# Java™ Performance Testing for Everyone

Presented By:

Shelley Lambert

(AdoptOpenJDK Committer, Eclipse OpenJ9 Committer, IBM Runtimes Test Lead)

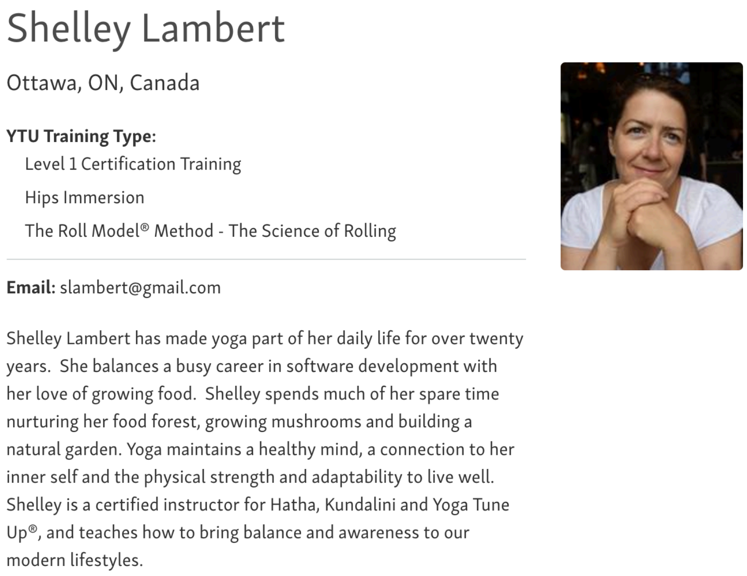
# Who Am I?

## Various Roles

Developer / Test Lead
Development Manager

Yoga Teacher

tuneupfitness.com/teacher/shelley-lambert



Shelley Lambert

Ottawa, ON, Canada

**YTU Training Type:**

Level 1 Certification Training

Hips Immersion

The Roll Model® Method - The Science of Rolling

**Email:** slambert@gmail.com

Shelley Lambert has made yoga part of her daily life for over twenty years. She balances a busy career in software development with her love of growing food. Shelley spends much of her spare time nurturing her food forest, growing mushrooms and building a natural garden. Yoga maintains a healthy mind, a connection to her inner self and the physical strength and adaptability to live well. Shelley is a certified instructor for Hatha, Kundalini and Yoga Tune Up®, and teaches how to bring balance and awareness to our modern lifestyles.

Food Forests, Networks and Saving the World

Chief Food Forester

ottawafoodforests.com
nanabushfoodforests.com

# The Scope

- Projects: Eclipse OMR, Eclipse OpenJ9, AdoptOpenJDK
- Ensuring Free and Verified Java™ for the Community
- 6+ Jenkins servers

---

- 18+ platforms
- 6+ versions, 4+ implementations
- 7+ test categories -> 100,000's of tests

- 18x6x4x100000 = ~43 million tests… nightly, around the world!

OMR OpenJ9 AdoptOpenJDK IBM Runtime Technologies

# Where to start with performance?

- The story of Java performance
  - No single recipe (Many factors: JVM implementation, hardware, application design)
  - JVMs evolve, performance improves

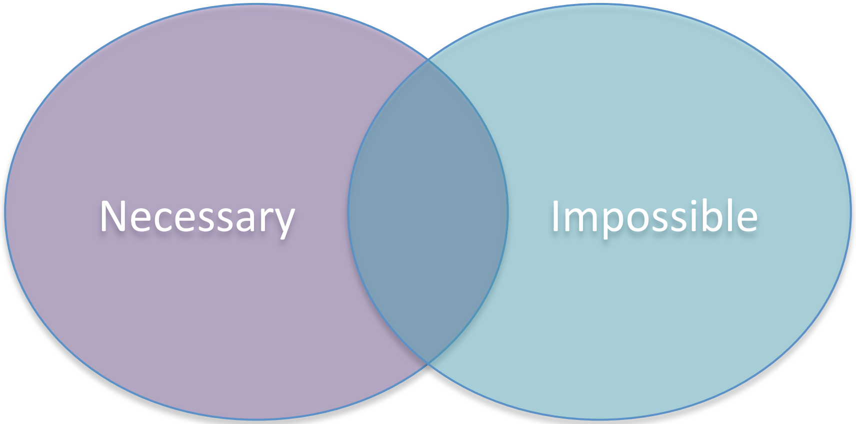
- JVMs "complex, intricate, subtle"

- Wouldn't it be great if it were simple?

  -XX:goFaster, -XX:useLessResources

# The Intersection

- **Necessary**
  - developers need to know if the code they write affects performance
  - currently using diverse set of tools and approaches, home-made scripts, duplication, lost learning opportunities
- **Impossible**
  - measuring performance often stated as “too hard” to do



# Defining the “Impossible”

- What is performance testing?
  - Often called ”Experimental science”
  - “Testing if a system **accomplishes its designated functions within given constraints** regarding processing time and throughput rate.”*

- Good performance? Speed, resources or a blend
  - Modern language runtimes care about many different metrics
    - Throughput, Startup Time, Ramp-up Time, Compile Time
    - Footprint
      - Average Resident Set Size
      - Compilation Memory Consumption
      - Peak Resident Set Size

