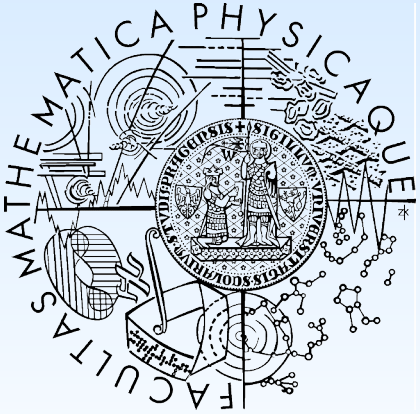
Measurement Accuracy

Expectations vs Reality in Performance Testing

Lubomír Bulej
Petr Tůma

Department of Distributed and Dependable Systems



FACULTY OF MATHEMATICS AND PHYSICS
CHARLES UNIVERSITY IN PRAGUE

What We Do

Performance regression testing

- running performance tests on each commit
- detecting and reporting changes in performance

Our particular **context**

- development of **Just-In-Time compiler**
- on top of standard **Java Virtual Machine**
- performance tested using many **benchmarks**
  - DaCapo
  - ScalaBench
  - SPECjvm2008
  - SPECjbb2015
  - In house (micro)benchmarks

We should not only prevent catastrophic failures but especially help direct development

Department of Distributed and Dependable Systems
D3S

Not Quite DevOps

Not pushing to live deployment

Not measuring live deployment (yet)

But (hopefully) some common points

- benchmark scores close to **request level metrics**
- performance testing part of **build pipeline**
  - for now only reporting
  - working on gating
- **automated** change detection
- reporting to developers

Accuracy Requirements

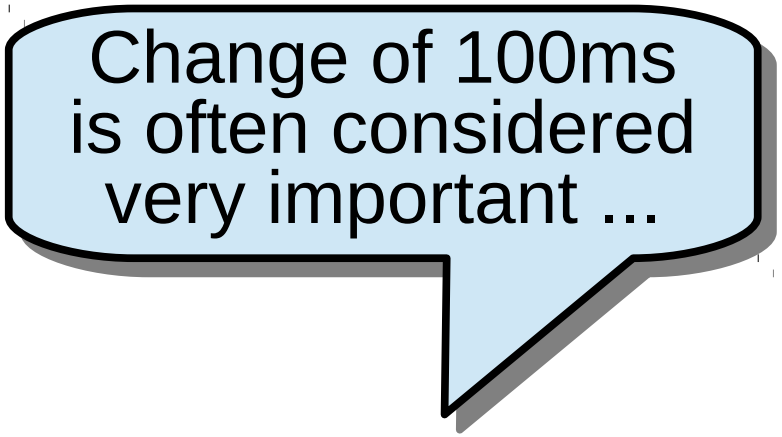
Some Anecdotal Requirements

Amazon

- Every 100ms load time increase is 1% sales decrease

Google

- Change from 0.4s to 0.9s reduces ad revenues by 20%



Walmart

- For every 100ms page load improvement there is 1% revenue growth

Microsoft

- Simple 2s delay in search results is 4.3% revenue drop

Curse of Ten Fingers

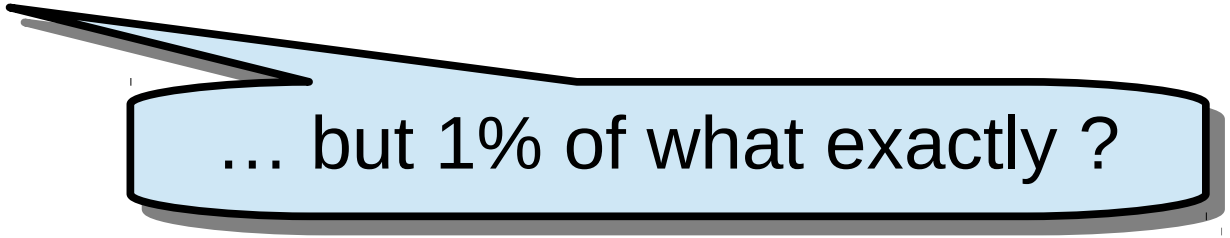
Do you need to detect **10%** performance change ?

Absolutly ! **Sounds like a lot**, we need to do better.

Do you need to detect **0.1%** performance change ?

Sounds like a tiny change, no …

So **1% is it** ?



Confidence Intervals

"**Average** benchmark execu

We will be wrong about 1% of the time.

