

A High-Performance Fault-Tolerant Software Framework for Memory on Commodity GPUs

Naoya Maruyama and Akira Nukada
GSIC, Tokyo Institute of Technology
JST CREST

Email: naoya,nukada@matsulab.is.titech.ac.jp

Satoshi Matsuoka
GSIC, Tokyo Institute of Technology
National Institute of Informatics
JST CREST
Email: matsu@is.titech.ac.jp

Abstract—As GPUs are increasingly used to accelerate HPC applications by allowing more flexibility and programmability, their fault tolerance is becoming much more important than before when they were used only for graphics. The current generation of GPUs, however, does not have standard error detection and correction capabilities, such as SEC-DED ECC for DRAM, which is almost always exercised in HPC servers. We present a high-performance software framework to enhance commodity off-the-shelf GPUs with DRAM fault tolerance. It combines data coding for detecting bit-flip errors and checkpointing for recovering computations when such errors are detected. We analyze performance of data coding in GPUs and present optimizations geared toward memory-intensive GPU applications. We present performance studies of the prototype implementation of the framework and show that the proposed framework can be realized with negligible overheads in compute intensive applications such as N-body problem and matrix multiplication, and as low as 35% in a highly-efficient memory intensive 3-D FFT kernel.

Keywords—GPGPU; fault tolerance; memory soft errors;

I. INTRODUCTION

Modern graphics processors have been successfully demonstrated to accelerate a wide variety of HPC applications by several orders of magnitude, including physical simulations [1, 2], bioinformatics [3], and medical analysis [4]. One of the side effects of this trend, however, is that the reliability of GPUs must be carefully reconsidered. Traditionally, it had not been given much attention since the dominant applications of GPUs, such as 3-D graphics games, favor performance over reliability. In contrast, many of the newly adopted scientific applications such as bioinformatics and medical analysis require more rigorous error detection and correction (EDAC) capabilities even with extra performance overheads. Standard HPC platforms where these applications are routinely run use hardware EDAC mechanisms for key vulnerable components, including DRAM [5], SRAM [6], and even arithmetic units and data paths in some RAS-conscious systems such as PowerPC [7] and SPARC processors [8]. In contrast, the reliability of the current generation of GPUs is quite limited: as far as we know, there are no EDAC mechanisms in GPU memory systems except for the newer DRAM interface called GDDR5, which has cyclic

redundancy checking in transmission links between GPU DRAMs and chips [9]. The problem is more prevalent in long-running parallel applications exploiting a large number of distributed GPU accelerators.

To overcome the reliability concern, this particular paper focuses on DRAM bit-flip errors and shows that *high-performance* fault tolerance for such errors can be realized by means of software-only techniques. Bit-flip errors are one of the well-known classical vulnerabilities in semiconductor memory, and can be caused by energetic particles, such as cosmic neutrons and alpha particles [10–15]. They can cause fail-stop program crashes and silent data errors [16, 17]. While the former are externally visible and thus relatively simple to detect, the latter are not: Unless data integrity is completely checked, they can modify program results without affecting the original control flow. Such silent errors on DRAM do not always cause incorrect program results (i.e., benign errors) [11], especially when the data size of applications is small; however, since GPUs are often used to accelerate scientific applications to process large-scale data, the probability of true errors detectability of such errors should not be optional but necessity. In fact, we had observed eight bit-flip errors in a 72-hour run of a memory-stress testing program on 60 NVIDIA GeForce 8800GTS 512.

Hardware solutions for such errors, such as error-correcting code (ECC) for DRAM [5] and redundant executions [18], can be implemented in GPUs. NVIDIA recently announced that their new Fermi-based GPUs will be equipped with ECC for both on- and off-chip memory. However, it is still unlikely that such solutions will be employed in the commodity cost-conscious mainstream GPUs; the majority of the applications of the GPUs are still graphics processing that does not require extensive reliability.

To achieve DRAM fault tolerance in commodity off-the-shelf GPUs, we propose a software framework that extends standard GPU applications with error detection and recovery. Before running a GPU program, it keeps a copy of input data on the host memory or disk. While the GPU program is executed, it checks data integrity by using a parity-based error-detection code. If errors are actually detected, the GPU program is re-executed by using the input data kept on host.