Contribution ID: **258** Type: **not specified**

containers and mountinfo woes

Monday, 24 August 2020 10:20 (20 minutes)

This summarizes my (not-so-good) experience wrt using the kernel API exposed as /proc/*/mount{s,info} in various container projects (docker, runc, aufs, cri-o, cilium etc.), and outlines various problems with this API and its (ab)use.

Mountinfo API is quite adequate for 10s of mounts (systems with
no containers). With containers, each one adds a few mounts, and there might be thousands of containers – so we now have 10.000s of mounts, for which mountinfo is just not working any more.

The following issues are illustrated with examples from real code
and/or real bugs.

(1) Some major problems with the current mountinfo API are:

- it is slow (since there is no way to get information about a specific mount point, or a specific subset of mounts – it's all or nothing); in my experience, it takes up to 0.1s to read mountinfo on a loaded system;
- it is text-based (so everyone writes their own parser, and many of them are slow and/or incorrect);
- it is racy (there is a mount but it can't be found) – and this leads to actual bugs.

(2) In addition to the above issues, there are cases when
mountinfo is abused by userspace developers (most can be fixed). Those would not cause issues if mountinfo is fast – alas
currently it's not the case.

- checking if a mount(2) has succeeded (not needed at all);
- checking if a mount is already there before calling mount(2):
  - not needed in many cases;
  - can be done using two stat(2) syscalls – for real fs;
  - unavoidable with bind mounts;
- checking if a mount is there before calling umount(2) (not needed at all);
- checking if umount(2) succeeded (not needed);
- finding mount root of a specified directory (an alternative approach is to traverse the directory tree up calling stat(2) until dev is no longer matches);
- parsing mountinfo multiple times in a loop ((runc did it 50 to 100+ times for a simple runc run call);

(3) So, we are in a desperate need of a new API.

Here are the typical use cases:

- check if a directory is a mount point (including or excluding bind mounts);
- find all mounts under a given path;
- get some info about a particular mount (same as mountinfo currently provides, e.g. propagation flags or Root directory aka field 4);
- …

I agree to abide by the anti-harassment policy

I agree

Primary author: KOLYSHKIN, Kir (Red Hat)

Presenter: KOLYSHKIN, Kir (Red Hat)

Session Classification: Containers and Checkpoint/Restore MC

Track Classification: Containers and Checkpoint/Restore MC