Duet Benchmarking:
Improving Measurement Accuracy in the Cloud

Speakers: Di Mo, Solmaz Seyed Monir, Andrew Lim

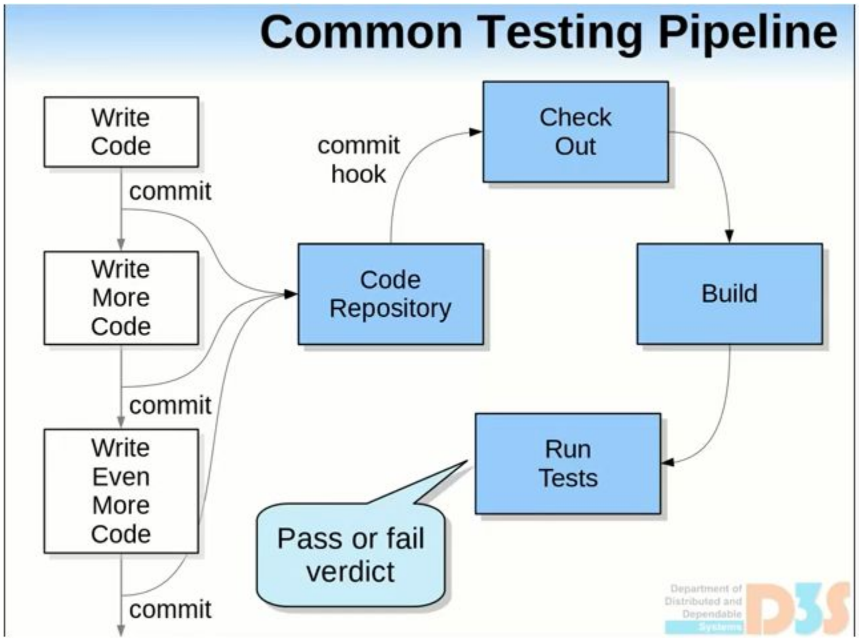
DEC 2, 2021

Bulej, Lubomír, et al. "Duet benchmarking: improving measurement accuracy in the cloud." Proceedings of the ACM/SPEC International Conference on Performance Engineering. 2020.

OUTLINE

- Introduction and Background (Di Mo)
- Experimental evaluation (Andrew Lim)
- Conclusion and Discussion (Solmaz Seyed Monir)
- Critique
- Evaluation
- Questions

Background: Regression test



Common Testing Pipeline

Write Code

commit

Write More Code

commit

Write Even More Code

commit

commit hook

Code Repository

Check Out

Build

Run Tests

Pass or fail verdict

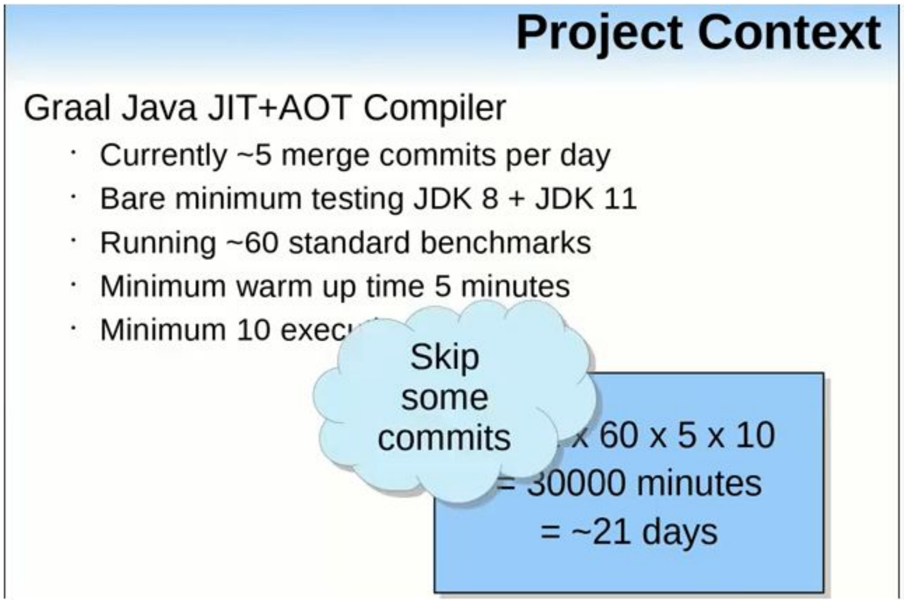
Department of Distributed and Dependable Systems D3S