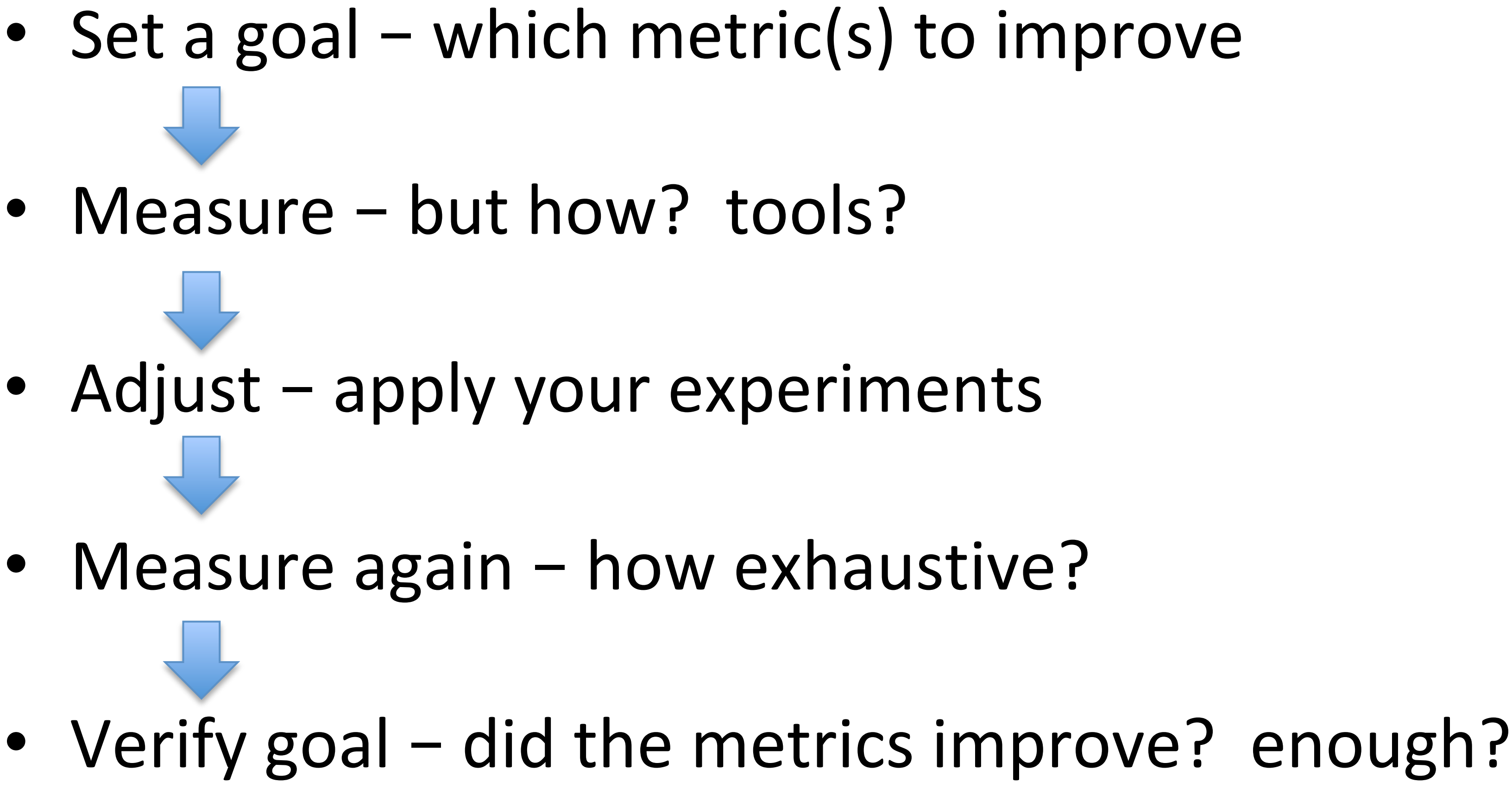
* Witteveen, Albert. Performance testing - a practical guide (Kindle Locations 176-177).

# What to measure

| Metric name | What to measure? | Constraints | Inputs to vary |
|---|---|---|---|
| Throughput | # of transactions | time | |
| Latency | Time for single transaction | # of transactions | Workload (increases) |
| Capacity | # of simultaneous transactions | Throughput or latency | Parallel load on the system |
| Utilization | Use of resources | workload | |
| Efficiency | Throughput/ utilization | | |
| Scalability | Throughput or capacity | | Resources (added) |
| Degradation | Latency or throughput | utilization | Workload (increases) |

Explicit or implicit ‘inputs’ to normalize: HW, OS, system setup

# Basic Steps



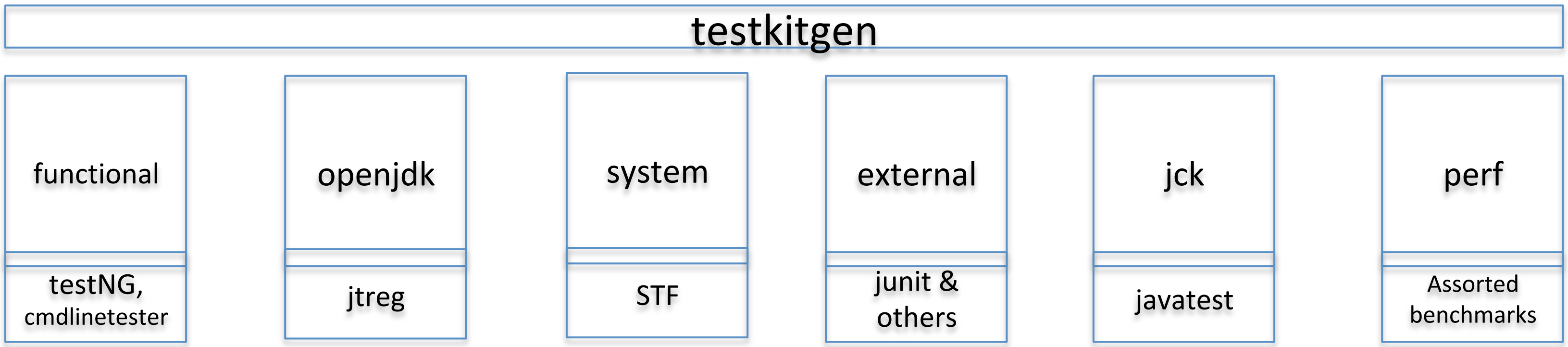
- Set a goal – which metric(s) to improve
- Measure – but how? tools?
- Adjust – apply your experiments
- Measure again – how exhaustive?
- Verify goal – did the metrics improve? enough?

