Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Descartes research

The Science of Systems Benchmarking

Samuel Kounev
University of Würzburg

Keynote Talk, CLOSER 2021, April 29, 2021
Slides available for download at http://descartes.tools

Agenda

- Background and Motivation
- Benchmarking Education
- Benchmark Standardization
- Cloud Benchmarking
  - Measuring and quantifying elasticity
  - Reproducibility of experimental evaluation

Definition: Benchmark

- Originally: "*a mark on a workbench used to compare the lengths of pieces so as to determine whether one was longer or shorter than desired*"

- For computers: "*a test, or set of tests, designed to compare the performance of one computer system against the performance of others*"
  From SPEC's Glossary

  - Performance: "the amount of useful work accomplished by a computer system compared to the time and resources used" (Wikipedia)

*"You can't **control** what you can't measure?"* (DeMarco)

*"If you cannot measure it, you cannot **improve** it"* (Lord Kelvin)

Definition: Benchmark

- Modern benchmarks can be been seen as evaluating performance in a broader sense

- **Broader Benchmark Definition**:
  - „A tool coupled with a methodology for the evaluation and comparison of systems or components with respect to specific characteristics, such as performance, reliability, or security.“

Systems Benchmarking: For Scientists and Engineers, S. Kounev, K.-D. Lange, and J. von Kistowski (2020). Springer International Publishing, 1st edition, ISBN: 978-3-030-41704-8, DOI: 10.1007/978-3-030-41705-5.

Motivating Example

- Execution times of two programs on three different servers. Assuming that both programs are equally important, which server is the fastest on average?

	Server 1	Server 2	Server 3
Program A	10 sec	10 sec	5 sec
Program B	1000 sec	500 sec	1000 sec
Average	505	255	502.5

⇩ Speedup in relation to Server 1

	Server 1	Server 2	Server 3
Program A	1	1	2
Program B	1	2	1
Average	1	1.5	1.5

Average Speedup

	Server 1	Server 2	Server 3
Average	1	1.5	1.5
Rank	2	1	1

Speedup relative to Server 1

	Server 1	Server 2	Server 3
Average	0.75	1.25	1.0
Rank	3	1	2

Speedup relative to Server 3

	Server 1	Server 2	Server 3
Average	0.75	1.0	1.25
Rank	3	2	1

Speedup relative to Server 2

Average Speedup using Geom. Mean

	Server 1	Server 2	Server 3
Geom. Mean	$1^{½}$	$2^{½}$	$2^{½}$
Rank	2	1	1

Speedup relative to Server 1

	Server 1	Server 2	Server 3
Geom. Mean	$0.5^{½}$	$1^{½}$	$1^{½}$
Rank	2	1	1

Speedup relative to Server 3

	Server 1	Server 2	Server 3
Geom. Mean	$0.5^{½}$	$1^{½}$	$1^{½}$
Rank	2	1	1

Speedup relative to Server 2

Benchmarking, not Benchmarketing!

"**benchmark, v. trans**. - *To subject (a system) to a series of tests in order to obtain prearranged results not available on competitive systems*."

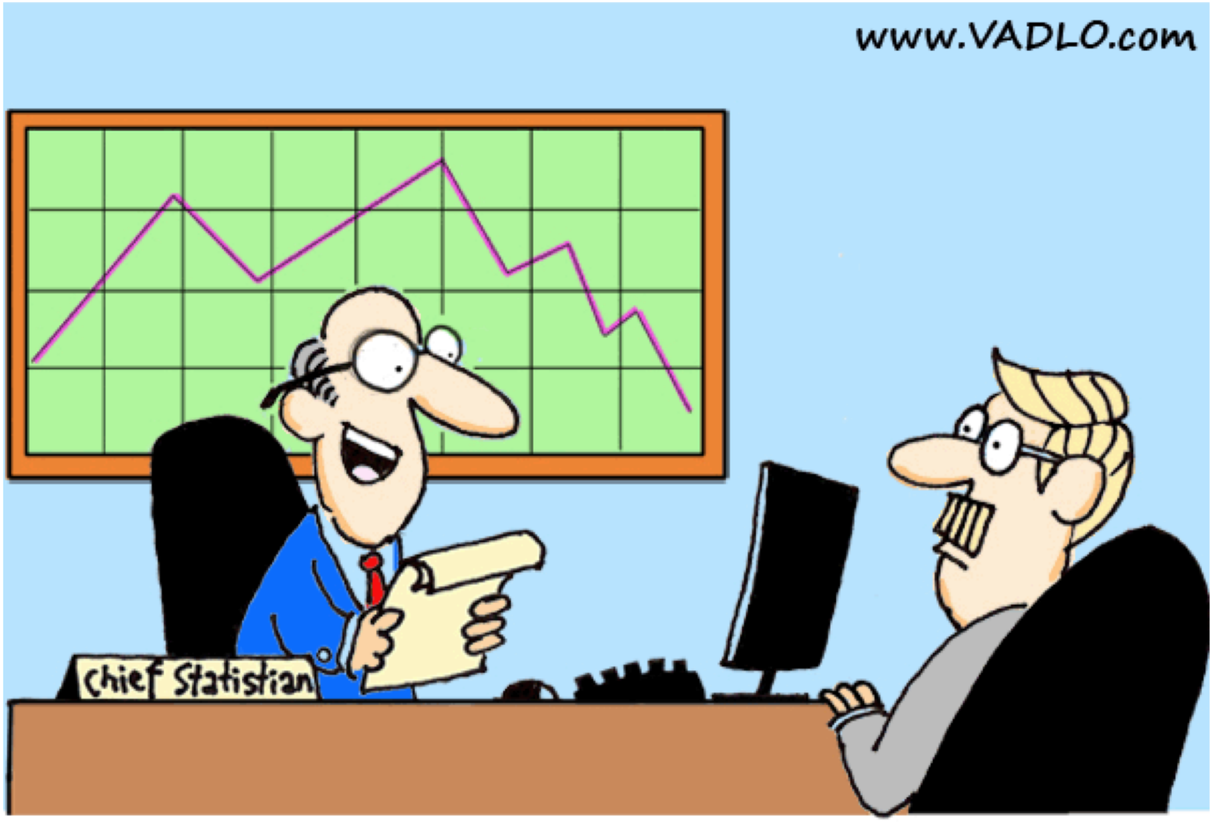
-- S.Kelly-Bootle
The Devil's DP Dictionary

"It is easy to lie

with statistics.

It is hard to tell the truth

***without statistics.*" — A. Dunkels**



"I can prove it or disprove it! What do you want me to do?"

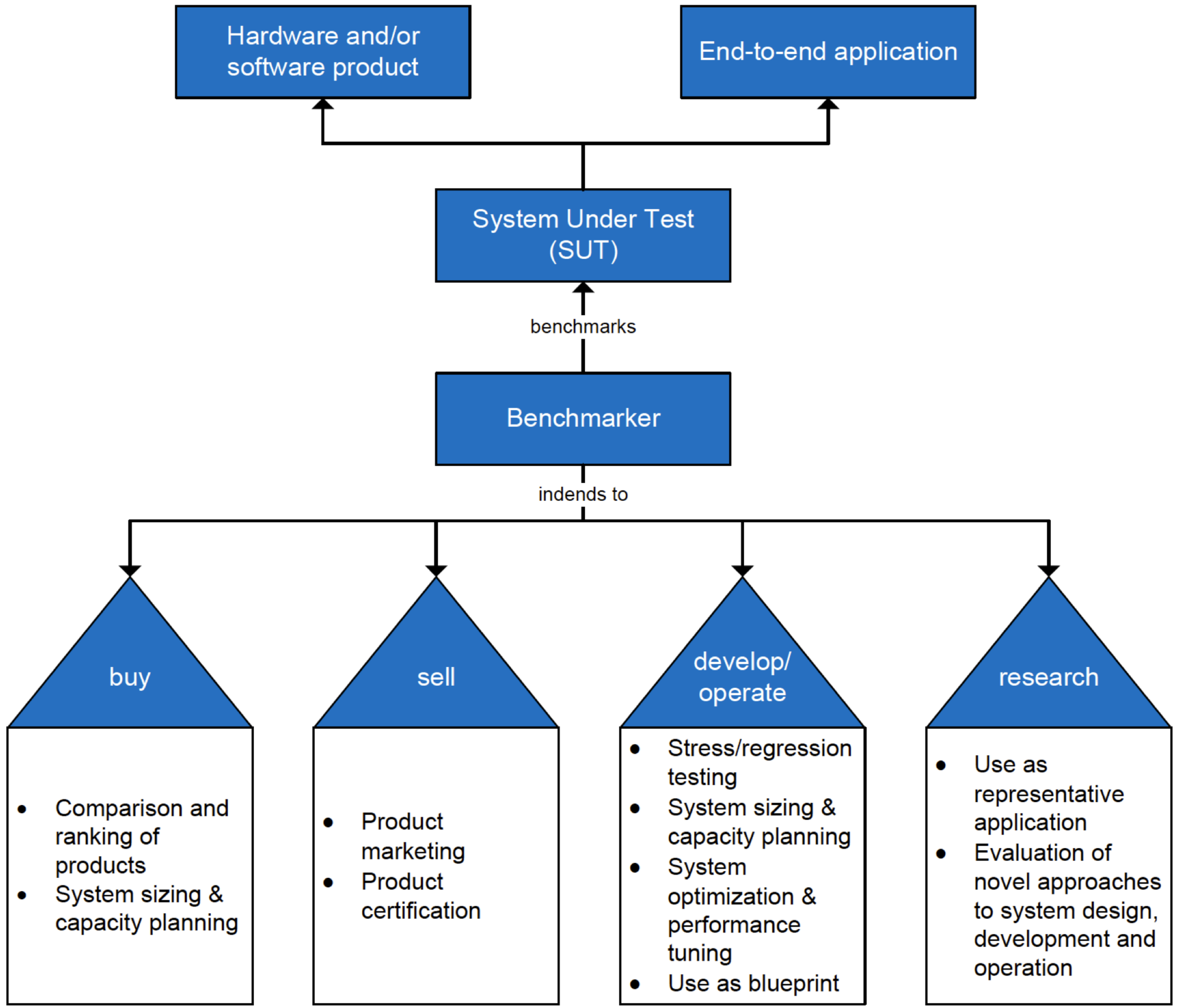
Beyond Marketing

From a Nature-published survey by Baker from 2016:

- 70% of the 1,500 researchers surveyed have tried and failed to reproduce prior work done by others, and
- over 50% failed to reproduce their own experimental results

M. Baker, “Is there a reproducibility crisis?” Nature, vol. 533, pp.452–454, 2016.

Scope of Benchmarking as a Discipline



Key Components of Each Benchmark

1. Reliable Metrics

What exactly should be measured and computed?

2. Representative Workloads

For which usage scenarios and under what conditions?

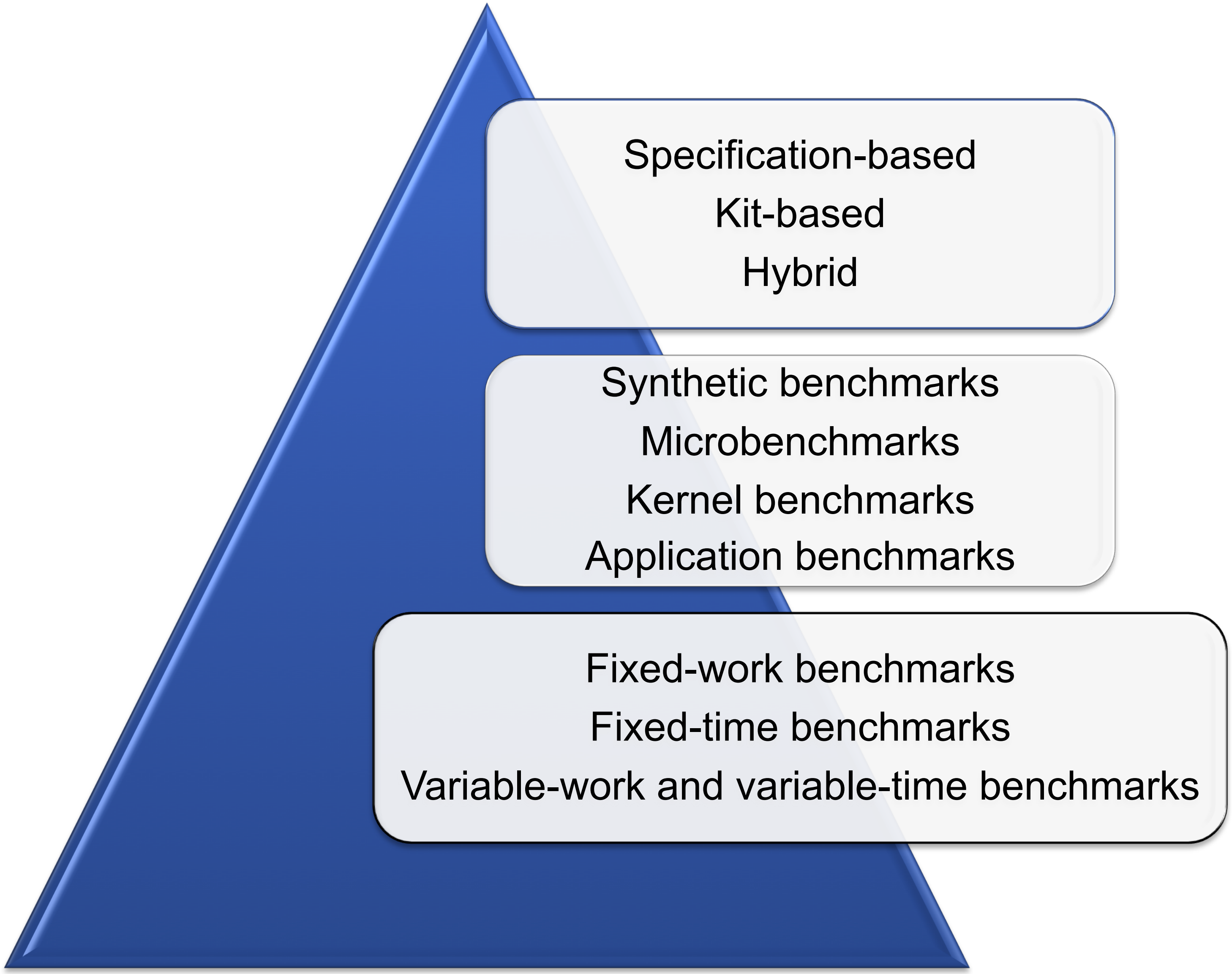
3. Sound Measurement Methodology

How should measurements be conducted?

