How’s it like working as a performance engineer ?

Outlines

- Performance engineer: who, why ?
- How to become a performance engineer
  - Technical skills
  - Soft skills
  - Learning paths and materials

Who is performance engineer ?

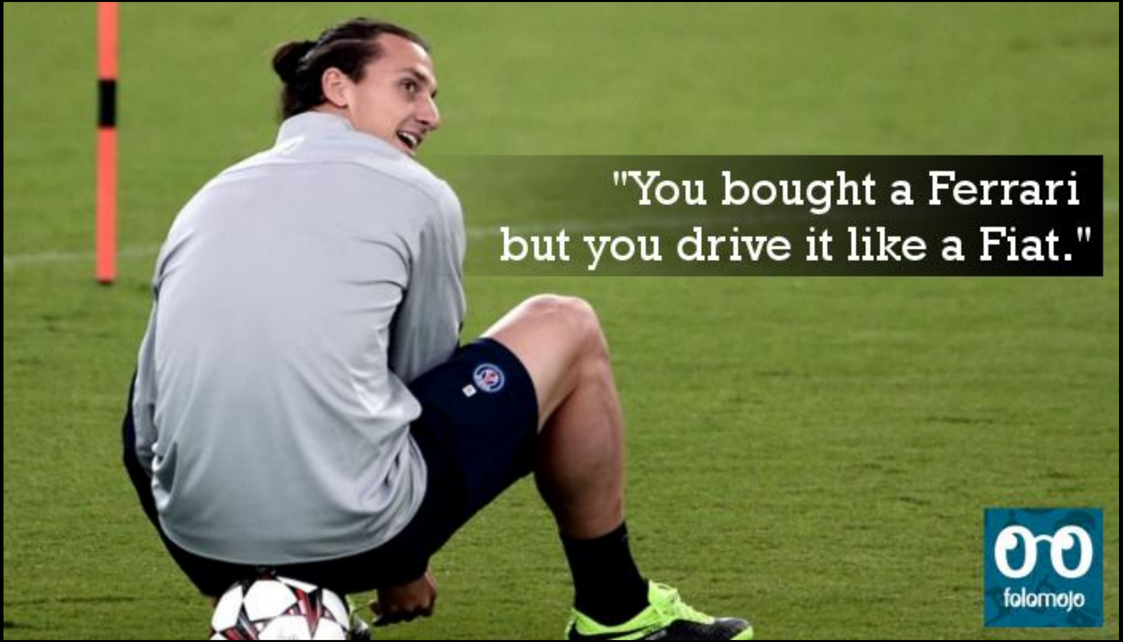
The person who answers these simple questions:

- Why is it slow ?
- What can we do do to make it faster ?
- Is it the best we can do with this machine / server / system ?

Why performance engineer ?

Performance is a feature

You must get the best out of your resources



Puzzles are intriguing

The truth: poor men have to work harder

Different ways to contribute

- Troubleshoot performance issues (SA / SRE / Expert in specific are)
- Develop tools to help others to spot issue quicker
- Research and advocate: share best practices, patterns / anti-patterns to proactively avoid potential problems

A perfect mix

- A developer who knows more than his own code
- A sysadmin who doesn’t mind to code
- A tech guy who can communicate well

→ A “real SRE” who’s not on-call and dedicated for performance tasks

My own ops path

Common knowledge

- Hardware
- OS (CPU / Mem / Disks / Filesystem ...)
- Network (mostly TCP/IP)
- Applications (third-party / in-house, DB, Web server, LB ...)