# AdoptOpenJDK Testing

github.com/AdoptOpenJDK/openjdk-tests

- The wildly different 'fruit', how to make them easily consumable

"Consolidate and Curate"



testkitgen

functional
testNG, cmdlinetester

openjdk
jtreg

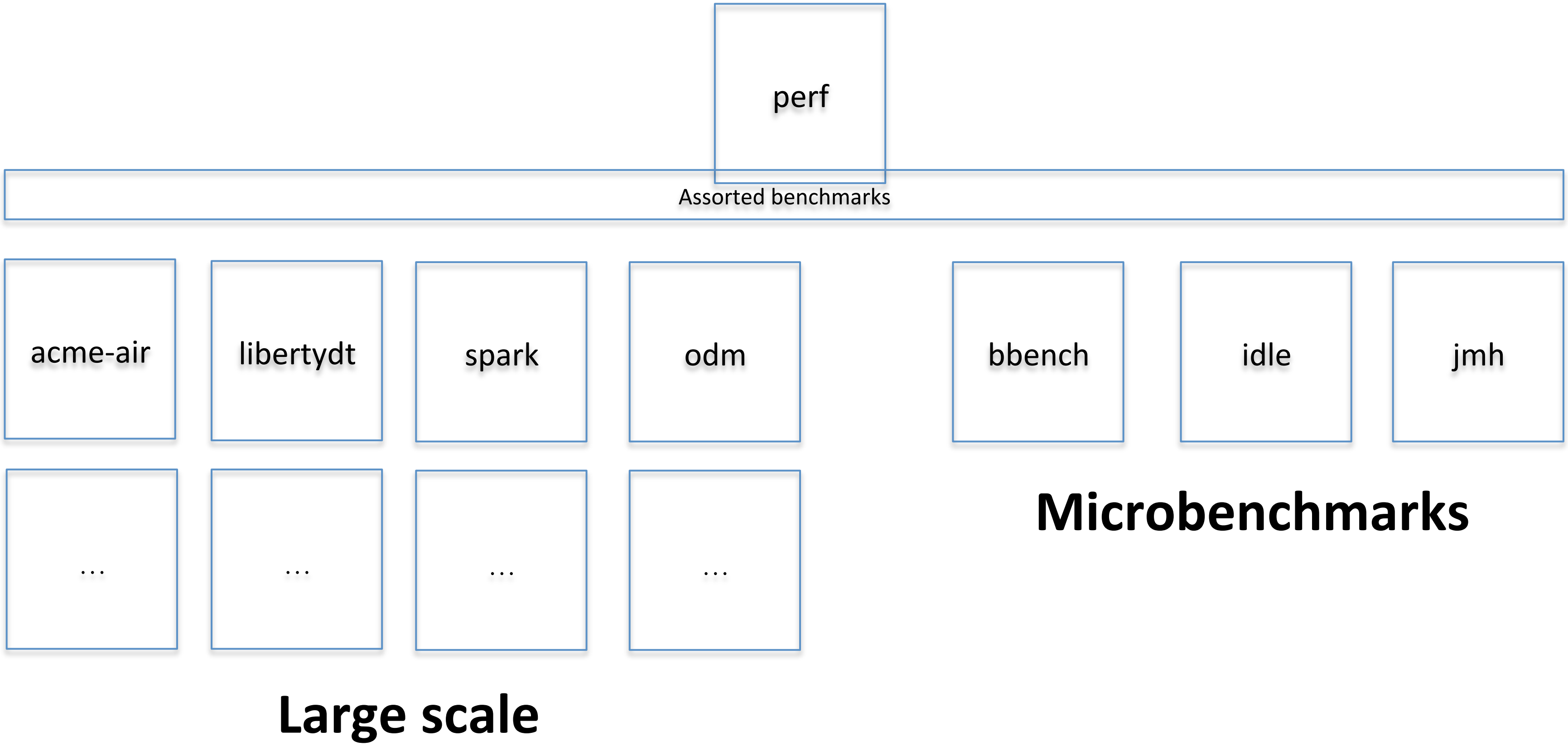
system
STF

external
junit & others

jck
javatest

perf
Assorted benchmarks

# Performance Benchmarks
## (Large-scale and Microbenchmarks at AdoptOpenJDK)



perf

Assorted benchmarks

acme-air

libertydt

spark

odm

…

…

…

…

**Large scale**

bbench

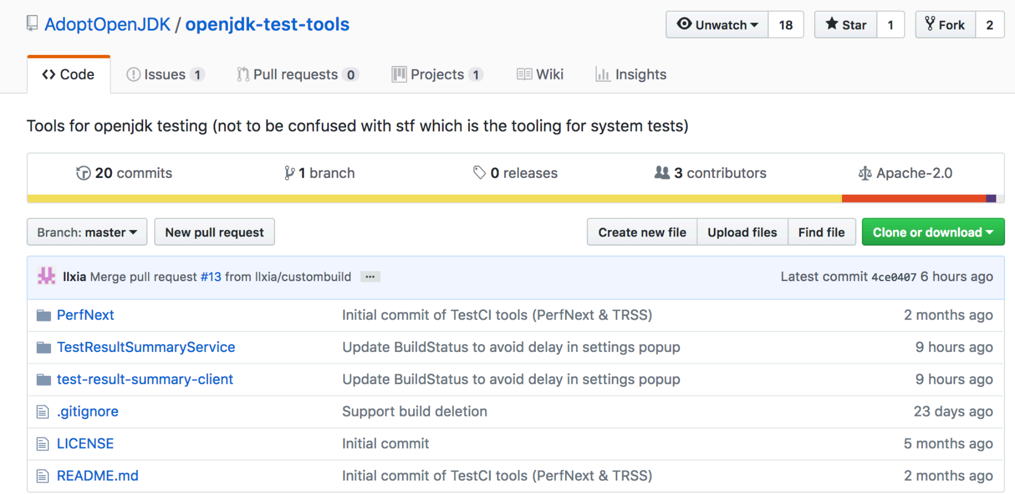
idle

jmh

**Microbenchmarks**

# Introducing

## github.com/AdoptOpenJDK/openjdk-test-tools

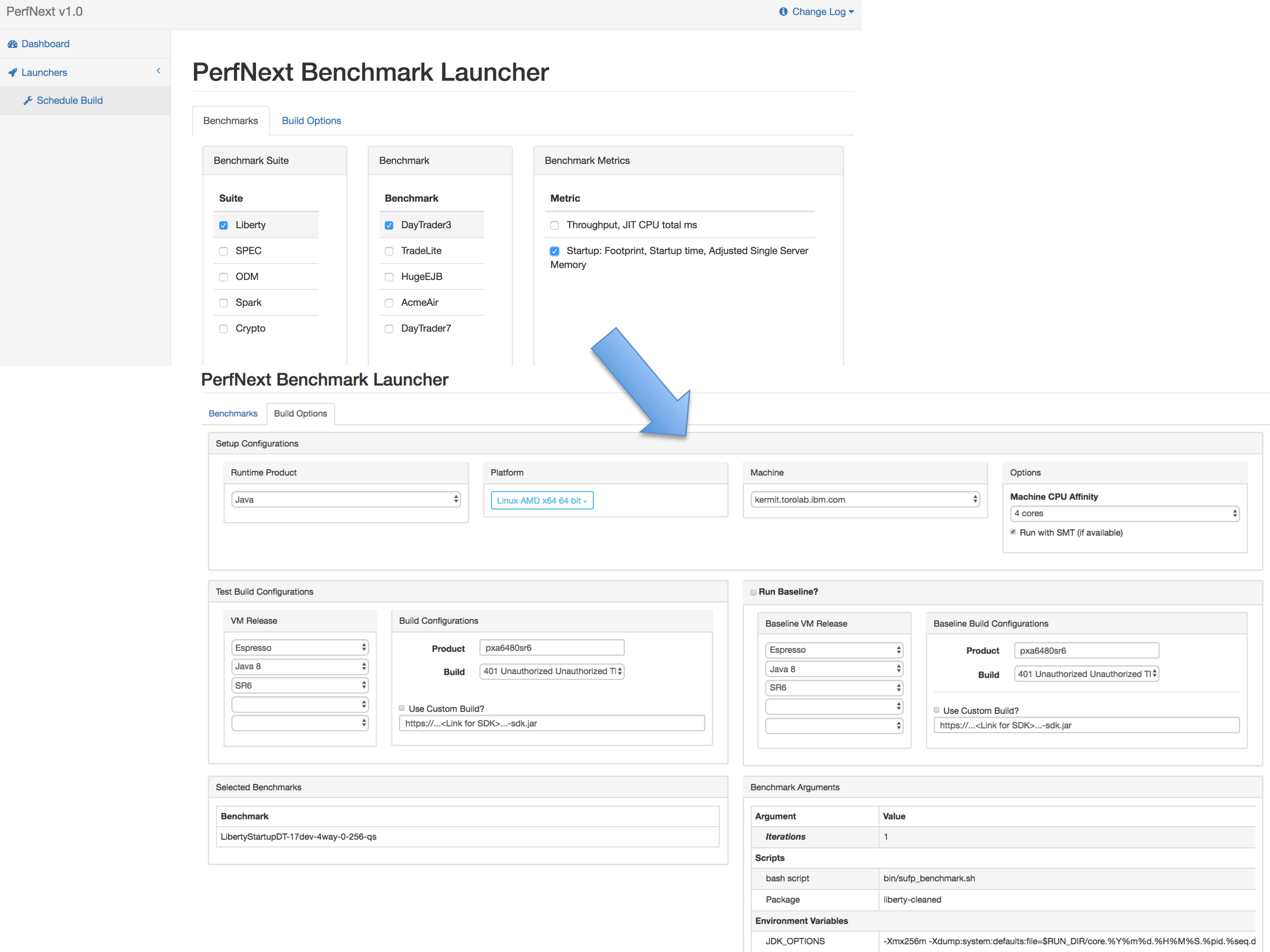


Tools for openjdk testing (not to be confused with stf which is the tooling for system tests)

llxia Merge pull request #13 from llxia/custombuild … Latest commit 4ce0407 6 hours ago

| | | |
|---|---|---|
| PerfNext | Initial commit of TestCI tools (PerfNext & TRSS) | 2 months ago |
| TestResultSummaryService | Update BuildStatus to avoid delay in settings popup | 9 hours ago |
| test-result-summary-client | Update BuildStatus to avoid delay in settings popup | 9 hours ago |
| .gitignore | Support build deletion | 23 days ago |
| LICENSE | Initial commit | 5 months ago |
| README.md | Initial commit of TestCI tools (PerfNext & TRSS) | 2 months ago |

- **PerfNext** - configure, tune and launch performance benchmarks.
