



Contribution ID: 45

Type: **Project talk**

Scalable GPU-Accelerated Incremental Checkpointing of Sparsely Updated Data

Wednesday 22 March 2023 16:20 (20 minutes)

Checkpointing large amounts of related data concurrently to stable storage is a common I/O pattern of many HPC applications in a variety of scenarios: checkpoint-restart fault tolerance, coupled workflows that combine simulations with analytics, adjoint computations, etc. This pattern is challenging because it needs to happen frequently and typically leads to I/O bottlenecks that negatively impact the performance and scalability of the applications.

Furthermore, checkpoint sizes are continuously increasing and overwhelm the capacity of the storage stack, prompting the need for data reduction. A large class of applications including graph algorithms such as graph alignment, perform sparse updates to large data structures between checkpoints. In this case, incremental checkpointing approaches that save only the differences from one checkpoint to another can dramatically reduce the checkpoint sizes, which reduces both the I/O bottlenecks and the storage capacity utilization. However, such techniques are not without challenges: it is non-trivial to transparently determine what data changed since a previous checkpoint and to assemble the differences in a compact fashion that does not result in excessive metadata. State-of-art deduplication techniques have limited support to address these challenges for modern applications that manipulate data structures directly on GPUs. Our approach builds a compact representation of the differences between checkpoints using Merkle-tree-inspired data structures optimized for parallel construction and manipulation.

Our previous talk introduced the project and focused on the challenge of making efficient incremental checkpoints on GPU-accelerated platforms. We presented our compact representation for representing incremental checkpoints. Our algorithm was implemented and initial testing was done with ORANGES, a graph alignment application with sparse update patterns.

For this project update, we have optimized and refactored our implementation and compared the performance of the following approaches.

Full Checkpoint: Copy all data from the GPU to the Host

Basic Incremental Checkpoint: Break data into chunks and save the chunks that have changed since the previous checkpoint

List Incremental Checkpoint: Identify and save a single copy of each new chunk along with a list of shifted duplicate chunks

Our approach: Expand on the List approach by storing shifted duplicates in a compact tree representation

We have analyzed the degree of deduplication for the checkpoint along with the runtime overhead for creating and saving the checkpoint to the Host. We have also examined various tradeoffs that affect checkpoint size and deduplication performance.

Our next steps are to compare performance with compression techniques, evaluate different applications or access patterns, and examine alternative hash functions. Locality-sensitive hash functions in particular are useful for lossy deduplication for floating-point data.

JLESC topic

Resilience and fault tolerance

Primary authors: TAN, Nigel Phillip (University of Tennessee Knoxville); Dr NICOLAE, Bogdan (Argonne National Laboratory); Dr LUETTGAU, Jakob (University of Tennessee Knoxville); Prof. BHOWMICK, Sanjukta

(University of North Texas); Dr TERANISHI, Keita (Oak Ridge National Laboratory); Dr MORALES, Nicolas (Sandia National Laboratories); Prof. TAUFER, Michela (University of Tennessee); Dr CAPPELLO, Franck (Argonne National Laboratory)

Presenter: TAN, Nigel Phillip (University of Tennessee Knoxville)

Session Classification: Project Talks on further topics

Track Classification: Resilience and compression