Karlsruhe Institute of Technology

Autotuning PostgreSQL

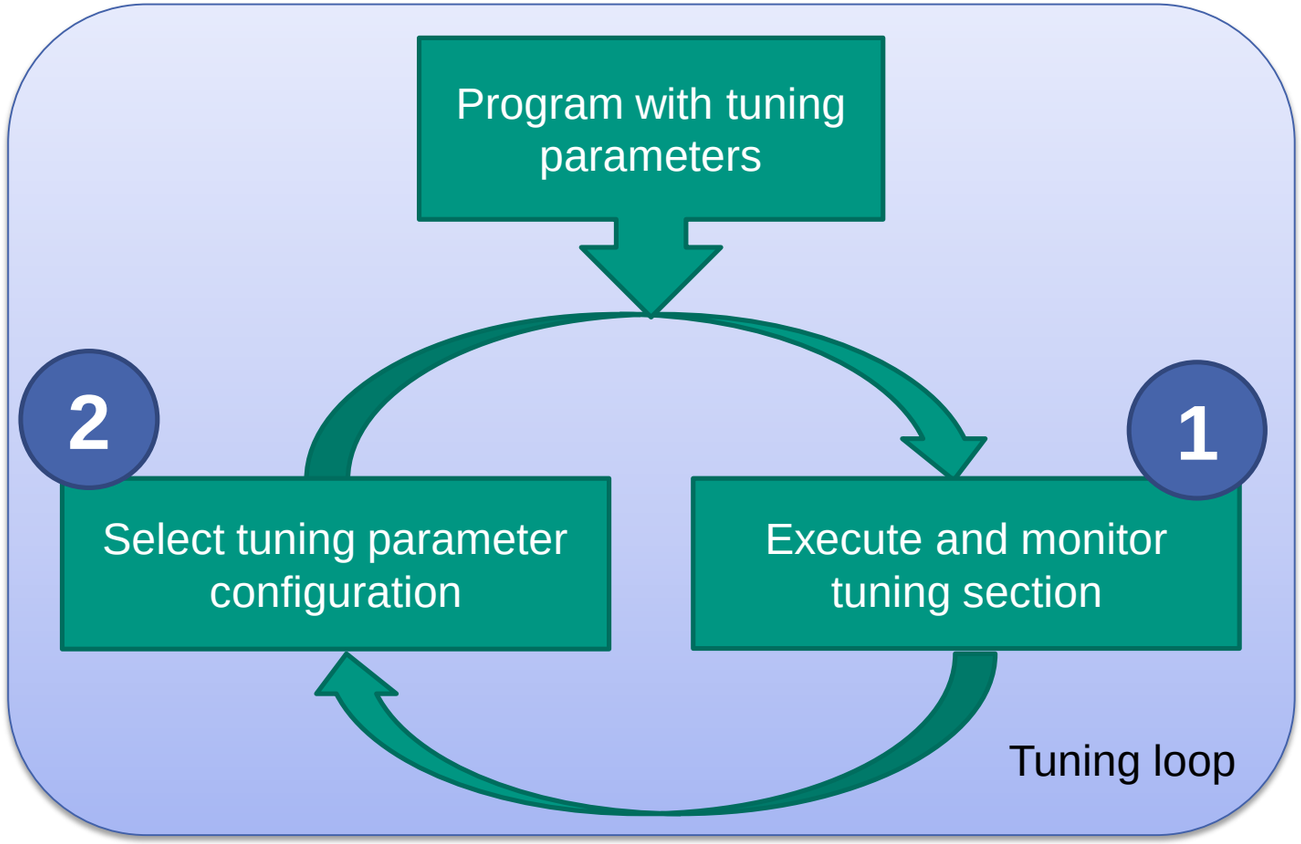
– A Blueprint for Successfully Tuning Real-World Applications –

Thomas Karcher, Mathias Landhäußer

KIT – Department of Informatics – Institute for Program Structures and Data Organization (IPD)

