving must be identified and energy saving hardware features must be integrated into MAGMA (e.g., what hardware context - how many CPUs are used and how many GPUs is being used to solve a problem, etc.). First and foremost this requires an investigation of the energy consumption effects of various algorithms. Future work includes the expansion of the current infrastructure for precise measurements and the development of energy consumption models. Tuning parameters must be identified and added to MAGMA’s activation framework. The newly released CUFFT (http://icl.cs.utk.edu/cufft/) CUDA Component offers a promising way to estimate structure power at runtime by enhancing methodology from design-time CPU power models like Watkins. This kind of technique counts on estimation of structures (e.g., cache hit rate) and cumulative power estimate based on a power-aware energy model. However, such direct computation of structure power on GPGPUs would require hundreds of Simulation Statistics. To address this challenge, we may use a structure-aware power model in hardware, we would like to extend our current power model for GPGPUs (AMoS), to estimate activity factors and power for micro-architectural structures on GPGPUs. This model will not only rely on real-time current measuring, or simulating hundreds of simulation statistics, but also predict performance vs power consumption. We expect only a few input statistics are sufficient to estimate per-structure dynamic power of a GPGPU because the myriad of per-structure events are related to a small set of global parameters, such as IPC and load. In the future, we may consider new methods to derive the development of AMoS. We will first analyze the correlation of system-level power metrics. Then we monitor only the least important terms and use monitored metrics to extrapolate the concerned metrics. After we obtain all of the required metrics about the structure events, we apply a power-aware energy model derived from a simple model to those structures in calculating the power of each structure. We believe that AMoS makes a further step towards understanding and reducing the power of GPGPUs systems through the usage of architecture level performance counters.