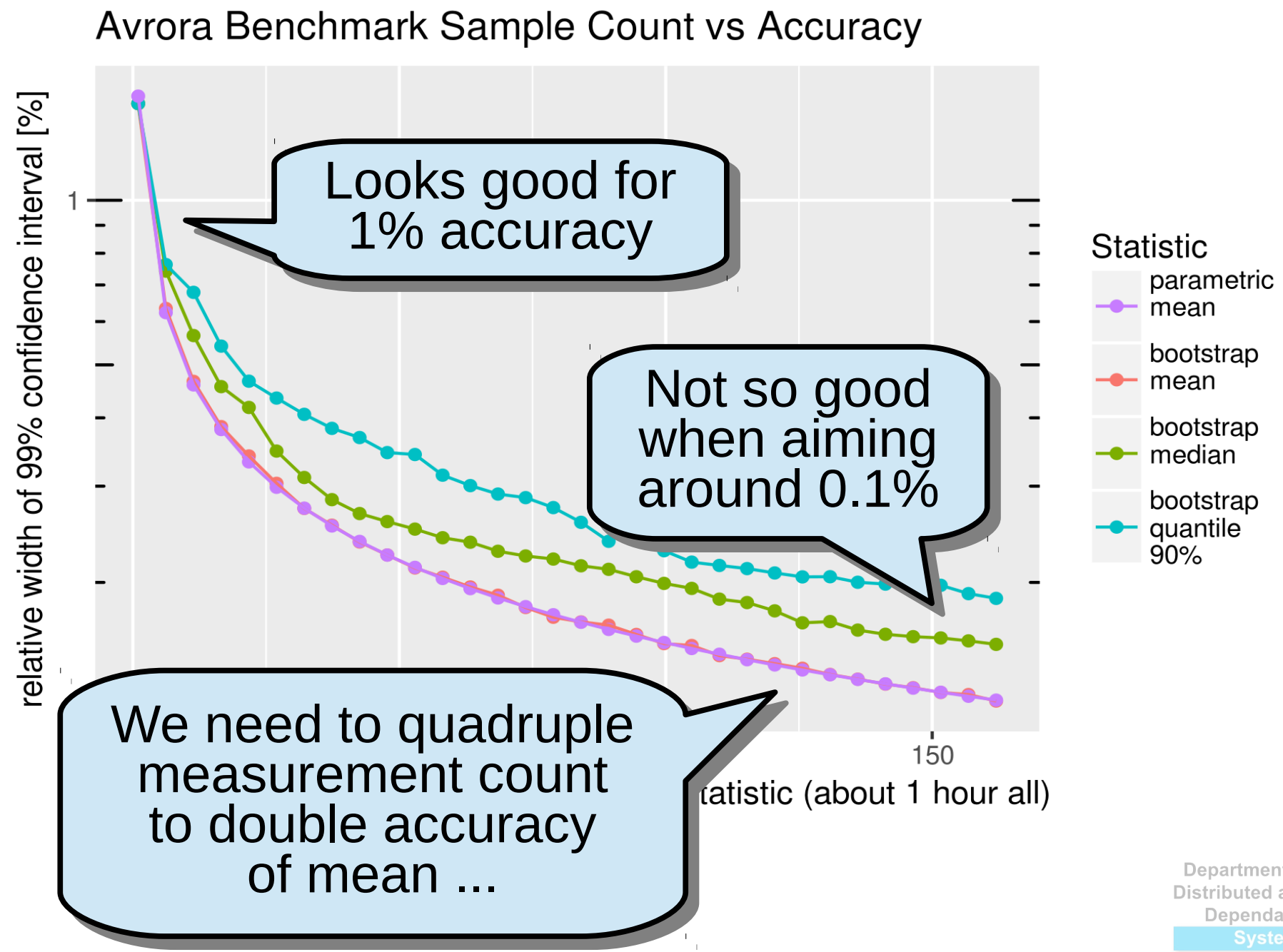
"True benchmark execution time **is between 570ms and 630ms** with **confidence 99%.**"

"The 99% confidence interval width **relative to the average** is 10%."