"*To **measure** is to **know**.*" -- Clerk Maxwell, 1831-1879

"*It is much easier to make **measurements** than to **know** exactly what you are measuring.*"
-- J.W.N.Sullivan (1928)

Benchmark Classification



Benchmarking Quality Criteria

1. Relevance

- How closely the benchmark behavior correlates to behaviors that are of interest to users

2. Reproducibility

- Producing consistent results when the benchmark is run with the same test configuration + the ability for another tester to independently reproduce the results in another but identical environment.

3. Fairness

- Allowing different test configurations to compete on their merits without artificial limitations

4. Verifiability

- Providing confidence that a benchmark result is accurate

5. Usability

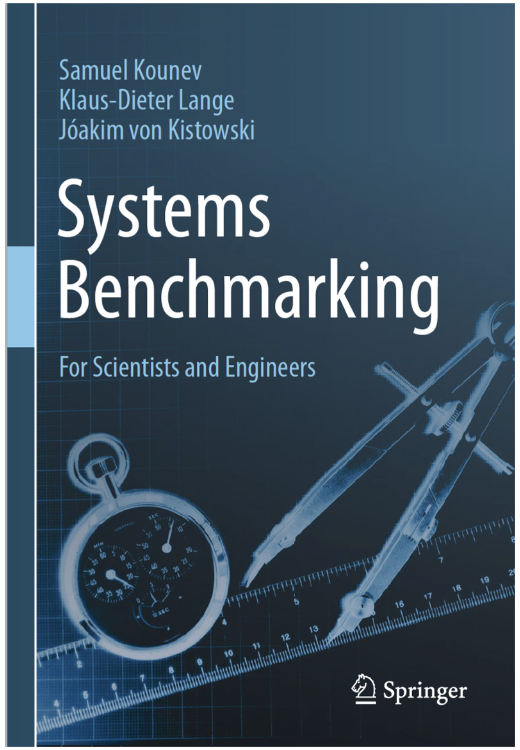
- Avoiding roadblocks for users to run the bench-mark in their test environments

Agenda

- Background and Motivation
- **<u>Benchmarking Education</u>**
- Benchmark Standardization
- Cloud Benchmarking
  - Measuring and quantifying elasticity
  - Reproducibility of experimental evaluation

New Course on Systems Benchmarking

- Theoretical and practical foundations
- Both a textbook and a handbook on benchmarking
- Two parts: foundations and applications
- Includes modern applications, case studies, and latest research based on input from over 40 experts
- Teaching materials will be made available soon!

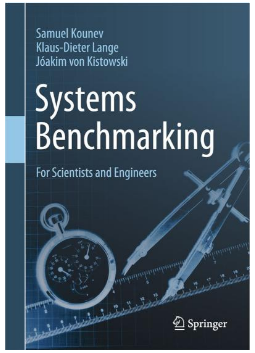


„This book should be required reading for anyone interested in making good benchmarks."
– David Patterson, 2017 ACM A.M. Turing Award Laureate

http://benchmarking-book.com

A Long-Term Joint Effort

http://benchmarking-book.com



Authors: Samuel Kounev, Klaus-Dieter Lange, and Jóakim von Kistowski

Foreword: David Patterson and John R. Mashey

Contributors:

Jeremy A. Arnold, André Bauer, John Beckett, James Bucek, Ken Cantrell, Don Capps, Alexander Carlton, Simon Eismann, Sorin Faibish, Johannes Grohmann, Karl Huppler, Nikolas R. Herbst, Rouven Krebs, Mary Marquez, Aleksandar Milenkoski, David Morse, Nick Principe, Meikel Poess, David Schmidt, Norbert Schmitt, Simon Spinner, and Sitsofe Wheeler

Review and Support:

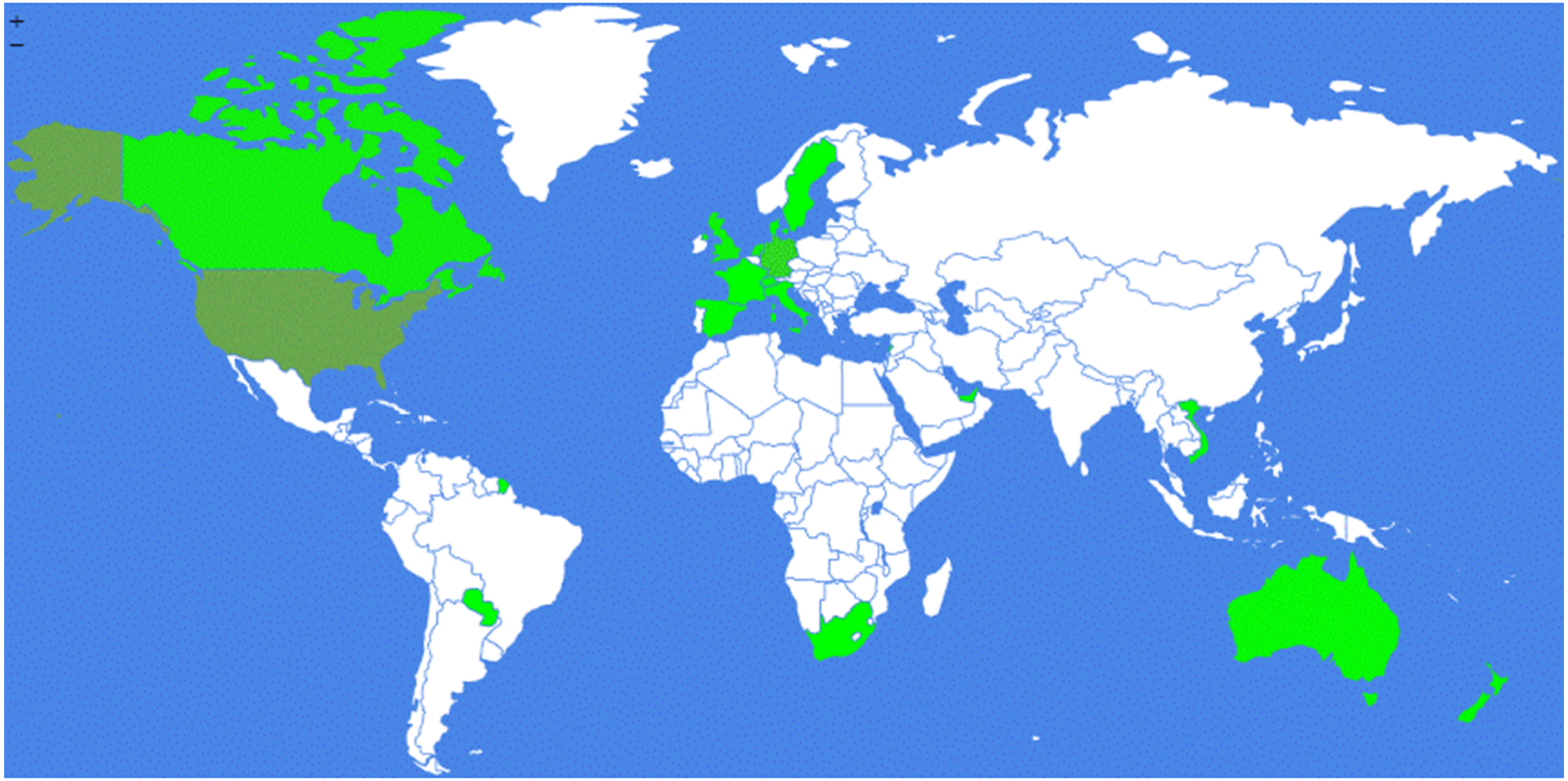
Walter Bays, Hansfried Block, Karla Orozco Bucek, John Henning, Scott Hinchley, Supriya Kamthania, Lorrie Crow Kimble, Mukund Kumar, Kris Langenfeld, Pranay Mahendra, John R. Mashey, Luis Mendoza, Sriranga Nadiger, Daniel Pol, Jesse Rangel, Nishant Rawtani, Jeff Reilly, David Reiner, Sanjay Sharma, and Rajesh Tadakamadla.

Numerous (under-)graduate students (2006-2020):

Julius-Maximilians-UNIVERSITÄT WÜRZBURG · KIT Karlsruhe Institute of Technology · UNIVERSITY OF CAMBRIDGE · UPC UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

University Library Adoption

World-wide: 147 libraries in 18 countries across 6 continents



5	Australia
12	Canada
2	Denmark
1	France
29	Germany
1	Italy
1	Lebanon
10	Netherlands
3	New Zealand
1	Paraguay
1	South Africa
3	Spain
1	Sweden
1	Switzerland
1	United Arab Emirates
9	United Kingdom
65	United States
1	Vietnam

Overview Part I: Foundations

1 Benchmarking Basics

2 Review of Basic Probability and Statistics

3 Metric

4 Statistical Measurements

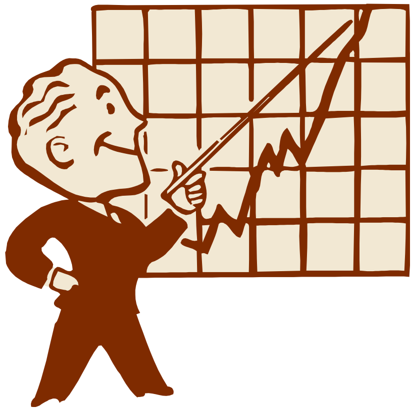
5 Experimental Design

6 Measurement Techniques

7 Operational Analysis and Basic Queueing Models

8 Workloads

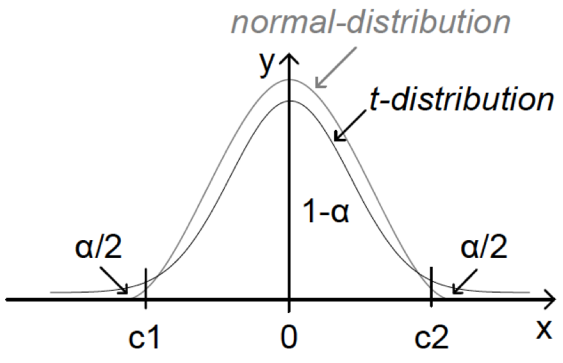
9 Standardization



1. Benchmarking Basics

- Definitions
- System quality attributes
- Types of benchmarks
- Performance benchmarking strategies (fixed-work, fixed-time,...)
- Benchmark quality criteria
  - Relevance
  - Reproducibility
  - Fairness
  - Verifiability
  - Usability
- Application scenarios for benchmarks

2. Review of Basic Probability and Statistics



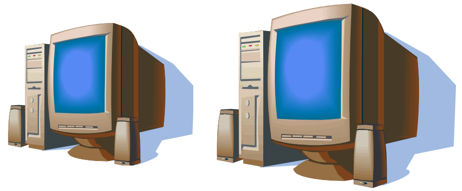
- Basic concepts
- Distributions of random variables
- Independent and dependent random variables
- Random samples and some important statistics
- Important continuous distributions and Central Limit Theorem
- The Bernoulli and Binomial bistributions
- Statistical techniques for parameter estimation
  - Regression analysis, Kalman filter, maximum likelihood estimation, Bayesian inference, mathematical optimization

3. Metrics

- Definitions: Measurement, Measure, Metric,...
- Scales of measurement
- Performance metrics
  - Speedup and relative change
  - Basic performance metrics
- Quality attributes of good metrics
- From measurements to metrics
  - Types of averages
  - Composite metrics
  - Aggregating results from multiple benchmarks

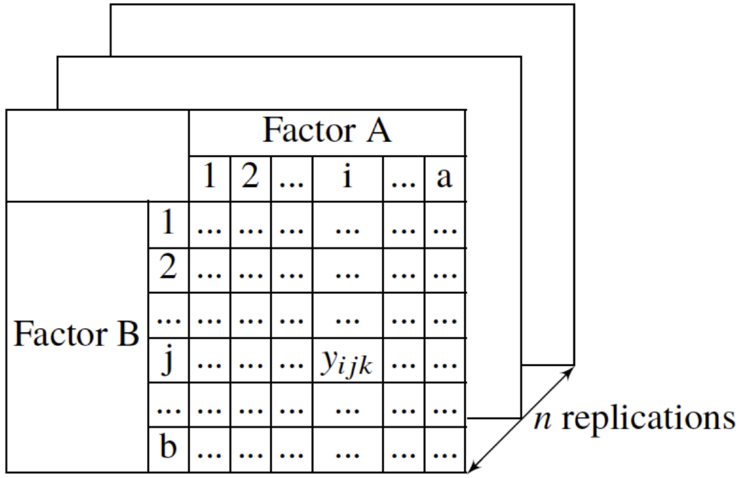
4. Statistical Measurements

- Measurement as a random experiment
- Quantifying precision of measurements
  - Experimental errors
  - A model of random errors
  - Estimating means
  - Estimating proportions
- Comparing alternatives
  - Non-corresponding measurements
  - Before-and-after comparisons
  - Comparing proportions



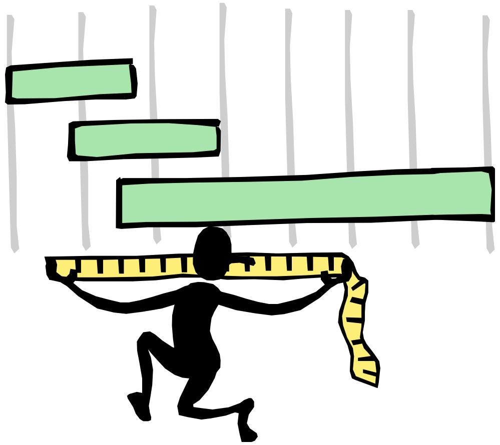
5. Experimental Design

- One-factor analysis of variance
- Method of contrasts
- Two-factor full factorial designs
- General m-factor full factorial designs
- Fractional factorial designs: Plackett–Burman
- Case studies



