FOSDEM

Level up your Linux gaming

[How sched_ext can save your fps]

Andrea Righi <arighi@nvidia.com>

NVIDIA

Agenda

- Gaming on Linux
- Linux scheduling
- sched_ext + gaming
- Conclusion

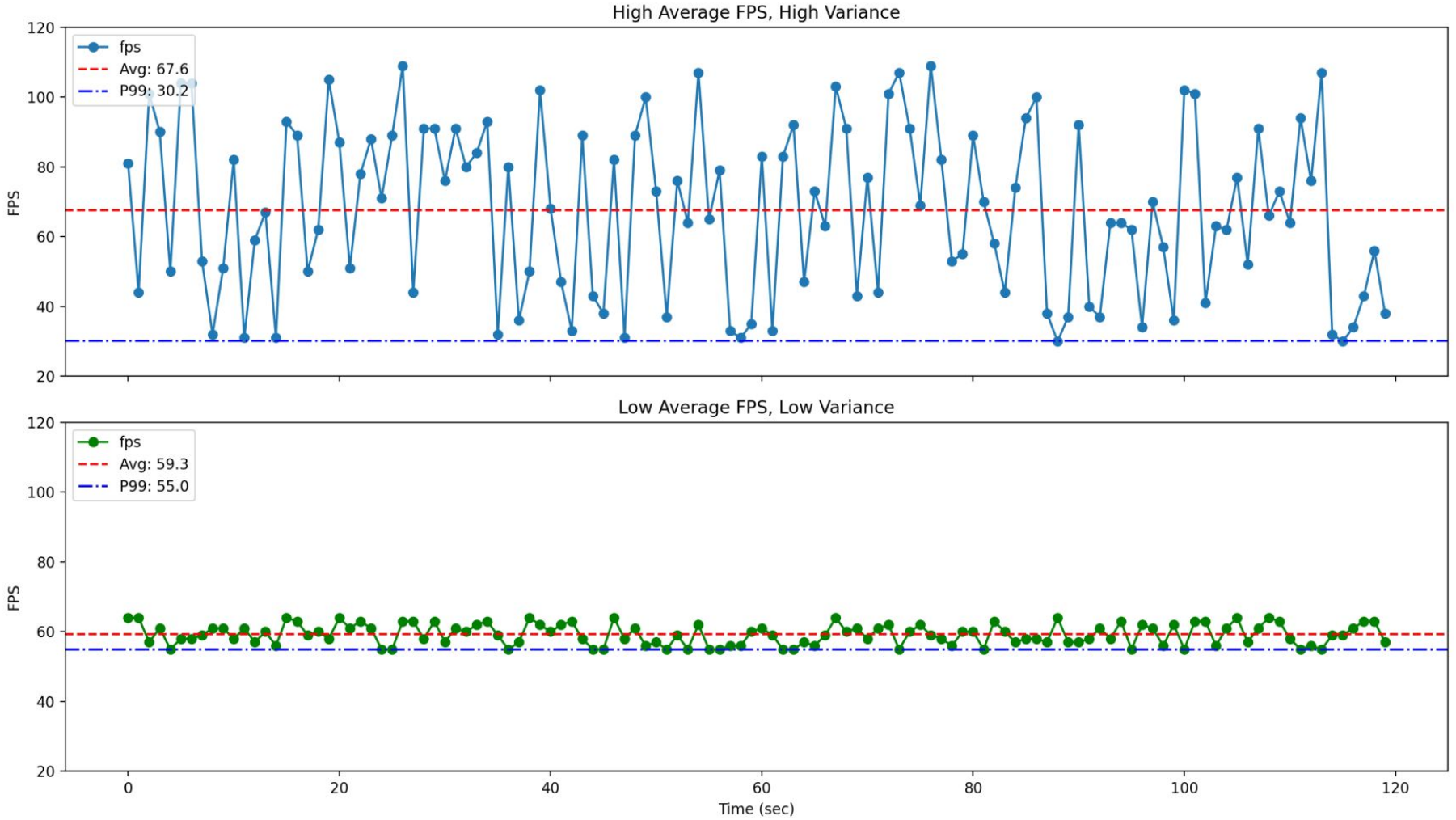
Gaming on Linux

Gaming on Linux is serious business

- Linux has become a viable gaming platform
  - SteamOS / SteamDeck
  - Vulkan API
  - Proton
  - DXVK
- High compatibility with AAA games
- Improved Linux GPU drivers: NVIDIA / AMD / Intel

