

ERIK OSTERMUELLER

Author of

Troubleshooting Java Performance

<http://bit.ly/2017tjp>

java architect

lead performance engineer

FIS Global

eostermueller@gmail.com

@Eostermueller

erikostermueller.com

SNAIL4J

A Simple Plan for Performance Education

PART I

- What is the most efficient way to learn Software Performance?

TEACHING PERFORMANCE (CURRENT APPROACHES)

- Self Taught, Learn on the job
- Read performance books
- 4 Year CS Degree / Coding Bootcamp
- Weeklong training from Perf Experts
- Hire performance people from outside



TEACHING PERFORMANCE

(PROBLEMS)

- Self Taught, Learn on the job
Takes' too long!
- Read performance books
Which books?
- 4 Year CS Degree / Coding Bootcamp
No room in curriculum for perf
- Weeklong training from Perf Experts
Few can afford \$\$\$ this
- Hire performance people from outside
Not enough people with right skills

SPECIALISTS OF GENERALISTS?

- Armies of developers creating perfect defects faster than they can be discovered
- Developers who can clean up most of their own messes?

WHAT COULD POSSIBLY GO WRONG WITH TEACHING PERFORMANCE?

- Bored, unmotivated students
- There isn't enough time/\$ for classes
- Not enough quality instructors
- Inconsistent curriculum
- Evaluating students takes work

WHAT IF WE

FLIPPED THE PROBLEM?

OLD WAY:

“EXPERTS IMPART
WISDOM”

NEW WAY:

“STUDENTS SEEK
ANSWERS”

...to well
stated problems.

Most efficient way to learn Software Performance?

“EXPERTS IMPART WISDOM”

- Pitiful track record of producing enough engineers with perf skills.
- Tough to find quality instructors
- Tough to find class time
- Tough to assess efficacy of training
- Tough to decide on curriculum

“STUDENTS SEEK ANSWERS”

- Self Paced
- Driven by curiosity
- Assessment is easier (just see who finishes early)
- Students learn from each other
- Optionally, can be supplemented with curriculum

PART 2

- What is snail4j? (it is NOT READY FOR RELEASE)



TODAY: hands-off

Devs have very little hands-on experience with systems under load



WITH SNAIL4j: hands-on

Devs identify the cause of live performance defects, ones running on their wkstn.

WHAT IS SNAIL4J?

- An easy to install “Load-Test-In-A-Box”
 - Load Generator
 - System Under Test
 - Metrics

snail4j implements “The Simple Plan“ for Performance Education:

1. Identify This Performance Defect.
2. Explain How the Optimization Improved Performance.
3. Teach others to use snail4j.

WHAT IS SNAIL4J?

- Comes with its own set of “PATH” perf defects:

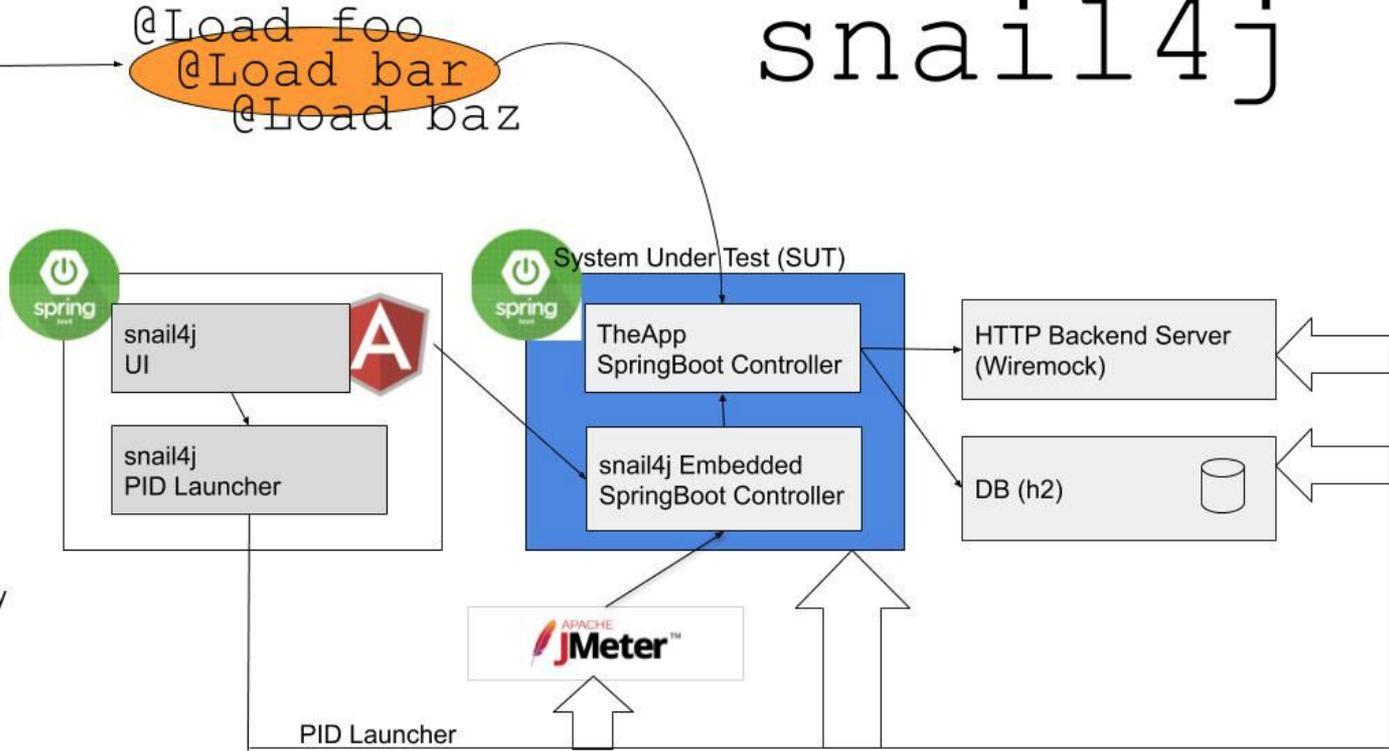
- P - Persistence (RDBMS)
- A - Alien Systems (any system over a network)
- t - threads
- h - heap/gc issues