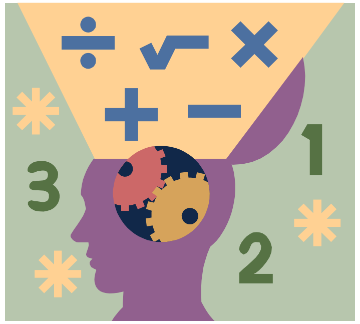
6. Measurement Techniques

- Basic measurement strategies
- Interval timers
  - Timer rollover
  - Timer accuracy
  - Measuring short intervals
- Performance profiling
- Event tracing
  - Call path tracing
  - Performance monitoring and tracing tools

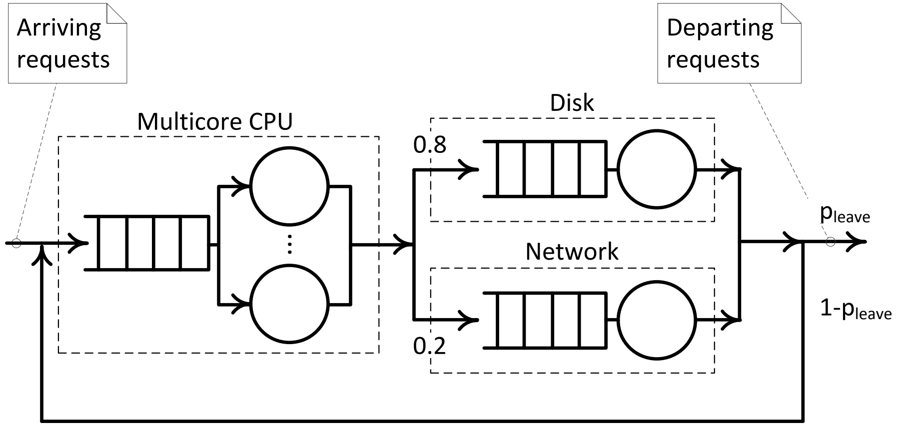


7. Operational Analysis and Basic Queueing Models

- Operational analysis
  - Utilization law, service demand law, forced flow law, Little's law, interactive response time law
  - Performance bounds

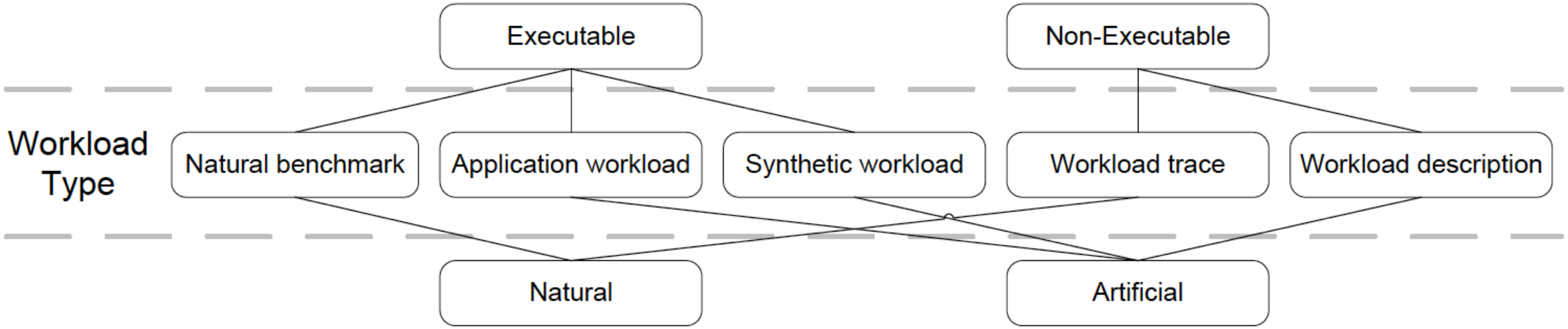


- Basic queueing theory
  - Single queues
  - Queueing networks
  - Operational laws
  - Response time equations
  - Solution techniques for queueing networks



8. Workloads

- Workload facets and artifacts
- Executable parts of a workload
- Non-executable parts of a workload
  - Workload traces
  - Workload descriptions
  - System-metric-based workload descriptions



9. Standardization

- Historical perspective on computer systems benchmarking

- Standard Performance Evaluation Corporation (SPEC)
  - SPEC’s origin, membership, structure and organization
  - Open Systems Group (OSG), Graphics and Workstation Performance Group (GWPG), High Performance Group (HPG), Research Group (RG)
  - Benchmark Development Cycle

- Transaction Processing Performance Council (TPC)
  - Beginning of TPC; From Engineering to Marketing, From Benchmarking to Benchmarketing; Progression of Benchmarks; Evolution of the TPC Model Over Time; Kit-Based Benchmarks; The Virtual World of Computing

Overview Part II: Applications

10 The SPEC CPU benchmark suite

11 Benchmarking the energy efficiency of servers

12 Virtualization benchmarks

13 Storage benchmarks

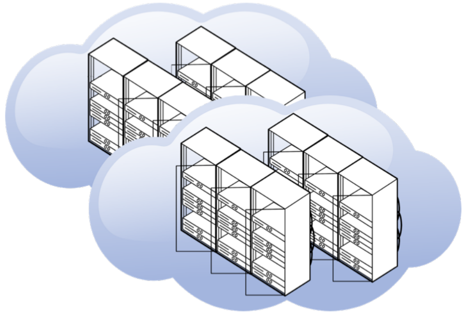
14 TeaStore: A microservice reference application

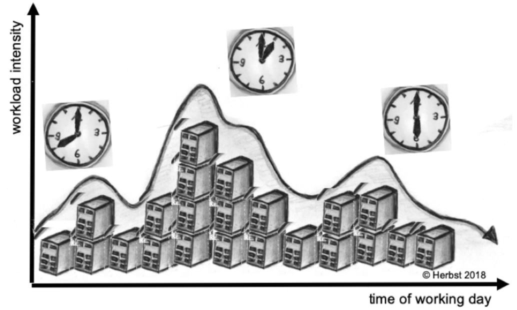
15 Elasticity of cloud platforms

16 Performance isolation

17 Resource demand estimation

18 Software and system security



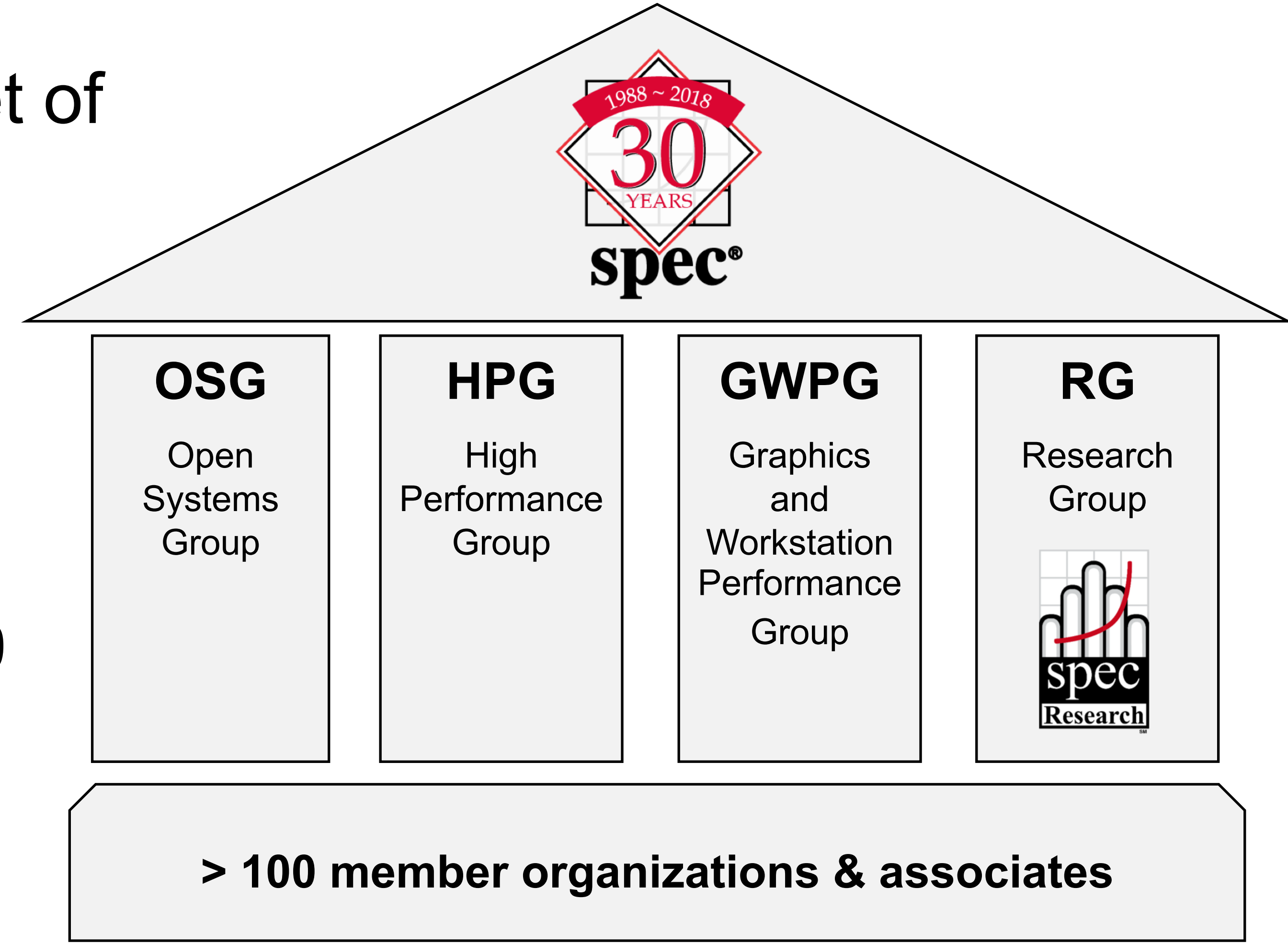


Background and Motivation

- Background and Motivation
- Benchmarking Education
- **Benchmark Standardization**
- Cloud Benchmarking
  - Measuring and quantifying elasticity
  - Reproducibility of experimental evaluation

Benchmark Standardization

- Standard Performance Evaluation Corporation (SPEC)
- Goal: provide standardized set of application benchmarks and standardized methodology for running them and reporting results
- First benchmark was SPEC89
  - 4 C programs
  - 6 Fortran programs



SPEC Research Group (RG)

http://research.spec.org

- Founded in March 2011
- Scope: Systems benchmarking, performance evaluation, and experimental system analysis
- Provide metrics, (research) benchmarks, methodologies and tools
- Foster transfer of knowledge and collaboration btw. industry and academia
- Beyond classical benchmarking
  - Dependability, elasticity, cost and energy efficiency
  - Evaluation methodologies and analysis tools
  - All stages of the system lifecycle
  - Both existing and newly emerging technologies

SPEC RG Members (April 2021) http://research.spec.org

VU Vrije Universiteit Amsterdam
University of Miami
Microsoft
Technische Universität München
Linköping University
Università di Pavia
Hewlett Packard Enterprise operated by Selectium
Universidade de Coimbra
Julius-Maximilians-Universität Würzburg
Concordia University
Charles University
The University of Texas at Austin
ESPOL
redhat
Chalmers University of Technology
Queen's University
Berkeley University of California
fortiss
York Université University
IBM
Stiftung Universität Hildesheim 2003
Imperial College London
FAU Friedrich-Alexander Universität Erlangen-Nürnberg
arm
intel
KIT Karlsruhe Institute of Technology
salesforce
Harokopio Univeristy
BEZNext
vmware
NovaTec
Cisco Systems
Mälardalen University Sweden
Oracle
CAU Christian-Albrechts-Universität zu Kiel
San Francisco State University 1899
Universität Ulm
TTA
RETIT Resource Efficient Technologies & IT Systems
UPT Universitatea Politehnica Timişoara
Technische Universität Darmstadt
University of Minnesota Driven to Discover
USI Università della Svizzera italiana
Uni Basel
University of Calgary
Universität Stuttgart
SINTEF
RWTH Aachen University
South China University of Technology 1934
mongoDB
MITRE
University of L'Aquila
Technische Universität Dresden
University of Alberta

Selected SPEC RG Activities

Working Groups

- RG Cloud (Chair: Alexandru Iosup, TU Delft)
- RG Security (Chair: Aleksandar Milenkoski , ERNW)
- RG DevOps (Chair: André van Hoorn, Uni-Stuttgart)
- RG Power (Chair: Norbert Schmitt, Uni-Würzburg)
- RG Quality of Experience (Chair: Florian Wamser, Uni-Würzburg)

Repository of peer-reviewed tools, experimental data & traces

Maintain a portal for all kinds of performance-related resources

Organization of **conferences and workshops**

https://icpe-conference.org

ICPE 2021
12th ACM/SPEC International Conference on Performance Engineering

SPEC Research Cloud WG

- Monthly group meetings plus very frequent activity-driven meetings and exchange via a Slack workspace
- Workshop series “**HotCloudPerf**”, again in conjunction with ICPE – good number of submissions, quality & attendance
- Organization of a **Dagstuhl Seminar** on “Serverless Computing” together with Ian Foster in May 2021

- Group officers:
  - Chair: Alexandru Iosup, VU Amsterdam
  - Co-Chair: Nikolas Herbst, University of Würzburg
  - Release Manger: André Bauer, University of Würzburg
  - Secretary: Sacheedra Talluri, VU Amsterdam

SPEC Research Cloud WG (cont.)

- **Group's Serverless activity - Tangible results in 2020:**
  - Technical Report: S. Eismann, J. Scheuner, E. van Eyk, M. Schwinger, J. Grohmann, N. Herbst, C. L. Abad, and A. Iosup. A Review of Serverless Use Cases and their Characteristics. May 2020.
  - IEEE Software article: S. Eismann *et al.*, **"Serverless Applications: Why, When, and How?,"** in *IEEE Software*, vol. 38, no. 1, pp. 32-39, Jan.-Feb. 2021.