Gaming performance

- Frames per second (fps)
  - Primary metric for gaming performance
- Ideal fps for smooth gameplay
  - 30 fps: acceptable
  - 60 fps: fluid gaming experience
  - 120 fps: competitive gaming


High Average FPS, High Variance
fps
Avg: 67.6
P99: 30.2
FPS
120
100
80
60
40
20
Low Average FPS, Low Variance
fps
Avg: 59.3
P99: 55.0
FPS
120
100
80
60
40
20
0
20
40
60
80
100
120
Time (sec)

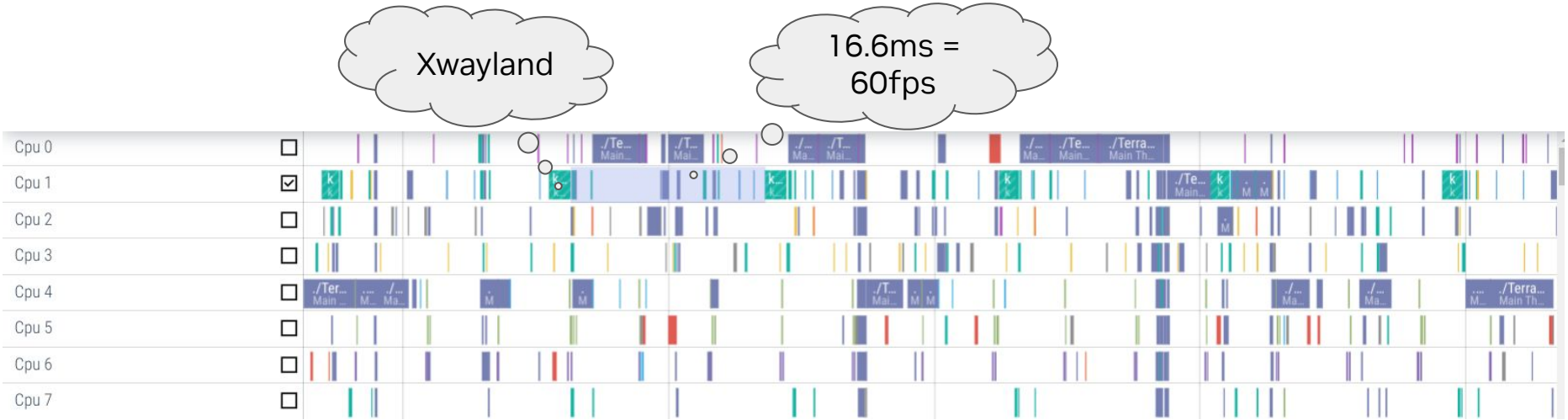
Throughput vs consistency

- Throughput
  - Average fps
- Consistency
  - P99 fps (slowest 1% fps)
- Prioritize low variance for a better gaming experience

Why scheduling is so important?

- Most games are GPU intensive, however...
- The CPU is always the manager
  - Game engine (logic and AI)
  - State management
  - Audio processing
- CPUs are feeding data to the GPUs

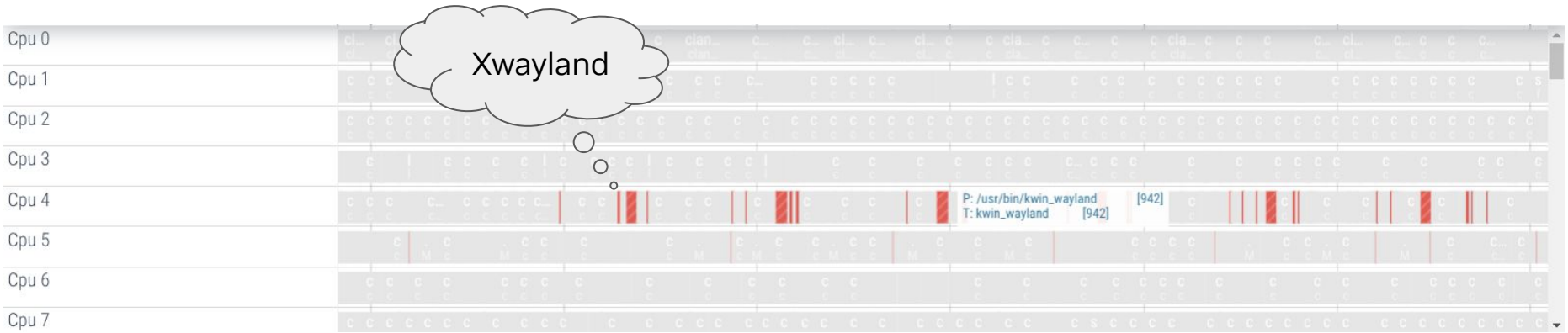
Gaming workload (normal condition)