- **Test Results Summary Service (TRSS)** - summarize and visualize different test results including perf results, push different sets of test results to a DB, search test and compare results across different platforms, report on differences between jobs
- **Future services** – result analytics, test generation, core analytics, bug prediction



PerfNext v1.0

Change Log

Dashboard

Launchers

Schedule Build

## PerfNext Benchmark Launcher

Benchmarks | Build Options

| Benchmark Suite |
|---|
| **Suite** |
| ☑ Liberty |
| □ SPEC |
| □ ODM |
| □ Spark |
| □ Crypto |

| Benchmark |
|---|
| **Benchmark** |
| ☑ DayTrader3 |
| □ TradeLite |
| □ HugeEJB |
| □ AcmeAir |
| □ DayTrader7 |

| Benchmark Metrics |
|---|
| **Metric** |
| □ Throughput, JIT CPU total ms |
| ☑ Startup: Footprint, Startup time, Adjusted Single Server Memory |

## PerfNext Benchmark Launcher

Benchmarks | Build Options

Setup Configurations

Runtime Product: Java

Platform: Linux AMD x64 64 bit

Machine: kermit.torolab.ibm.com

Options

Machine CPU Affinity: 4 cores

☑ Run with SMT (if available)

Test Build Configurations

VM Release: Espresso / Java 8 / SR6

Build Configurations

**Product** pxa6480sr6

**Build** 401 Unauthorized Unauthorized Tl

□ Use Custom Build?

https://...<Link for SDK>...-sdk.jar

□ **Run Baseline?**

Baseline VM Release: Espresso / Java 8 / SR6

Baseline Build Configurations

**Product** pxa6480sr6

**Build** 401 Unauthorized Unauthorized Tl

□ Use Custom Build?

