Implementing storaged

Android O adds support for `storaged`, an Android native daemon that collects and publishes storage metrics on Android devices.

- For daily diskstats, `storaged` periodically parses `/sys/block/mmcblk0/stat` (eMMC storage devices) or `/sys/block/sda/stat` (non-eMMC devices).
- For eMMC lifetime, `storaged` parses `/d/mmc0/mmc0:001/ext_csd` (if available).
- For application I/O blaming, `storaged` periodically traverses `/proc/uid_io/stats` and maintains parsed data, which includes data from all applications (not just running applications). `dumpsys` can call `storaged` to log the application I/O usage in a bug report.

Diskstat (including stalled diskstats) and eMMC information is logged to the Android event log, where a platform checkin service collects the logs.

`storaged` operations occur automatically and are handled entirely by the Android framework, so you don't need to do any implementation work. This page describes the design of `storaged` (including new interfaces) and how to use it to get I/O status from the kernel.

storaged design

For accounting and permission flexibility, `storaged` is implemented as a kernel module that returns per-uid I/O information (instead of using standard `proc/PID/io`). Raw I/O data for each I/O request continues to be stored and updated in kernel `task_struct`, and the kernel keeps track of when a process exits so it doesn't miss I/O usage that occurs from the last `storaged` polling event.

The module reads raw data and processes it only when the framework notifies it of a uid foreground/background switch or when the `storaged` daemon requests a report. At that time, the module exports a file node from kernel for communication with framework and `storaged` daemon.

`storaged` introduces the `/proc/uid_io/stats` interface, which returns a list of I/O stats for each UID in the system. The format is:

```
<uid>: <foreground read bytes> <foreground write bytes> <foreground read char
```

- read/write bytes are I/O events from a storage device.
- read/write chars (also in bytes) are data requested by read/write syscalls.

Getting I/O status from the kernel

To dump I/O usage from the kernel, use the `storaged` command with the `-u` option.

Command: `storaged -u`

Command output format: `name/uid fg_rchar fg_wchar fg_rbytes fg_wbytes bg_rchar bg_wchar bg_rbytes bg_wbytes fg_fsync bg_fsync`

Note: This output is similar to the output for **`proc/uid_io/stats`**. This is because **`storaged`** processes data from **`/proc/uid_io/stats`** and generates its own data.

Example output:

```
com.google.android.backuptransport  2269  60  0  0  1719845663  143912573  14
com.android.vending  2170  60  0  0  219904796  38693092  174436352  18944000
```