



Fast Checkpoint Restore for GPUs

Felix Kuehling

Rajneesh Bhardwaj

David Yat Sin

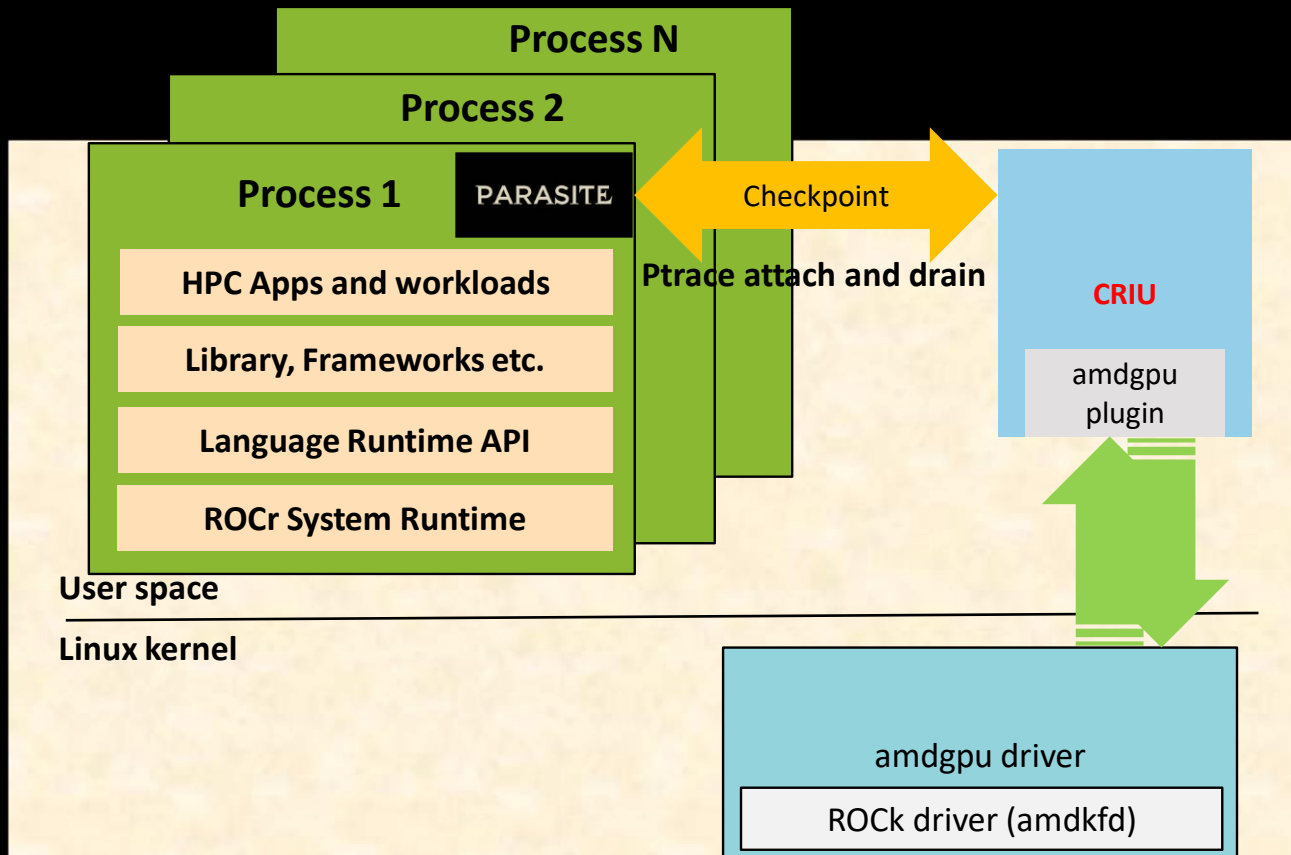
Device files and CRIU

- ▶ Cannot Checkpoint processes with device files (yet)
 - ▶ Device specific state unknown / invisible to CRIU
 - ▶ E.g. GPU-specific state:
 - ▶ VRAM buffer objects
 - ▶ GPU VA mappings
 - ▶ User mode queues
- ▶ CRIU plugin mechanism
 - ▶ Shared object, loaded before dumping or restoring
 - ▶ No real device plugin that exists
 - ▶ More new hooks required for GPU devices