https://...<Link for SDK>...-sdk.jar

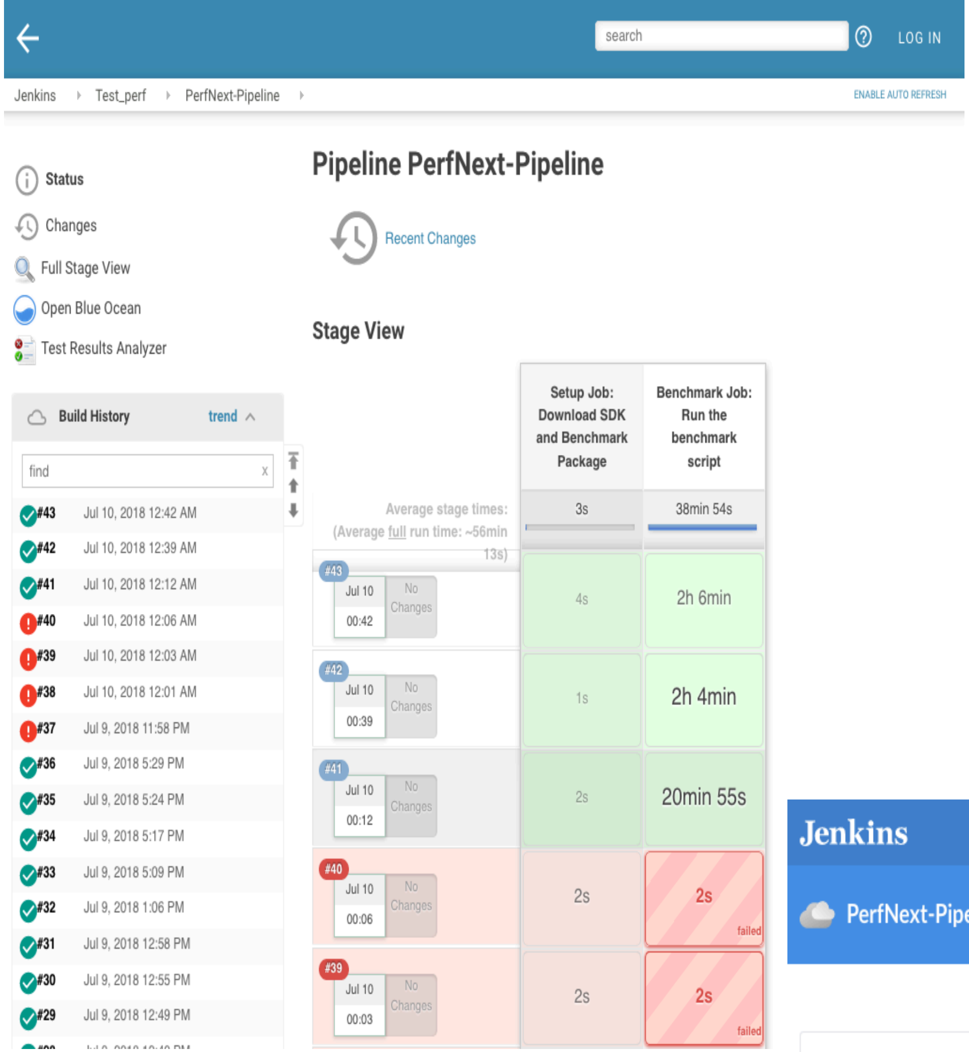
Selected Benchmarks

| Benchmark |
|---|
| LibertyStartupDT-17dev-4way-0-256-qs |

Benchmark Arguments

| Argument | Value |
|---|---|
| ***Iterations*** | 1 |
| **Scripts** | |
| bash script | bin/sufp_benchmark.sh |
| Package | liberty-cleaned |
| **Environment Variables** | |
| JDK_OPTIONS | -Xmx256m -Xdump:system:defaults:file=$RUN_DIR/core.%Y%m%d.%H%M%S.%pid.%seq.d |

# Track the progress of benchmark runs and verify their output



Jenkins › Test_perf › PerfNext-Pipeline ›

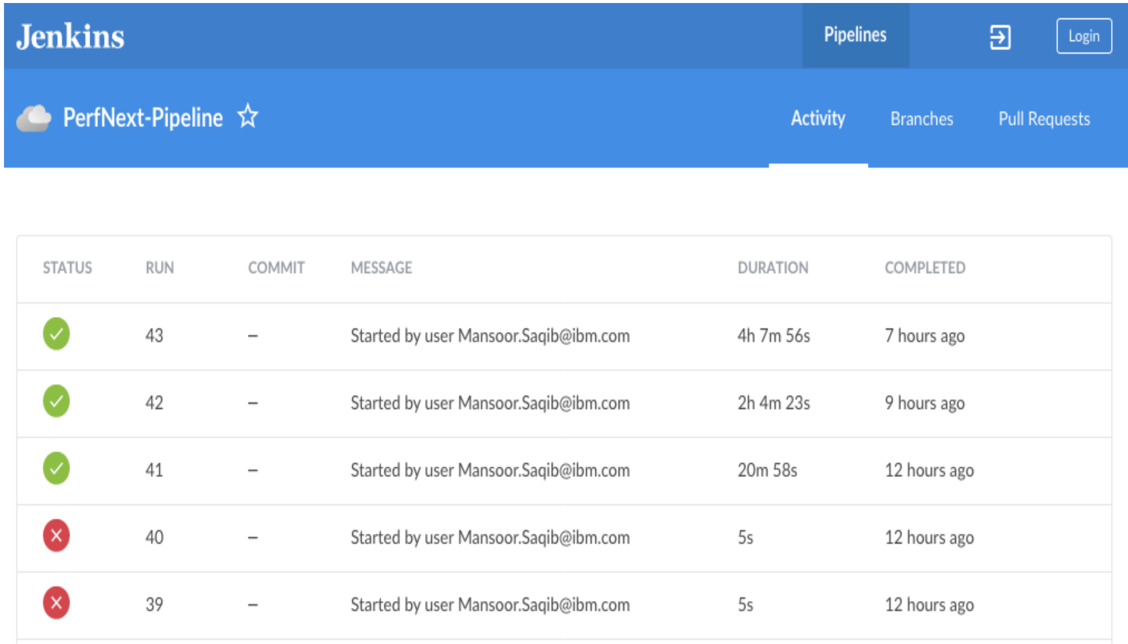
## Pipeline PerfNext-Pipeline

### Build History

| | |
|---|---|
| ✓ #43 | Jul 10, 2018 12:42 AM |
| ✓ #42 | Jul 10, 2018 12:39 AM |
| ✓ #41 | Jul 10, 2018 12:12 AM |
| ! #40 | Jul 10, 2018 12:06 AM |
| ! #39 | Jul 10, 2018 12:03 AM |
| ! #38 | Jul 10, 2018 12:01 AM |
| ! #37 | Jul 9, 2018 11:58 PM |
| ✓ #36 | Jul 9, 2018 5:29 PM |
| ✓ #35 | Jul 9, 2018 5:24 PM |
| ✓ #34 | Jul 9, 2018 5:17 PM |
| ✓ #33 | Jul 9, 2018 5:09 PM |
| ✓ #32 | Jul 9, 2018 1:06 PM |
| ✓ #31 | Jul 9, 2018 12:58 PM |
| ✓ #30 | Jul 9, 2018 12:55 PM |
| ✓ #29 | Jul 9, 2018 12:49 PM |

