Profiling energy consumption on Linux systems

Ondřej Míchal
xmicha80@stud.fit.vutbr.cz

VYSOKÉ UČENÍ FAKULTA
TECHNICKÉ INFORMAČNÍCH
V BRNĚ TECHNOLOGIÍ

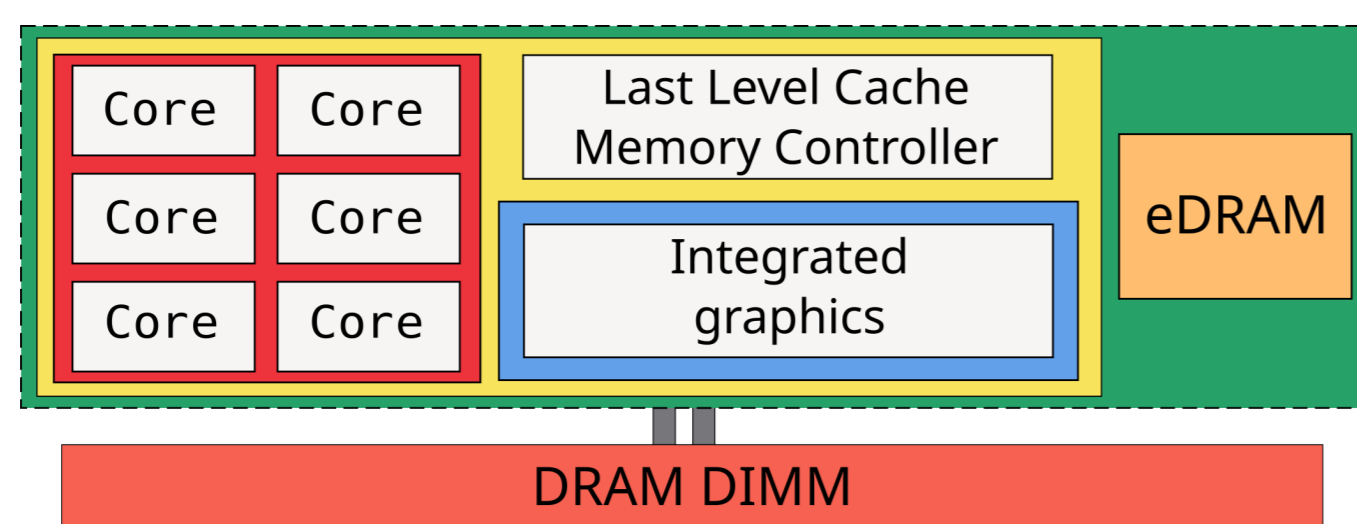Excel@FIT 2023

Motivation: Untapped potential of energy profiling

- Modern devices can have problems with **energy consumption**
  - Mobile devices, IoT or smart devices

- **Developers do not know the performance of their software**
  - They need help of profilers

