How to Tune your Python Analysis Pipeline:

A Profiler Guide

Jonathan Striebel

Hi, I’m **Jonathan Striebel**.

@jostriebel

jonathan@aignostics.com

aignostics

Profiling

- Measure
- Speed & Memory Bottlenecks
- to Identify & Analyse them
- for Mitigation

What?

Why?

Profiling vs Benchmarking

Profiling

- Measure **individual parts** of a program
- Analysis **within** program

Benchmarking

- Measure **whole program**
- **Compare** different programs

Time Profiling

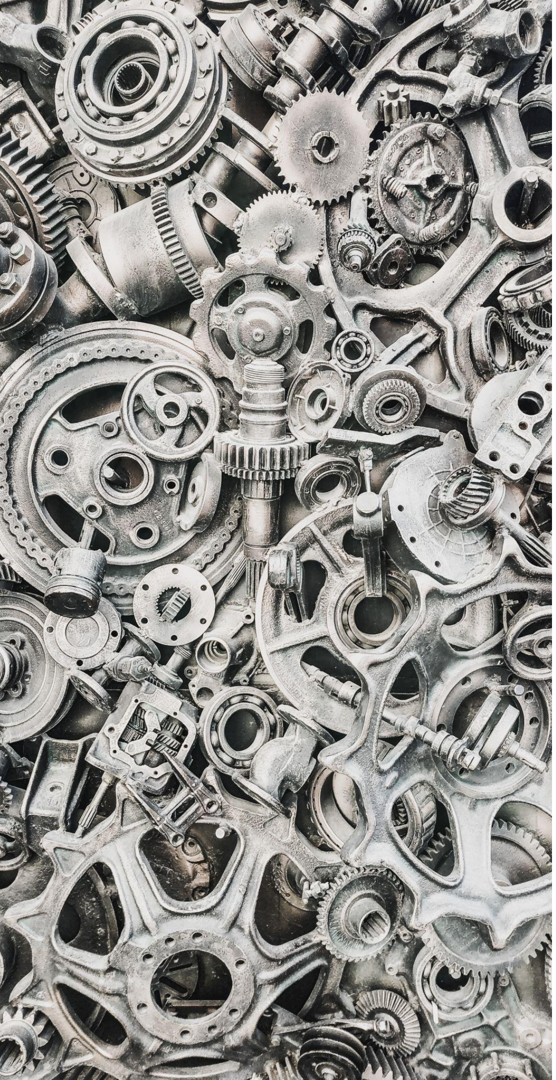
Slow Code

- Where?
- How slow?
- Why?

Memory Profiling

High Memory Usage

- Where?
- How much?
- Why?



Instrumenting vs Sampling

Deterministic Instrumenting

- Measure **each function or code block** of a program
- Potential overhead
- Inaccuracies for calls with lots of instrumentation inside

Statistical Sampling

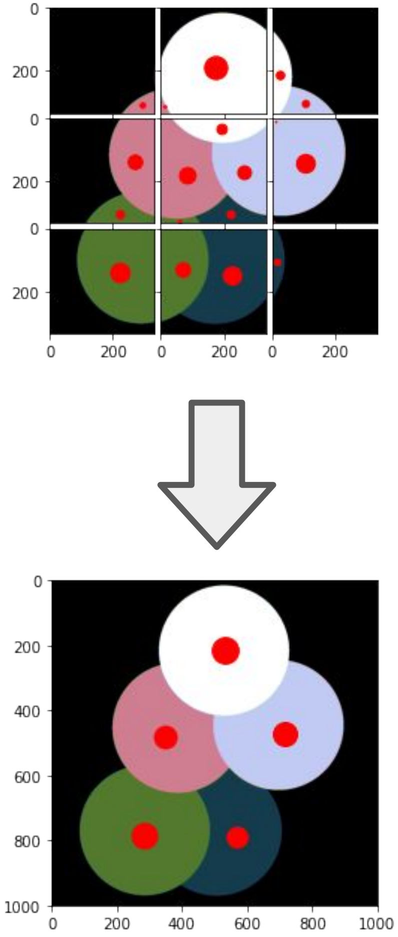
- **Periodically sample** the program's state
- Can miss brief invocations
  (usually fine for timing, problematic for short memory spikes)