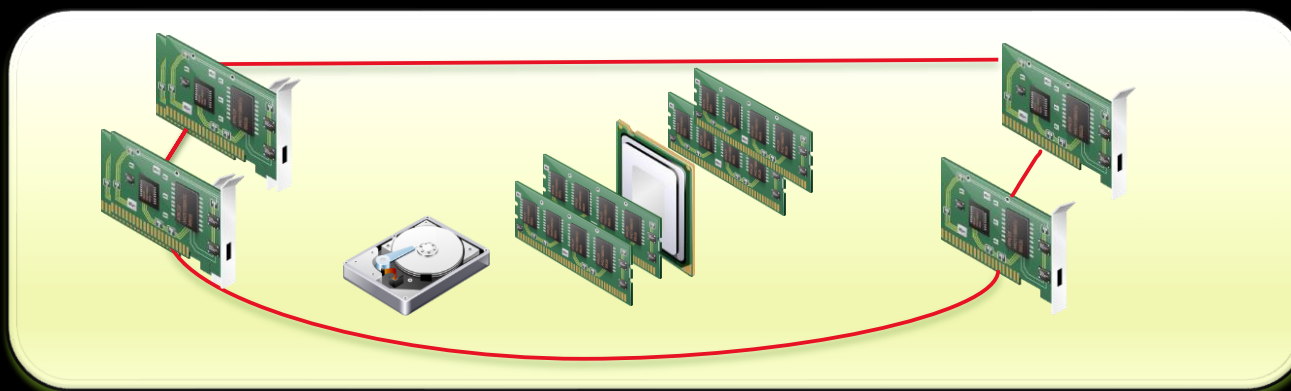
CRIU support for ROCm™

- ▲ CRIU modifications
 - ▲ 3 new plugin hooks in CRIU
 - ▲ Support for device file VMAs
 - ▲ Under review by CRIU community: <https://github.com/checkpoint-restore/criu/pull/1556>
- ▲ AMDGPU plugin (amdgpu_plugin.so)
 - ▲ Built as part of CRIU
 - ▲ Expected to be upstreamed to CRIU
 - ▲ Current WIP: <https://github.com/RadeonOpenCompute/criu>
- ▲ KFD ioctl APIs to support plugin
 - ▲ Expected to be upstreamed to Linux kernel
 - ▲ Under review by DRI and AMDGPU community
 - ▲ Current WIP: <https://github.com/RadeonOpenCompute/ROCK-Kernel-Driver/commits/fxkamd/criu-wip>



- *Topology and device cgroups*
- *Memory Allocations*
- *GPU virtual address mappings, including memory and doorbells*
- *HMM virtual address range attributes*
- *User mode queues*

Migration →



AMDGPU PLUGIN (amdgpu_plugin.so)

- ▲ Supports saving of GPU device files:
 - ▲ /dev/kfd
 - ▲ /dev/dri/renderD*
- ▲ Drains GPU state from KFD
 - ▲ Uses new KFD ioctls
 - ▲ Copies VRAM contents using SDMA engine
 - ▲ Supports multiple GPUs
 - ▲ Supports ROCm™ compute applications only
 - ▲ Does not support
 - ▲ Vulkan compute
 - ▲ OpenGL
 - ▲ Video decode/encode acceleration
- ▲ Saves GPU state to new protobuf image file (kfd-<id>.img)

AMDGPU PLUGIN (amdgpu_plugin.so)

Plugin Hook	Status	Description
DUMP_EXT_FILE	EXISTS	Saves all KFD state and VRAM contents for the process
RESTORE_EXT_FILE	EXISTS	Prepares all the KFD state and restores VRAM contents for the process
UPDATE_VMA_MAP	NEW	Updates restored memory mappings, file paths
RESUME_DEVICES_LATE	NEW	Restarts queues, MMU Notifiers, Restores SVM ranges, makes process ready to run in almost the last CRIU restore phase
HANDLE_DEVICE_VMA	NEW	Detect a suitable plugin to handle device file VMA with PF IO mappings

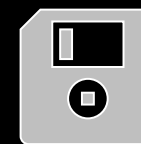
AMDKFD IOCTLs (amdgpu kernel driver)

- ▲ API definitions are currently under community review
- ▲ Authorization checks for ioctl calls affecting remote processes or restoring privileged state
 - ▲ CAP_SYS_ADMIN and PTRACE_ATTACHED flags

Name	Caller Context	Description
CRIU_PAUSE	CRIU (ptrace)	ioctl to make sure all queues are evicted
CRIU_PROCESS_INFO	CRIU (ptrace)	ioctl to determine current process information such as various objects and their types
CRIU_DUMPER	CRIU (ptrace)	ioctl to save BO metadata and other KFD state
CRIU_RESTORER	Target process	ioctl to restore the BOs and other KFD state
CRIU_RESUME	CRIU (remote)	ioctl registers MMU Notifiers, restores SVM ranges, restarts queues (late stage after VMAs locations are finalized by CRIU)