Introduction: Performance Comparison

Tradeoff

- Longer experiment times vs shorter experiment times
- Average noise and accurate results, expensive vs loss of sensitivity or report false alarms

=> TESTING IN THE CLOUD



Project Context

Graal Java JIT+AOT Compiler

- Currently ~5 merge commits per day
- Bare minimum testing JDK 8 + JDK 11
- Running ~60 standard benchmarks
- Minimum warm up time 5 minutes
- Minimum 10 execu

Skip some commits

x 60 x 5 x 10
30000 minutes
= ~21 days

Introduction: Performance measurements in the cloud

Performance measurements in the cloud are noisy

Reasons:

- lack of control over hardware configuration
- overhead of virtualization
- colocated workloads of other tenants

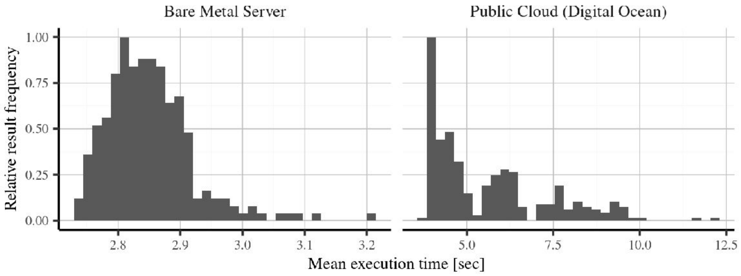


Figure 1: Distribution of observed mean execution times of the avrora benchmark, running on an otherwise idle bare-metal server and on a public cloud machine. Note the min-max range, which is about 16 % of the mean on the bare-metal server and about 150 % in the cloud.

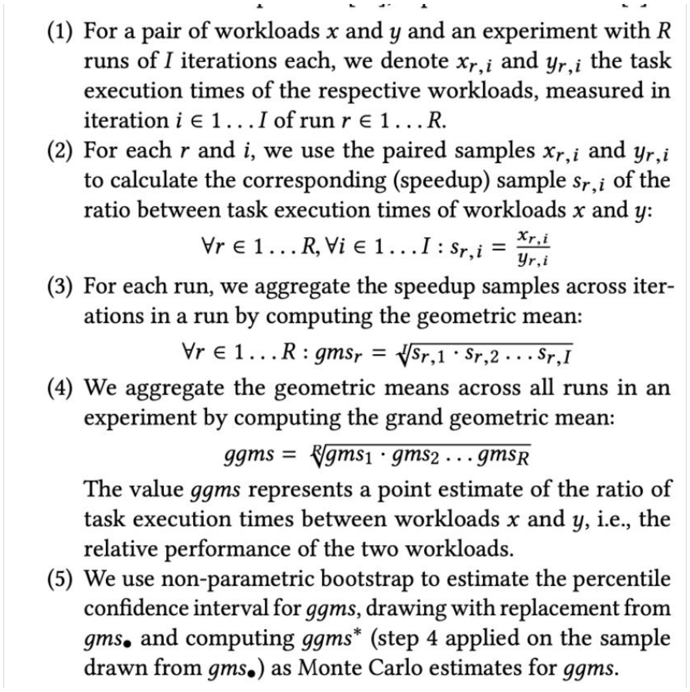
Background: duet measurement procedure

- Reduce noise in cloud:
  - Repeat the measured operation enough times
- Duet measurement procedure
  - Make 2 workloads run in parallel, inside a virtual machine with two virtual cores, with each workload restricted to one virtual core.
  - The workloads are synchronized using a shared memory barrier, so that their measured operations always start at the same time.
  - This setting ensures that any external interference on the virtual machine impacts both workloads simultaneously, which equalizes the probability of interference between the workloads for each paired measurement and thus avoids the bias immediately—rather than only for a long enough experiment.

Introduction: an overview

- Duet Benchmarking:
  - Improve the accuracy of performance comparison experiments conducted on shared machines
  - Orchestrate measured artifacts in parallel to facilitate evaluating relative performance together
- Assumption
  - Performance fluctuations due to interference tend to impact similar tenants equally
  - Minimize performance variance by maximizing the likelihood of such equal impact by executing the measured artifacts in parallel.
- Research questions:
  - RQ1. Are the performance comparisons made with the duet procedure more accurate than performance comparisons done using standard methods?
  - RQ2. Can we attribute the improved accuracy exhibited by the duet procedure to both workloads suffering from synchronized interference?
  - RQ3. Is the presence of synchronized interference associated with the existence of other workloads that share the same computing platform?
  - RQ4. How does uneven resource utilization impact the estimated workload execution time ratio ?