- Further activities and plans
  - Serverless benchmark
  - Cloud experiment methodology and reproducibility
  - Edge computing workloads and benchmarks

RG DevOps Performance WG

- **Mission**
  - Consolidate concepts and tools to better integrate activities on the **interplay of DevOps and performance engineering**
- **Current Subgroups**
  - Performance testing of *next-generation cloud* applications
  - Model extraction and refinement in *continuous software engineering*
  - Performance of *continuous delivery* infrastructures
  - **new** *Resilience engineering* for cloud-native applications
- **Regular Contributors**

http://research.spec.org/devopswg/

RG DevOps Performance WG

2020 Highlights

- **11 monthly regular calls with 10 invited presentations**
- **3 Joint Publications in international conferences:**
  - *“Microservices: A Performance Tester's Dream or Nightmare?“* (ICPE 2020)
  - *“Incremental Calibration of Architectural Performance Models with Parametric Dependencies”*(ICSA 2020)
  - “*Optimizing Parametric Dependencies for Incremental Performance Model Extraction*“ (QUDOS 2020)
- **Jointly-organized community activities:**
  - Co-organized 6th Int. Workshop on Quality-Aware DevOps (QUDOS)
  - Co-initiated special issue on software performance in top-ranked EMSE journal (2021)

http://research.spec.org/devopswg/

- HySyringe: **Hypercall injection framework**
  - Test results based on the Microsoft Hyper-V hypervisor
    - Measured hypercall execution times when under stress
    - Discovered dependabiliy/security issues
      - MSRC Case 60869: *The Hyper-V hypervisor crashes with a critical error*
      - MSRC Case 60849: *The Hyper-V hypervisor eventually crashes with a critical error*
      - Acknowledged by Microsoft, to be fixed in a future release of Hyper-V

- Planned papers:
  - Full paper on HySyringe
  - Letter discussing challenges in benchmarking low-level system interfaces

RG Power

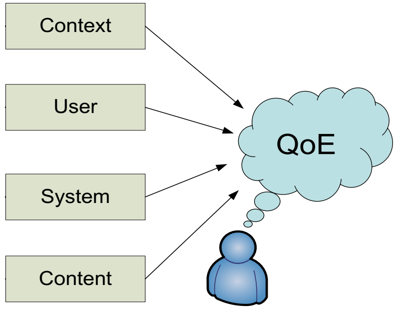
Chair: Norbert Schmitt, Uni-Würzburg

- 2020 in Retrospect
  - **Work-In-Progress and full paper** at ICPE 2020 / UCC 2020 about the influence of compiler settings on SPEC CPU 2017
  - **Industry Track paper** at ICPE 2021 about the current development of the upcoming SPECpowerNext benchmark
  - **Vision paper** at ICPE 2021 for a software resource efficiency benchmark (under review)
  - **Third party funding** for a software resource / energy efficiency benchmark demonstrator (under review)

- New Activities in 2021
  - Paper on the energy efficiency and performance of group encryption algorithms
  - More research towards software energy efficiency

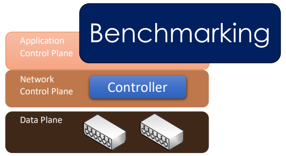
RG QoE WG: Quality of Experience

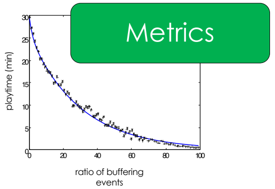
- Promote QoE as end-user evaluation metric
  - assesses how end-users ultimately perceive a service or system



Objectives

1. **Standardization of benchmarking *wrt.* QoE**
2. **Performance evaluation *wrt.* end-user**
3. **Discussions on approaches and best practices**





RG QoE WG: 2020 Activities

- QoE **modeling and metrics,** crowdsourcing for QoE evaluation, **benchmarking** scenarios for different metrics
- QoE **fairness**
  … definition of a universal **comparison metric** to weigh different user ratings and experiences
- **Crowdsourcing-based benchmarking**
  … massive amount of user ratings, validity & analysis methods
- Benchmarking of **Music Streaming** towards Quality of Experience

Quality of Experience Working Group

- **Florian Wamser**, University of Würzburg, Germany
- eMail: florian.wamser(at)informatik.uni-wuerzburg(dot)de

- Website https://research.spec.org/working-groups/rg-quality-of-experience.html
- Mail: rgqoe@spec.org

Agenda

- Introduction
- Benchmarking Education
- Benchmark Standardization
- **Cloud Benchmarking**
  - Measuring and quantifying elasticity
  - Reproducibility of experimental evaluation

Case Study: Cloud Benchmarking

- Based on:

  Quantifying Cloud Performance and Dependability: Taxonomy, Metric Design, and Emerging Challenges

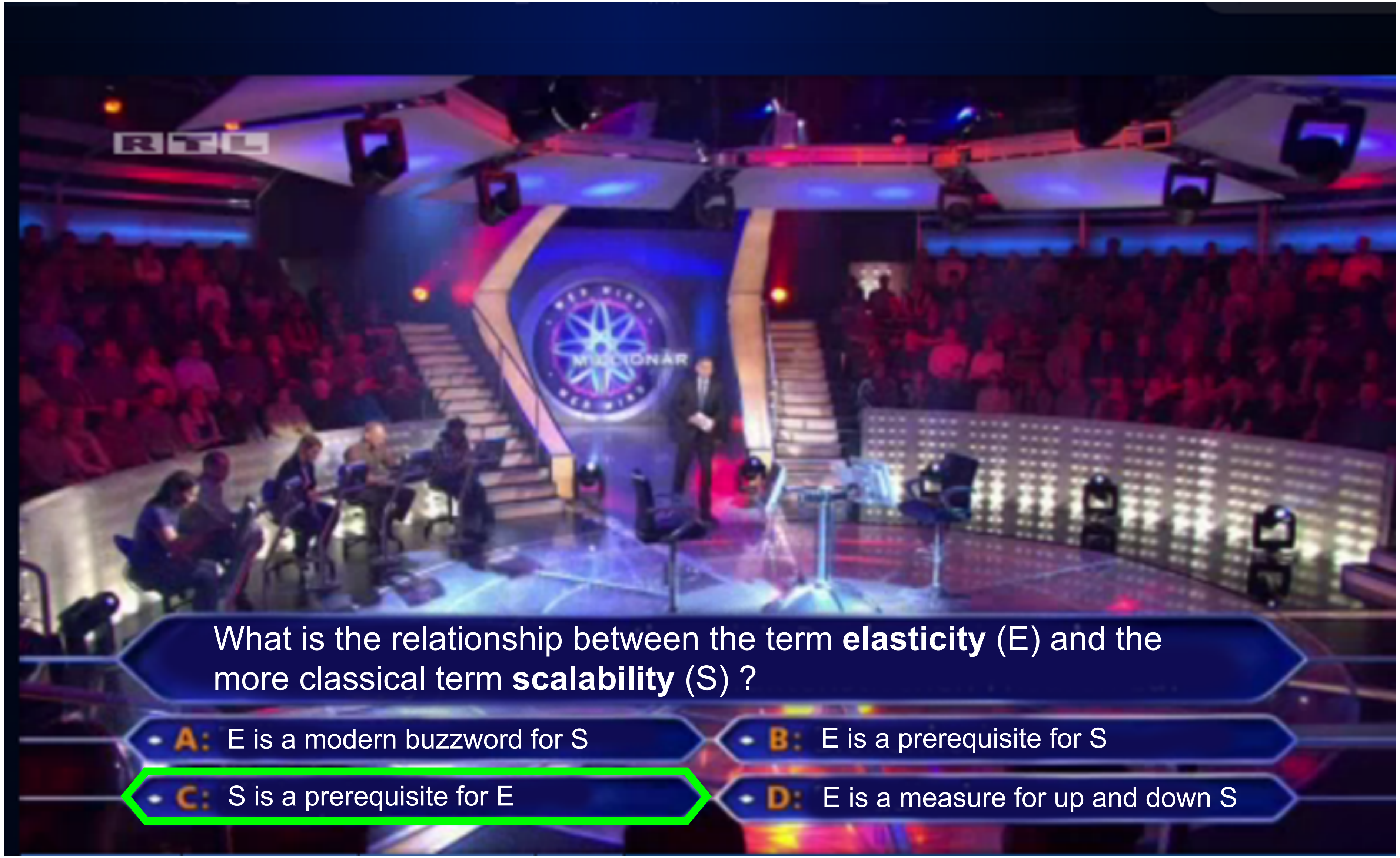
  N. Herbst, A. Bauer, S. Kounev, G. Oikonomou, E. van Eyk, G. Kousiouris, A. Evangelinou, R. Krebs, T. Brecht, C. L. Abad, A. Iosup.

  ACM Transactions on Modeling and Performance Evaluation of Computing Systems (ToMPECS) (2018). **3**(4) 19:1–19:36. ACM, New York, NY, USA.

- Covered Aspects:
  - **Elasticity** of the cloud service to accommodate large variations in the amount of service requested
  - **Performance isolation** between users of shared cloud systems and resulting performance variability
  - **Availability** of cloud services and systems
  - **Operational risk** of running a production system in a cloud environment

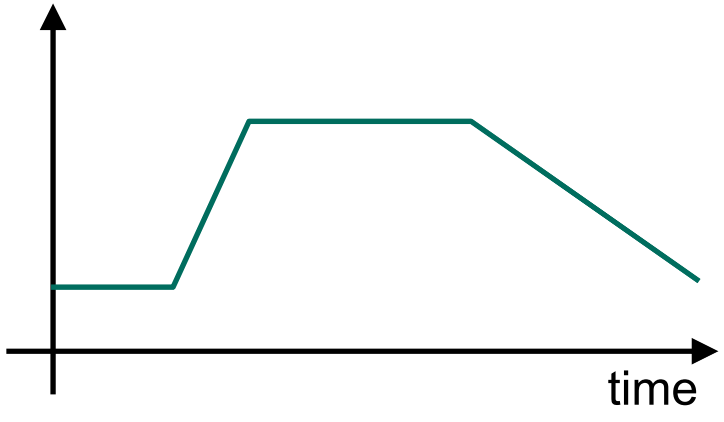
- Focus here: elasticity

Elasticity vs. Scalability



Elasticity (in Cloud Computing)

Workload intensity (e.g., # requests / sec)



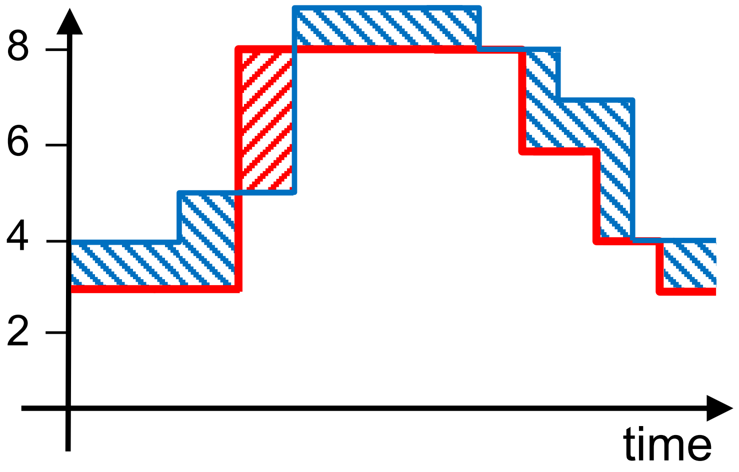
Service Level Objective (SLO)

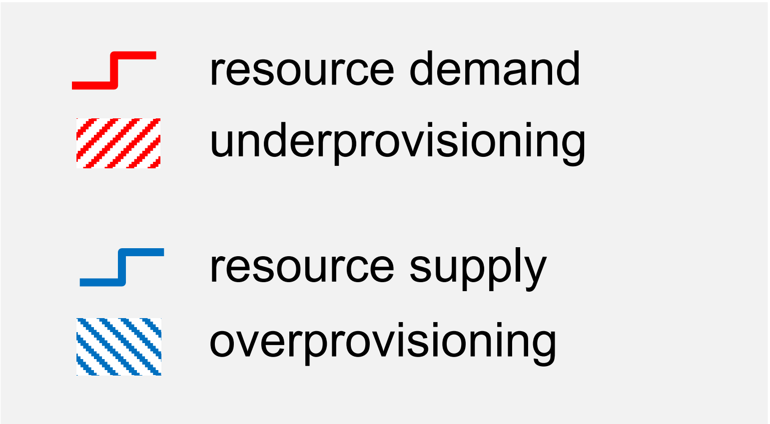
(e.g., resp. time ≤ 2 sec, 95%)

Resource Demand

Minimal amount of resources required to ensure SLOs

Amount of resources (e.g., # VMs)





Elasticity (in Cloud Computing)

Def: The degree to which a system is able to **adapt** to **workload changes** by **provisioning and deprovisioning** resources in an **autonomic manner**, such that at each point in time the **available resources match** the **current demand** as closely as possible.

N. Herbst, S. Kounev and R. Reussner
Elasticity in Cloud Computing: What it is, and What it is Not.
in Proceedings of the 10th International Conference on Autonomic Computing (ICAC 2013), San Jose, CA, June 24-28, 2013.

Key Components of Each Benchmark

1. Reliable Metrics

What exactly should be measured and computed?

2. Representative Workloads

For which usage scenarios and under what conditions?

