Deploying and Managing sched_ext Schedulers in CachyOS

Peter Jung <ptr1337@cachyos.org>
Piotr Gorski <piotrgorski@cachyos.org>

Linux plumbers Conference, Sept 18-20,2024

CachyOS @ Plumbers

who • what • why

CachyOS

WHO ARE WE

Peter Jung

packages • tests • infra • community
day job: test engineer, automotive

Vladislav Nepogodin

software development
day job: university student, CS

Piotr Górski

kernel backports & maintenance
day job: university lecturer, CS

WHAT WE DO

- Started in **May 2021**
- **Arch Linux** derivative
- Focus on **performance** and **user experience**
- Ship **out of tree schedulers** for years
  - BORE, CacULE, BMQ, TT and more
- Adopted **sched_ext** in late 2023

WHY ARE WE HERE

- Share **field experience**
- **sched_ext integration**
- Make CachyOS **even better**

CachyOS project

CachyOS

Intuitive

- User friendly (installation, configuration, …)
- Performance optimization
- Implementing new features early in testing repository

Community

- ~3,100 members on Discord, ~700 Telegram, ~850 Forum, ~1800 Reddit
- Very active community of users and enthusiasts
- ~175 Terabytes traffic per month
- Servers around the world: CDN77, Cloudflare, Tebi

CachyOS project

Optimizations and tools

- Cachybuilder - archlinux package compilation
- Leveraging modern hardware capabilities
- Balancing optimization with user-friendliness and stability
- Package optimization with PGO, BOLT and others
- Integration of out of tree schedulers (BORE, TT, Echo)
- CachyOS hardware detection (chwd)

sched_ext: system integration

select • start • stop schedulers:

- with systemd
- with GUI

select • start • stop schedulers

manually from command line

CachyOS

- scx_bpfland
- scx_central
- scx_lavd
- scx_layered
- scx_nest
- scx_qmap
- scx_rlfifo
- scx_rustland
- scx_rusty
- scx_simple
- scx_userland

```
# scx_rusty

07:03:57 [INFO] Running scx_rusty (build ID: 1.0.2-gfd6aa...)
07:03:57 [INFO] ---
07:03:57 [INFO] NUMA[00] mask= 0b1111111111111111
07:03:57 [INFO]   DOM[00] mask= 0b1111111111111111
07:03:57 [WARN] libbpf: map 'rusty': BPF skeleton version is old, ...
07:03:57 [INFO] Rusty scheduler started!
07:04:00 [INFO] Counters:
07:04:00 [INFO]   dispatched_tasks_total: 442 [147.3/s]
07:04:00 [INFO]   prev_idle: 332 (75.1%) [110.7/s]
07:04:00 [INFO]   wsync_prev_idle: 42 (9.5%) [14.0/s]
07:04:00 [INFO]   direct_dispatch: 37 (8.4%) [12.3/s]
07:04:00 [INFO]   dsq: 31 (7.0%) [10.3/s]
07:04:00 [INFO]   wsync: 0 (0.0%) [0.0/s]
07:04:00 [INFO]   greedy_xnuma: 0 (0.0%) [0.0/s]
07:04:00 [INFO]   direct_greedy_far: 0 (0.0%) [0.0/s]
07:04:00 [INFO]   direct_greedy: 0 (0.0%) [0.0/s]
07:04:00 [INFO]   greedy_idle: 0 (0.0%) [0.0/s]
07:04:00 [INFO]   greedy_local: 0 (0.0%) [0.0/s]
07:04:00 [INFO]   pinned: 0 (0.0%) [0.0/s]
...
```

