Fault Tolerance for GPU Accelerated HPC Systems

Saurabh Hukerikar
{hukerika@usc.edu}
Ming Hsieh Department of Electrical Engineering

Motivation

- Graphics Processing Units (GPUs) are high-performance many-core processors that can be used to accelerate a wide range of HPC applications.
- GPU memories typically did not include error detection/correction capabilities until very recently (ECC protection).
- Number of nodes (both CPUs and GPUs) in HPC clusters increasing.
  - E.g. Shanghai National Supercomputing Center’s Tianhe-1A (2.57 Petaflop/s) and 2nd on TOP 500, uses 7168 NVIDIA Tesla GPUs and 14336 Xeon CPUs.
- Higher density in components results in failures becoming more frequent because of an increased number of Points of Failure (PoF) in the system.

Approach

- We must capture the state of the system so that it is possible to recover from failure and replay the application from the exact point of failure.
- Non-trivial to capture the state which is distributed across multiple nodes and also distributed between the host and GPU accelerator.
- Need a failure detection and recovery trigger mechanism at node level and accelerator-level.
- We approach the problem using a multi-tiered checkpoint and recovery method.

The System Architecture

- Cluster level failure detection and recovery by Heartbeat frames – captures state of global communication.
- Node level recovery using Berkeley Labs Checkpoint & Rollback library – captures state of process on node as seen by OS.
- GPU detection and recovery service:
  - Modeled as intercept library
  - Captures state of the GDRAM and transfers to host for checkpoint creation
  - Failure detection based on CUDA SMI

GPU Checkpointing: Intercept all GPU memory based calls to the CUDA runtime: cuMem*(), cuArray*(), cuModule*(), CUDA context is destroyed and checkpoint created into filesystem.

GPU Rollback to previous checkpoint: Context is recreated on the device and memory snapshot from file reloaded into GPU memory.

Metrics of Success

Availability of HPC System = \[
\frac{\text{Mean Time to Failure}}{\text{Mean Time to Failure} + \text{Mean Time to Repair/Recover}}
\]

- The intercept library based GPU increases MTTF of the HPC system by adding capability to tolerate GPU failures.
- In a multi-tiered architecture, another important metric of success is correctness in recovery across multiple nodes and within the node (between GPU and host).
- Managing policy v mechanism – how often to checkpoint and granularity of checkpoints.

References

1. TOP500, http://www.top500.org