Resumen Multi-Level Feedback Queue - SistOp

Lautaro Bachmann

Contents

Scheduling: The Multi-Level Feedback Queue

Definitions

starvation:

if there are "too many" interactive jobs in the system, they will combine to consume all CPU time, and thus long-running jobs will never receive any CPU time (they starve).

game the scheduler.

Gaming the scheduler generally refers to the idea of doing something sneaky to trick the scheduler into giving you more than your fair share of the resource.

voo-doo constants,

Constants that seem to require some form of black magic to set them correctly.

TIP: USE ADVICE WHERE POSSIBLE

it is often useful to provide interfaces to allow users or administrators to provide some hints to the OS.

We often call such hints **advice**, as the OS need not necessarily pay attention to it, but rather might take the advice into account in order to make a better decision.

MLFQ: Summary

why it is called that:

it has multiple levels of queues, and uses feedback to determine the priority of a given job.

History is its guide:

Pays attention to how jobs behave over time and treat them accordingly.

refined set of MLFQ rules,

Rule 1:
If Priority(A) > Priority(B), A runs (B doesn't).

Rule 2:
If Priority(A) = Priority(B), A & B run in round-robin fashion using the time slice (quantum length) of the given queue.

Rule 3:
When a job enters the system, it is placed at the highest priority (the topmost queue).

Rule 4:
Once a job uses up its time allotment at a given level (regardless of how many times it has given up the CPU), its priority is reduced (i.e., it moves down one queue).

Rule 5:
After some time period S, move all the jobs in the system to the topmost queue.

MLFQ usefulness

it manages to achieve the best of both worlds: it can deliver excellent overall performance for short-running interactive jobs, and is fair and makes progress for long-running CPU-intensive workloads.