Xwayland runs consistently every 16.6ms

https://perfetto.dev

Gaming workload (overloaded system)



Xwayland execution is inconsistent, missing the 16.6ms intervals

https://perfetto.dev

Linux scheduling

Scheduling in Linux

- One scheduler to rule them all
  - CFS < v6.6
  - EEVDF >= v6.6
- Really difficult to conduct experiments
- Really difficult to upstream changes
- Multiple out-of-tree schedulers

Proportional weight-based CPU allocation: fairness

- Each task T_i has a weight w_i
- The runtime assigned to each task T_i is proportional to its weight w_i divided by the sum of all the runnable tasks' weight

$$runtime(T_i) = \int_{t_0}^{t_1} \frac{w_i}{\sum_{j=0}^{N} w_j} dt \simeq \frac{w_i}{\sum_{j=0}^{N} w_j} \cdot (t_1 - t_0)$$

How fairness is implemented: vruntime

- Virtual runtime (vruntime)
  - Charge each task a runtime proportional to w_{base} and inversely proportional to its weight w_i
- Tasks are scheduled in order of increasing vruntime

$$V_{T_i}(t_1) = \frac{w_{base}}{w_i} \cdot (t_1 - t_0)$$

EEVDF: Earliest Eligible Virtual Deadline First

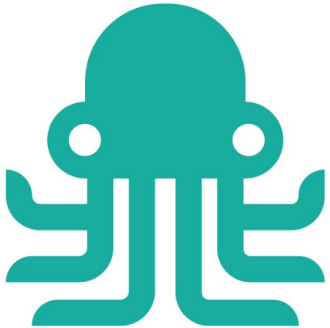
- Lag: difference between the ideal runtime and the actual runtime of a task
- Eligibility: a task is eligible to run if its lag >= 0
- Virtual deadline: vruntime + requested time slice (scaled)

$$lag_T(t_1) = V_{avg}(t_1) - V_T(t_1) \geq 0$$

$$D_T(t_1) = V_T(t_1) + \Delta t_T \cdot \frac{w_{base}}{w_i}$$

sched_ext: the extensible scheduling class

- Technology in the Linux kernel that allows to implement scheduling policies as BPF programs (GPLv2)
- Available since Linux v6.12
- Key features:
  - Bespoke scheduling policies
  - Rapid experimentation
  - Safety (can’t crash the kernel)



sched_ext

sched_ext + gaming

Design sched_ext scheduler(s) to prioritize latency

- Latency-sensitive tasks tend to block often
- Relax the fairness constraint and prioritize latency behavior
  - Boost priority in function of voluntary context switch rate
  - Track of sleep / wake up frequency per-task
  - Track average partial runtime per-task
  - Scale time slice inversely proportional to the number of tasks waiting to be scheduled

VDER: Virtual Deadline with Execution Runtime

- Virtual deadline: total vruntime + partial vruntime accumulated since the task was blocked on an event
  - t_1 = current time
  - t_0 = time when the task was blocked

$$D_{T_i}(t_1) = V_{T_i}(t_1) + \Delta v_{T_i}(t_1)$$

$$\Delta v_{T_i}(t_1) = \frac{w_{base}}{w_i} \cdot (r_{T_i}(t_1) - r_{T_i}(t_0))$$

Demo

- Gaming under pressure
  - https://www.youtube.com/watch?v=ZuWyIrKJA38

Conclusion

Can sched_ext help gaming on Linux?

- Gaming devices are getting more complex
  - Topology complexity
  - Power saving
- Workload is getting more complex
  - Multiple high-priority activities
- Scheduling specialization could be the key to improve gaming experience on Linux

FOSDEM

Questions?

Andrea Righi <arighi@nvidia.com>

NVIDIA