Performance

How to get it right.

Know your Algorithms

Go back to math class

O(1) rulez

Know your Resources

non-linearity makes guessing hard

Contention

I/O

CPU (Memory Bandwidth)

Memory

Locks

„premature optimization is the root of all evil.“

–Donald Knuth, 1974

Design

first . . . . . then optimize

Make a working System fast is easy

Make a fast System working is hard

A non-working system can be made arbitrarily fast

Measure!

Do not guess.

Think

create a hypothesis

coincidence is not causality

Divide & Conquer

Test

Modify. Measure. Repeat.

Stop

You must not fool yourself

and you are the easiest person to fool.
-Richard P. Feynman

Tools

- Java - jpg, jstack, jhat, … => jcmd
- Java - console => jvisualvm => jmc & flightrecorder
- Logging - @system boundaries, gc, metrics, hibernate, Spring
- Tracing - Prometheus, Zipkin
- Linux, AIX - nmon
- Windows - perfmon
- btrace
- linux perf tools, Solaris Dtrace, Intel Performance Counter Monitor
- flamegraph

Hands on Labs

not so much „real“ analysis, but learn what different problems look like

Primes

JPA

ByteSum

github.com/tmohme/Performance.git