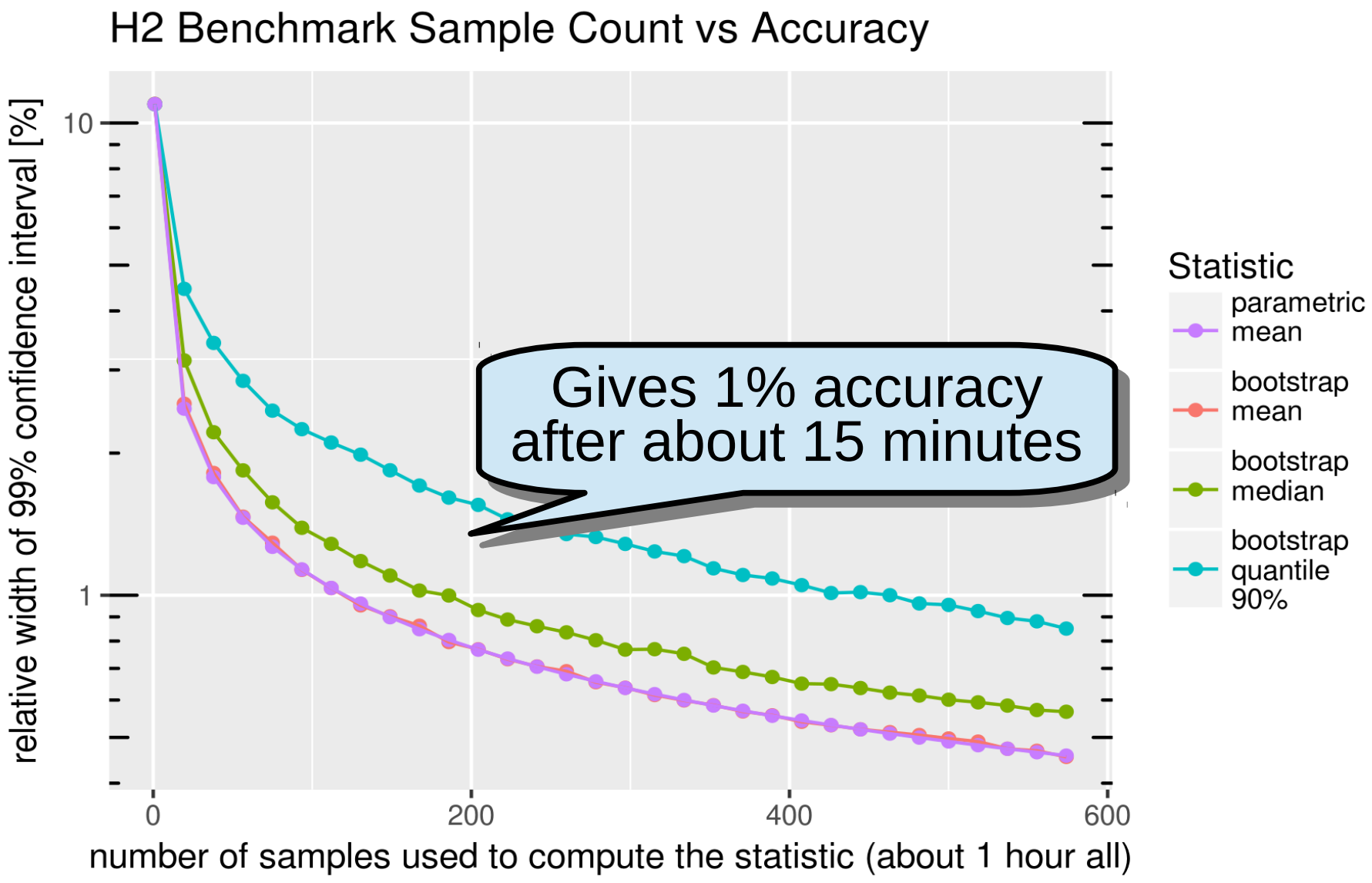
Makes most sense with symmetric intervals.