### Stage View

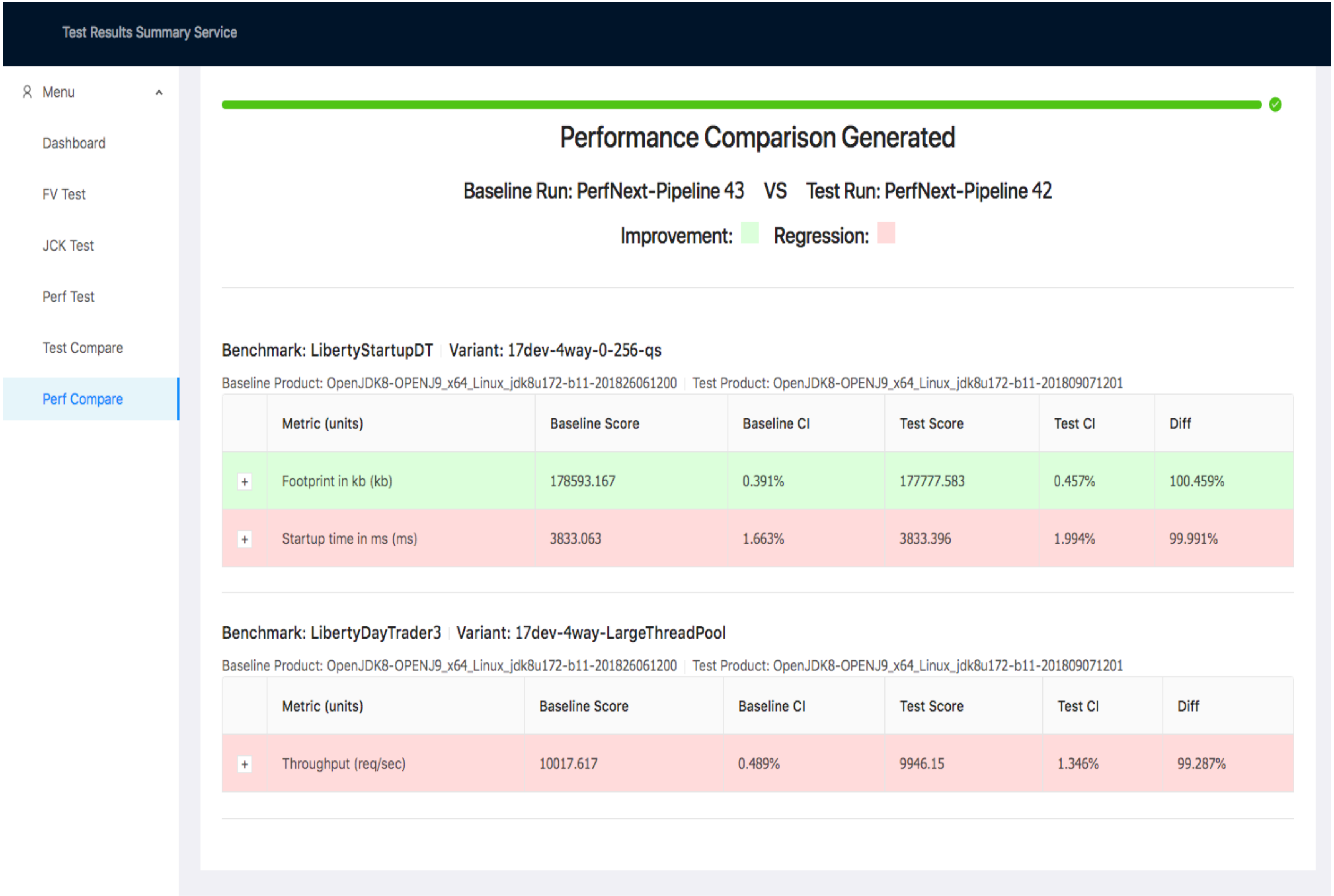
| | Setup Job: Download SDK and Benchmark Package | Benchmark Job: Run the benchmark script |
|---|---|---|
| Average stage times:<br>(Average full run time: ~56min 13s) | 3s | 38min 54s |
| #43 Jul 10 00:42 No Changes | 4s | 2h 6min |
| #42 Jul 10 00:39 No Changes | 1s | 2h 4min |
| #41 Jul 10 00:12 No Changes | 2s | 20min 55s |
| #40 Jul 10 00:06 No Changes | 2s | 2s failed |
| #39 Jul 10 00:03 No Changes | 2s | 2s failed |



PerfNext-Pipeline ☆

| STATUS | RUN | COMMIT | MESSAGE | DURATION | COMPLETED |
|---|---|---|---|---|---|
| ✓ | 43 | – | Started by user Mansoor.Saqib@ibm.com | 4h 7m 56s | 7 hours ago |
| ✓ | 42 | – | Started by user Mansoor.Saqib@ibm.com | 2h 4m 23s | 9 hours ago |
| ✓ | 41 | – | Started by user Mansoor.Saqib@ibm.com | 20m 58s | 12 hours ago |
| ✕ | 40 | – | Started by user Mansoor.Saqib@ibm.com | 5s | 12 hours ago |
| ✕ | 39 | – | Started by user Mansoor.Saqib@ibm.com | 5s | 12 hours ago |

# TRSS – Performance Comparison



## Performance Comparison Generated

Baseline Run: PerfNext-Pipeline 43 VS Test Run: PerfNext-Pipeline 42

Improvement: 🟩 Regression: 🟥

**Benchmark: LibertyStartupDT | Variant: 17dev-4way-0-256-qs**

Baseline Product: OpenJDK8-OPENJ9_x64_Linux_jdk8u172-b11-201826061200 | Test Product: OpenJDK8-OPENJ9_x64_Linux_jdk8u172-b11-201809071201

