

Hey, psst, try this.

The underground culture around custom CPU schedulers.

Masahito Suzuki <firelzrd@gmail.com>

Alfred Chen <cchalp@alpha@gmail.com>

Petr Jung <ptr1337@cachyos.org>

Piotr Górski <piotrgorski@cachyos.org>

Giovanni Gherdovich <ggherdovich@suse.cz>





Burst-Oriented Response Enhancer

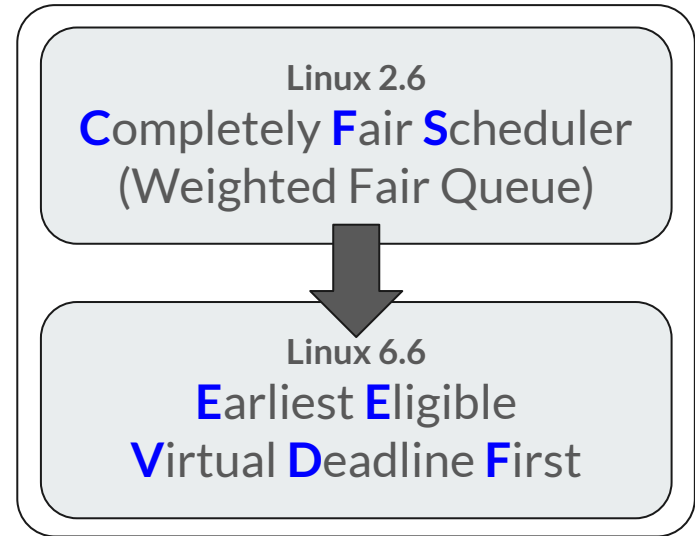
CPU Scheduler Modification

Masahito Suzuki <frelzrd@gmail.com>



Linux mainline scheduler is great

- Sophisticated
- Straightforward
- High performance
- Continuously improved



BORE supports CFS and EEVDF

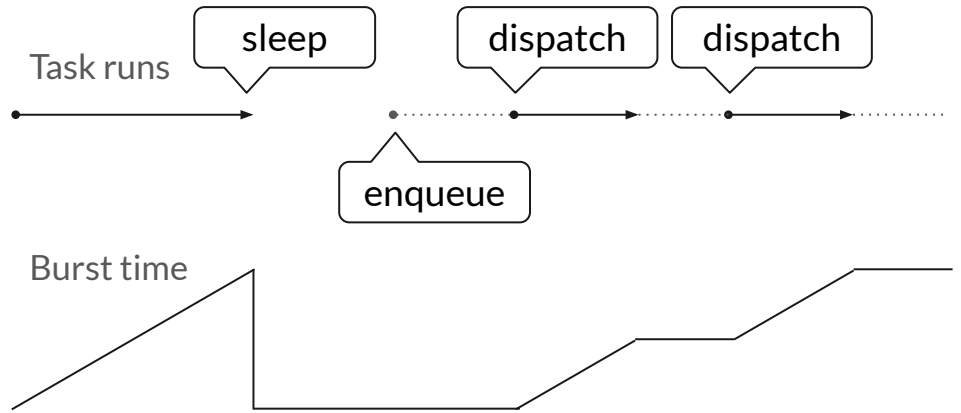


Penalization of greedy tasks

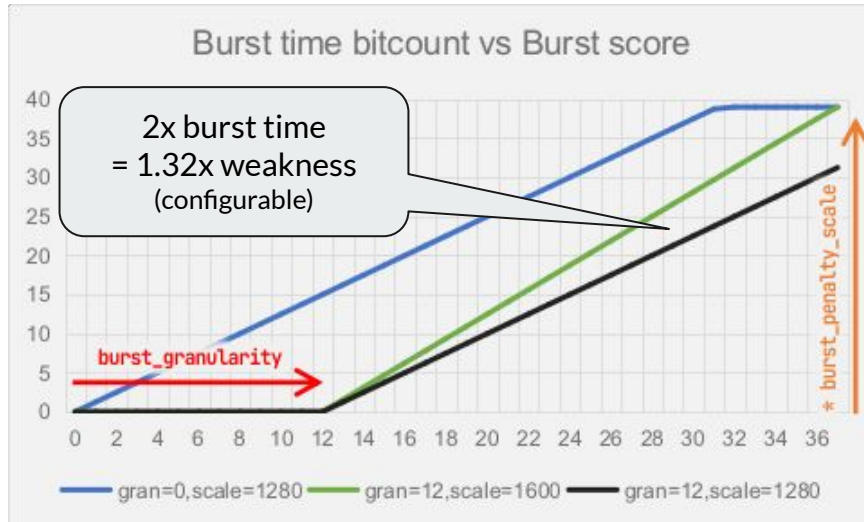


Burst Time

- Task's continuous CPU occupancy time is tracked.
- Burst time is reset by **sleep**, **I/O wait**, or **yield**.