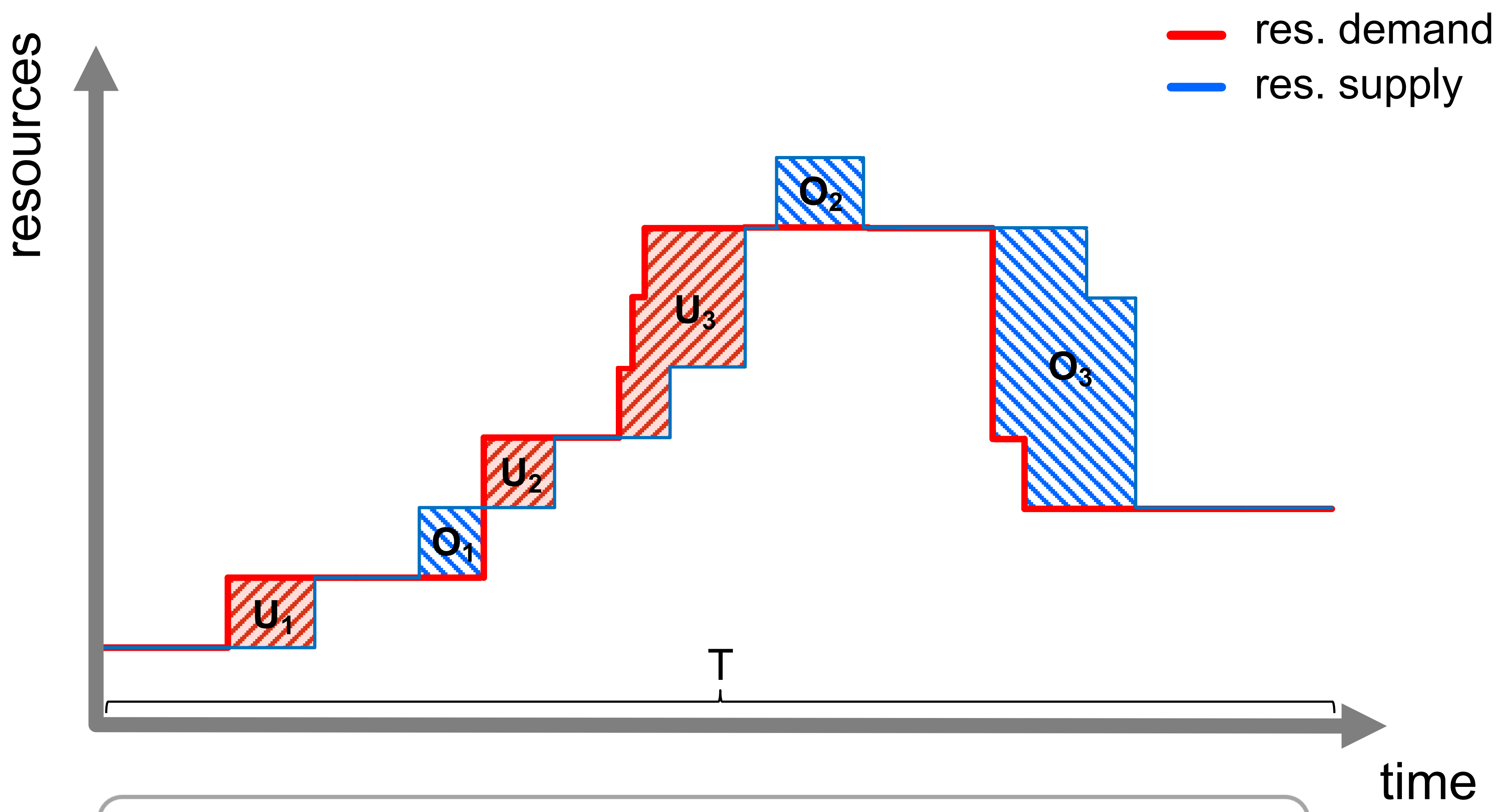
3. Sound Measurement Methodology

How should measurements be conducted?

"*To **measure** is to **know**.*" -- Clerk Maxwell, 1831-1879

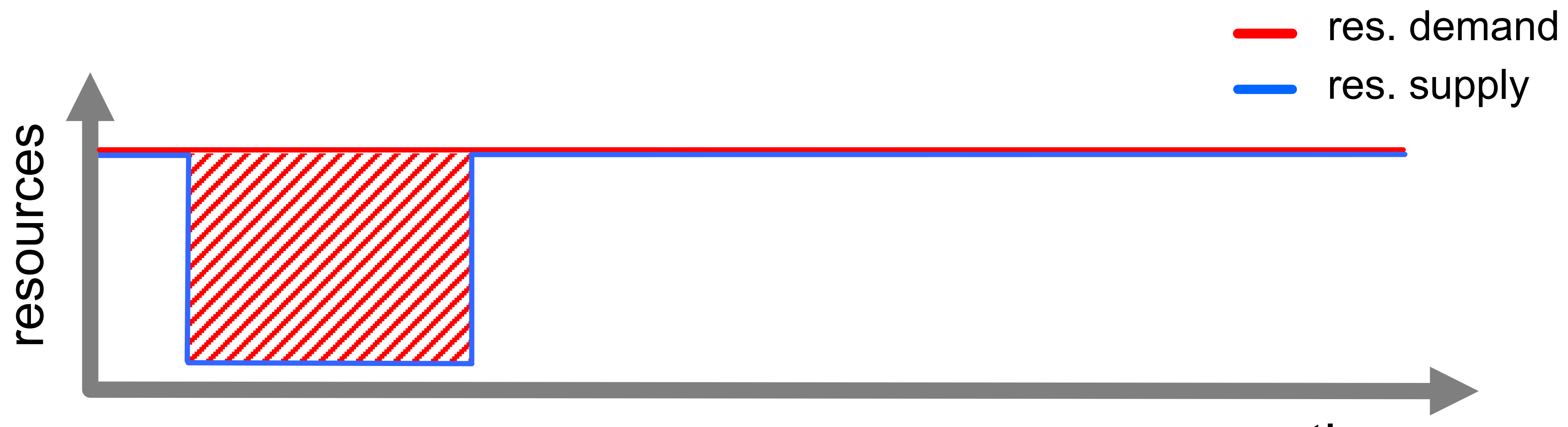
"*It is much easier to make **measurements** than to **know** exactly what you are measuring.*"
-- J.W.N.Sullivan (1928)

Metric 1: Accuracy

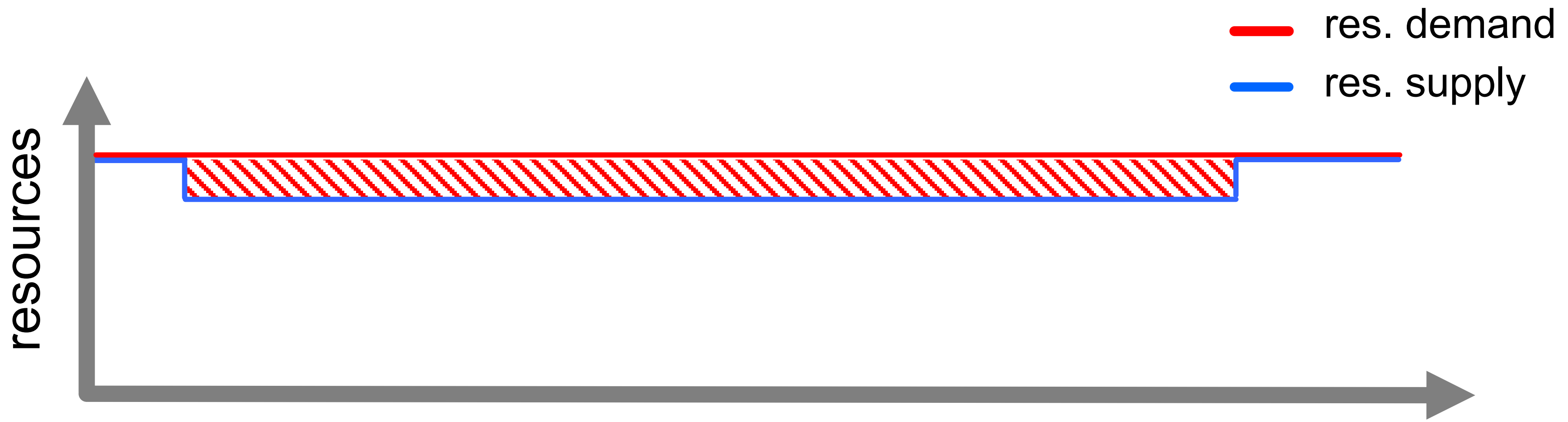


(1) $accuracy_U$: $\frac{\sum U}{T}$ (2) $accuracy_O$: $\frac{\sum O}{T}$

Same Metric Values - Different Behavior!