select • start • stop schedulers

manually from command line

CachyOS

- scx_bpfland
- scx_central
- scx_lavd
- scx_layered
- scx_nest
- scx_qmap
- scx_rlfifo
- scx_rustland
- scx_rusty
- scx_simple
- scx_userland

```
# scx_rusty

07:03:57 [INFO] Running scx_rusty (build ID: 1.0.2-gfd6aa...)
07:03:57 [INFO] ---
07:03:57 [INFO] NUMA[00] mask= 0b1111111111111111
07:03:57 [INFO]   DOM[00] mask= 0b1111111111111111
07:03:57 [WARN] libbpf: map 'rusty': BPF skeleton version is old, ...
07:03:57 [INFO] Rusty scheduler started!
07:04:00 [INFO] Counters:
07:04:00 [INFO]   dispatched_tasks_total: 442 [14
07:04:00 [INFO]   prev_idle: 332 (75.1%) [110.7/
07:04:00 [INFO]   wsync_prev_idle: 42 (9.5%) [1
07:04:00 [INFO]   direct_dispatch: 37 (8.4%) [12
07:04:00 [INFO]   dsq: 31 (7.0%) [10.3/s]
07:04:00 [INFO]   wsync: 0 (0.0%) [0.0/s]
07:04:00 [INFO]   greedy_xnuma: 0 (0.0%) [0.0/s]
07:04:00 [INFO]   direct_greedy_far: 0 (0.0%) [0.0/s]
07:04:00 [INFO]   direct_greedy: 0 (0.0%) [0.0/s]
07:04:00 [INFO]   greedy_idle: 0 (0.0%) [0.0/s]
07:04:00 [INFO]   greedy_local: 0 (0.0%) [0.0/s]
07:04:00 [INFO]   pinned: 0 (0.0%) [0.0/s]
...
```

not starting at boot

logs not stored

select • start • stop schedulers

with systemd: one service per scheduler?

- scx_bpfland
- scx_central
- scx_lavd
- scx_layered
- scx_nest
- scx_qmap
- scx_rlfifo
- scx_rustland
- scx_rusty
- scx_simple
- scx_userland

scx_rusty.service

```
...
[Service]
ExecStart=scx_rusty
...
```

scx_bpfland.service

```
...
[Service]
ExecStart=scx_bpfland
...
```

scx_lavd.service

```
...
[Service]
ExecStart=scx_lavd
...
```

```
# systemctl start scx_rusty
...
# systemctl stop scx_rusty
...

# systemctl start scx_bpfland
...
# systemctl stop scx_bpfland
...

# systemctl start scx_lavd
...
# systemctl stop scx_lavd
...
```

select • start • stop schedulers

with systemd: one service per scheduler?

CachyOS

- scx_bpfland
- scx_central
- scx_lavd
- scx_layered
- scx_nest
- scx_qmap
- scx_rlfifo
- scx_rustland
- scx_rusty
- scx_simple
- scx_userland

scx_rusty.service

```
...
[Service]
ExecStart=scx_rusty
...
```

scx_bpfland.service

```
...
[Service]
ExecStart=scx_bpfland
...
```

scx_lavd.service

```
...
[Service]
ExecStart=scx_lavd
...
```

✅ can start at boot
✅ log management
❌ option flags
❌ convenience

```
# systemctl start scx_rusty
...
x_rusty

cx_bpfland

x_bpfland


# systemctl start scx_lavd
...
# systemctl stop scx_lavd
...
```

select • start • stop schedulers

with systemd: one single, parametric service

CachyOS

scx.service

```
[Unit]
ConditionPathIsDirectory=/sys/kernel/sched_ext
...
[Service]
EnvironmentFile=/etc/default/scx
ExecStart=/bin/bash -c 'exec ↩
↪  ${SCX_SCHEDULER_OVERRIDE:-$SCX_SCHEDULER} ↩
↪  ${SCX_FLAGS_OVERRIDE:-$SCX_FLAGS}'
LogNamespace=sched-ext
...
```

/etc/default/scx (environment file)

```
SCX_SCHEDULER=scx_bpfland
SCX_FLAGS=--lowlatency
```

Unit file contributed to github.com/sched-ext/scx • thanks to Pietro Righi for improvements

start the service (eg. at boot)

```
# systemctl start scx
```

which scheduler? scx_bpfland, our default

```
# systemctl status scx
(scx_bpfland)
```

replace the scheduler

```
# systemctl set-environment SCX_SCHEDULER_OVERRIDE=↩
  ↪scx_lavd
```

restart the service, for the change to take effect

```
# systemctl restart scx
```

now on scx_lavd

```
# systemctl status scx
(scx_lavd)
```

restore default scx scheduler (ie scx_bpfland)

```
# systemctl unset-environment SCX_SCHEDULER_OVERRIDE
```