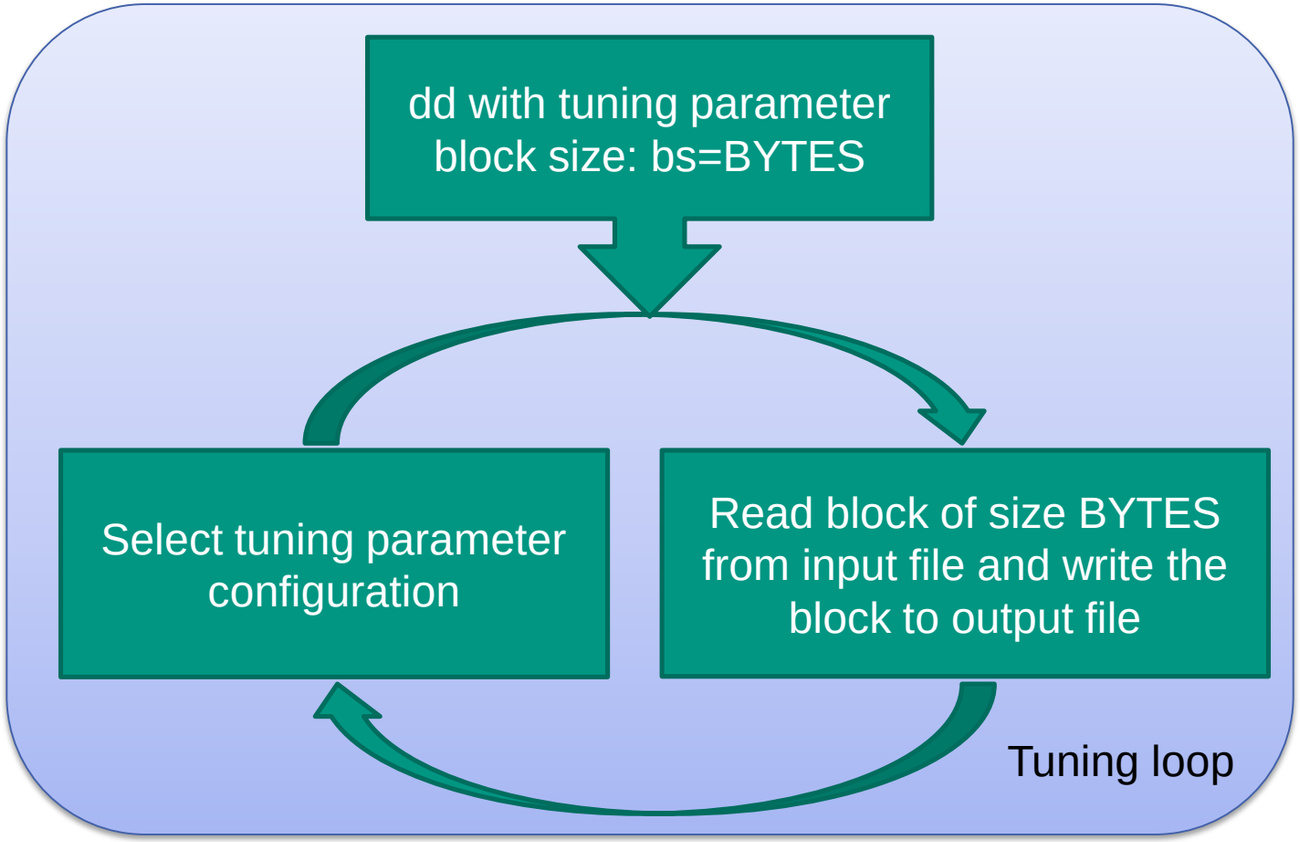
Autotuning in a Nutshell

- Performance optimization by manipulating program variables (“tuning parameters”)
- Offline (between runs),
  Online (while running)
- Auto-tuning scientific applications for ~15 years
- “But what about my everyday programs?“
- You need:
  - At least one tuning parameter
  - A tuning section

