

# Alternative ways to extract information about processes

Monday, 20 September 2021 10:00 (25 minutes)

CRIU uses many different interfaces to get information about kernel resources, to extract sockets data `sock_diag` subsystem is used, for mounts/mount namespaces, `proctfs` per-pid mountinfo files are used, to get some file type-specific info we use `proctfs fdinfo` interface (which allows to get `mnt_id` from which file was opened, file flags and so on).

One of the most important and time-consuming stages in CRIU dump is getting process memory mappings information. Let's discuss that problem and approaches to optimize the performance of this stage. There was a prototype implementation of netlink-based interface to get information about a task [1]. We suggest to use eBPF iterators framework [2] to create CRIU-optimized interface to get task VMAs data.

Another interesting thing is mounts information acquisition. For simple cases mountinfo file seems sufficient. Previous year we introduced support of checkpoint-restoring nested containers. Main goal was to have ability to C/R OpenVZ containers with Docker containers inside. And here we met problem with overlayfs mounts. CRIU needs to get real overlayfs paths from the kernel (`mnt_id`+full path for each source directory) and these paths may be very long (like `PAGE_SIZE`). And this is the problem because of serious limitations which implied by mountinfo interface (limited size of lines, bad extendability). Some overlayfs-specific patches were proposed [3] earlier, but it's worth to have some universal approach to query mounts information for all file systems. There was a great subsystem called `fsinfo` [4] proposed by David Howells. But for some reasons it wasn't merged. There is idea to get some progress by creating some eBPF helpers which allows to get mounts information.

Thanks a lot to Andrei Vagin for advices and help.

Links:

[1] <https://github.com/avagin/linux-task-diag/commits/v5.8-task-diag>

[2] [https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/kernel/bpf/task\\_iter.c?h=v5.13#n472](https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/kernel/bpf/task_iter.c?h=v5.13#n472)

[3] overlayfs: C/R enhancements <https://lkml.org/lkml/2020/10/4/208>

[4] `fsinfo` <https://lwn.net/Articles/827934/>

## I agree to abide by the anti-harassment policy

I agree

**Primary authors:** MIKHALITSYN, Alexander (Virtuozzo); VAGIN, Andrei

**Presenters:** MIKHALITSYN, Alexander (Virtuozzo); VAGIN, Andrei

**Session Classification:** Containers and Checkpoint/Restore MC

**Track Classification:** Containers and Checkpoint/Restore MC