restart after changing settings

```
# systemctl restart scx
```

yup, back to scx_bpfland

```
# systemctl status scx
(scx_bpfland)
```

select • start • stop schedulers

with systemd: separate log namespace

CachyOS

scx.service

```
[Unit]
ConditionPathIsDirectory=/sys/kernel/sched_ext
...
[Service]
EnvironmentFile=/etc/default/scx
ExecStart=/bin/bash -c 'exec ↩
↪  ${SCX_SCHEDULER_OVERRIDE:-$SCX_SCHEDULER} ↩
↪  ${SCX_FLAGS_OVERRIDE:-$SCX_FLAGS}'
LogNamespace=sched-ext
...
```

journald@sched-ext.conf

```
SystemMaxUse=50M
MaxLevelStore=info
```

```
# journalctl --namespace sched-ext
...
07:03:57 [INFO] ---
07:03:57 [INFO] NUMA[00] mask= 0b1111111111111111
07:03:57 [INFO]   DOM[00] mask= 0b1111111111111111
07:03:57 [WARN] libbpf: map 'rusty': ...
07:03:57 [INFO] Rusty scheduler started!
07:04:00 [INFO] Counters:
07:04:00 [INFO]   dispatched_tasks_total: 442 [147.3/s]
07:04:00 [INFO]   prev_idle: 332 (75.1%) [110.7/s]
07:04:00 [INFO]   wsync_prev_idle: 42 (9.5%) [14.0/s]
07:04:00 [INFO]   direct_dispatch: 37 (8.4%) [12.3/s]
07:04:00 [INFO]   dsq: 31 (7.0%) [10.3/s]
07:04:00 [INFO]   wsync: 0 (0.0%) [0.0/s]
07:04:00 [INFO]   greedy_xnuma: 0 (0.0%) [0.0/s]
07:04:00 [INFO]   direct_greedy_far: 0 (0.0%) [0.0/s]
07:04:00 [INFO]   direct_greedy: 0 (0.0%) [0.0/s]
07:04:00 [INFO]   greedy_idle: 0 (0.0%) [0.0/s]
07:04:00 [INFO]   greedy_local: 0 (0.0%) [0.0/s]
07:04:00 [INFO]   pinned: 0 (0.0%) [0.0/s]
...
```