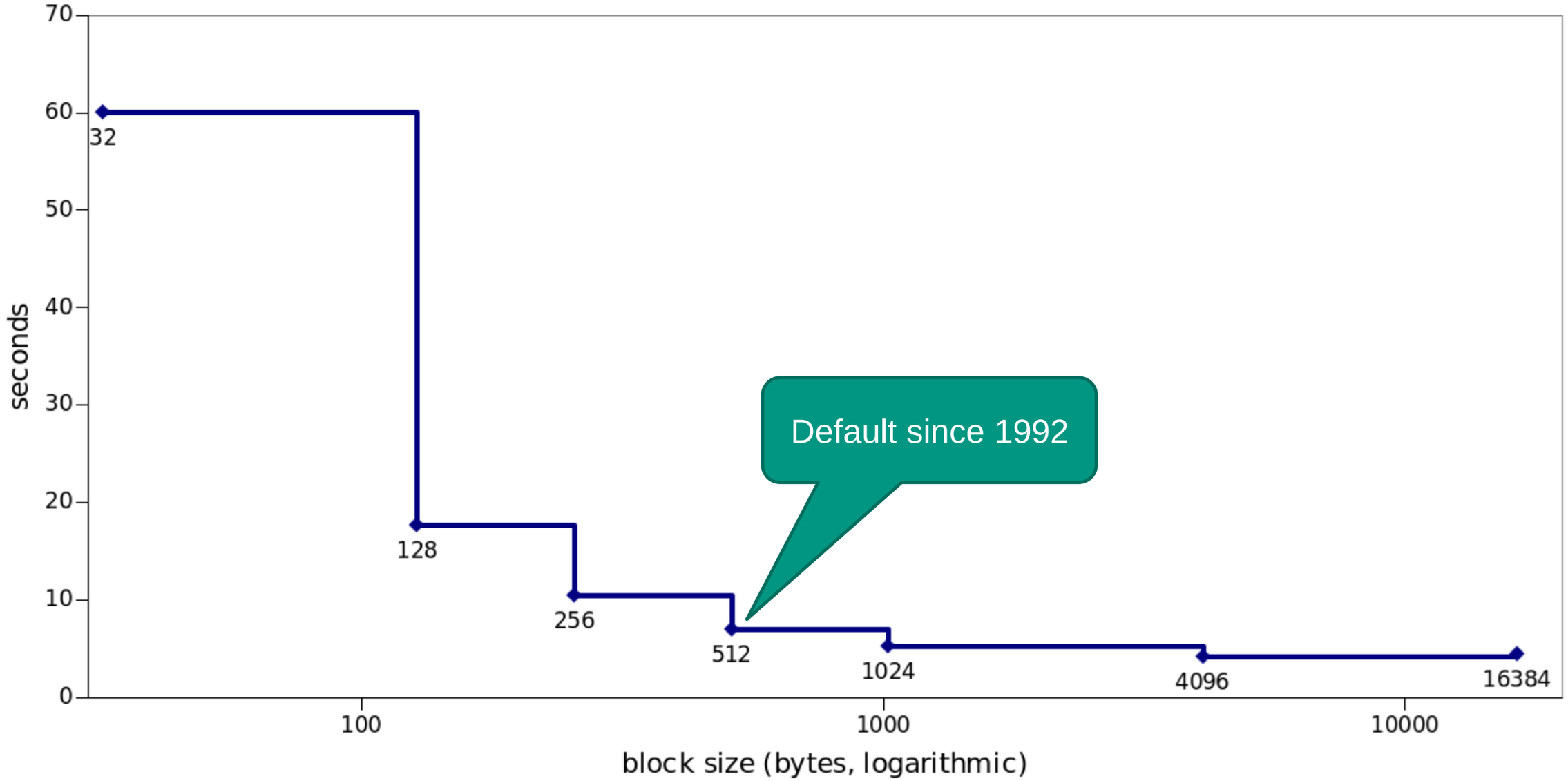


Tuning dd (from GNU coreutils)
A Simple Example

Command line: `dd if=inputfile of=outputfile bs=512`



Tuning dd (from GNU coreutils)
Manually Measuring the Effects of Block Size (with cold cache)




How about tuning a recent software
w/ 261 parameters
w/ less obvious optimal settings?
37 potential tuning parameters
25 query preparation
12 query execution
PostgreSQL 9.6

Tuning PostgreSQL 9.6
Choosing Tuning Parameter Candidates

- Parameter pre-selection: “SARD: A Statistical Approach for Ranking Database Tuning Parameters”