Confidence interval for the ratio of task execution times



(1) For a pair of workloads x and y and an experiment with R runs of I iterations each, we denote $x_{r,i}$ and $y_{r,i}$ the task execution times of the respective workloads, measured in iteration $i \in 1 \dots I$ of run $r \in 1 \dots R$.

(2) For each r and i, we use the paired samples $x_{r,i}$ and $y_{r,i}$ to calculate the corresponding (speedup) sample $s_{r,i}$ of the ratio between task execution times of workloads x and y:

$$\forall r \in 1 \dots R, \forall i \in 1 \dots I : s_{r,i} = \frac{x_{r,i}}{y_{r,i}}$$

(3) For each run, we aggregate the speedup samples across iterations in a run by computing the geometric mean:

$$\forall r \in 1 \dots R : gms_r = \sqrt[I]{s_{r,1} \cdot s_{r,2} \dots s_{r,I}}$$

(4) We aggregate the geometric means across all runs in an experiment by computing the grand geometric mean:

$$ggms = \sqrt[R]{gms_1 \cdot gms_2 \dots gms_R}$$

The value $ggms$ represents a point estimate of the ratio of task execution times between workloads x and y, i.e., the relative performance of the two workloads.

(5) We use non-parametric bootstrap to estimate the percentile confidence interval for $ggms$, drawing with replacement from $gms_{\bullet}$ and computing $ggms^*$ (step 4 applied on the sample drawn from $gms_{\bullet}$) as Monte Carlo estimates for $ggms$.

Experimental Evaluation

- Duet measurements target shared resource environments (cloud).

Public cloud

- Amazon Elastic Cloud: t3.medium, t3a.medium, m5.large, m5a.large
- Travis CI: unspecified Google Compute Engine
- GitLab CI: Digital Ocean

Private cloud

- Proxmox Virtual Environment

Bare metal

- To represent most stable baseline for comparison

Experimental Evaluation (continued)

Benchmark suites

- SPEC CPU 2017: statically compiled and optimized workloads
- ScalaBench (with DaCapo): dynamically compiled and optimized workloads

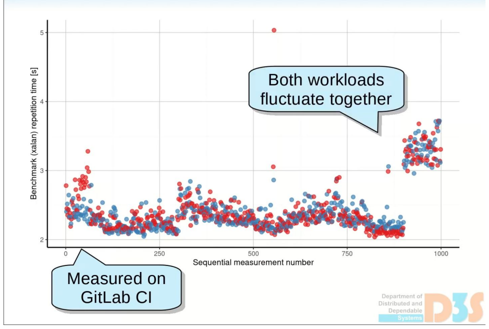
Result variance

- Execute all benchmarks multiple times
- Use random samples of 10 runs for all computations
- Faster execution: timing of first 100 iterations or first 10 minutes
- Slower execution: timing of first 100 iterations or first 60 minutes
- Filter outliers that are further than 20% away from the min-max range of the remaining observations.

RQ1: Accuracy Improvements

Are the performance comparisons made with the duet procedure more accurate than performance comparisons done using standard methods?

Two Measurements In Parallel



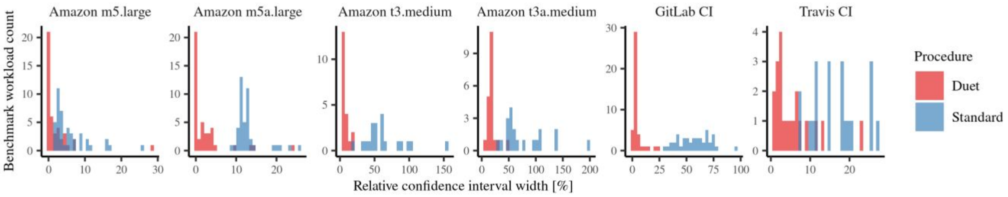
Both workloads fluctuate together

Measured on GitLab CI

Measurement accuracy

"We want to reliably detect 5% slowdowns ..."

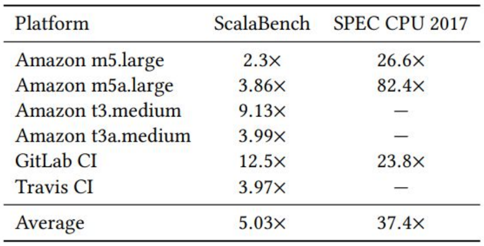
- Look at ratios instead of absolute values.
- Duet: use 99% confidence intervals for the mean of ratios.
- Standard/sequential: use 99% confidence intervals for the difference of means.
- Compare CI width relative to mean.