- Existing tools have **low granularity** and **lack runtime context**

poll-psys
futex-psys
read-psys
poll-cores
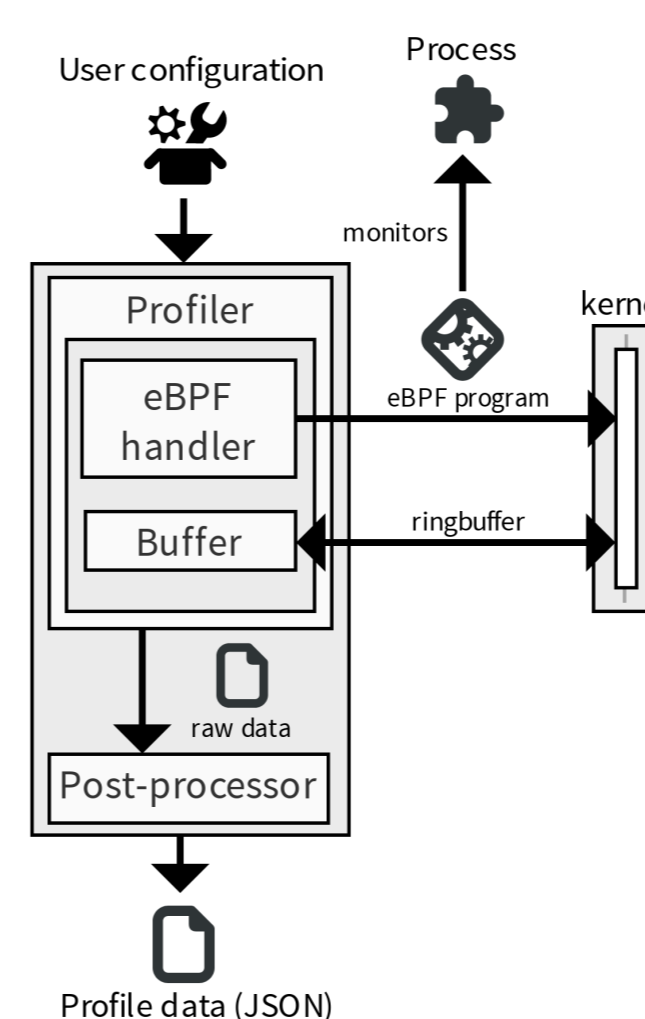futex-cores
read-cores
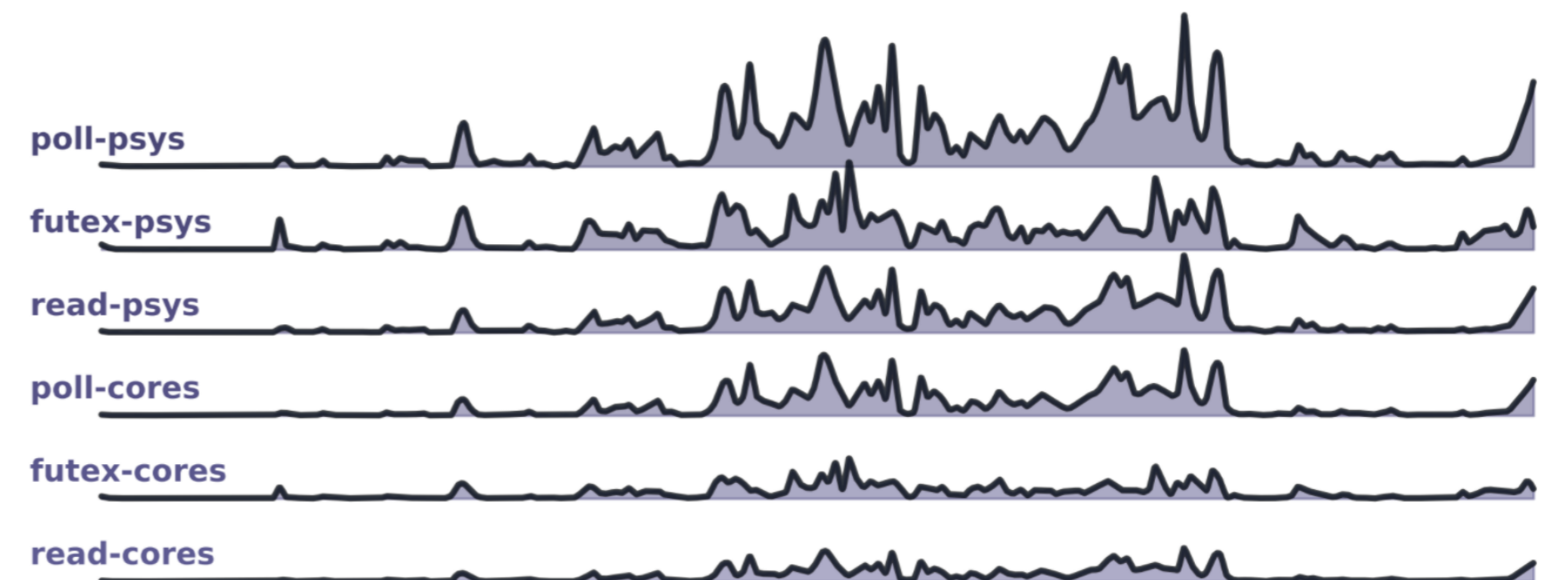
Measuring energy consumption

- It's impossible to cover every energy domain

- Modern SoCs support voltage monitoring
  - *Running Average Power Limit* in Intel/AMD SoCs

Core Core | Last Level Cache
Core Core | Memory Controller | eDRAM
Core Core | Integrated graphics
DRAM DIMM

Energy profiler

User configuration
Process
monitors
Profiler
eBPF handler
eBPF program
kernel
Buffer
ringbuffer
raw data
Post-processor
Profile data (JSON)

- Novel energy profiler

- Uses system calls as context

- Samples at **high frequency**

- Uses **eBPF** for monitoring

- Profiles available in JSON format

Experimental evaluation: Comparison of different consumptions

- Evaluated on *GNOME Shell* with different workloads:
  - Figure 1: Clean session

