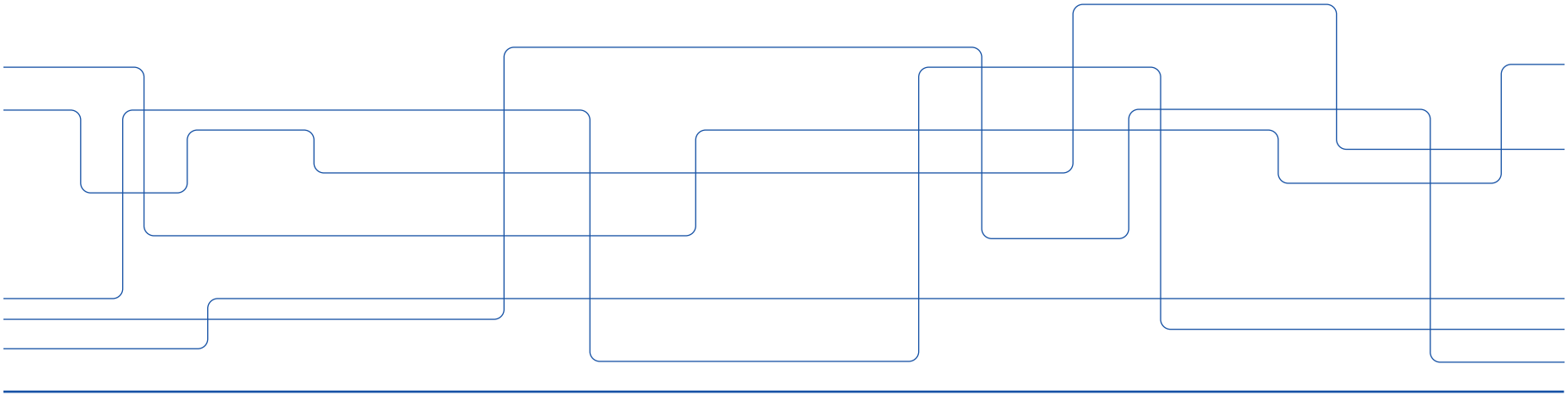
DD2358 – Introspecting an Existing Process with `PySpy`

Stefano Markidis

KTH Royal Institute of Technology



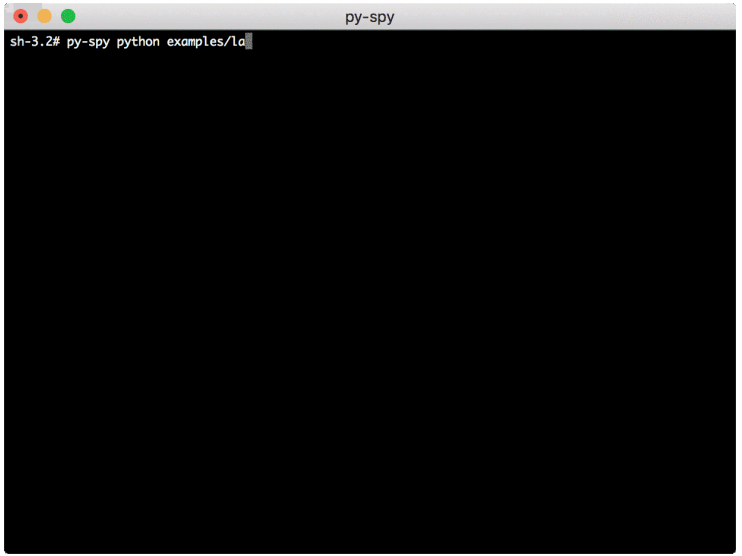
Intended Learning Outcomes

- Use the `py-spy` tool to profile running processes

py-spy

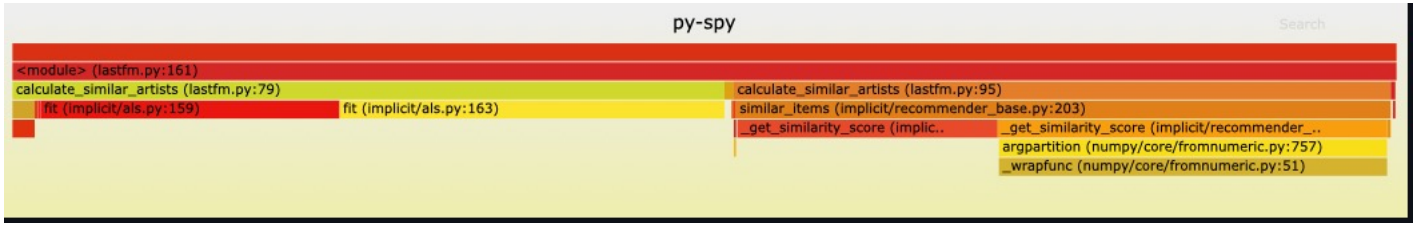
- `py-spy` is a sampling profiler
  - Rather than requiring any code changes, it introspects an already-running Python process and **reports in the console with a top-like display**.
- Being a sampling profiler, it has almost no runtime impact on your code.
- It is written in Rust
- It requires elevated privileges to introspect another process.
  - Use with `sudo python …` to acquire super-user privilege
  - You can do on your own computer but not in the lab computers
- This tool could be very useful in a production environment with **long-running processes** or **complicated installation requirements**.

Using py-spy with top and record



- `py-spy top -- python myprogram.py`
  - It creates a top-like graphics with updates.
  - his updates every second.

- `py-spy record -o profile.svg -- python myprogram.py`
  - It creates a plot with profiling. The width of the display represents the entire program's runtime, and each layer moving down the image represents functions called from above.



To Summarize

- `pyspy` is a sampling profiler to provide profiling information using a top-like window or a plot
- It requires super user privileges to run
- Useful for monitoring long-run processes and installation