- Add-your-own-defects

- Compare performance of your own implementations
- Great for code reviewing performance
- Got software with your Java book? Use snail4j to display it under load.

snail4j

- 1) Annotate and deploy the code to load test. Sample perf defects provided!!
- 2) UI presents list of all annotated code: foo, bar, baz.
- 3) JMeter applies load to any/all of foo/bar/baz, based on your live selection
- 4) Change/experiment with JVM parameters and restart from UI.
- 5) Monitoring provided by glowroot.org and Spring Boot Actuator.



SNAIL4J ARCHITECTURE

PART 3

- Demo of snail4j, which
is NOT READY FOR
RELEASE

- Developers author performance defects every day of the week

A CONUNDRUM

- ...but lack the skills to locate those same defects once they've meandered into a live environment

IT IS FRUSTRATING, LIKE LOSING YOUR CAR KEYS,
THEY WERE HERE JUST A MINUTE AGO.

PART 4

- Snail4j Roadmap

SNAIL4J ROADMAP (PAGE 1)

- Showcase publicly available source code (under load, of course) from popular java books, especially java perf books. Here are a few interesting ones:
 - <https://jcip.net>
 - <https://github.com/vladmihalcea/high-performance-java-persistence>
 - <http://bit.ly/2017tjp>
- Demonstrate performance implications of remotely deploying things. Ex: how does performance change if load generator runs from a data center that is 500 miles away?

Takeaway: must be able to deploy the four tiers of the system under test (load gen, java app, wiremock, h2) across multiple machines.

SNAIL4J ROADMAP (PAGE 2)

- Nicely annotated time series graphs that clearly show on a timeline what processing was running when, and how performance differs. This will tell you, for example, “oh, my first implementation of this interface (which I ran for 5 minutes) was much faster than the second one I wrote (which ran for the second 5 minutes of the test).”
- More sophisticated/detailed metrics from the load generation. Why is this important? If your load generation is deployed 500 miles away, need to see/understand performance diff between throughput/response time measured on the server side and on the load generator.
- Docker image distribution.
- Command line packaging that lets you run performance defects headlessly. Great for seeing whether your monitoring can identify the defects.

SNAIL4J ROADMAP (PAGE 3)

- `javaagent` packaging of this tool that loads with your container (`-javaagent:/full/path/to/havoc.jar`), and provides a REST interface that lets PerfGoat enumerate the classes that implement the `@Load` interface
- For all `@Load` annotated code, `snail4j` displays source code, decompiled, and displayed right alongside the option (radio buttons and check boxes) to run, or not to run the code.
- Provide a hardware assessment to the end user, indicating whether the host machine has enough CPU/RAM/DISK to run `snail4j` native examples. Would be helpful to as a launch task.

PART 5

- Snail4j / Conclusion



Troubleshooting Java Performance

Detecting Anti-Patterns with
Open Source Tools

A repeatable method for addressing
performance problems in Java-based
applications

Erik Ostermueller

Let's make providers of
curriculum compete to
better answer students
questions.

- Please welcome a new
- member to the snail4j
development team!!

THAT'S NOT ALL

GERMAN GONZALEZ-MORRIS

<http://devwebcl.blogspot.com>

<https://github.com/devwebcl>

CONCLUSION

What is the most efficient way to learn software performance?

- Ignite curiosity with a simple, open ended question.
- Provided discrete, self-paced work
- Provide hands-on experience with live performance defects
- Be wary of betting the farm on structured curriculum.
- Share the love: Encourage people to share snail4j with others.