Measurement Variability

More Data Gives More Accuracy

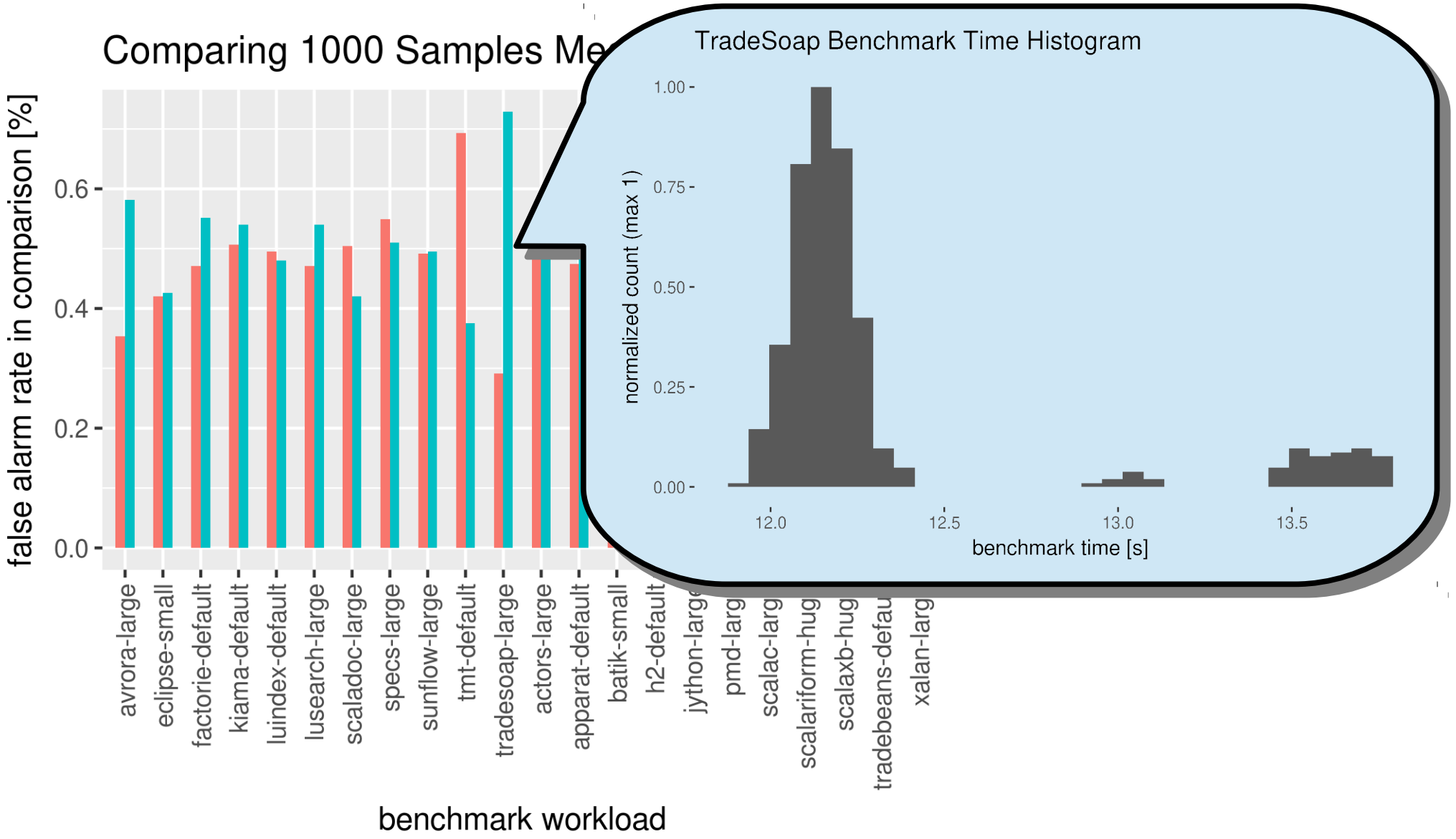


Some Benchmarks Need More Data



Department of
Distributed and
Dependable
Systems

Right Skew Plays Tricks



Sample Dependency

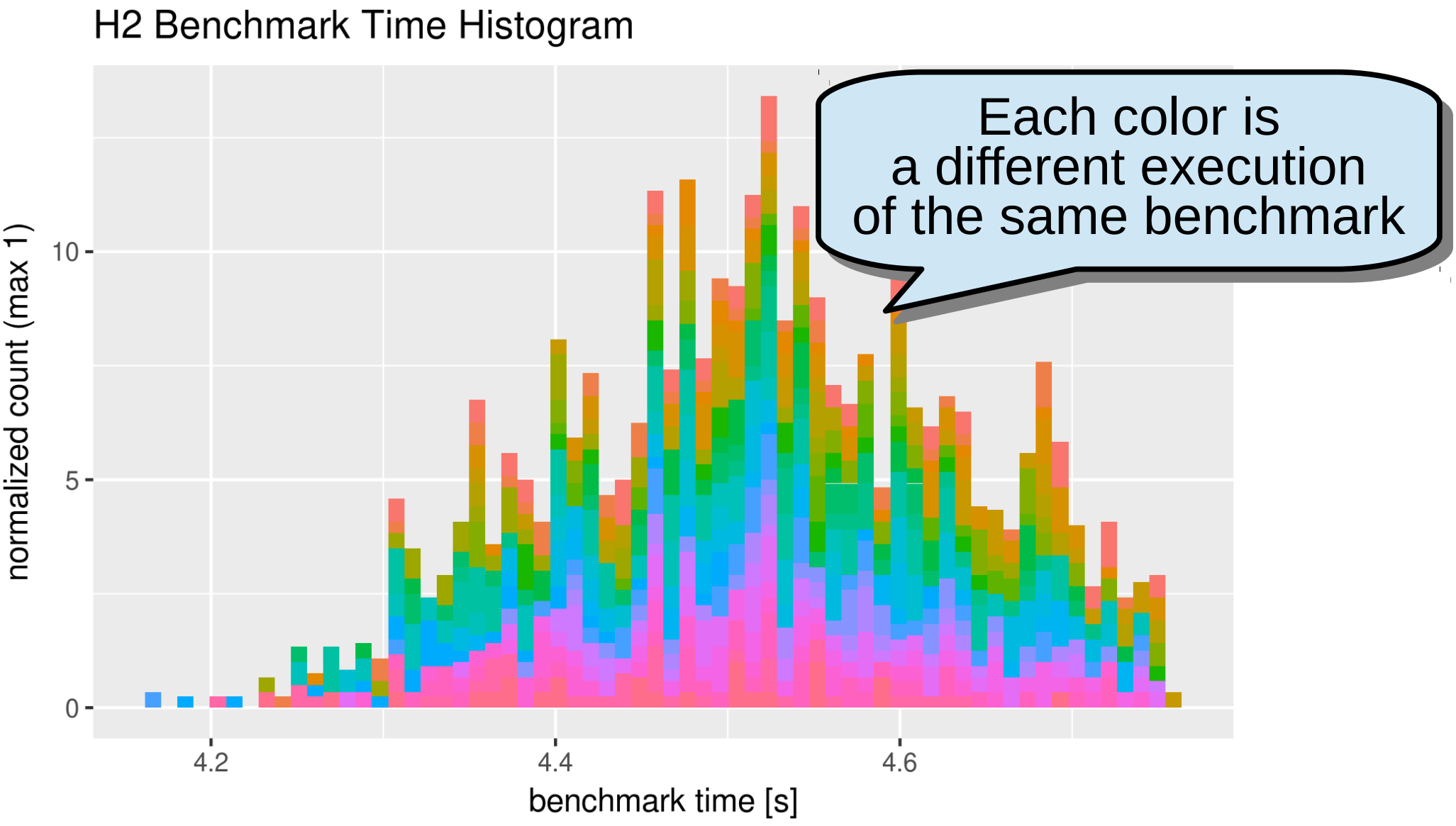