  - Figure 2: Firefox plays a video

- Visualizations (*waterfall graph*) highlight higher consumption of Shell during video playback

- Highest consumption is during shell transitions (e.g., virtual desktop switch) and composit-ing windows with moving content (*read, write, ioctl*)
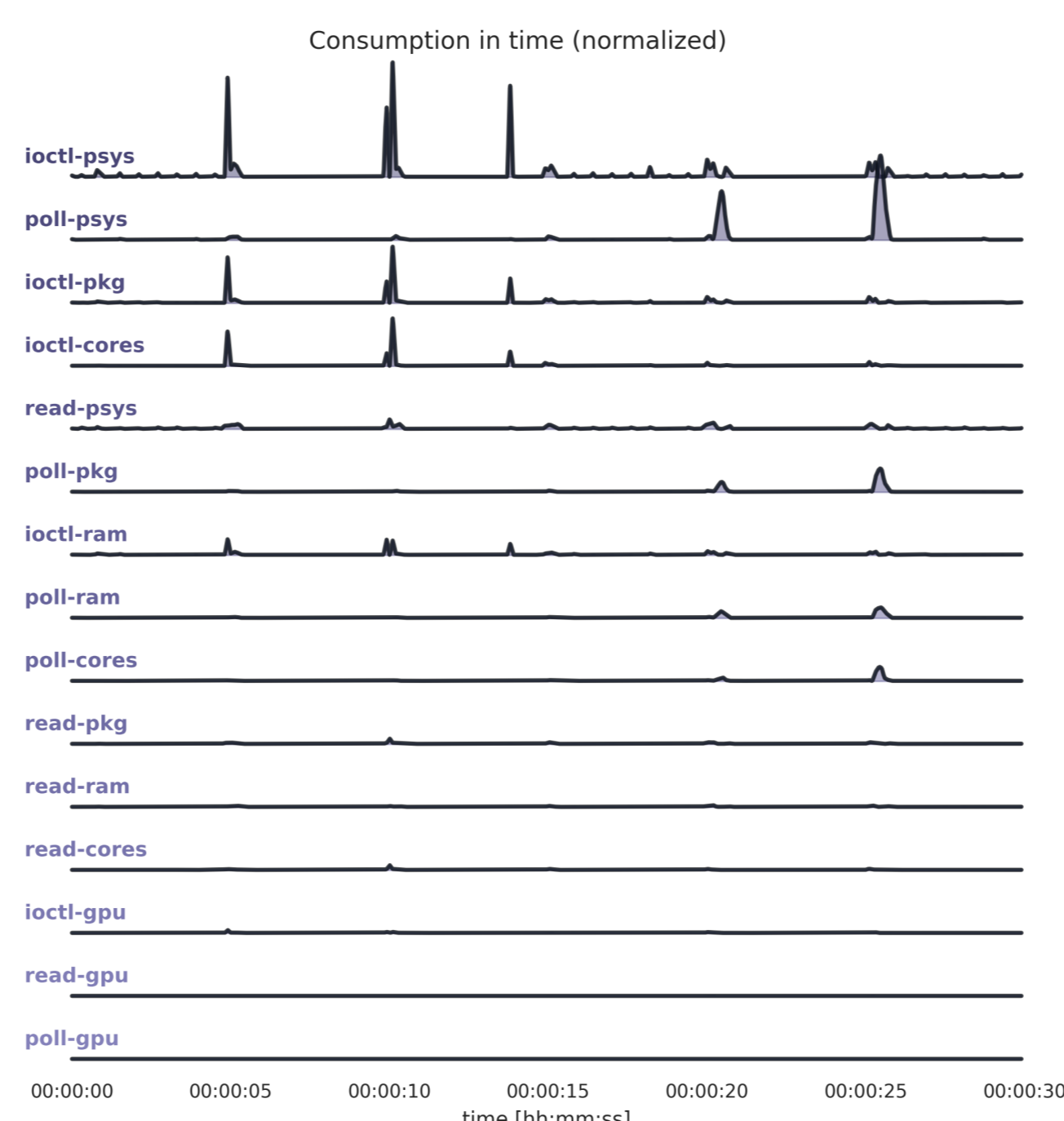

Consumption in time (normalized)

ioctl-psys
poll-psys
ioctl-pkg
ioctl-cores
read-psys
poll-pkg
ioctl-ram
poll-ram
poll-cores
read-pkg
read-ram
read-cores
ioctl-gpu
read-gpu
poll-gpu

00:00:00 00:00:05 00:00:10 00:00:15 00:00:20 00:00:25 00:00:30
time [hh:mm:ss]

