Project Title

A Cycle-Level Dataset Enabling GPU Optimization using Machine Learning

Background

Performance improvements for processor hardware are becoming increasingly challenging. The problem is that conventional system design overlooks the interactions that can be foreseen and mitigated by taking a holistic look at the interactions occurring across the entire system. While designers would like to ideally take a holistic picture into account, the challenge lies in manually understanding and rationalizing the interactions at design time. A data-driven AI approach to computer architecture design can enable the development of efficient, scalable designs that can learn complex processor behaviors and events from processor data.

In our previous work [1,2], we validated this data-driven approach in the context of GPU execution of arbitrary programs. We showed that complex behaviors can be mined from processor data that can in turn lead to data-dependent (or software driven) optimizations to the hardware execution of _your_ code. With this document, we release a subset of these data along with the following guide that can be used to generate analogous data for arbitrary code and a range of GPU architectures. We hope that computer architects can use the data to train Machine Learning models to predict and improve processor performance, power consumption, etc.

The dataset consists of programs from the RODINIA benchmark suite [3], a collection of a variety of workloads written in CUDA. For each workload, the data set is further divided in to a dataset for each kernel and core of the GPU. Each such dataset consists of (1) instruction counters (integer) corresponding to the total number (over threads, warps, SPs) of running operations of each PTX opcode such as add, mul, mov etc. These are in comma-separated format in files named "instructions_kernel_<id>_core_<id>"(2) hardware events such as a variety of cache hits and misses (binary e.g. data cache miss, instruction cache hit), number of idle cores, warp instructions, floating point operations etc. These are in files named "metric_kernel_<id>_core_<id>". In total, there are 62 features of the first type and 21 features of the second type. The below video shows an example of these features over time.

We modified the simulator GPGPUSim to extract these features at the cycle level. As mentioned above, we provide datasets for workloads in the RODINIA benchmark by simulating an NVIDIA GTX480 GPU. However, it is easy to generate these data for any GPU architecture and any workload. In addition to the dataset, we make our fork of GPGPUSim available along with config files, PTX compilations and scripts to extract the dataset.

References to related work using this data:
[1] A. Raghavan, et al. "GPU Activity Prediction using Representation Learning", ML Systems Workshop, arXiv:1703.09146 (2017)
[2] T. Mathew, et al. “Computer Architecture Design using Deep Temporal Models”, Workshop on Energy Efficient Machine Learning and Cognitive Computing for Embedded Applications (EMC2), March 2018.
[3] Che, et al., “A Characterization of the Rodinia Benchmark Suite with Comparison to Contemporary CMP Workloads”, IISWC2010
[4] Bakhoda et al. Analyzing Cuda Workloads Using a Detailed GPU Simulator. ISPASS2009

Prerequisites

The dataset does not require any additional software or processing.

Installing

The dataset is provided as a compressed zip file.

Description

The dataset does not require any additional software or processing. Users can use their favorite machine-learning framework to ingest the dataset. Detail on this approach is described in [1-2].

Versioning

Version 1, May 10, 2018

Authors

Aswin Raghavan, Sek Chai

Copyright

Acknowledgments

This material is based upon work supported by NSF #1526399.