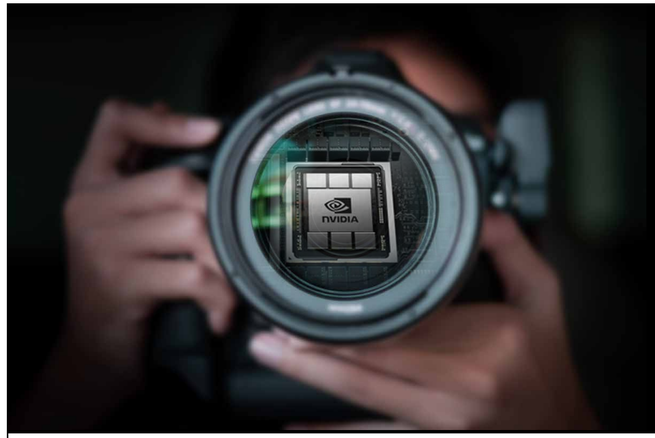
Poster ID: **P63184** | Topic / Category: **Data Center / Cloud - Data Center / Cloud Security**

Bernie Wu: bernie.wu@memverge.com



Achieving K8S and Public Cloud Operational Efficiency Using a New Checkpoint/Restore for GPUs

Author: B. Wu, T. Zhang, Z. Xiong, and Y. Tian

MemVerge

NVIDIA Inception Program

Introduction and Problem Statement

Many AI/ML frameworks and related workflow platforms have built-in checkpoint/restore mechanisms that enable periodic checkpointing of training epochs, MLOp pipeline stages, etc. to provide some level of fault tolerance, tuneability, cloning, and versioning. However, other types of CPU/GPU-based applications may lack such capabilities and/or have either extremely long run times or restart times. Complex initiation sequences and large CPU/GPU memory footprints can contribute to prolonged restart times. Furthermore, there may be a need for system administrators/infrastructure operators to transparently pre-empt running applications to allow re-prioritization and migration workloads with minimum loss of productivity while running at scale. Operators will also need to drain nodes for maintenance/upgrades or for cost-containment reasons, desire to run AI/ML workloads on preemptible public cloud spot instances. These types of use cases require a system-level, application-transparent checkpoint-restart mechanism to be implemented.

A new CUDA driver (12.x) is being developed by Nvidia to support the system-level checkpoint/restore of GPU-specific memory and machine state within and across Nvidia's GPU product lines. The driver deposits the checkpointed state into system memory where it can then be transferred to files as needed. MemVerge and Nvdia have collaborated to modify the open-source CRIU (**C**heckpoint **R**estore **I**n **U**serspace) project to work with this driver and will be upstreaming the changes to the CRIU open-source repository.

Furthermore, MemVerge is characterizing and optimizing the performance and overhead of the checkpoint/restore function as well as integrating it into proof-of-concept demonstrations intended to show how operational efficiency and resiliency can both be increased for K8s and public clouds. The initial Proof-of-Concept showing coordinated CPU-GPU machine state and memory checkpointing and restoration is described below.

Architecture

MemVerge software supercharges checkpointing for applications running on CPU and Nvidia GPU, with no application change and minimal downtime. Here's how:

a. Simultaneous Freeze: pauses both CPU and GPU processes at the same time, leveraging a new CUDA driver function to copy GPU memory to system memory.

b. Quick Resume: The application resumes running immediately while MemVerge efficiently constructs checkpoint metadata, while the complete image is offloaded to storage in the background.

This performance boost enables frequent checkpoints without significant overhead, ensuring application resilience and flexibility.

The restore works in reverse, as follows:

c. Restore CPU memory, GPU memory, and file content from checkpoint image in storage

d. Restore process states in CPU and GPU, and restart the application.

