facebook
Resource Control @ FB

Tejun Heo
Software Engineer
10MB/s growth on both
10MB/s growth only on fbtax2
10MB/s growth only on fbtax2
oomd kills the memory bomb
oomd kills the memory bomb
oomd kills the memory bomb
base recovering after reboot
The Goal

Work-conserving full-OS resource isolation
What does that mean?

• Work-conserving:
  • Don’t keep machine idle if there’s work to do

• Full-OS:
  • Transparent
  • Keep doing what you’ve been doing and overlay isolation
  • No need for direct IO, hard-allocate mem, separate FS
The Challenges

Memory Control

• *memory.high* and *max* aren’t work-conserving
• Adding restrictions to already over-subscribed systems
  -> more brittle systems
• Kernel OOM killer doesn’t protect workload health
The Challenges

IO Control

- No good IO controller to use
- Accounting of FS metadata and swap IOs
The Challenges

Priority Inversions

- Filesystem operations (e.g. ext4)
- FS metadata, swap IO spikes lead to priority inversions
- `mmap_sem` and readahead
- Misc – squashfs, fuse...
The Solutions
memory.low and memory.min
Lift up, don’t push down

- Work-conserving best-effort protection
- More forgiving, allowing for ball-park configurations
- Proportional pressure (being worked on)
PSI – resource pressure metric

Who’s slowing us down?

*If I had more of this resource, I might have been able to run this percentage faster*

- On memory, IO and CPU
- System-wide and per-cgroup
- Reliable and intuitive understanding of workload health
- Used for resource allocation, load-shedding, oomd
Watch workload health with PSI, remEDIATE contentions
Workload QoS and context-aware decisions and actions
Helps kernel when resource isolation breaks down

oomd
The gentler and more perceptive grim reaper
io.latency

Completion latency based IO control

- Best-effort avg (or p90) completion latency guarantee
- More work-conserving
- Can be used both on hard disks and SSDs
- Supports do-first-pay-later for metadata and swap IOs
- Works on blk-mq
The Hunt for Priority Inversions

Kernel gotta be able to handle a part of system being really slow

• Switch to btrfs and fix priority inversions
• \textit{mmap\_sem}: readahead aborts, early breakout
• Shared IOs: do-first-pay-later, dirtier/allocator throttling
• Other misc config change and fixes
The Setup
fbtax2 - btrfs

No FS priority inversions, easier management

- Multi-100k machines running on btrfs (HDD and SSD)
- All priority inversions fixed, all metadata annotated
fbtax2 - swap
If there’s no swap, all anon memory is memlocked

- Better use of memory
- Allows memory pressure to build up gracefully
- Enabled everywhere except for the main workload
fbtax2 - cgroup

- hostcritical.slice  (mem.min=352M, io.latency=50ms)
 ources, sshd, systemd-journald, rsyslog
- workload.slice     (mem.low=17G, io.latency=50ms)
  workload-wdb.slice (mem.low=2.5G)
  workload-tw.slice  (mem.low=max)
- system.slice       (io.latency=75ms)
fbtax2 - oomd

- Kill a memory hog in *system* if
  - *workload* under moderate and *system* under high mempress
  - *system* under prolonged high mempress
- Kill an IO hog in *system* if
  - *workload* under moderate and *system* under high iopress
- Kill a swap hog from *system* or *workload-wdb* if
  - Swap is running out
The Results on HDDs
The Results on SSDs
The Possibilities & Todos

• We now have working full-OS resource isolation
• Batch workload side-loading
• Better thread-pool and resource consumption mgmt
• Upstreaming
• Proportion IO control for complex workload stacking
opensource.fb.com/linux