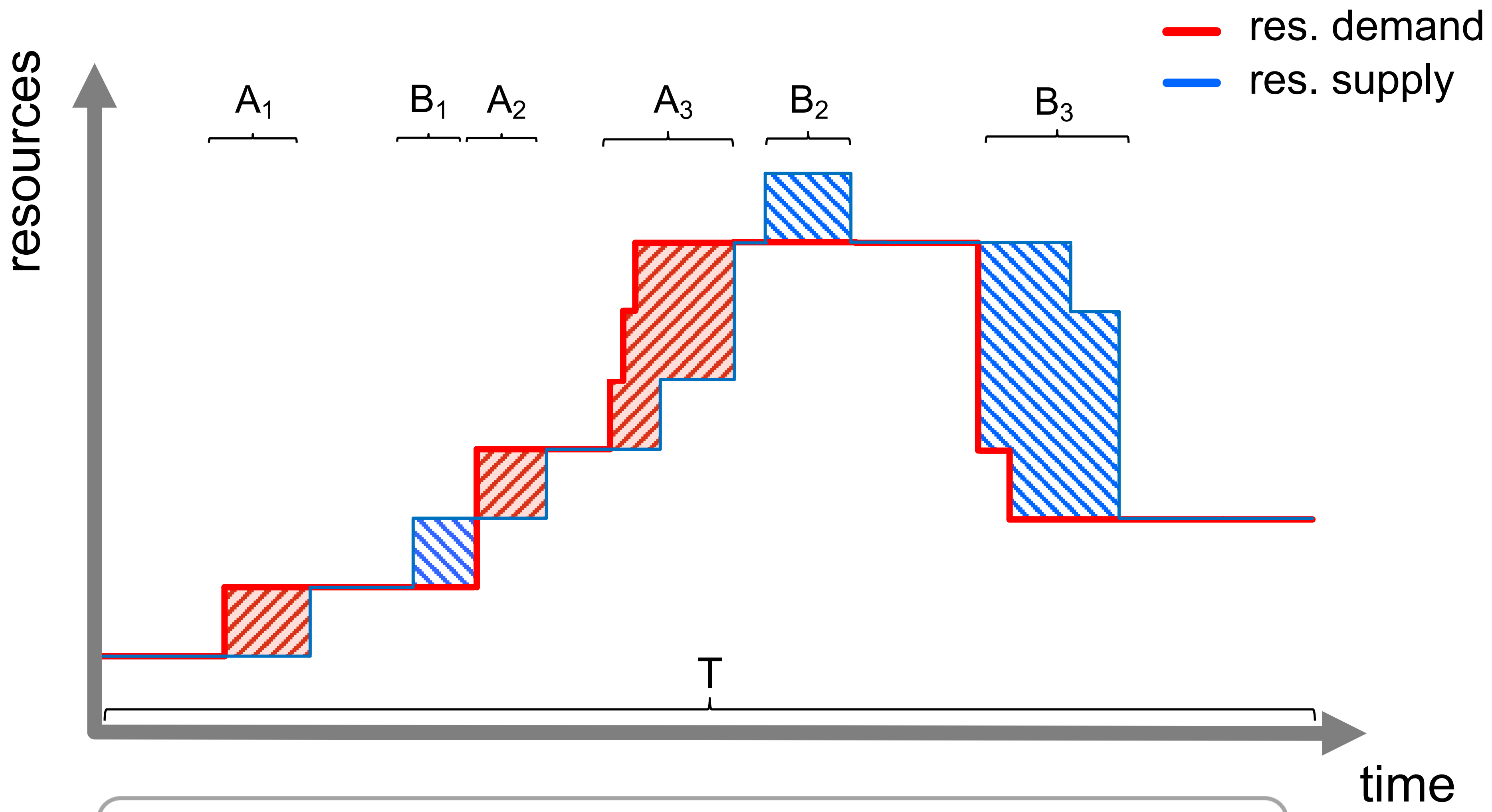


System A



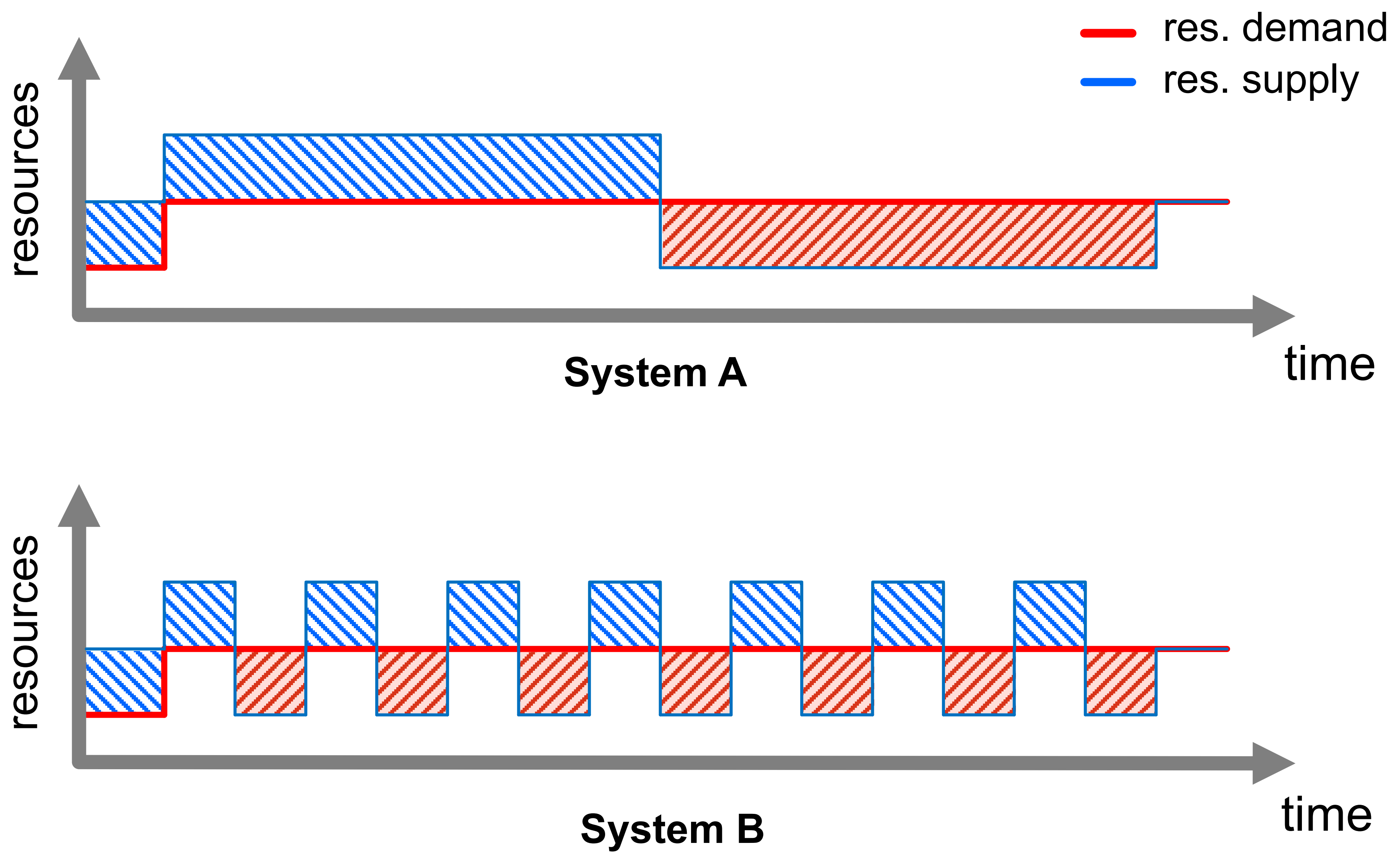
System B

Metric 2: Timeshare

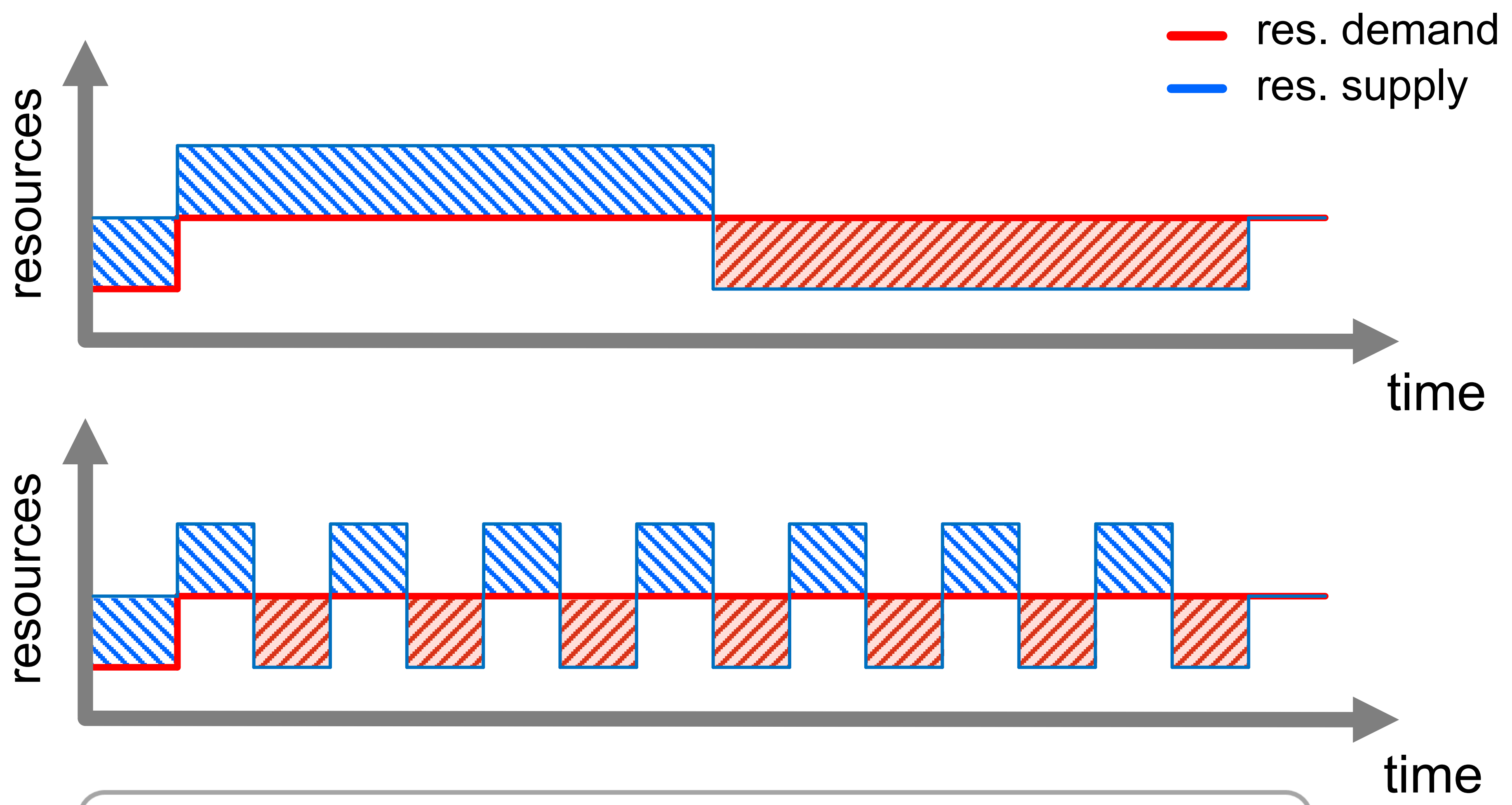


(3) timeshare$_U$: $\frac{\sum A}{T}$ (4) timeshare$_O$: $\frac{\sum B}{T}$

Same Metric Values - Different Behavior!



Metric 3: Jitter



(5) jitter: $\frac{E_S - E_D}{T}$

E_D: # demand changes

E_S: # supply changes

Key Components of Each Benchmark

1. Reliable Metrics

What exactly should be measured and computed?

2. Representative Workloads

For which usage scenarios and under what conditions?

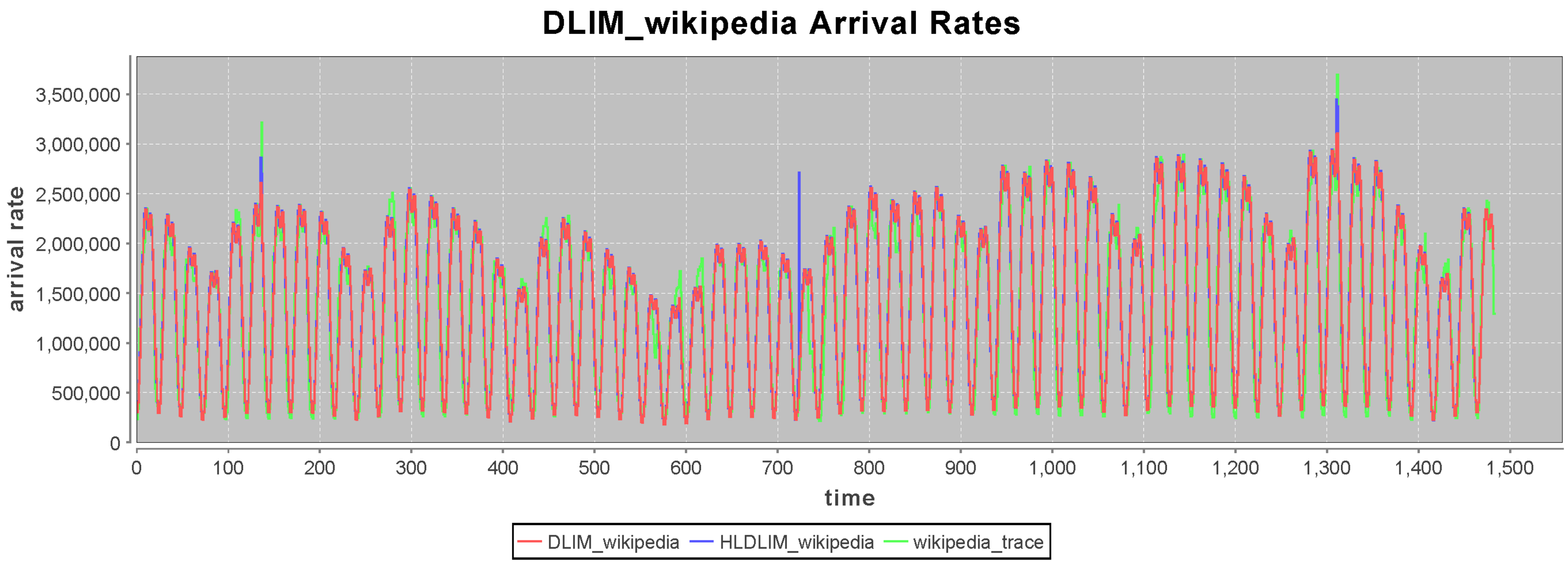
3. Sound Measurement Methodology

How should measurements be conducted?

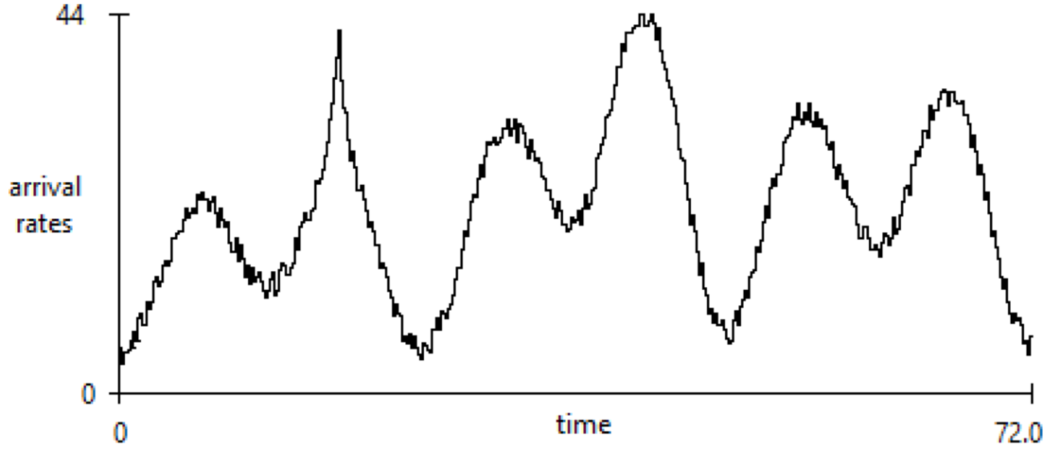
"*To **measure** is to **know**.*" -- Clerk Maxwell, 1831-1879

"*It is much easier to make **measurements** than to **know** exactly what you are measuring.*"
-- J.W.N.Sullivan (1928)

Example: Wikipedia Workload Trace



Extracting Models of Real-Life Traces

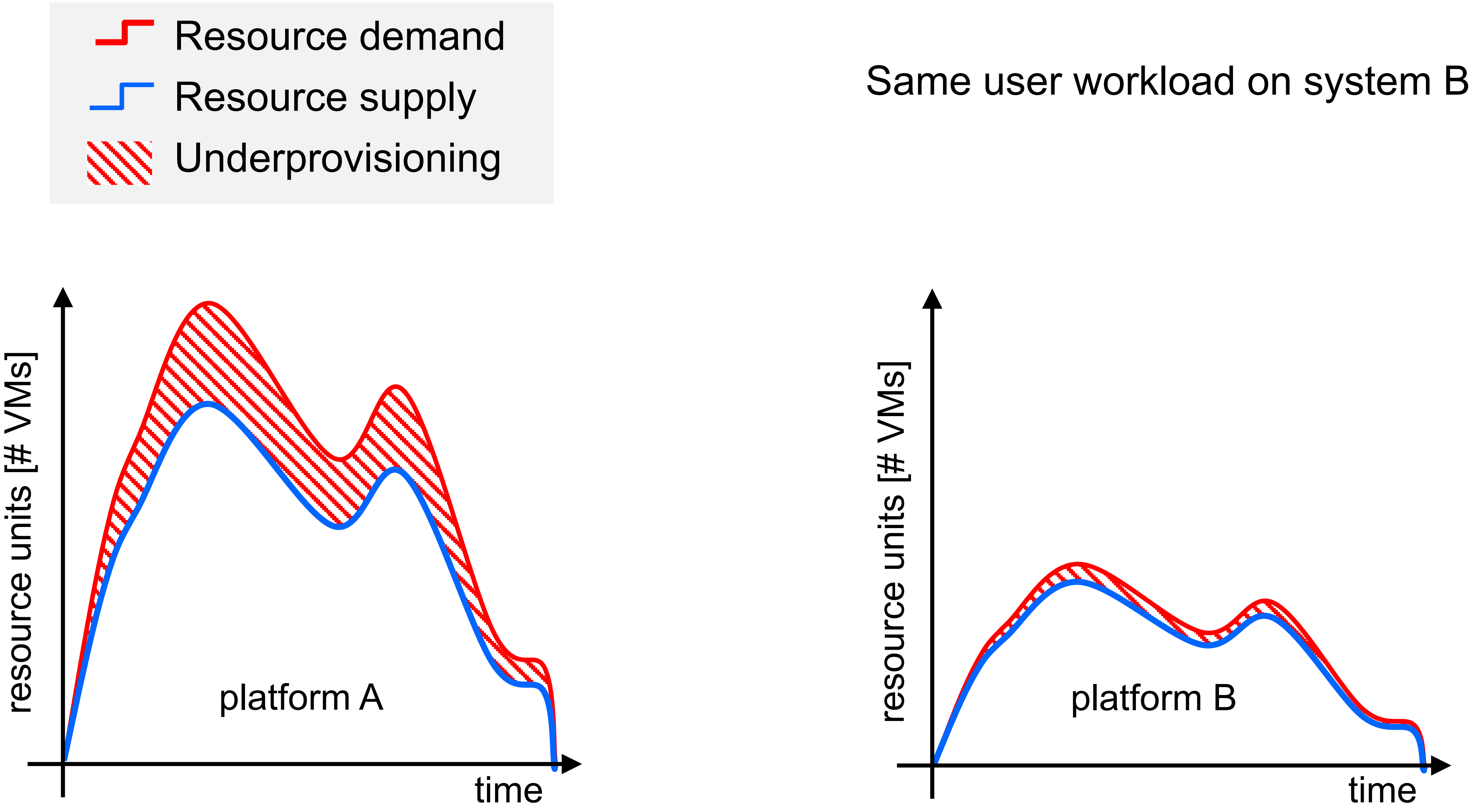


http://descartes.tools/limbo

Modeling and Extracting Load Intensity Profiles. J. von Kistowski; N. Herbst; S. Kounev; H. Groenda; C. Stier; S. Lehrig; in *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* (2017). **11**(4) 23:1–23:28.