CR_PLUGIN_HOOK__HANDLE_DEVICE_VMA

- ▲ *int amdgpu_plugin_handle_device_vma(int fd, const struct stat *st_buf)*
- ▲ Assists CRIU to deal with device file VMAs
 - ▲ Registered device plugins identifies their VMAs
 - ▲ Treat VMAs with PF | IO flags as regular VMAs
 - ▲ CRIU just stores the metadata of such VMAs



Early Checkpoint Stage

CR_PLUGIN_HOOK__UPDATE_VMA_MAP

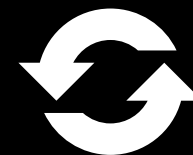
- ▲ *int amdgpu_plugin_update_vmamap(const char *old_path, char *new_path, const uint64_t addr, const uint64_t old_offset, uint64_t *new_offset)*
- ▲ Updates Memory mappings and Paths
 - ▲ CRIU pre maps all the VMAs
 - ▲ Saved mmap offsets and file paths may not be valid on restore
- ▲ Runs before PIC SYS_MMAP phase
 - ▲ VMAs with VMA_UNSUPP flags
 - ▲ Returns new mmap offsets and paths



Late Restore Stage

CR_PLUGIN_HOOK__RESUME_DEVICES_LATE

- ▲ *int amdgpu_plugin_resume_devices_late(int target_pid)*
- ▲ VMA locations are not yet finalized in restore phase
 - ▲ GPU MMU notifiers cannot be sanely setup at this stage
- ▲ Run in very late restore stage
 - ▲ CRIU is ptrace detached
 - ▲ PID needs to be validated in kernel mode
 - ▲ Last chance for all resuming tasks
- ▲ For each resuming task
 - ▲ AMD GPU driver (KFD)
 - ▲ Checks whether the PID is a valid kfd process
 - ▲ Extracts the resuming task's mm_struct
 - ▲ Restarts the queues and sets up MMU Notifiers
 - ▲ Restores the SVM ranges



Very Late Restore Stage

VRAM COPY METHODS

- ▲ PROCPIDMEM
 - ▲ Reliable but very slow
 - ▲ Always available
- ▲ MMAP on Large BAR GPUs
 - ▲ Faster than PROCPIDMEM
 - ▲ Not always available
- ▲ SDMA
 - ▲ Always available
 - ▲ Extremely fast
 - ▲ Depends on libdrm and libdrm_amdgpu

SECURITY CONCERNS

Threats:

- Read access to remote process state
- Control of remote process execution
- Write access to privileged HW state

Mitigation:

- Control of remote process execution
 - Requires ptrace attached caller during checkpoint
 - Requires CAP_SYS_ADMIN to resume execution of restored process
- Write access to privileged state requires CAP_SYS_ADMIN
- Read access to remote process requires ptrace attached caller

Open issues:

- How does CRIU ensure image integrity?
 - Do we need to sanitize privileged HW state?
- Is CAP_SYS_ADMIN a reasonable requirement long-term?

CHALLENGES

- ▲ CRIU ID caching and multiple independent processes
 - ▲ Not ideal for device files
- ▲ Protobuf cannot handle > 4G data size
 - ▲ Buffer Objects in VRAM are huge for Tensorflow and Pytorch
- ▲ Improve image size and speed up further
 - ▲ On a typical 4 GPU server, total VRAM image size ~32-128GB
 - ▲ Need to figure out dedup and incremental checkpoint



Demo

UPSTREAM STATUS

- ▲ Yes, It will be available upstream!
- ▲ Plugin and APIs are under review
- ▲ Finalizing work on HMM support and CRIU Image streamer
- ▲ Next Steps:
 - ▲ Work with the Linux kernel community to so they accept the new APIs
 - ▲ Get the CRIU Pull requests merged
 - ▲ Improve image size and speed up further



DISCLAIMER AND ATTRIBUTIONS

DISCLAIMER

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS.' AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

©2021 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, Radeon and combinations thereof are trademarks of Advanced Micro Devices, Inc. Linux is a trademark of Linus Torvalds and OpenCL is a trademark of Apple Inc. CRIU is an open-source project licensed under <https://github.com/checkpoint-restore/criu/blob/criu-dev/COPYING> GNU GPL v2. The titanium falcon picture used on slide #2 is licensed under GNU free documentation license <https://www.gnu.org/licenses/fdl-1.3.html> and is used to represent CRIU v3.15. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