If samples within execution depend on each other, then perhaps **no single execution is entirely representative ?**

Some reasons this can happen

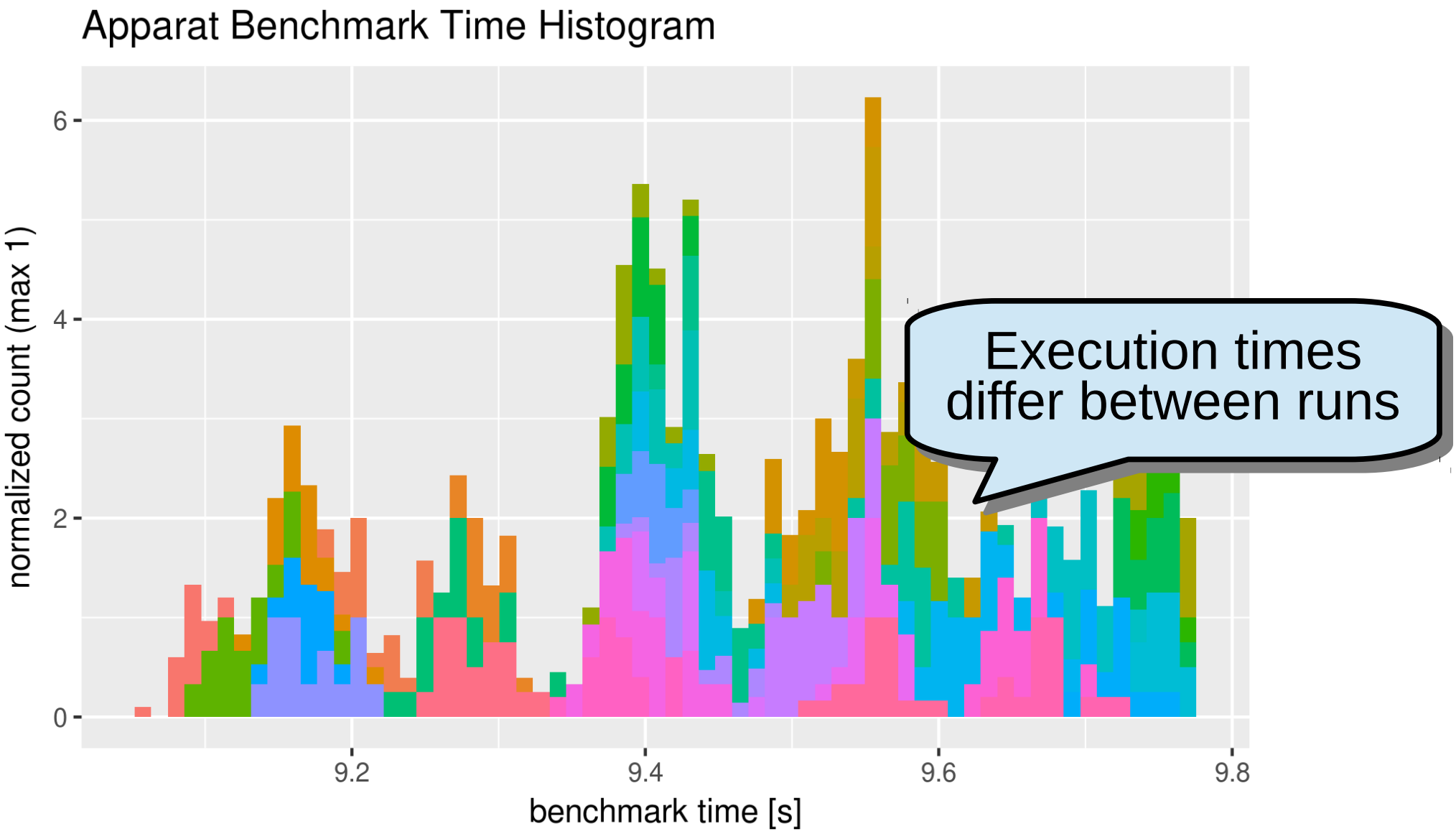
- Different compilation decisions
- Virtual machine ergonomics
- Physical memory allocation
- …