new statistics library in scx schedulers

since v1.0.4 • default log namespace • statistics on demand

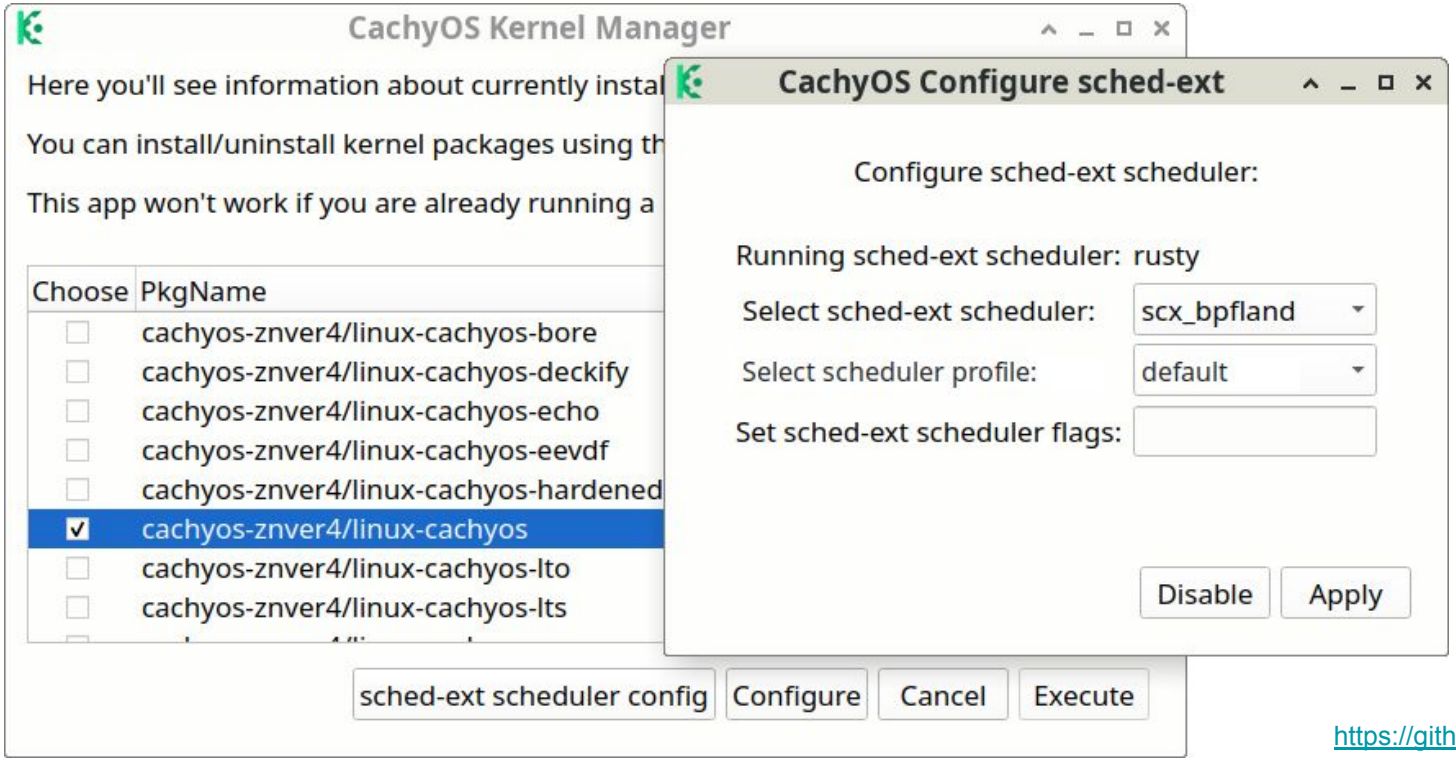

```
# scx_bpfland
... nothing!
```

```
# scx_bpfland --monitor $INTERVAL
...
[scx_bpfland] tasks -> run:  4/4  int: 3  wait: 2 | nvcsw: 6 | dispatch -> dir: 0    ...
[scx_bpfland] tasks -> run:  2/4  int: 0  wait: 0 | nvcsw: 3 | dispatch -> dir: 1269 ...
[scx_bpfland] tasks -> run:  2/4  int: 0  wait: 0 | nvcsw: 8 | dispatch -> dir: 236  ...
[scx_bpfland] tasks -> run:  2/4  int: 0  wait: 0 | nvcsw: 6 | dispatch -> dir: 227  ...
[scx_bpfland] tasks -> run:  3/4  int: 0  wait: 0 | nvcsw: 8 | dispatch -> dir: 243  ...
[scx_bpfland] tasks -> run:  1/4  int: 0  wait: 0 | nvcsw: 3 | dispatch -> dir: 239  ...
[scx_bpfland] tasks -> run:  2/4  int: 0  wait: 0 | nvcsw: 7 | dispatch -> dir: 242  ...
...
```

```
# journalctl --unit scx.service --boot 0
...
```