Skills set

- Monitoring
- Debugging
- Profiling / Tracing
- Performance testing
- Programming

Monitoring for high level view

- Logs, metrics, traces
- R.E.D method
- U.S.E method
- Visualization
- “Smart” alerting

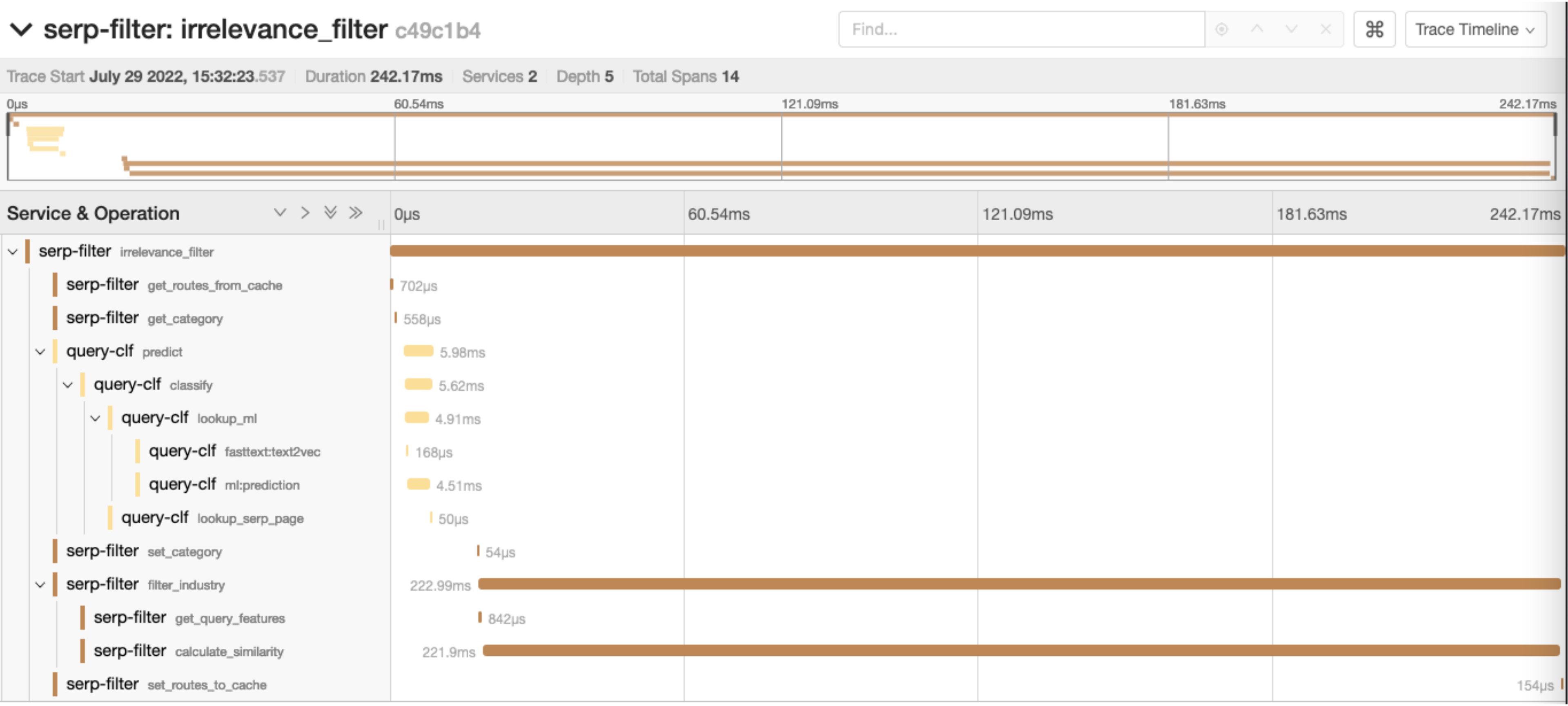
Debugging

- Clarify the situation by right questions
- Reproduce
- Debug/profile/monitor tools to collect info
- Brainstorm ideas and verify it
- Analyze impact of your fix
- Release gradually and observe afterward

Peter: "Where is Gamora?"
Tony: "Yeah. I'll do you one better. Who's Gamora?"
Drax: "I'll do you one better. Why is Gamora?"