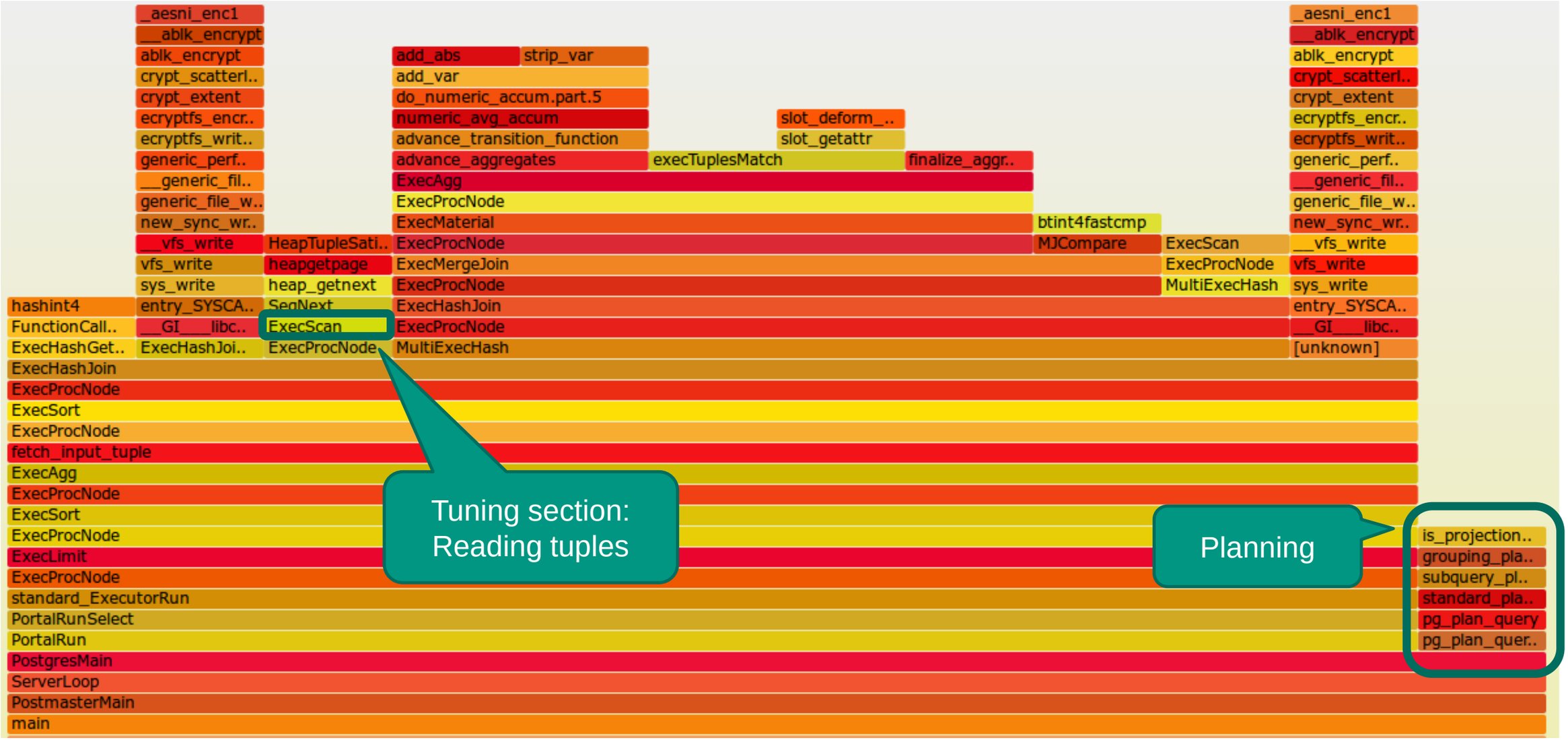
Experiment-based sorting of tuning parameter candidates by performance impact

