



Contribution ID: 246

Type: **not specified**

Time Namespaces

Tuesday, 13 November 2018 11:00 (30 minutes)

Discussions around time namespace are there for a long time. The first attempt to implement it was in 2006. From that time, the topic appears on and off in various discussions.

There are two main use cases for time namespace:

1. change date and time inside a container;
2. adjust clocks for a container restored from a checkpoint.

“It seems like this might be one of the last major obstacles keeping migration from being used in production systems, given that not all containers and connections can be migrated as long as a time dependency is capable of messing it up.” (by github.com/dav-ell)

The kernel provides access to several clocks: `CLOCK_REALTIME`, `CLOCK_MONOTONIC`, `CLOCK_BOOTTIME`. Last two clocks are monotonous, but the start points for them are not well defined (currently it is system start up time, but the POSIX says “since an unspecified point in the past”), and are different for each system. When a container is migrated from one node to another, all clocks are to be restored into consistent states; in other words, they have to continue running from the same points where they have been dumped.

We are going to present a patch set to support offsets for monotonic and boottime clocks. There are a few points to discuss:

- Clone flags exhaustion. Currently there is only one unused clone flag bit left, and it may be worth to use it to extend arguments of the clone system call.
- Realtime clock implementation details. Is having a simple offset enough? What to do when date and time is changed on the host? Is there a need to adjust vfs modification and creation times?

Any other features of time namespaces can be also discussed there.

Git: <https://github.com/0x7f454c46/linux/tree/wip/time-ns>

Wiki: https://criu.org/Time_namespace

I agree to abide by the anti-harassment policy

Presenters: VAGIN, Andrei; SAFONOV, Dmitry (Arista Networks)

Session Classification: Containers MC