Fingerprinting Anomalous Computation with RNN for GPU-Accelerated HPC Machines

Pengfei Zou, Rong Ge
{pzou,rge}@clemson.edu
Clemson University

Ang Li, Kevin Barker
{ang.li,kevin.barker}@pnnl.gov
Pacific Northwest National Laboratory

Keywords

HPC security, GPU accelerated systems, workload classification.

1 Introduction

The integration of GPGPUs complicates the security issues in HPC systems [4]. By exploiting GPU-accelerator based HPC systems, attackers get the needed high hash rate and avoid paying for the computing resource and energy bills. Attackers have exploited HPC systems for bitcoin mining [1], brute force password cracking [2] and GPU-inflated denial-of-service (DoS) attack [7]. Such HPC security incidents not only deprive mission-critical and scientific applications of execution cycles, but also increase the chance for attackers to steal data, damage systems, and leverage the high computation and network bandwidth for attacking other sites.

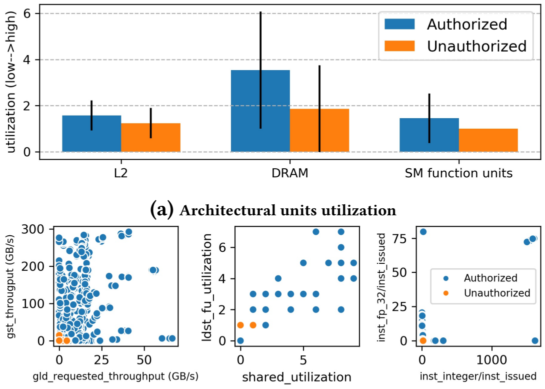
GPU accelerated HPC systems require different security measures from traditional IT and homogeneous HPC systems. Existing measures for homogeneous HPC systems detect illicit computation using CPU execution patterns [3]. Their applications to GPU accelerated systems are problematic because illicit computations are offloaded to GPUs without much CPU involvement [5, 6]; a GPU-side monitoring and detection approach is thus highly desired.

Fortunately, regarding HPC workloads, we can leverage their unique features and patterns to effectively mitigate risks for (open) systems and compute nodes. HPC systems often present a small set of programs with specific resource usage patterns that are more predictable [4]. Such workloads have a high chance to invoke certain functions such as linear algebra operations and fast Fourier transform. They have distinctive microarchitectural activities and system behaviors that can be monitored, collected, and identified.

In this paper, we present a machine learning framework for classifying GPGPU workloads based on their microarchitectural and system behaviors. We achieve over 95% accuracy to detect illicit workloads with the following contributions: **1.** We demonstrate the feasibility and accuracy of using workload behavioral data to fingerprint illicit computations and anomalous workloads in GPU-accelerated HPC systems. **2.** We investigate multiple online and offline machine learning methods for anomalous workload detection. **3.** We evaluate our workload analysis and classification framework with 83 applications and kernels over 3 generations of GPU architecture.

2 GPU Workload Profiling

To correctly classify the running applications, we collect workload profiles and behavioral data. As the characteristics of GPU-accelerated HPC workload have rarely been studied, features that can best identify the workloads are unknown. To address this issue, we collect as many features as possible with available profilers. Specifically, we profile workload execution at both system and microarchitectural levels, and collect multiple types of workload behavioral data from multiple sources.



(a) Architectural units utilization

(b) Architectural component throughput

Figure 1: Different patterns for measured performance metrics between authorized and unauthorized workloads.

2.1 Microarchitectural Events Behavior

Microarchitectural events are activities in hardware including processing units, memory, and caches, and can be monitored with Performance Monitor Counters (PMCs). Prior work shows that PMC measured events are strong indicators of workload characterization for CPU workloads [3]. Our initial study shows that PMC are also good indicators to classify GPU workloads. As shown in Figure 1, for unauthorized programs such as cryptocurrency and password cracking, hashing functions on integer units are intensively executed. Moreover, they tend to show a much smaller utilization of the devices in the memory hierarchy including L2 cache, load/store functional units, and DRAM. In addition, they demonstrate much smaller global load/store throughput compared to authorized workloads.

2.2 Data Movement Behavior

In addition to architectural events monitoring, we also collect time series data. Such data complement the cumulative event counts collected from PMCs, and are particularly suitable for online detection. We collect data movement between host and device over time. Unauthorized workloads periodically transfer data from host to device, and barely transfer data from device to host. In contrast, authorized programs are optimized, offloading larger amount of data to device, and transfer data back from device. Such differences in kernel execution time, data transfer direction and volume can be leveraged to identify illicit computations and anomalous programs.

2.3 Resources Utilization Behavior

Figure 2 shows an example of resource usage traces for linear algebra, data processing, and password cracking applications. Matrix-multiply consumes significantly higher and stable power compared to the other two. While radix-sort and password-crack show varying patterns with similar frequency for both power and memory