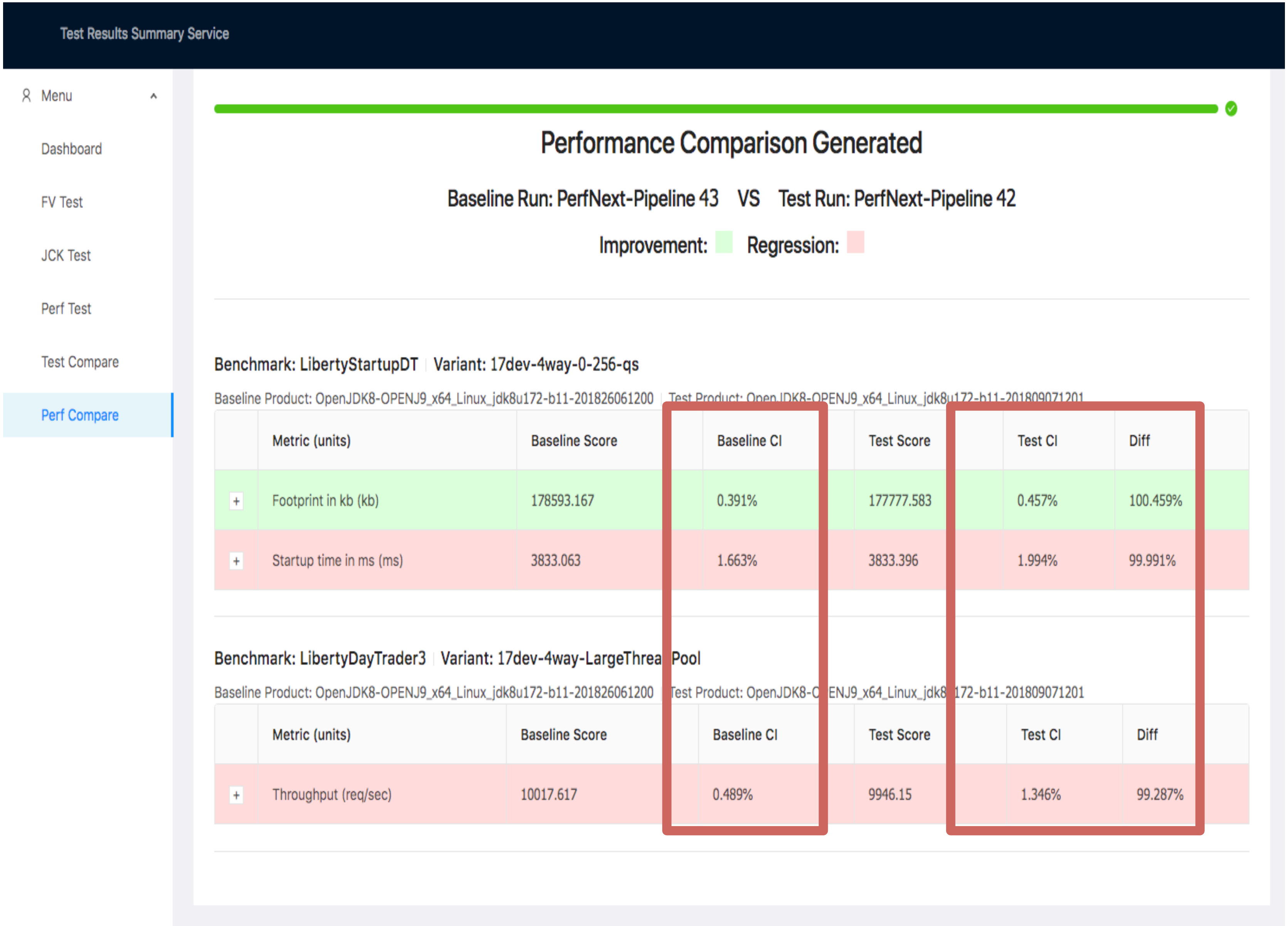
| | Metric (units) | Baseline Score | Baseline CI | Test Score | Test CI | Diff |
|---|---|---|---|---|---|---|
| + | Footprint in kb (kb) | 178593.167 | 0.391% | 177777.583 | 0.457% | 100.459% |
| + | Startup time in ms (ms) | 3833.063 | 1.663% | 3833.396 | 1.994% | 99.991% |

**Benchmark: LibertyDayTrader3 | Variant: 17dev-4way-LargeThreadPool**

Baseline Product: OpenJDK8-OPENJ9_x64_Linux_jdk8u172-b11-201826061200 | Test Product: OpenJDK8-OPENJ9_x64_Linux_jdk8u172-b11-201809071201

| | Metric (units) | Baseline Score | Baseline CI | Test Score | Test CI | Diff |
|---|---|---|---|---|---|---|
| + | Throughput (req/sec) | 10017.617 | 0.489% | 9946.15 | 1.346% | 99.287% |

# TRSS – Performance Comparison



## Performance Comparison Generated

Baseline Run: PerfNext-Pipeline 43 VS Test Run: PerfNext-Pipeline 42

Improvement: 🟩 Regression: 🟥

**Benchmark: LibertyStartupDT | Variant: 17dev-4way-0-256-qs**

Baseline Product: OpenJDK8-OPENJ9_x64_Linux_jdk8u172-b11-201826061200 | Test Product: OpenJDK8-OPENJ9_x64_Linux_jdk8u172-b11-201809071201

| | Metric (units) | Baseline Score | Baseline CI | Test Score | Test CI | Diff |
|---|---|---|---|---|---|---|
| + | Footprint in kb (kb) | 178593.167 | 0.391% | 177777.583 | 0.457% | 100.459% |
| + | Startup time in ms (ms) | 3833.063 | 1.663% | 3833.396 | 1.994% | 99.991% |

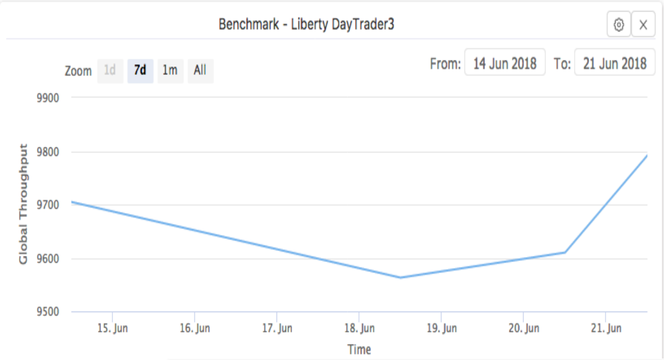
**Benchmark: LibertyDayTrader3 | Variant: 17dev-4way-LargeThreadPool**

