



Contribution ID: 41

Type: **not specified**

Optimizing sysfs and procfs

Monday, 13 November 2023 15:15 (45 minutes)

Problem - Sysfs and procfs are implemented as on-demand file systems. An on-demand file system maintains meta-data about the entries and creates inodes/dentries on access and keeps them till there is enough memory available in the system.

During system boot, it was observed that about 40% of sysfs and procfs entries were accessed. This means significant number of inodes/dentries have been allocated along with their meta-data. Once these inodes/dentries are allocated they remain in memory even if there is no immediate need for them.

This increases the memory requirement during system boot and impacts the systems with low memory (embedded/IOT) and virtualised environments (VMs competing for memory)

Solution - We propose that once the access to a sysfs/procfs file/folder is completed (after close() is called) the Kernel can delete the corresponding inode/dentry, this can be achieved by using DCACHE_DONTCACHE flag. This ensures memory is released sooner than a memory crunch situation and reduces the overall memory required during boot.

During POC it was observed there was slight impact on boot time because of same folder hierarchy was recreated on every access. In order to mitigate this, we have changed the behaviour so that only the inodes/dentries of files were removed upon close() call. This would ensure that memory footprint is conserved while not affecting the boot time.

Primary authors: KAHER, Ajay; Mr BRAHMAJOSYULA, Vamsi Krishna

Presenters: KAHER, Ajay; Mr BRAHMAJOSYULA, Vamsi Krishna

Session Classification: LPC Refereed Track

Track Classification: LPC Refereed Track