oomd2 and beyond: a year of improvements

Daniel Xu
Facebook
Overview

- Motivations & past development
- Present state
- Future plans
- Q&A
Motivations & past developments
Resource control @ FB

- Goal: resource isolation across applications
- Active area of development
- Use cases:
  - Protecting the workload
  - Side-loading batch workloads (eg transcoding)
- Deployed to several internal machine pools
- oomd steps in when kernel resource isolation breaks down
What is oomd?

- Out-of-memory killing in userspace
- Faster, more accurate
- Uses cgroup2, PSI, other system stats
- [https://github.com/facebookincubator/oomd](https://github.com/facebookincubator/oomd)
  - GPL2
Why oomd?

- Configuration not very intuitive (what's with all the numbers?)
  - `/proc/[pid]/oom_adj`
  - `/proc/[pid]/oom_score`
  - `/proc/[pid]/oom_score_adj`
  - `/proc/sys/vm/oom_kill_allocating_task`
  - `/proc/sys/vm/panic_on_oom`
- Slow to act; often it's already too late by the time the kernel reacts
- Tries to protect kernel health; user-space could be livelocked but the kernel could still be happily churning pages in and out
Why oomd? (cont.)

- Little context on logical composition of system
  - What should be killed together, what shouldn’t be, etc.
- No way to customize kill action (modulo OOM eventfd; still slow though)
  - For some processes, a SIGTERM/SIGKILL is fine. Other might want a song and dance
    - Eg. a persistent process that manages containers
- Non-deterministic (Or at least really hard to get deterministic)
oomd deployment

panic_on_oom rate before and after oomd rollout
Present State
oomd2

- Essentially a rule engine
- What we unsuccessfully tried
  - Monolithic config was not flexible enough
  - Plugin-only was too much work
- “Core plugins” was just right
oomd2

- Gotcha-free configurations
  - Of course, you can still make mistakes. But it should be clear they’re your mistakes
  - Plugins inherently encode domain knowledge
- Example: `swap_free` plugin
  - `/proc/swaps` and `/proc/meminfo` present slightly different information when draining swap
oomd2 config example
oomd2 config example (simplified)

```json
{
  "rulesets": [
    ..., 
    {
      "name": "user session protection",
      "detectors": [
        "<FIRE IF USER.SLICE, WORKLOAD.SLICE, or WWW.SLICE SLOWS BY OVER 60%>
        <FIRE IF SYSTEM.SLICE SLOWS BY OVER 80%>
      ],
      "actions": [
        "<KILL THE LARGEST MEMORY HOG ON THE SYSTEM>"
      ]
    },
    ..., 
  ]
}
```
Drop-in configurations

- Alters base configuration settings without having to modify base
- Useful for when containers can move on and off hosts
- Containers can carry around specialized oomd configuration
- Path not taken: in-container oomd
Lessons learned

- Most people (speaker included) are hazy on memory management internals
  - Thus it’s important that someone does it right and the work can be reused
- OOMing not a widely solved problem
- Lots of things can trigger an OOM
  - Understandable diagnostics are crucial
Future improvements

- epoll(2)-able pressure files
  - O(1) memory.stat access
- iocost
- systemd-oomd?
  - systemd is possibly in a good position to do sane autoconfiguration
Q&A