Baseline Product: OpenJDK8-OPENJ9_x64_Linux_jdk8u172-b11-201826061200 | Test Product: OpenJDK8-OPENJ9_x64_Linux_jdk8u172-b11-201809071201

| | Metric (units) | Baseline Score | Baseline CI | Test Score | Test CI | Diff |
|---|---|---|---|---|---|---|
| + | Throughput (req/sec) | 10017.617 | 0.489% | 9946.15 | 1.346% | 99.287% |

# TRSS – Regression Analysis

7 days

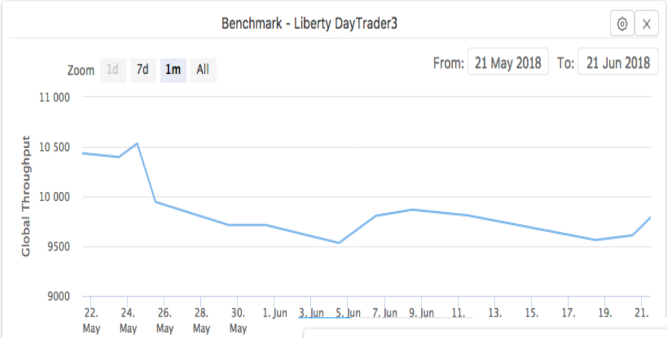


Benchmark - Liberty DayTrader3

Zoom 1d 7d 1m All

From: 14 Jun 2018 To: 21 Jun 2018

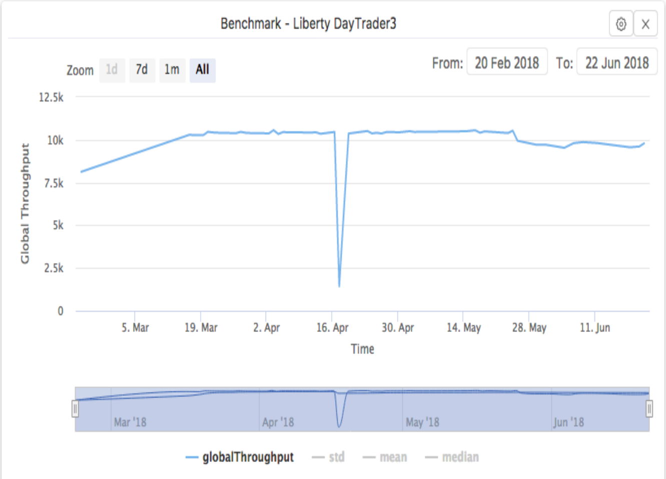
1 month



Benchmark - Liberty DayTrader3

Zoom 1d 7d 1m All

From: 21 May 2018 To: 21 Jun 2018



Benchmark - Liberty DayTrader3

Zoom 1d 7d 1m All

From: 20 Feb 2018 To: 22 Jun 2018

All data

# BumbleBench

“Microbenchmarks Simplified”

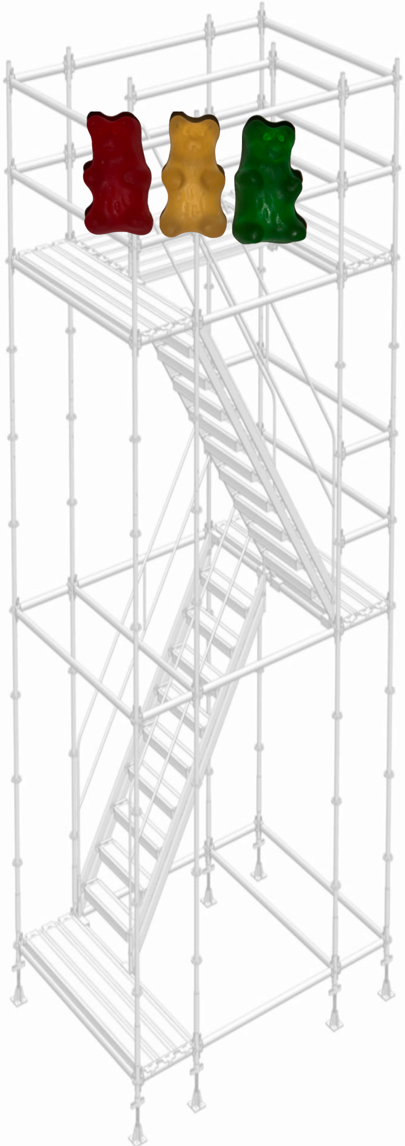
github.com/AdoptOpenJDK/bumblebench

- Writing a good microbench with an optimizing JIT running your code is hard
  - are you measuring what you think you are measuring?
- BumbleBench is a Java framework that provide a hook point to implement the benchmark payload
- Framework provides the outer timing loop, scoring infrastructure, etc.

# Conclusion

- Perf is hard (not impossible)
  - High resource requirements for full-scale testing
  - Microbenchmarks difficult to write
  - Data is noisy and subject to interpretation
- Building tools to make perf easier
  - TRSS / PerfNext / BumbleBench
    - AdoptOpenJDK git repos: openjdk-tests, openjdk-test-tools, bumblebench
    - Coming soon -> trss.adoptopenjdk.net
- Open Collaboration leads to greater Innovation
  - **“Innovation is creativity with a job to do.”** – John Emmerling

# Connect & Collaborate!



| Website | Github | Twitter |
|---|---|---|
| adoptopenjdk.net | AdoptOpenJDK/openjdk-tests | @adoptopenjdk |
| eclipse.org/openj9 | eclipse/openj9 | @openj9 |
| eclipse.org/omr | eclipse.org/omr | @eclipseomr |
| 8thdaytesting.com | smlambert | @ShelleyMLambert |

***Upcoming Talks*:**

Performance Testing for Everyone
AdoptOpenJDK: Ensuring Free Java for the Community
Fuzzy Plans and Other Test Integrations
Shaking Sticks and Testing OpenJDK Implementations