CachyOS kernel manager GUI

```
# cachyos-kernel-manager
```



https://github.com/CachyOS/kernel-manager

Improvements of sched-ext

bpfland • lavd

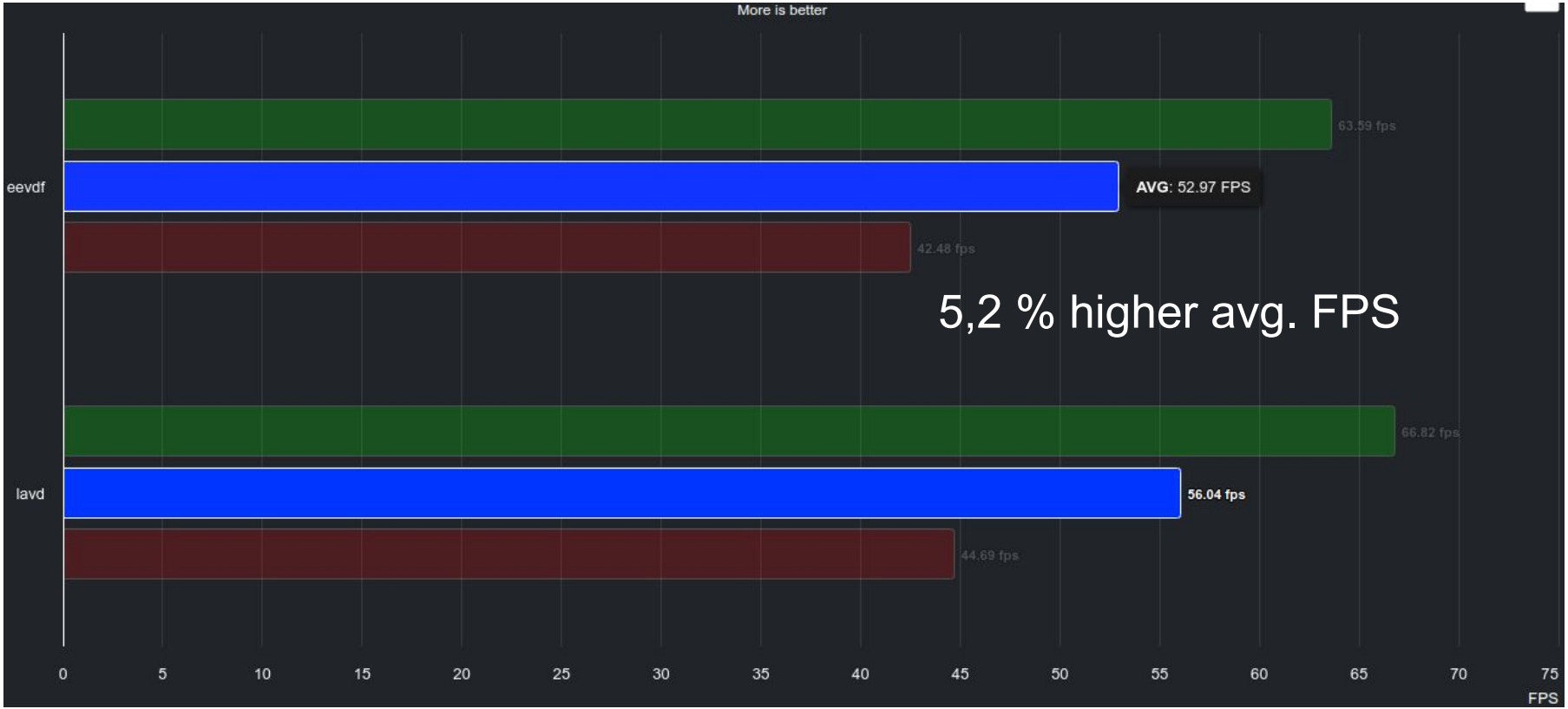
CachyOS

1. Bpfland
   - Demonstrated improvements in interactivity
   - Excels under heavy load scenarios
   - Provides substantial FPS stability in gaming workloads
   - Hybrid CPU Support (Cache, E/P Cores)
   - Predefined profiles (lowlatency, gaming, powersave)
2. LAVD (Latency-criticality Aware Virtual Deadline)
   - Focused on optimizing gaming performance
   - Notable improvements in 1% lows and average FPS compared to the default Linux scheduler
   - Particularly effective for handheld devices like the Steam Deck
   - Hybrid CPU Support
   - Rolled out for the CachyOS Handheld Edition
   - “Autopilot” for different modes (Performance, Balanced, Powersave)

LAVD benchmark

steam deck - Baldur's Gate 3

CachyOS



scx_loader - scheduler manager

Why • Features

CachyOS

Why

- Integration with common tools is not working (power-profiles-daemon)
- Launching sched-ext requires “sudo” permissions
- Dynamic Scheduler Usage depending on the workload/profile

Features

- Start | Stop | Current | Mode | Supported Scheduler
- Started via systemd service and waiting for signals
- Signals via dbus

QUESTIONS?

FEEL FREE TO ASK

CACHYOS!
BLAZINGLY FAST

Credits

- The CachyOS Community
- Vladislav Nepogodin
- Giovanni Gherdovich
- Andrea Righi
- Eric Naim
- Matias (Aarrayy)
- Harsh Peshwani
- Mashaito
- Luca (Nextworks) Paglia
- Szymon Mytych
- WSEI Linux Users Group