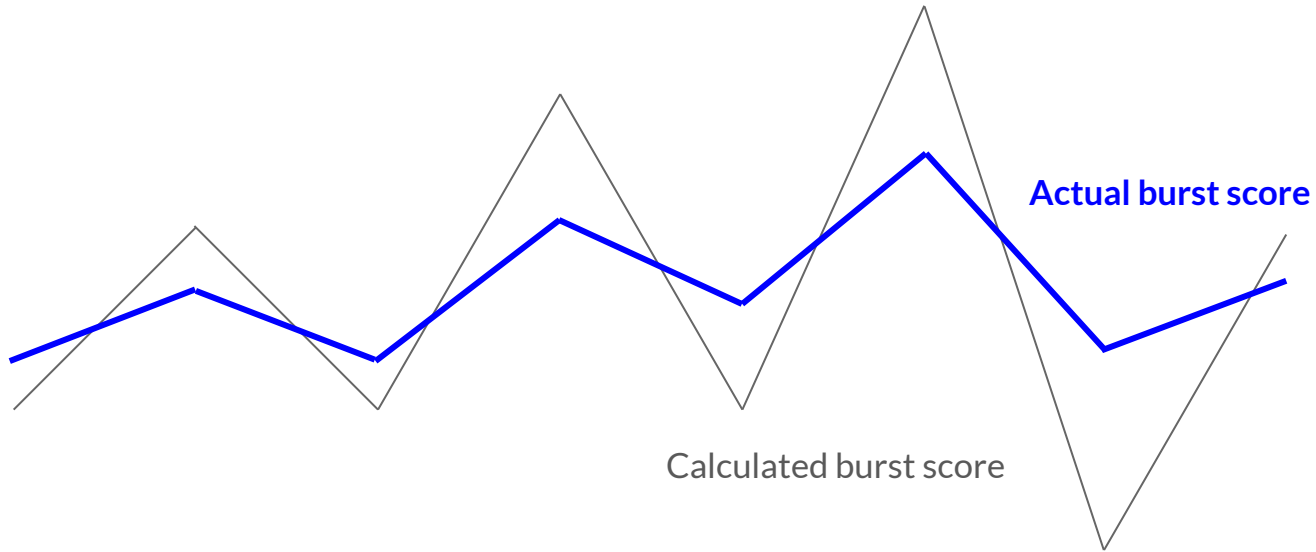
Burst Score



- Burst score grows by logarithmic scale
- Tasks are penalized based on per-task burst scores
- Greedy tasks get less chance and less time



Burst Score Smoothing





Demo



Project C - Brief Introduction

Project C

The Project C community started in 2014 and slowly grew from a personal development blog to an open source community on Gitlab. The community is committed to developing and maintaining alternative CPU scheduling algorithms for the Linux kernel.

Project C, C stands for “cross”, and the combination of two CPU schedulers in the project. Now, based on the latest Linux kernel branch, two CPU scheduling algorithms, **BMQ** (Bitmap Queue) and **PDS** (Priority and Deadline based Scheduler), are provided in one single Linux kernel patch. Last two LTS Linux kernels are also supported.

The Project C has participated in the OSPP (<https://summer-ospp.ac.cn/>) event since 2021.

Gitlab kernel repository: <https://gitlab.com/alfredchen/linux-prj>



BMQ & PDS Schedulers

BMQ (Bitmap Queue)

BMQ is a bitmap queue based simple scheduler Linux implementation, inspired by the CPU scheduling algorithm in the zircon kernel of the Google Fuchsia system.

- Bitmap for double-linked queue available state
- **Boost priority**

PDS (Priority and Deadline based Scheduler)

PDS is an efficient CPU scheduling algorithm deeply evolved from the BFS algorithm. Used to be skip-list based, now it's bitmap queue based.

- Bitmap for double-linked queue available state
- **Moving time slots** for deadline, mapping to bitmap queue

Common features of BMQ & PDS

In Project C, BMQ and PDS shares common scheduler codes which are improvement vs linux mainline scheduler code

- Task wake up
- CPU task selection
- CPU load balance
- CPU number Scalability

