Contribution ID: **300** Type: **not specified**

CRIU: Reworking vDSO proxification, syscall restart

Tuesday, September 10, 2019 7:30 PM (20 minutes)

We have a number of unsolved time and vdso related issues in CRIU.

- Syscall restart: if a task Checkpoint interrupted a syscall, on restore CRIU blindely starts again the syscall (executing SYSCALL/SYSENTER/INT80/etc instruction with the original regset). It works OK-ish, but not with time blocking syscalls i.e., poll(), nanosleep(), futex() and etc. For this purpose, Glibc and vDSO use restart_syscall(). Which won't work in CRIU as kernel is not aware of interrupted syscall. To solve those issues I suggest to extend PTRACE_GET_SYSCALL_INFO with information from task_struct->restart_block. This way on restore criu will be able to adjust syscall arguments on application Restore.
- vDSO proxification. There is a chance that between Checkpoint and Restore events vDSO code may change. That may be in example, migration to another node or updating the kernel on the very same node. The old vDSO code can't be used anymore as vvar physical page can be missing [migration to an older kernel] or it may have different offsets. CRIU deals with that by mmaping old vdso code and patching entries with jumps to a new vdso. That's far from being perfect: the original application could have being Checkpointed while executing vdso code, but luckily we haven't got any reports about crashes on restore so far! Addressing this problem, we could add symbol table to vvar and got/plt tables to vdso, allowing CRIU to do linker job on restore by patching relocations on older vdso to newer vvar. The other approach would be making proxification process more correct: we could single-step application on Checkpoint from bytes that might be patched on Restore (JUMP_PATCH_SIZE). But additional trouble would be signals which may have being delivered while application was executing the very same bytes. That can be solved probably with hijacking SA_RESTORER..

I agree to abide by the anti-harassment policy

Yes

Primary authors: SAFONOV, Dmitry; VAGIN, Andrei

Presenters: SAFONOV, Dmitry; VAGIN, Andrei

Session Classification: Containers and Checkpoint/Restore MC