Figure 1: Consumption of GNOME Shell without any windows

Consumption in time (normalized)

ioctl-psys
write-psys
read-psys
ioctl-pkg
write-pkg
read-pkg
ioctl-ram
write-ram
ioctl-cores
read-ram
write-cores
read-cores
ioctl-gpu
write-gpu
read-gpu

00:00:00 00:00:05 00:00:10 00:00:15 00:00:20 00:00:25 00:00:30
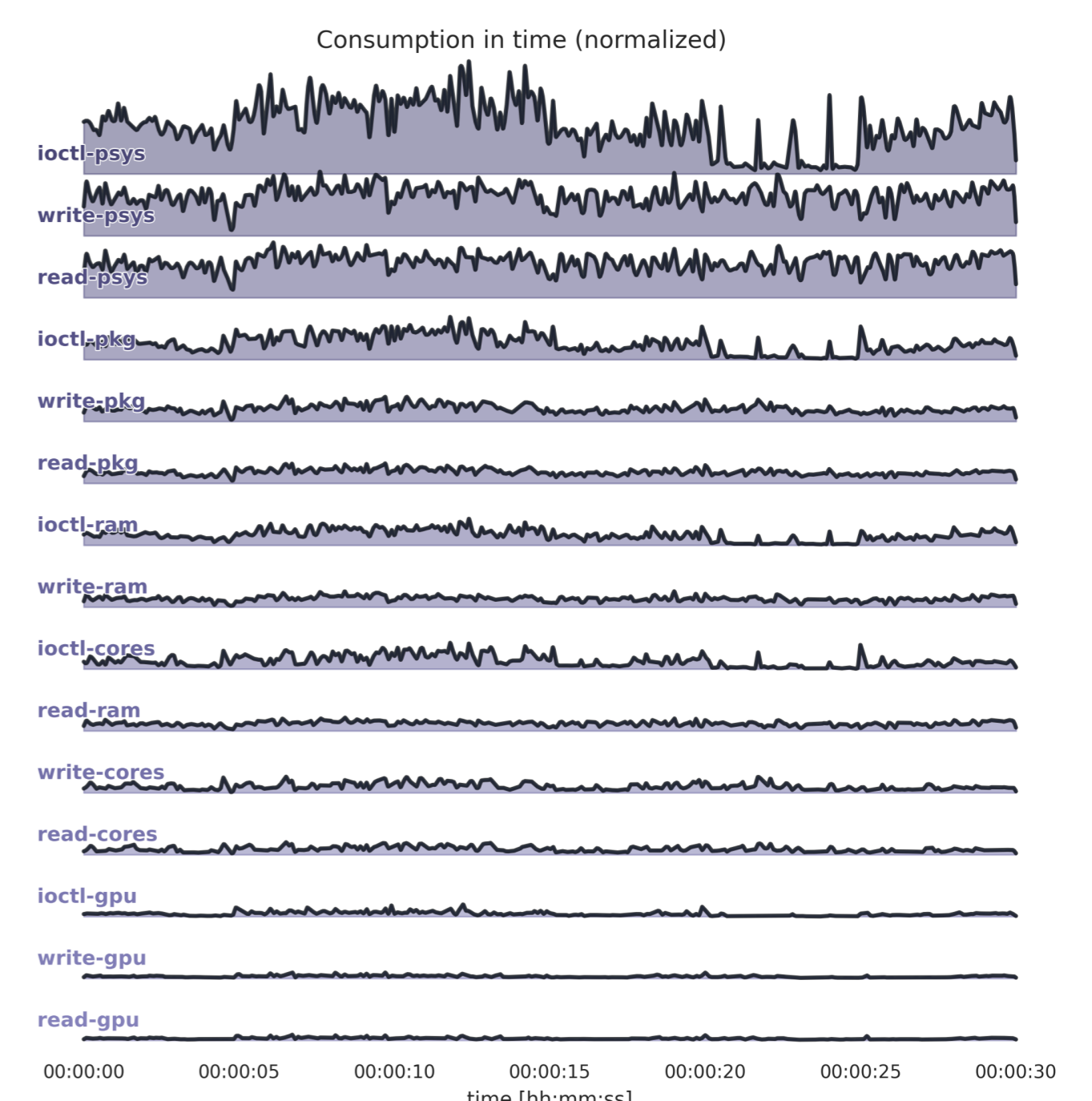time [hh:mm:ss]

Figure 2: Consumption of GNOME Shell with Firefox playing a YouTube video