containers and mountinfo woes

Linux Plumbers, 24 Aug 2020

Kir Kolyshkin
Principal Software Engineer, Red Hat

1. Problems with mountinfo kernel API

2. [Ab]use cases

3. Solutions

1. Text-based (and not simple)
   - some path characters are encoded
   - some fields are optional
   - those ^^ are in the middle (͡° ͜ʖ ͡°)

This is a problem of /proc in general

2. All or nothing approach
    - no way to say what we’re interested in
    - want to get info about 1 mount?
        go parse 100000 of them (͡° ͜ʖ ͡°)

3. Slow
    - 0.1s for reading is not unreal
    - not sure why (b2t, locking?)

4. Racy
    - uses kernel’s seq_file interface
    - in case the next mount to show is gone, everything after it gets lost
    - found when debugging customer issues with aufs (the fix is to re-read the file)
    - more details and repro at: https://github.com/kolyshkin/procfs-test

6 PROBLEMS WITH MOUNTINFO API cont’d

1. Those are only issues on a system with too many mounts.
2. Every container adds at least 2-4 mounts to the initial mount namespace (overlayfs, shared /dev/shm, nsfs).
3. Many containers (and thus mounts) are short-lived (see “Race” above).

1. Is the directory a mount point?

Can be answered by doing stat of directory, its parent, and comparing dev_t fields.

but it is not always working
for bind mounts (͡° ͜ʖ ͡°)

2. Find the mount point for a given file under

Can be solved by traversing up a tree while doing stat, until dev_t differs.

...but it is not always working for bind mounts!

3. Check if a dir is mounted – before mount()

- same as (1) above, but it’s not needed

4. Check if dir is mounted – before umount()

- same as (3) above, also not needed:
  do umount(), ignore EINVAL meaning
  “not mounted”
  … but EINVAL also means “bad flags” (͡° ͜ʖ ͡°)

5. Check if dir is mounted after failed umount()

- same as above, also not needed
  (if umount(2) failed, the mount is still there)

6. Recursive unmount of a directory

- little known fact:
  umount(2) with MNT_DETACH
  is already recursive...

- unless a dir is not a mount point (͡° ͜ʖ ͡°)

7. Get info about a particular mount

- mount propagation flags
- sb Root field (dind vs cgroup mounts case)

No way to get it without parsing mountinfo.

Example: runc

- 5 different mountinfo parsers in the code, optimized for different cases;
- all 5 had incorrect assumptions about number of optional fields;

Example: runc

```
# strace -f -e%file -oout \
> runc run -d ctid
# grep -c mountinfo out
116
```

- most are from cgroup v1 mounts; now fixed

Example: dockerd

- mount.Mount() was parsing mountinfo
  (fixed by https://github.com/moby/moby/pull/40656)
- mount.Unmount() was parsing mountinfo
  (before and, in case of an error, after)
  (fixed by https://github.com/moby/moby/pull/40637 etc.)
- used fmt.Sscanf() which is 8x slower than
  strings.Split and Atoi (fixed by PR #36091)

Userspace: fix it already

1. Do not use mountinfo unless abs necessary
2. Cache it if needed multiple times
3. Make sure your parser is correct and fast
    - No Go to fmt.Sscanf() and strings.Fields()
4. Use someone else’s parser:
    - https://github.com/moby/sys/mountinfo

Kernel: give us a good API already

1. fsinfo patches by David Howells
   https://lwn.net/Articles/827934/

2. task_diag by Andrey Vagin
   https://github.com/avagin/linux-task-diag

Do talk to me about your mountinfo woes

- kolyshkin (͡° ͜ʖ ͡°) gmail
- twitter.com/kolyshkin
- github.com/kolyshkin
- linkedin.com/in/kolyshkin/