Proof of Concept

System setup:

- CPU: Intel Xeon Gold 5120 with 512 GB DRAM
- Operation System: Ubuntu 20.04.6 LTS
- GPU: x2 NVIDIA V100 with 16 GB GPU RAM
- CUDA Version: 12.X
- Server: Supermicro SYS-1029GQ-TVRT

Application: Tensorflow *mnist* sample with 14GB Data Set

Testing Process:

1. Start *mnist* sample, and confirm the application is running on both CPU and GPU.
2. Use MemVerge *MemoryMachine* to checkpoint the *mnist* process, and save the image to storage. Note that the *mnist* process is killed upon checkpoint completion, so we can show restore on the same node.
3. Restore *mnist* from the checkpoint image. Check the log to confirm the process starts from where it was left off.
4. After *mnist* completes, confirm correctness by checking the result.

Result

The Checkpoint was initiated at Epoch 12/20 , completed in 21 seconds, and then the job was killed manually. The restoration process took 18 sec and the process ran to completion. Correctness was verified.

Please see this QR code to see a recording of the demo.

Conclusion

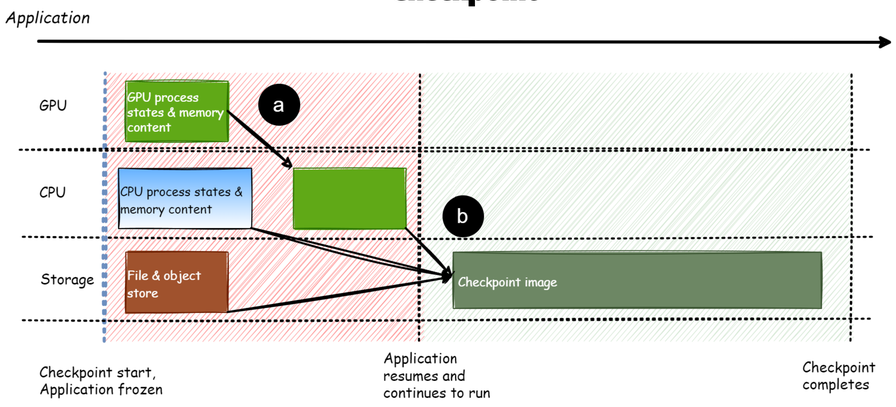
This Proof-of-Concept has successfully demonstrated that the entire CPU/GPU memory state can now be checkpointed/restored transparently. An asynchronous snapshot approach can also be implemented to minimize production interruptions and reduce the snapshot quiescent period. A K8S operator has also been prototyped, however, further development/testing is needed to solidify GPU snapshot error handling The updated CUDA driver is scheduled to be released in H1'24. MemVerge and Nvidia expect to upstream the corresponding changes to the CRIU project in that same timeframe.

This new feature allows Infrastructure architects and MLOps teams to supplement existing application/pipeline-specific snapshot tools by facilitating the automation of other system-level use cases such as spot instances, cloud bursting, job/batch prioritization, and instance rightsizing.

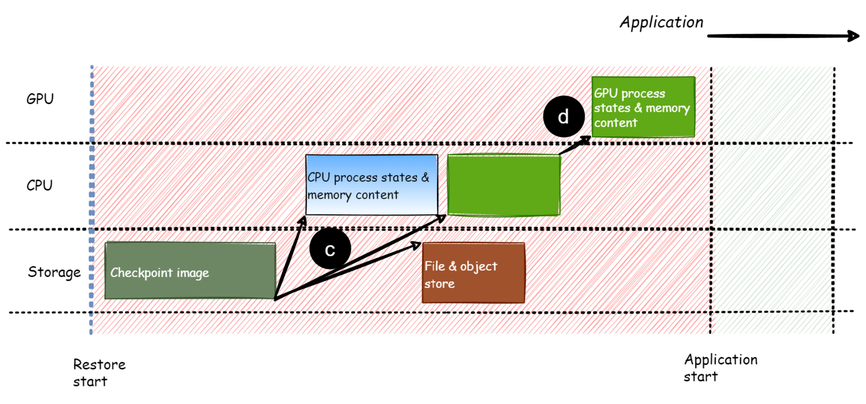
Acknowledgments

The authors would like to acknowledge the contributions of S. Gurfinkel and J. Ramos from Nvidia. Inquiries on this poster may be addressed to bernie.wu@memverge.com

Checkpoint



Restore



Watch Demo