Part II: Contemporary software

What is the purpose of `systemd-oomd`?

`system-oomd` is a system service that monitors a processes memory usage in order prevent out of memory errors in kernel space. It does this by polling process groups (cgroup) data to detect when corrective action needs to occur. It was originally developed by Facebook

https://man7.org/linux/man-pages/man8/systemd-oomd.service.8.html#:~:text=systemd%2Doomd%20is%20a%20system,ManagedOOMMemoryPressure%3D%20to%20the%20appropriate%20value

---

How does `systemd-oomd` make its decisions?

The `system-oomd` deamon only will poll cgroups in which the out of memory deamon has been enabled. It will monitor these groups and kill them based on memory/swap pressure is above the defined limits, as configured in the `oomd.conf` configuration file. This daemon will only kill the processes which have been enabled as to avoid killing random, necessary processes over memory use.

The processes aer killed based on memory/swap pressure also known as the pressure stall information (PSI). This PSI tool was created by Facebook and is the first "canonical way to see resource pressure increases as they develop, with new pressure metrics for three major resources—memory, CPU, and IO." (https://facebookmicrosites.github.io/psi/docs/overview). This PSI metric allows you to detect real time resource shortages.

The PSI is determined by how long tasks are delayed because there is a lack of resources. This PSI is measured by monitoring three catagories of system pressure: CPU, I/O and memory.

First I will discuss CPU pressure. Within the `proc/pressure/cpu` file there are four fields, `avg10`, `avg60` and `avg300` and `total`. The `avg*` fields represent the percentage of time in the last 10, 60 and 300 seconds respectively that processes were starved of CPU. The `total` field represents the total time in microseconds that processes were starved for CPU.

Next is how memory is determined. Within the `proc/pressure/memory` file there are two lines: the `some` and `full` metrics. If a single task has to wait due to a lack of memory then its `some` score is increased. In other words the `some` tracks the percentage of the time that at least one process could be running if it weren't waiting for memory resources. "In particular, the time spent for swapping in, refaulting pages from the page cache, and performing direct reclaim is tracked in this way." However the system's `full` metric is increased when all tasks are delayed by a lack of resources. Therefore, the "`full` number indicates a loss of overall throughput – the total amount of work done decreases due to lack of resources." Again, put differently `full` tracks the time that no user process is able to use the CPU for actual work due to memory pressure. "If the `full` numbers are much above zero, it's clear that the system lacks the memory it needs to support the current workload."

The `proc/pressure/io` file is how IO pressure is determined. This file tracks the time lost waiting for I/O.

Finally `proc/pressure` "tracks the state of the system as a whole... [It] can be used to ensure that the resource limits for each cgroup make sense; they should also make it easier to determine which processes are thrashing on a busy system."

Please let this be enough information on how PSI is determined. If you would like me to get more technical I am happy to and would love to meet with you after the due date.

https://lwn.net/Articles/759781/

https://unixism.net/2019/08/linux-pressure-stall-information-psi-by-example/

https://facebookmicrosites.github.io/psi/docs/overview

https://www.phoronix.com/scan.php?page=news_item&px=Systemd-247-Lands-OOMD

https://www.phoronix.com/scan.php?page=news_item&px=systemd-247

https://fossbytes.com/new-systemd-247-is-out-for-linux-operating-system-as-major-release/

What information from the kernel does `systemd-oomd` use and how is this information gathered?

The information that `system-oomd` uses is the pressure stall information (PSI). This statistic was also developed by Facebook and is a quantification of "lost wall clock time due to resource shortages." This PSI information is exported through a file in `/proc/pressure/cpu`, `/proc/pressure/memory`, and `/proc/pressure/io`. Then using `poll()` responses can be added to trigger when these pressures get above their thresholds.

https://events19.linuxfoundation.org/wp-content/uploads/2017/12/oomd-A-Userspace-OOM-Killer-Daniel-Xu-Facebook.pdf

https://www.kernel.org/doc/html/latest/accounting/psi.html

In what context is `systemd-oomd` expected to be most beneficial?

`system-oomd` is deterministic, faster, and more flexible than kernel OOM killer. Allowing more flexibility in determining what processes and cgroups have what thresholds. This allows for better strategic pausing or killing of the low priority and restartable processes. Ultimately allowing for more optimized usage of memory.

Specifically, browsers are notorious for poor memory management. The article below mentions how this new `system-oomd` monitoring would allow enforcement of browsers to reduce their memory hogging without the need to randomly kill processes; allowing for a much more elegant and performant solution to the browser problem.

https://news.ycombinator.com/item?id=26450960

Bonus funny article I came across: airline metaphors for kernel OOM killing:

https://lwn.net/Articles/104185/