A Framework for Resilient and Energy-efficient Computing for GPU-accelerated Systems

Zheng Miao, Jon C. Calhoun (advisor), Rong Ge (advisor)
Clemson University
{zmiao, jonccal, rge}@clemson.edu

Objectives

Problems: Resilience and energy-efficiency are two main challenges for large-scale GPU-based heterogeneous systems. Resilience techniques allow applications to successfully finish executions in the presence of failures but incur performance and energy costs. Energy efficiency changes from being desirable to being mandatory as power consumption of HPC systems increases.

- Higher power on GPUs lead to higher error rates and more job interruptions, requiring more advanced resilience techniques
- Resilience incurs power and time overhead and exacerbates the power challenge for GPU-based heterogeneous systems

Research goal: To improve both resilience and energy-efficiency for GPU-accelerated heterogeneous HPC systems

- Provide resilience for GPU computing at scale.
- Improve energy-efficiency and time-efficiency for resilient GPU applications.

Opportunities & Methodology

Opportunities:

- Plenty of hardware resources (CPUs, GPUs) for redundancy.
- Adjustable computation precision and power management.

Methodology

- Leverage abundant, various available resources (low/high-power CPUs/GPUs) to provide redundancy and support resilience
- Leverage algorithm and architectural technologies (low precision and mixed precision on CPUs, GPU, and TPUs) to reduce energy cost of redundancy
- Leverage hardware power saving technologies (DVFS, power capping) to further improve energy efficiency

The state-of-the-art:

- Resilience for GPU computing is relatively less studied, let alone energy efficiency.
- Redundancy [1] promises energy-efficient resilience for large scale systems, but is only studied for CPU computing.

Contributions

- We build the first of its kind framework to provide redundancy in GPU-accelerated systems.
- The framework allows main and replica processes to take over the role of each other alternatively upon failures.
- It supports redundant computing with configurable hardware resources (low/high-power CPUs or low/high-power GPUs), flexible precision [2] and power management to meet user's requirements.
- It further optimizes resilience overhead [3] and GPU communications.

Framework Design

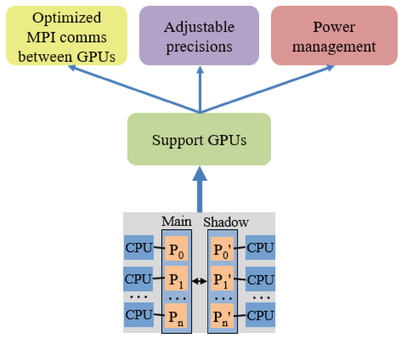

Figure 1: Design of our framework.

Key Features

- For each main process of a GPU program, there is a configurable number (typically [0.5-1]) of replica processes, which can run on GPUs or CPUs.
- Redundant computing and fault handling are implemented through MPI messaging and hashing [4] as Figure 2 shows.
- Upon failures, main and replica processes can take over the role of each other and adjust their speeds as Figure 3 shows.
- GPU communications leverage GPUDirect for high bandwidth and low latency.
- Replica processes can use low (single, half, mixed) precision to reduce power consumption.
- Power management technologies including DVFS and power capping are exploited by replica processes.

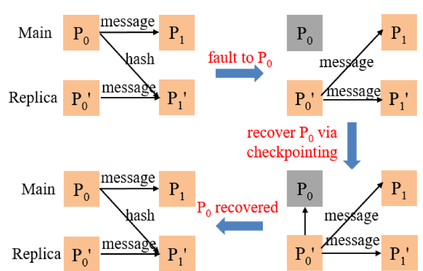

Figure 2: MPI implementation for redundancy and fault recovery. The faulty main MPI process is replaced by the corresponding replica process until the faulty process is recovered from checkpointing.

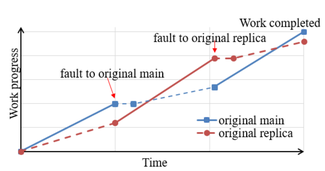

Figure 3: Resilience to multiple faults. Solid lines show current main processes.