Hands On

Combining statistics

per segment

for each chunk



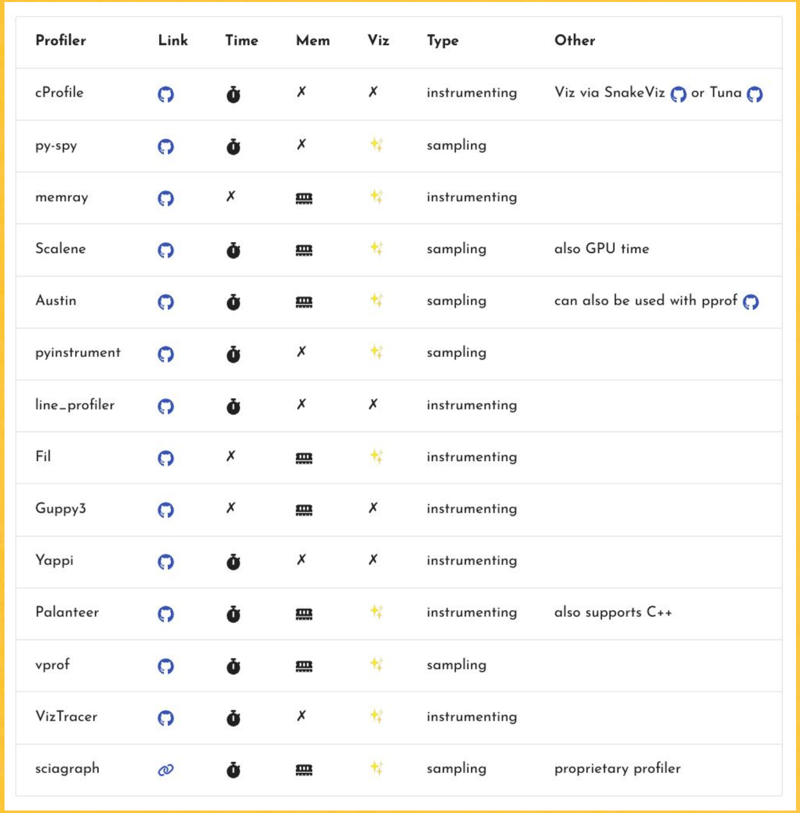
Python Time Profilers

- cPython (instrumenting) + SnakeViz
- py-spy (sampling)
- Scalene (sampling, also GPU time)
- ...

Python Memory Profilers

- memray (instrumenting)
- Scalene (sampling)
- ...

More



Profiler	Link	Time	Mem	Viz	Type	Other
cProfile	🐙	⏱	✗	✗	instrumenting	Viz via SnakeViz 🐙 or Tuna 🐙
py-spy	🐙	⏱	✗	✨	sampling	
memray	🐙	✗	▦	✨	instrumenting	
Scalene	🐙	⏱	▦	✨	sampling	also GPU time
Austin	🐙	⏱	▦	✨	sampling	can also be used with pprof 🐙
pyinstrument	🐙	⏱	✗	✨	sampling	
line_profiler	🐙	⏱	✗	✗	instrumenting	
Fil	🐙	✗	▦	✨	instrumenting	
Guppy3	🐙	✗	▦	✗	instrumenting	
Yappi	🐙	⏱	✗	✗	instrumenting	
Palanteer	🐙	⏱	▦	✨	instrumenting	also supports C++
vprof	🐙	⏱	▦	✨	sampling	
VizTracer	🐙	⏱	✗	✨	instrumenting	
sciagraph	🔗	⏱	▦	✨	sampling	proprietary profiler

codical.org

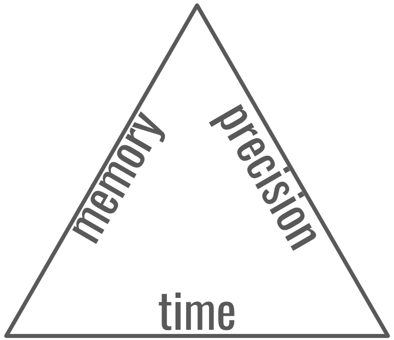
Open Questions

- Benchmark Overhead
- Measure Accuracies
- Visualization Comparison
- Support for
  - async, threading, multiprocessing
  - compiled extensions
  - Linux, MacOS, Windows

codical.org

Bottleneck Mitigation

- Efficient IO
  - less csv, json, yaml, ...
  - more zarr, parquet, sqlite, hdf5, ...
- Vectorization
  - less loops, more numpy
- Memory-Precision-Time Tradeoffs
  - data-type, compression, look-up tables,...
- Upgrade Libs & Runtime
- Closer to the Metal
  - Jitting with Numba
  - Cython, pybind11, cffi, PyO3, ONNX
- Parallelization
  - async, threading, multiprocessing, Spark/Dask/Ray, ...



https://github.com/
jstriebel/data-analysis-speedup

Future

- from Python 3.12: special mode to support Linux perf profiler
- Continuous Profiling: Pyroscopy & Grafana Phlare

How to Tune your Python Analysis Pipeline:

A Profiler Guide

More info:

codical.org/profilers

Jonathan Striebel jonathan@aignostics.com @jostriebel