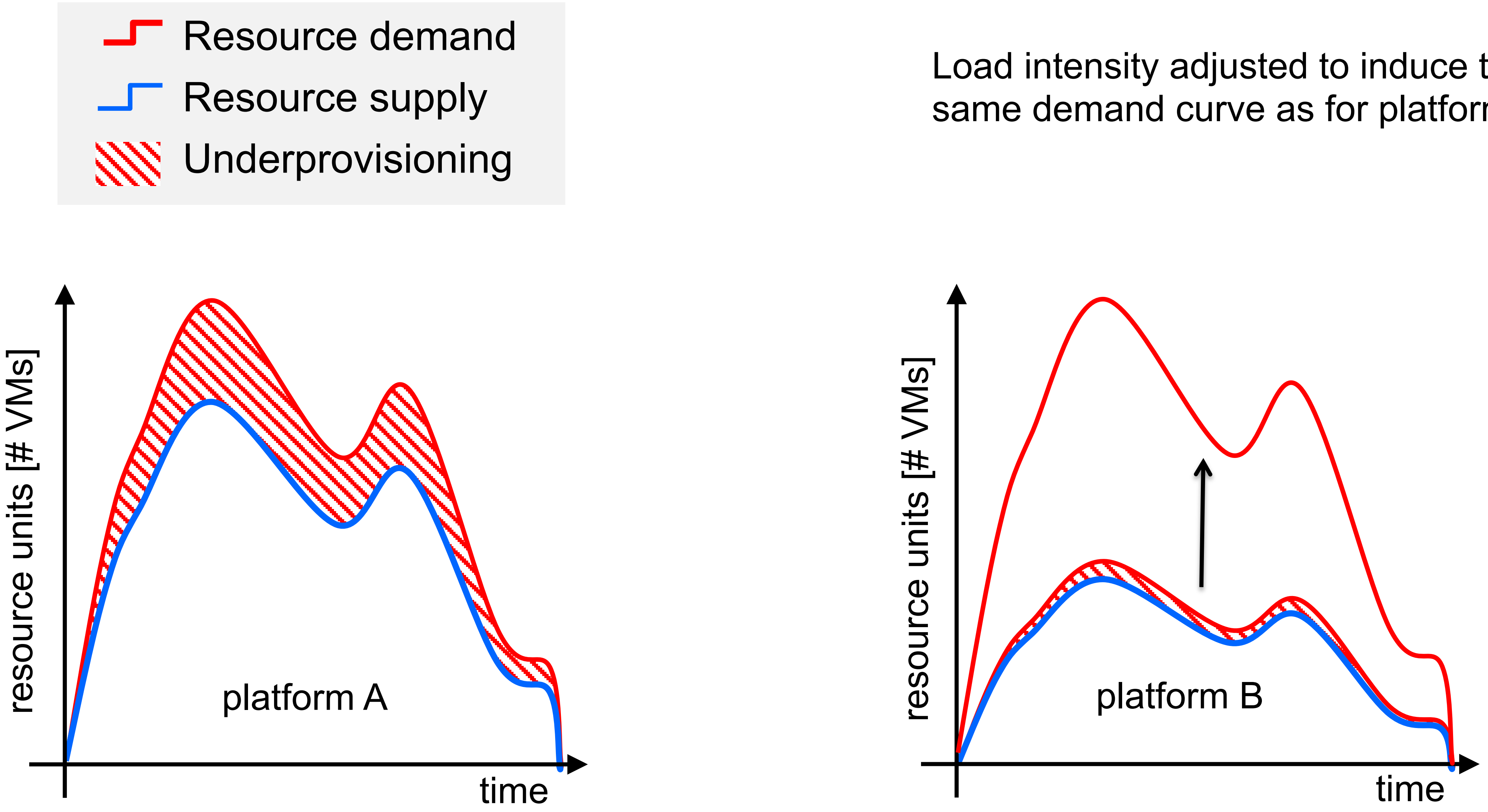
Same Workload on Two Platforms

Same user workload on system B

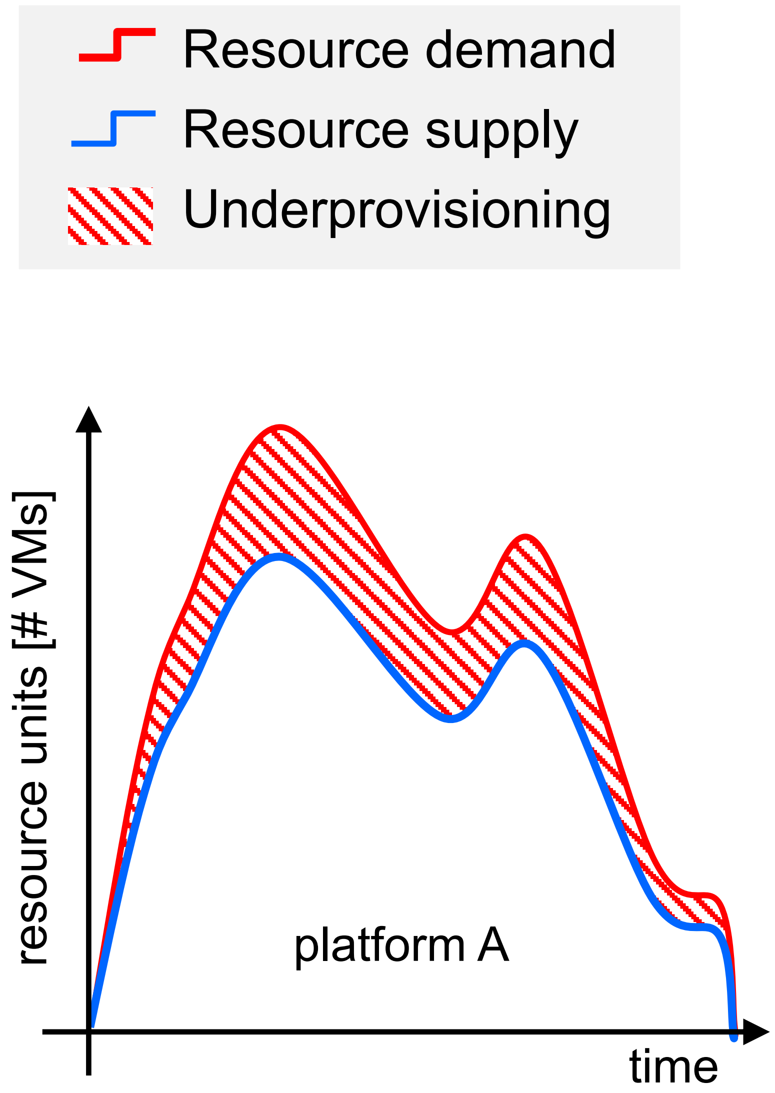


Same Workload on Two Platforms

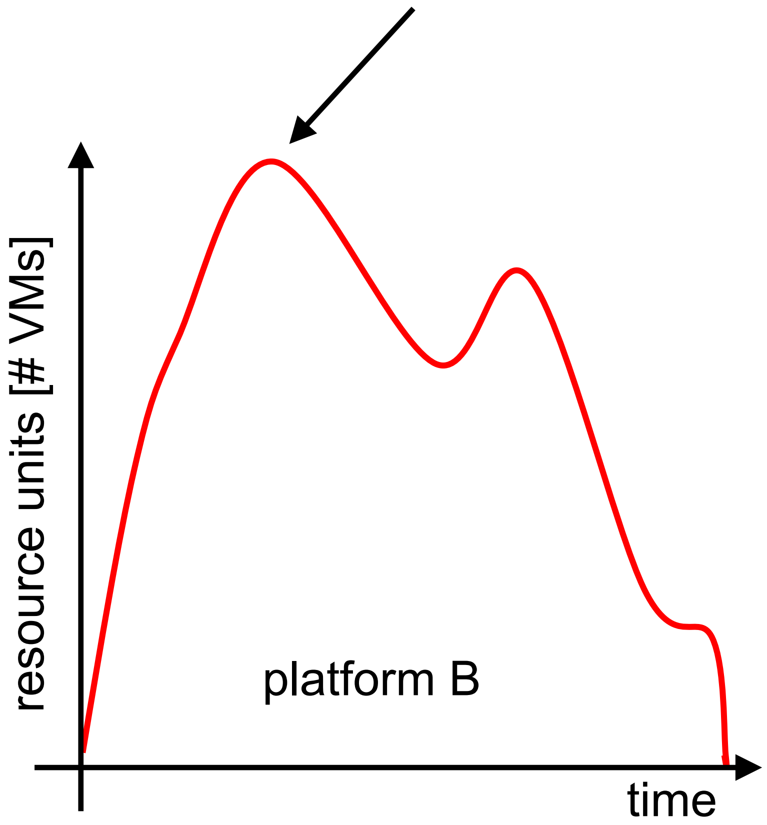
Load intensity adjusted to induce the same demand curve as for platform A



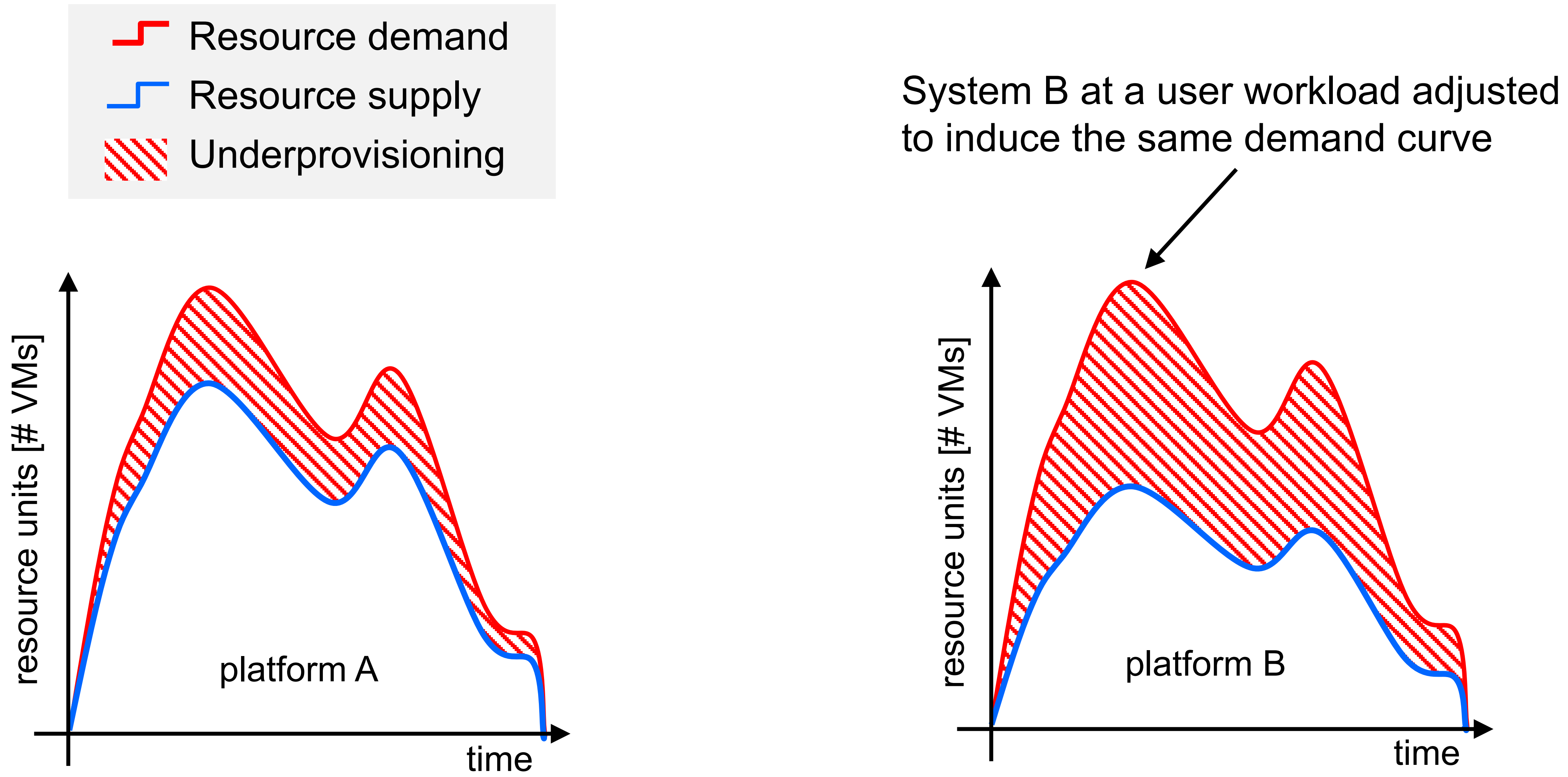
Same Demand Variations on Two Platforms



System B at a user workload adjusted to induce the same demand curve



Same Demand Variations on Two Platforms



Key Components of Each Benchmark

1. Reliable Metrics

What exactly should be measured and computed?

2. Representative Workloads

For which usage scenarios and under what conditions?

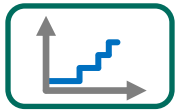
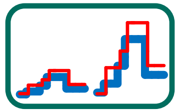
3. Sound Measurement Methodology

How should measurements be conducted?