Profiling to avoid guessing

- Examine source code in action
- Different types of tracing / profiling:
    - Low-level profiling
    - OS
    - Application
    - ON / OFF CPU
    - Continuous profiling
    - Distributed tracing



serp-filter: irrelevance_filter c49c1b4

Trace Start July 29 2022, 15:32:23.537 | Duration 242.17ms | Services 2 | Depth 5 | Total Spans 14

Service & Operation		
serp-filter	irrelevance_filter	
serp-filter	get_routes_from_cache	702μs
serp-filter	get_category	558μs
query-clf	predict	5.98ms
query-clf	classify	5.62ms
query-clf	lookup_ml	4.91ms
query-clf	fasttext:text2vec	168μs
query-clf	ml:prediction	4.51ms
query-clf	lookup_serp_page	50μs
serp-filter	set_category	54μs
serp-filter	filter_industry	222.99ms
serp-filter	get_query_features	842μs
serp-filter	calculate_similarity	221.9ms
serp-filter	set_routes_to_cache	154μs

Profiling obstacles

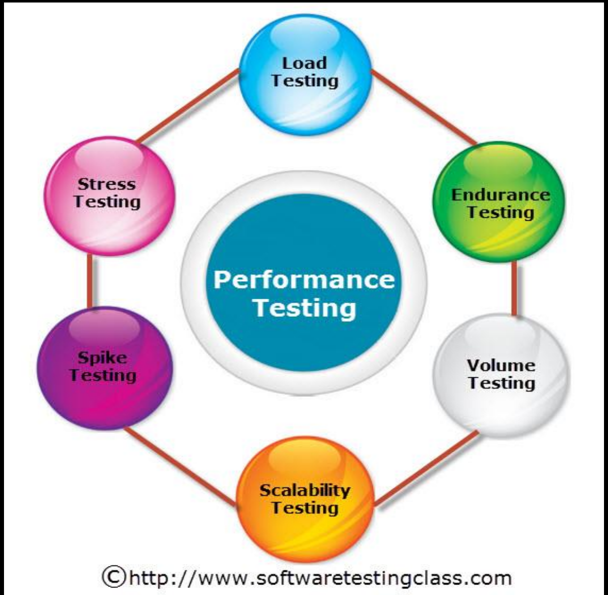
- No data
- Too much data (experience required)
- Overhead
- Requires extra setup sometimes

Analyze data

- Traces: Hot code and slow code
- Visualization: flamegraph, histogram, heatmap, gantt chart

Performance Testing

- Performance testing is an (dark) art
- Stress test, analyze, tune, benchmark
- Report / results visualization is important
- Always tune with benchmarking afterward



©http://www.softwaretestingclass.com

Programming

- ~~Secondary skill~~
- Eventually, source code is the truth. You have to read a lot !
- Should know both compiled and interpreted language
- Data structure and Algorithms is not mandatory but a very big plus
- To analyze (read) other’s code or write our own tools

Soft skills

- Know when to start and stop *
- Communication **
- Embrace failures

Challenges

- No “industry standard” training course / certification
- Lack of playground
- Not beginner friendly

Community

Follow these monsters:

- Linux tracing in general @brendangregg
- Low-level CPU perf @dandibakh
- Cloudflare network ninja @majek04
- Distributed tracer @YuriShkuro
- Always-on tracer @0xTanelPoder
- Continuous profiler @fredbrancz
- He knows it all @bcantrill

Books

- Any Linux book by Brendan Gregg (you may not want to read book about Solaris/FreeBSD OS)
- Methodology / Entry level: Faster
- Debugging
- Distributed tracing (a bit out-date with OpenTracing)
- Tuning modern CPUs
- Programming pearls (not very easy to read with ops guys, but is worth reading)

Conference & labs

Conferences:

- P99conf
- Syscon :D

Labs & tools:

- Bpf-perf-tools-book
- Awesome Low Latency
- FREE perf-ninja course (My TO-DO in Q4 :D)

Projects

- Low-level profiler: pmu-tools, pcm
- Common tracing tools: bpftrace
- Continuous profilers: parca, prodfiler
- Distributed tracing: jaeger, opentelemetry

Thank you !