Does this matter or do various effects **average each other out** ?

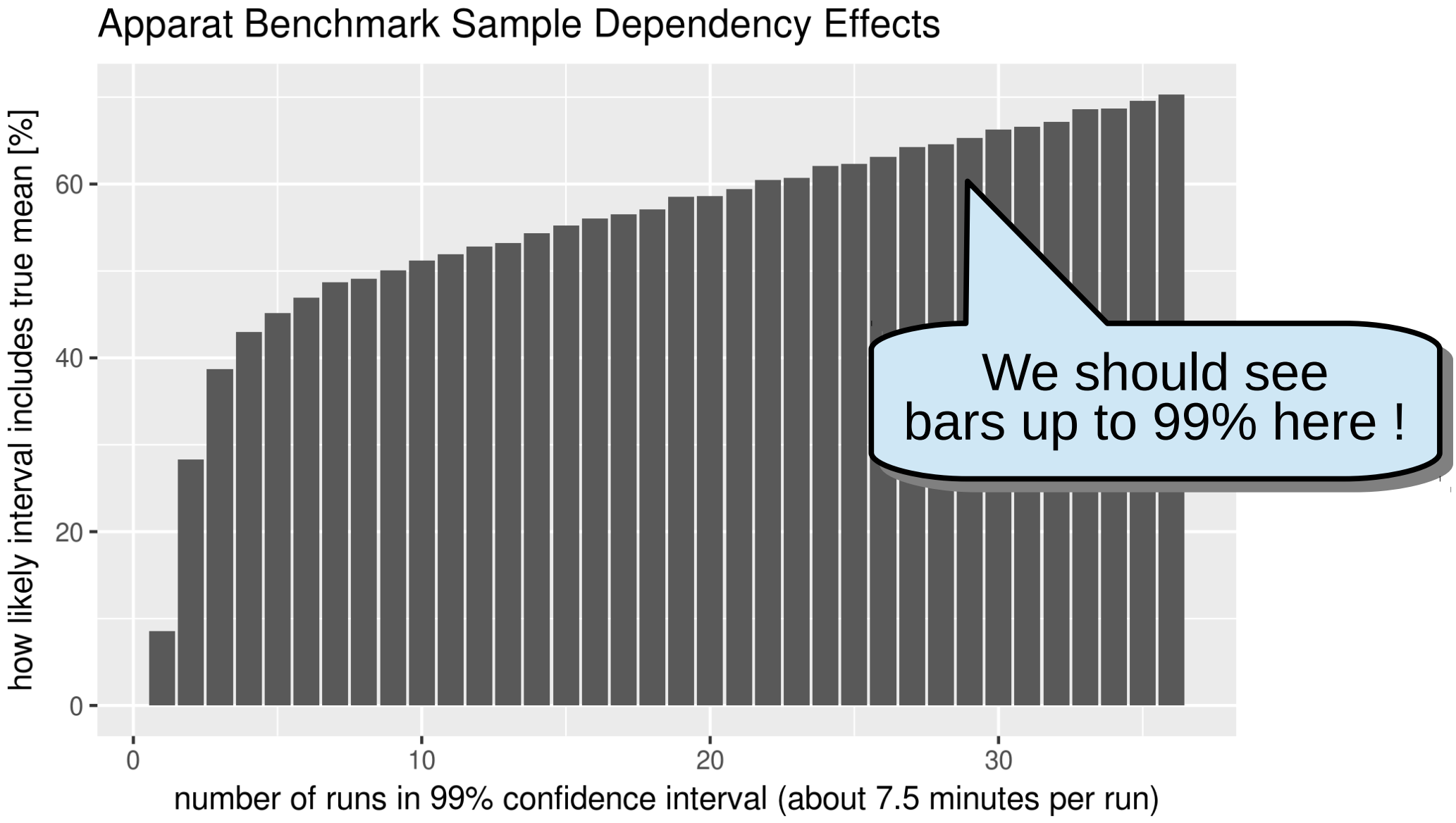
Sample Dependency



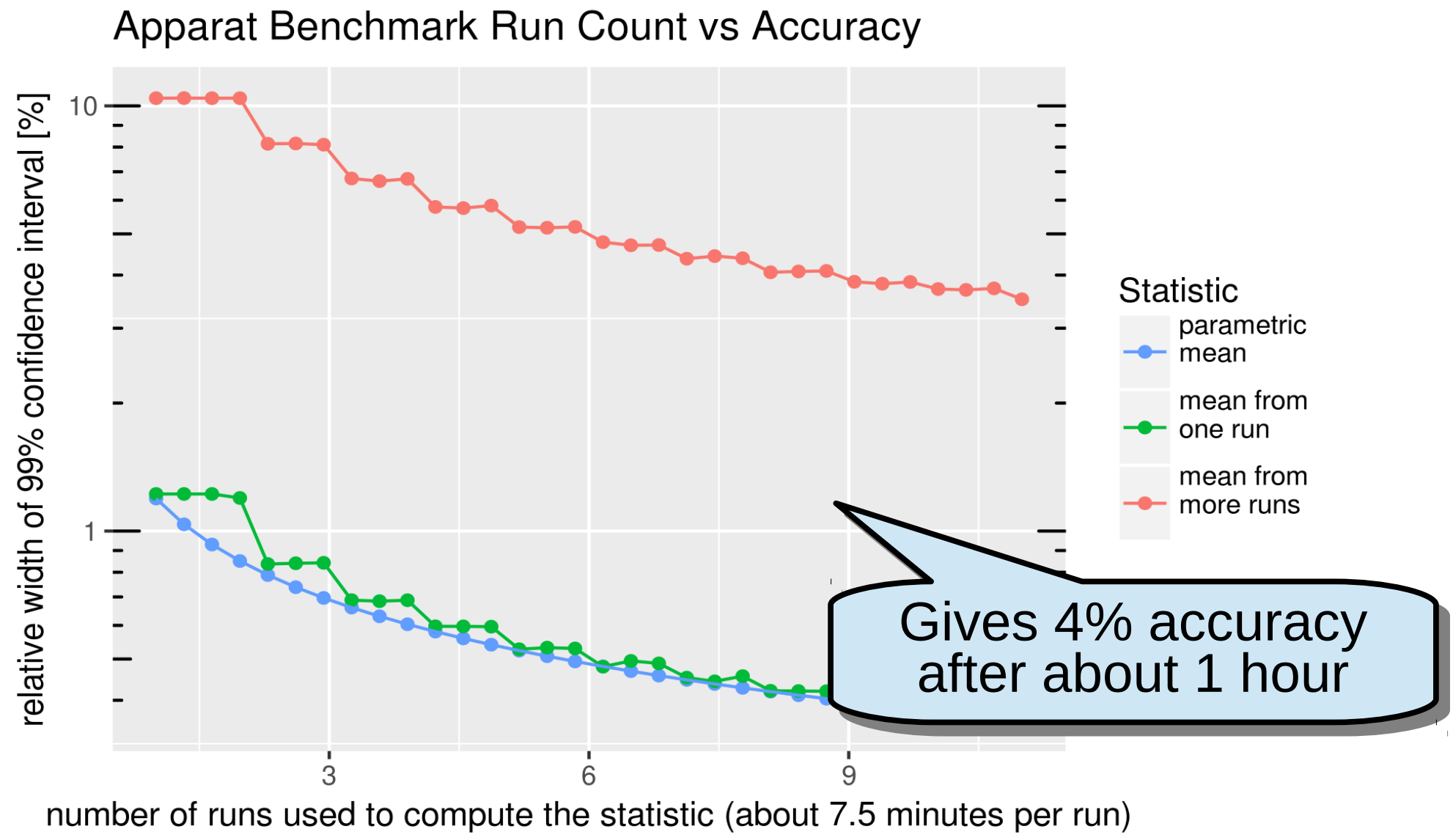
Sample Dependency



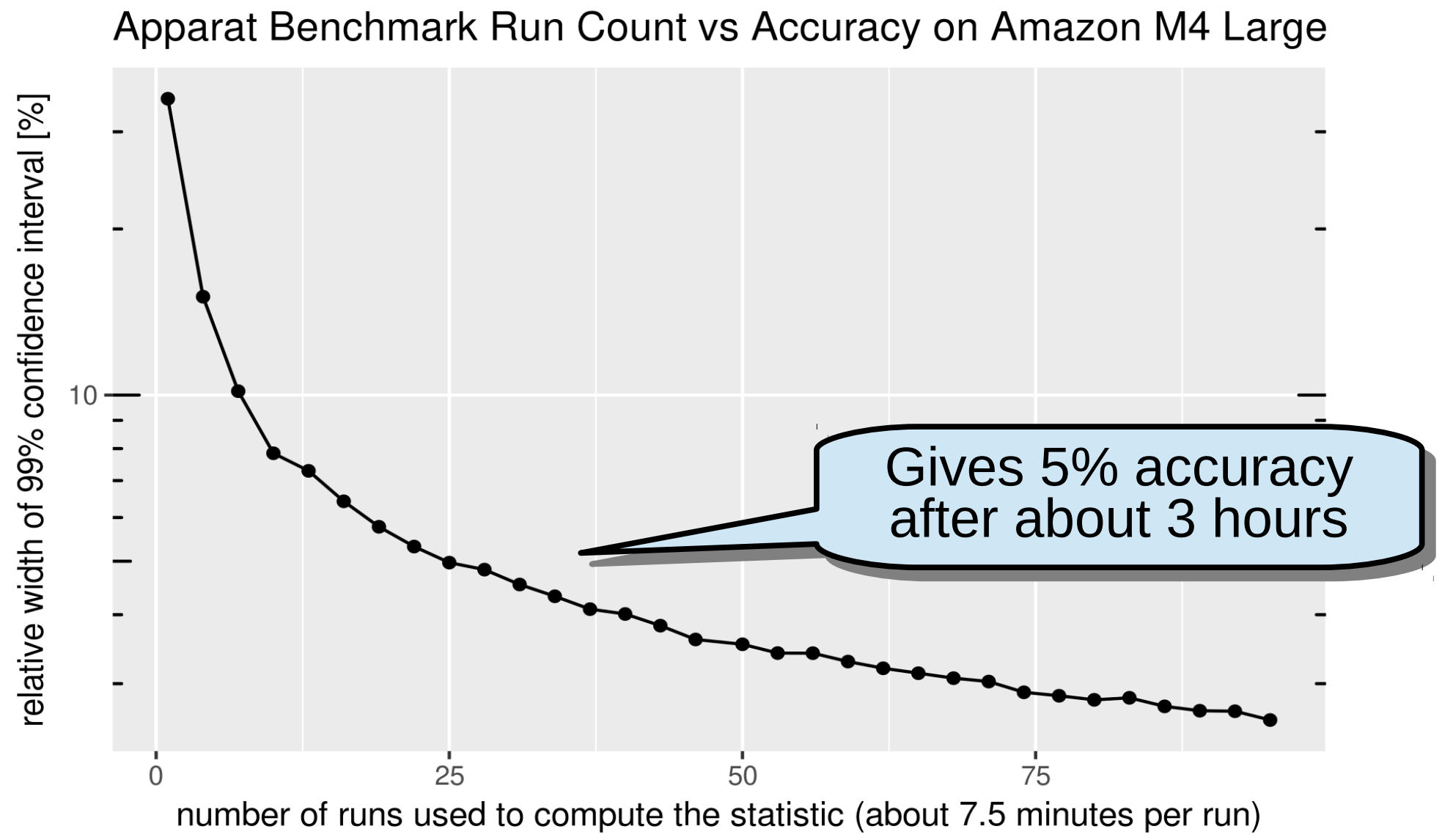
Sample Dependency Effects



Accuracy With More Executions



Accuracy With Cloud Executions



Take Away ?

Accuracy Is Expensive !

Many **tools not helpful** at all

- single pass per commit in continuous build systems
- test designs do not support easy repetition
- improper statistical computations

Limit number of performance **tests**

- good notion of test coverage ?
- global impact of local changes ?

Limit work done on each commit

- selective testing rather than each commit
- only basic tests inside deployment pipeline

Configuration versioning

- changes due to configuration difficult to track

Thank you !

More information at
http://d3s.mff.cuni.cz

Department of Distributed and Dependable Systems