- Benchmark: TPC-H

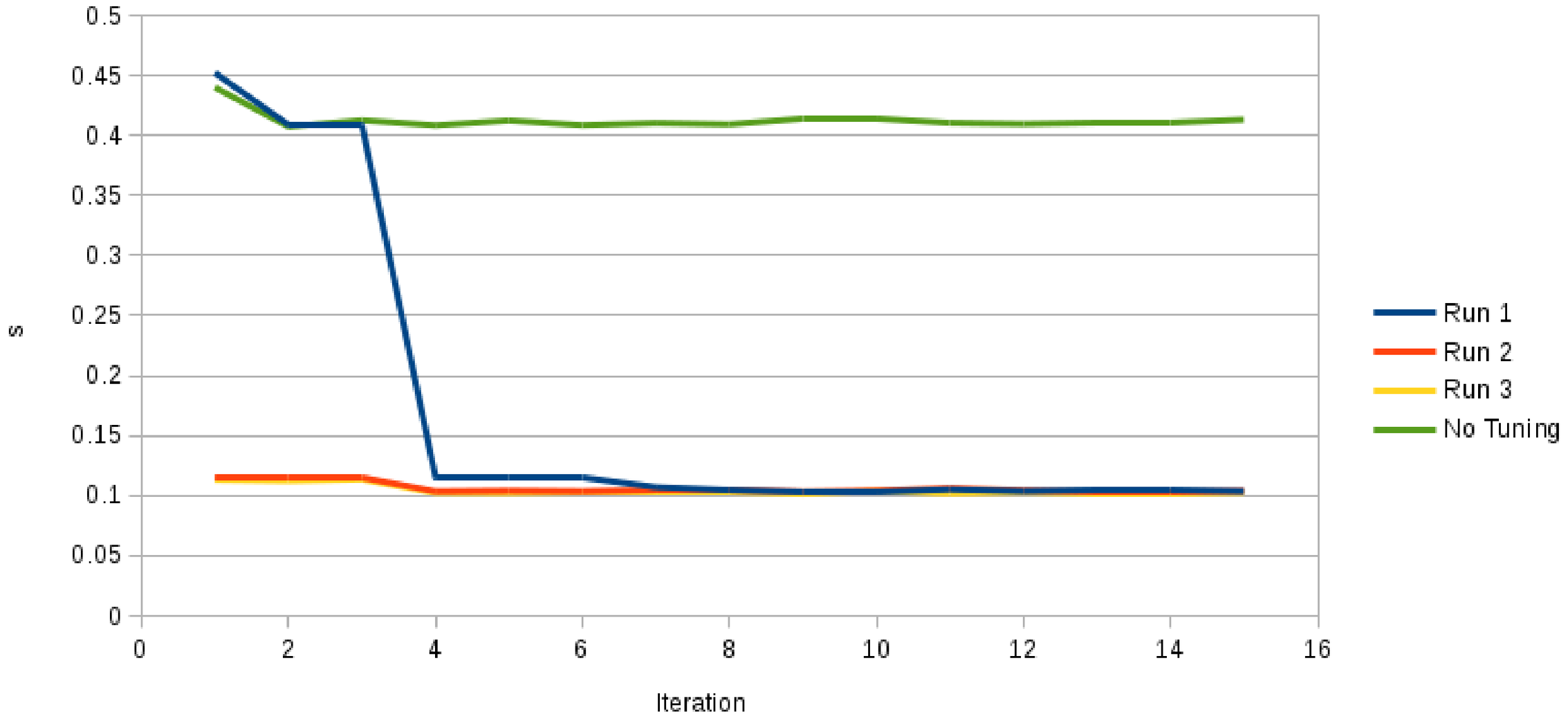
9 tables
23 SELECT queries

- Top 4 candidate parameters:
  - work_mem
  - effective_io_concurrency
  - temp_buffers
  - maintenance_work_mem
- Finding tuning section?

Tuning PostgreSQL 9.6
Profiling TPC-H Query #18 with a Flame Graph



Tuning PostgreSQL 9.6
Results TPC-H query #3



Want to Know More?

- More details about our auto-tuning blueprint can be found in our paper @ICSE ’18

- In the demo in Gothenburg we’ll show
  - how to set up parameter ranking with SARD,
  - how to deal w/ PostgreSQL’s large codebase,
  - how to choose the proper tools/views on the SW when searching for the tuning loop, and
  - how to actually implement the tuning loop in the source code.