Contribution ID: **164** Type: **not specified**

FS stacking with FUSE: performance issues and mitigations

Wednesday 22 September 2021 07:40 (20 minutes)

Stacking file systems based on FUSE are intended to go through complicated code paths implemented by the FUSE service, to enforce special access policies or manipulate data at runtime, based on what is the request received by the FUSE file system and the data in the lower file system.
Android relies on FUSE to enforce fine-grained access policies depending on file contents and requesting users, and may modify file contents at run-time.

These benefits come with the cost of an increased overhead to traverse the whole FUSE pipeline, worsened by the multiple switches between kernel-space and user-space. FUSE performance may result in less than 30% compared with direct access to the lower file system files.

FUSE passthrough is a first solution that has been proposed upstream to reduce this performance gap, allowing the FUSE service to provide some files with direct access to the lower file system, that would be internally rerouted by the FUSE driver. This solution is already available in a number of Android devices, but is still under discussion in the mailing list.

Another work-in-progress extension to FUSE passthrough is the extension of the FUSE driver with additional logic based on BPF, still managed by the FUSE service. This solution aims at extending the FUSE passthrough performance benefits also to file system operations and improving the FUSE driver flexibility and updatability with BPF programs without the need for modifying the kernel.

I agree to abide by the anti-harassment policy

I agree

Primary authors: BALSINI, Alessio (Google); LAWRENCE, Paul (Google)

Presenters: BALSINI, Alessio (Google); LAWRENCE, Paul (Google)

Session Classification: Android MC

Track Classification: Android MC