Narrower width = more accurate (Duet is narrower than standard)

Figure 2: Accuracy expressed as relative 99% confidence interval width, 10 runs, aggregated across all workloads.

Table 1: Average reduction in relative 99% confidence interval width from the standard procedure to the duet procedure, geomean.

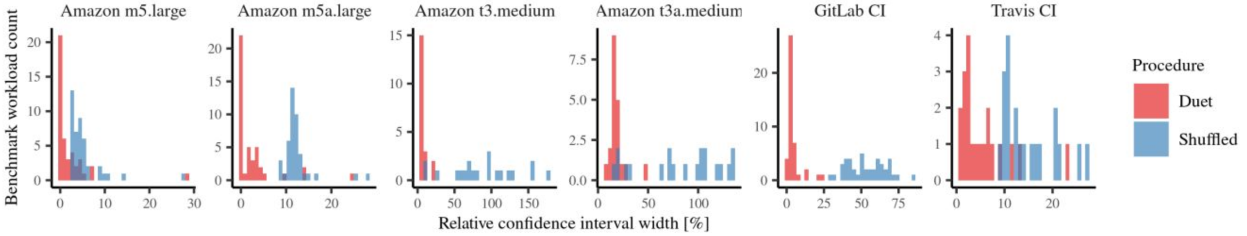


Platform	ScalaBench	SPEC CPU 2017
Amazon m5.large	2.3×	26.6×
Amazon m5a.large	3.86×	82.4×
Amazon t3.medium	9.13×	—
Amazon t3a.medium	3.99×	—
GitLab CI	12.5×	23.8×
Travis CI	3.97×	—
Average	5.03×	37.4×

RQ2: Synchronized Interference

Can we attribute the improved accuracy exhibited by the duet procedure to both workloads suffering from synchronized interference?

- Perform random shuffle and use ratios from unrelated measurements.
- Preserve all other aspects of the duet procedure, but obtain results that do not benefit from synchronized interference.
- The distribution demonstrates that the duet procedure indeed benefits particularly from synchronized interference.



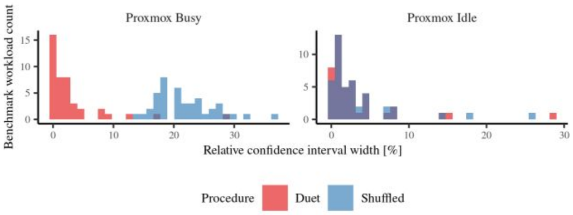
Synchronized interference is narrower (more accurate) than shuffled

Figure 4: Impact of random shuffling on relative 99% confidence interval width, 10 runs, aggregated across all workloads.

RQ3: Resource Sharing

Is the presence of synchronized interference associated with the existence of other workloads that share the same computing platform?

- Use private cloud measurements and control the utilization of the physical servers backing the virtual machine instances.
- In one set of measurements, we make sure each physical server runs only the measured workload.
- In the other set of measurements, we add a competing workload with the potential to saturate the physical server.
- Resource contention: shuffling changes the confidence intervals significantly.
- No resource contention: shuffling has almost no effect.
- Synchronized interference with duet procedure is indeed due to resource sharing.



Narrower width = better (duet is narrower during resource contention)

Figure 5: Impact of resource sharing on random shuffling in private cloud, idle vs busy with competing workload, expressed as relative 99% confidence interval width, 10 runs, aggregated across all workloads.

RQ4: Measuring Differences

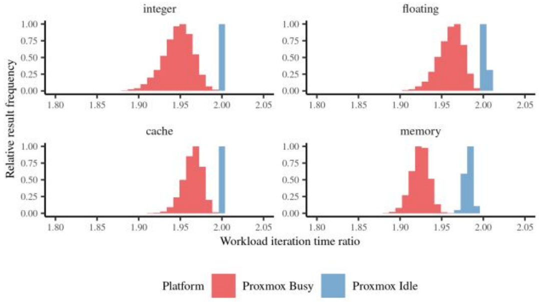
How does uneven resource utilization impact the estimated workload execution time ratio?

Private cloud

- Workloads A/B where B is twice as long as A.
- Concurrent phase: both A and B executed.
- Isolated phase: A finished and B executed in isolation.
- Desire to observe iteration times with the ratio of 2.0.
- Observed ratio is very close to 2.0 → the impact of uneven resource utilization is negligible.

Public cloud

- Compare execution time of A/A workloads to standard isolated measurements.
- Desire the ratio to be 1.0.
- Ratios either found to be close to 1.0 or bounded.
- Does not prevent detection of performance regression.



