

Supporting ROCm with CRIU

CRIU can checkpoint and restore processes using standard kernel interfaces. However, out of the box, it cannot support processes using device driver interfaces for devices like GPUs or compute accelerators.

CRIU already has a plugin architecture to support processes using device files. Using this architecture we added a plugin for supporting CRIU with GPU compute applications running on the AMD ROCm software stack. This requires new ioctls in the KFD kernel mode driver to save and restore hardware and kernel mode driver state, such as memory mappings, VRAM contents, user mode queues, and signals. We also needed a few new plugin hooks in CRIU itself to support remapping of device files and mmap offsets within them, and finalizing GPU virtual memory mappings and resuming execution of the GPU after all VMAs have been restored by the PIE code.

The result is the first real-world plugin and the first example of GPU support in CRIU.

We are going to present the architecture of our plugin, how it interacts with CRIU and our GPU driver during the checkpoint and restore flow. We can also talk about some security considerations and initial test results and performance stats.

Further reading: <https://github.com/RadeonOpenCompute/criu/tree/criu-dev/plugins/amdgpu#readme>

Our work-in-progress code: <https://github.com/RadeonOpenCompute/criu/tree/amd-criu-dev-staging>

I agree to abide by the anti-harassment policy

I agree

Primary authors: BHARDWAJ, Rajneesh (AMD); Mr KUEHLING, Felix; YAT SIN, David

Session Classification: Containers and Checkpoint/Restore MC

Track Classification: Containers and Checkpoint/Restore MC