

Планировщик процессов для гетерогенных процессорных архитектур

Шаго Павел Евгеньевич

Научный руководитель: Корхов Владимир Владиславович

Факультет прикладной математики – процессов управления
Кафедра моделирования электромеханических и компьютерных систем

8 декабря 2025 г.

Цель и задачи

Цель

Разработать более эффективный планировщик процессов для гетерогенного процессора (aka Hybrid CPU, big.LITTLE)

Задачи

- Переписать EEVDF на sched_ext

Тут будет обоснование почему это не очень хорошо для ARL, как это можно было бы поправить с точки зрения алгоритма



Введение

- Алгоритм построен на концепции сведения параметров задачи в одну функцию – виртуальное время.
- Решение о планировании принимается так: выбирается процесс у которого есть доступный запрос с самым ранним виртуальным дедлайном.
- Главный результат: гарантия справедливости алгоритма EEVDF, задержка строго ограничена размером временного кванта q .
- Показано, что ограничение является оптимальным для алгоритмов пропорционального распределения и улучшает все предыдущие ограничения.

Earliest Eligible Virtual Deadline First : A Flexible and Accurate Mechanism for Proportional Share Resource Allocation

Ion Stoica, Hussein Abdel-Wahab
Old Dominion University 1996

- Процессы конкурируют за общий разделяющийся по времени ресурс (CPU).
- Ресурс распределяется во временных квантах не больше чем q .
- В начале каждого кванта времени для использования ресурса выбирается процесс.
- После того как процесс захватил ресурс он может использовать его весь квант времени либо освободить до окончания кванта времени.
- Каждому процессу назначается вес w_i определяющий относительную долю времени которая ему положена.
- Только процессы могут порождать временные кванты.

1. Доля клиента (Client Share, $f_i(t)$) Доля ресурса, которую клиент i должен получить в момент времени t , определяется отношением его веса (w_i) к сумме весов всех активных клиентов ($A(t)$):

$$f_i(t) = \frac{w_i}{\sum_{j \in A(t)} w_j}$$

Тут будут результаты работы

Тут будет короткое напоминание что такое eBPF и sched_ext

Сведения о реализации, сравнительный график EEVDF (sched_ext) vs EEVDF

Спасибо за внимание, вопросы?

Репозиторий с реализацией

 <https://github.com/shgpavel/A1349>

Репозиторий с другими scx планировщиками

 <https://github.com/sched-ext/scx>

Дополнительный слайд. Обозначения

n – количество процессов

n_a – количество активных процессов

w_i – вес i -го процесса

q – квант времени

t – количество выделенных временных квантов

r – запрошенное процессом время на исполнение

r_{max} – максимально возможное время на исполнение

$r_i^{(k)}$ – продолжительность исполнения k -го запроса i -го процесса

$u_i^{(k)}$ – время которое фактически получает процесс i на k -ом запросе

t – момент в реальном времени

t_a – момент реального времени когда процесс i становится активным

t^- – время прямо перед наступлением события

t^+ – время сразу после наступления события

$A(t)$ – множество всех активных в момент t процессов

$W(t)$ – сумма весов всех активных процессов

Дополнительный слайд. Обозначения

$f_i(t)$ – доля процесса i во время t

$S_i(t_0, t_1)$ – время обслуживания которое должен получить процесс i в идеальной системе

$s_i(t_a, t)$ – время обслуживания которое клиент i на самом деле получает

$V(t)$ – виртуальное время системы

e – реальное подходящее время времени запроса

d – реальный дедлайн запроса

B – множество активных процессов дедлайн которых в $[e, d]$

C – множество активных процессов дедлайн которых больше d

$ve_i^{(k)}$ – виртуальное подходящее время k -го запроса i -го процесса

$vd_i^{(k)}$ – виртуальный дедлайн k -го запроса i -го процесса

$lag_i(t)$ – задержка обслуживания процесса i в момент времени t , $(S - s)$

d_{neg} – наибольший дедлайн среди всех процессов с отрицательными lag которые активны в момент времени t_1