Closer to 2.0 = better (lesser impact of resource utilization)

Figure 6: Distribution of observed mean iteration time ratios for individual artificial workloads in private cloud, idle vs busy with competing workload, 10 runs.

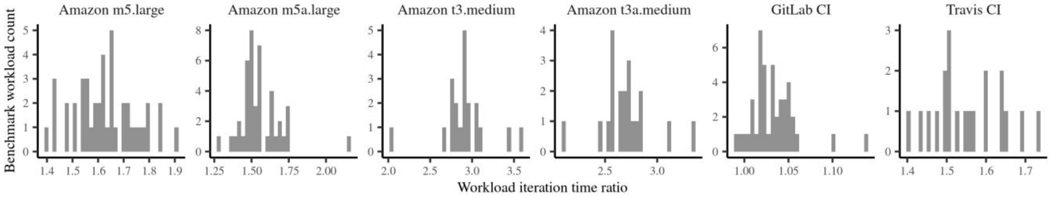


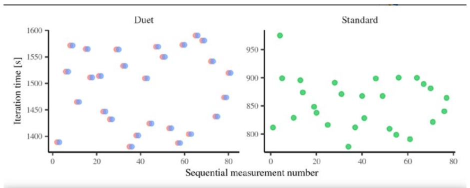
Figure 7: Distribution of observed ratios of mean iteration times between A/A duet procedure measurements and standard isolated measurements.

CONCLUSION

- For SPEC CPU 2017, ScalaBench and DaCapo workloads, duet measurement in the cloud appears to be more accurate than current methods.
- The improved accuracy is due to the paired workloads being subjected to synchronized external disturbance
- Duet measurement might result in resource competition between paired workloads and inequitable resource consumption patterns. As a result, these effects are either minor or constrained, and hence do not prevent performance regressions from being detected.

Critique: Strengths

1. Existing approaches[1] based on sequential measurements are much less precise than the new approach.
   a. Accuracy achievable in the cloud with standard measurement methods.
   b. Performance comparisons produced with the duet approach are more accurate



Individual measurement samples for the 503.bwaves r workload on the Amazon m5a.large platform. Colors in the duet procedure distinguish samples collected in parallel.

2. The A/A test results show experiments with small sample sizes suffer from high false-positive rates, irrespective of which statistical test, sampling strategy, and execution environment
   a. Duet procedure improves on this result.
3. The duet measurement approach eliminates systematic bias by equalizing interference probability, however it is specifically designed for experiments evaluating the performance of two (related) workloads

[1] Laaber et al

Critique: Weaknesses

1. The Abstract should clearly present your thesis statement
2. Well written academic articles are based on a great deal of research and the author has drawn conclusions from a range of sources.
3. As a solution, randomized trial ordering[2] is proposed. The assignment of workloads to processors is also random in duet measurements. Using same approach
4. Figure 7 and Figure 1 are not well present visualizations. Applying color to different parts of plot visualization lets you tell a more effective story.
5. The authors appear to only test with ScalaBench and SPEC CPU 2017. Will the results hold up for other pairs of benchmarks? Renaissance[3] shows that the performance differences are more significant than on existing suites such as such as DaCapo, ScalaBench
6. The measurement accuracy metrics may not work with performance expressed as a ratio, combining this work with duet approach is not always straightforward.
7. When comparing workloads with drastically different bottleneck resources, such as CPU-bound and I/O-bound workloads, the duet approach may not increase accuracy.

[2] A. Abedi and T. Brecht. 2017. Conducting Repeatable Experiments in Highly Variable Cloud Computing Environments. In ICPE. ACM.
[3] https://dl.acm.org/doi/abs/10.1145/3314221.3314637

Future Work

- Without dedicated instances, this strategy can also increase the correctness of CI/CD pipelines ?

References

1. https://www.semanticscholar.org/paper/Duet-Benchmarking%3A-Improving-Measurement-Accuracy-Bulej-Hork%C3%BD/0512a6a26efea1e91992b408e4ed3e0140cf3bc2/figure/3
2. https://dl.acm.org/doi/10.1145/3358960.3379132
3. S. He, G. Manns, J. Saunders, et al. 2019. A Statistics-Based Performance Testing Methodology for Cloud Applications. In ESEC/FSE. ACM, New York, NY, USA
4. C. Laaber, J. Scheuner, and P. Leitner. 2019. Software Microbenchmarking in the Cloud. How Bad is it Really? Empirical Software Engineering (2019)

Questions