Experiments & Results

We implement a prototype of the framework and evaluate it on a NUMA cluster built with dual 12-core Intel Haswell processers nodes and a total of 4 Titan V GPUs. We use an iterative CG with a 27-point stencil matrix as a study case.

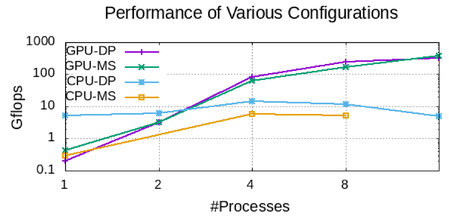

Figure 4: Performance of various resources and precisions. DP—double precision, MS—mixed with single precision. GPU with mixed precision shows higher performance at large problem sizes, even with precision conversion overhead.

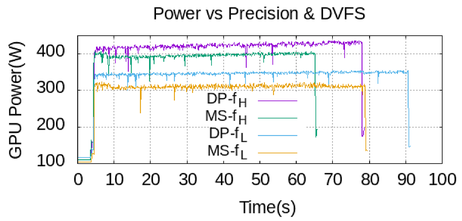

Figure 5: Power reduction with mixed-precision and lower GPU frequency. f_H—high frequency, f_L—low frequency. DP-f_H and MS-f_L have similar execution time, but the latter has a 30% lower power consumption.

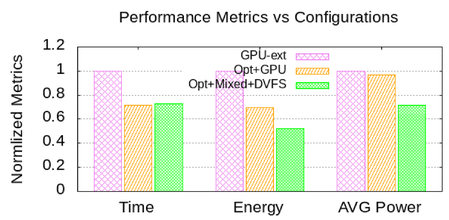

Figure 6: Normalized performance metrics for various configurations. Base is a simple extension of CPU replication to support GPUs. The others represent the replica processes run on GPUs but adopt the optimization techniques.

Main Observations

By applying multiple techniques, our framework simultaneously addresses performance and energy efficiency for GPU applications in faulty environments.

- As problem size increases, mixed-precision for replicas on GPUs shows better performance.
- Combined mixed-precision and power management achieve significant power savings.
- When replica processes run on GPUs, mixed-precision with a high frequency shows the best performance while mixed-precision with a lower frequency shows the best energy saving.

Discussion

Decision of main and replica execution rates.

- The choice of execution rates depends on various considerations, such as resource utilization or energy-efficiency.
- To maximum resource utilization, users can select execution rates with a smaller difference (for example, both GPUs).
- To get shortest time-to-solution, users can use all GPUs for main process and all CPUs for replica process.

Decision of precision for main and replica.

- Applying different precisions for main or replication processes generates an "incorrect" hash for duplicated main-replica communications.
- A possible solution to this issue is replacing hash with checksum under a certain tolerance.

Conclusions

We present a novel framework for resilient and energy-efficient HPC on GPU-accelerated systems. This framework

1. provides replica redundancy based on MPI implementation for GPU applications.
2. significantly reduces the resilience overhead of GPU applications while maintaining energy-efficiency.

Future work:

- We will optimize our resilience mechanism in recovering faulty processes to improve reliability.
- We will incorporate more advanced accelerators such as tensor cores or FPGAs, and explore mixed-precision with FP16 in those advanced accelerators for higher energy-efficiency.

References

[1] B. Mills, T. Znati, and R. Melhem, "Shadow computing: An energy-aware fault tolerant computing model," in *ICNC*, IEEE, 2014, pp. 73–77.

[2] A. Buttari, J. Dongarra, J. Kurzak, P. Luszczek, and S. Tomov, "Using mixed precision for sparse matrix computations to enhance the performance while achieving 64-bit accuracy," *ACM Transactions on Mathematical Software (TOMS)*, vol. 34, no. 4, p. 17, 2008.

[3] J. Kraus, "An introduction to cuda-aware mpi," *Weblog entry]. PARALLEL FORALL*, 2013.

[4] D. Fiala, F. Mueller, C. Engelmann, R. Riesen, K. Ferreira, and R. Brightwell, "Detection and correction of silent data corruption for large-scale high-performance computing," in *SC12*.

Acknowledgements

This work is supported in part by the U.S. National Science Foundation under Grants CCF-1551511 and CNS-1551262.