"*To **measure** is to **know**.*" -- Clerk Maxwell, 1831-1879

"*It is much easier to make **measurements** than to **know** exactly what you are measuring.*"
-- J.W.N.Sullivan (1928)

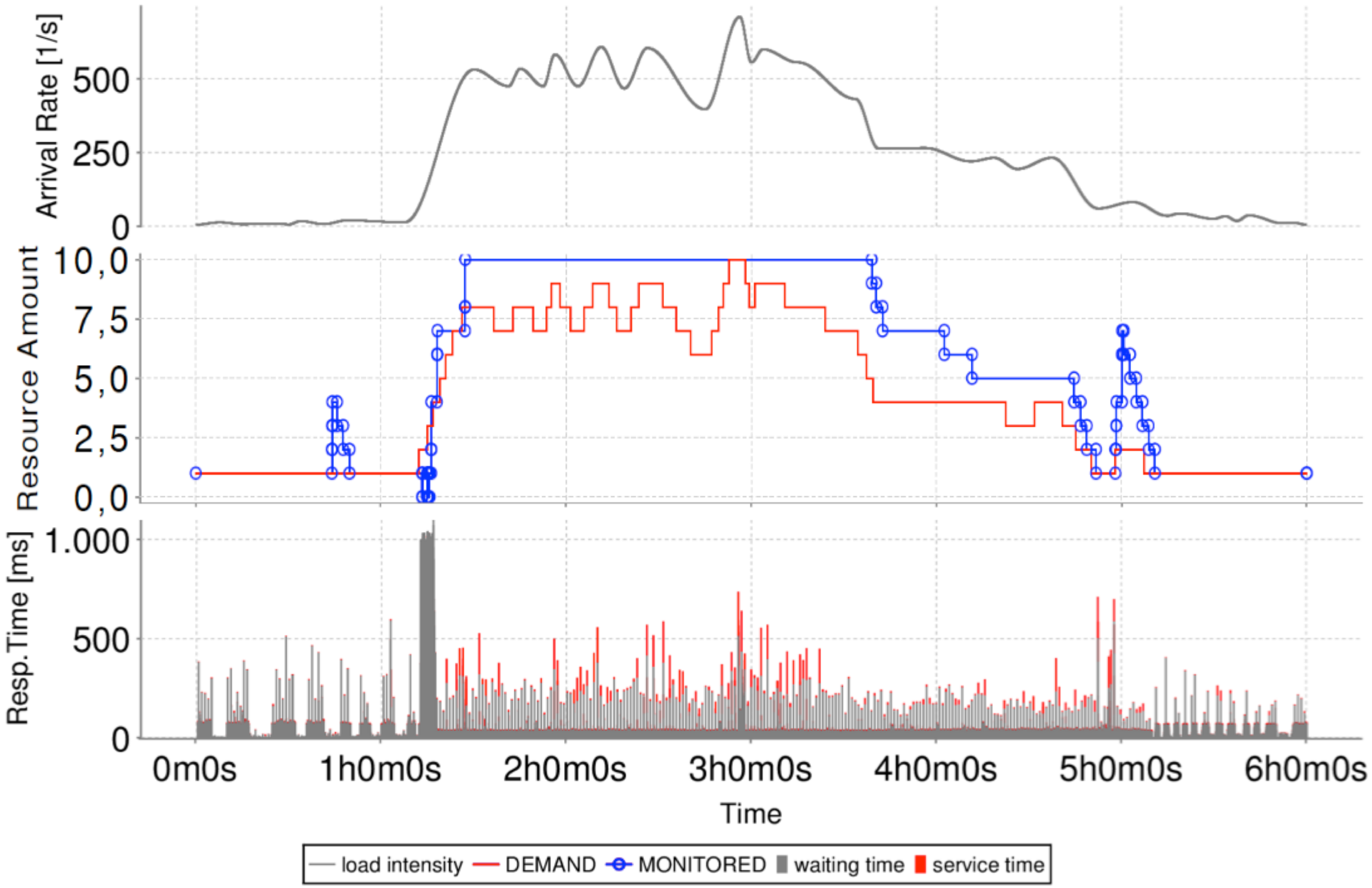
Elasticity Benchmarking Approach

System Analysis	Analyze efficiency & scaling behavior of underlying resources	
Benchmark Calibration	Adjust load profile	
Measurement	Expose SUT to varying load & monitor resource supply & demand	
Metric Evaluation	Compute elasticity metrics (accuracy & timing)	

N. Herbst, S. Kounev, A. Weber and H. Groenda. **BUNGEE: An Elasticity Benchmark for Self-Adaptive IaaS Cloud Environments**. In *10th Intl. Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2015)*, Firenze, Italy, May 18-19, 2015.

Chameleon: A Hybrid, Proactive Auto-Scaling Mechanism on a Level-Playing Field. Bauer, André; Herbst, Nikolas; Spinner, Simon; Ali-Eldin, Ahmed; Kounev, Samuel; in *IEEE Transactions on Parallel and Distributed Systems* (2019). **30**(4) 800–813. IEEE.

Case Study: Amazon Web Services vs. CloudStack



Configuration	$accuracy_O$ [res. units]	$accuracy_U$ [res. units]	$timeshare_O$ [%]	$timeshare_U$ [%]	jitter [adap/min.]	elastic speedup	violations [%]
CS – 1Core	2.423	0.067	66.1	4.8	-0.067	1.046	7.6
CS – 2Core adjusted	2.508	0.061	67.1	4.5	-0.044	1.025	8.2
AWS - m1.small	1.340	0.019	61.6	1.4	0.000	1.502	2.5

Agenda

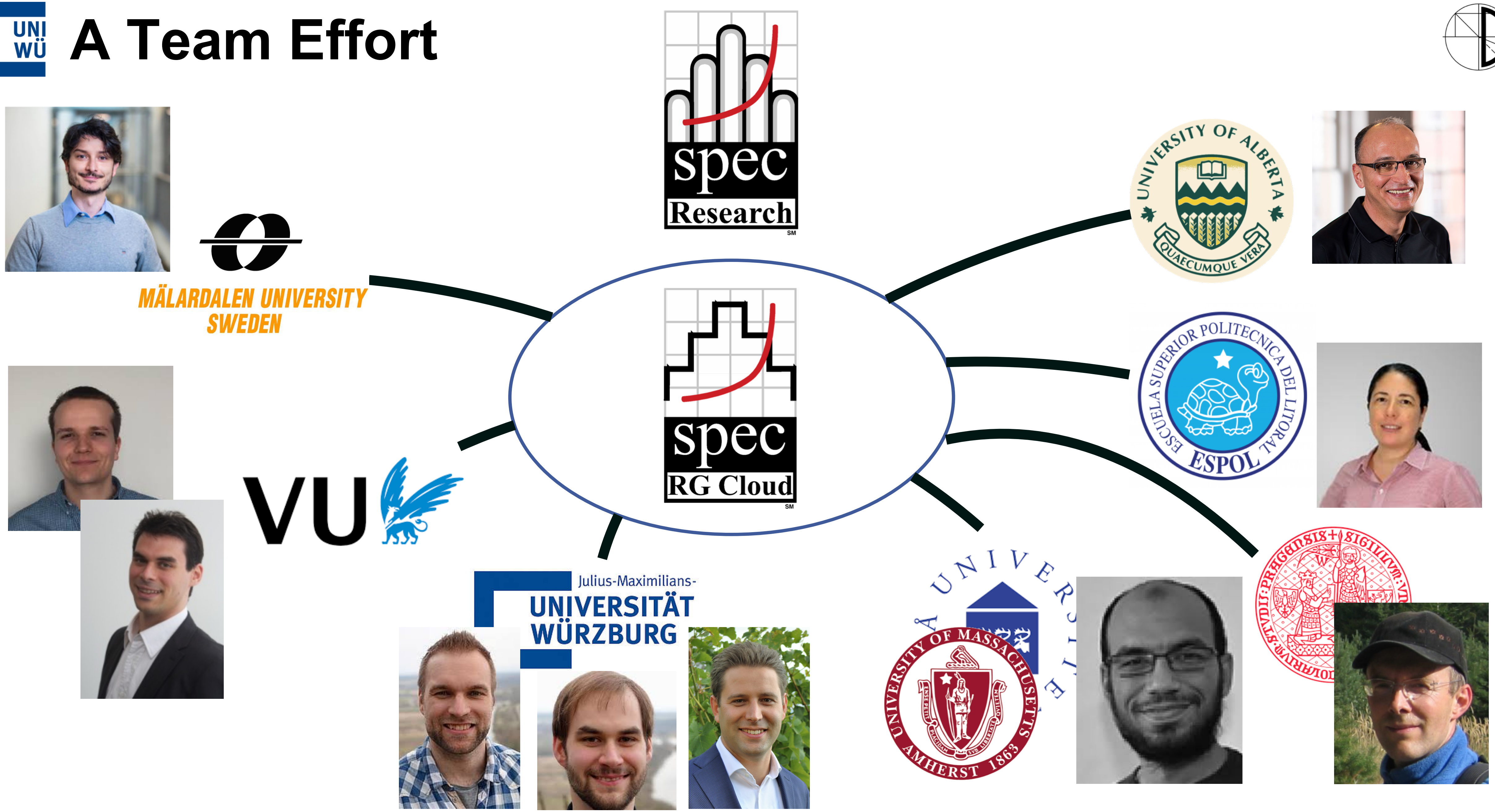
- Introduction
- Benchmarking Education
- Benchmark Standardization
- **Case Study on Cloud Benchmarking**
  - Measuring and quantifying elasticity
  - Reproducibility of experimental evaluation

Case Study: Reproducibility

Methodological Principles for Reproducible Performance Evaluation in Cloud Computing
A. V. Papadopoulos; L. Versluis; A. Bauer; N. Herbst; J. von Kistowski; A. Ali-Eldin; C. Abad; J. N. Amaral; P. Tuma; A. Iosup.
IEEE Transactions on Software Engineering (2019).

spec RG Cloud SM

A Team Effort



We Know Him All: the Ghost of Paper Past

From a Nature-published survey by Baker from 2016:

- 70% of the 1,500 researchers surveyed have tried and failed to reproduce prior work done by others, and
- over 50% failed to reproduce their own experimental results

M. Baker, “Is there a reproducibility crisis?” Nature, vol. 533, pp.452–454, 2016.

Reproducibility in Computer Science

Open-source code, versioning, virtualization, …

→ obvious techniques to support reproducibility

Some technical solutions:

▪ PlanetLab	functional reproducibility only
▪ Containers	e.g. data center networking experiments
▪ APT, EmuLab, FlexLab	towards distributed system experiment repro.
▪ DataMill	control experiment variability
▪ Jupyter Notebooks, IEEE CodeOcean	data processing algorithms
▪ Zenodoo	artifact archiving with DOIs

Reproducibility in Computer Science (2)

Collberg and Proebsting (2016, CACM) showed:

- 50% of works published in top CS venues
  (incl. ASPLOS, VLDB, SOSP)
  not reproduceable due to missing or uncompilable code
- Authors fail sometimes to reproduce their own results

Let us distinguish concepts of

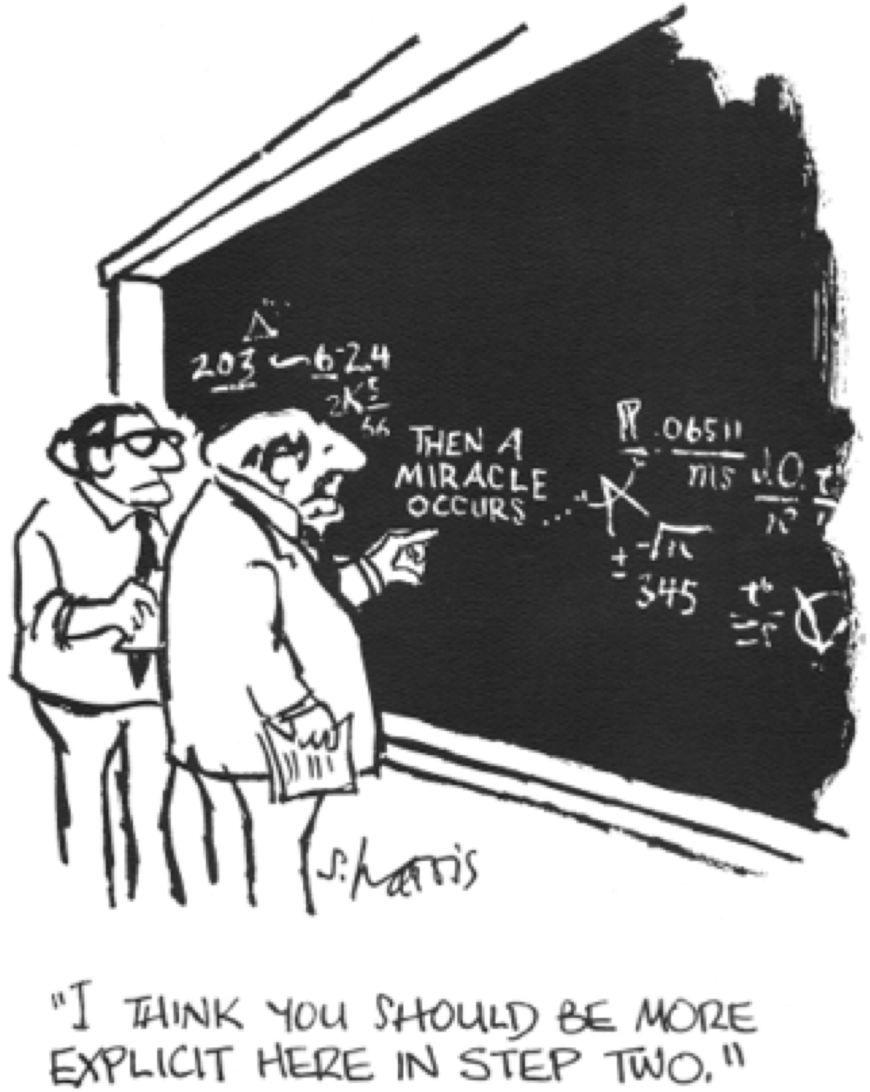
technical reproducibility ←→ reproducibility of claims

C. Collberg and T. A. Proebsting, “Repeatability in computer systems research,” Commun. ACM, vol. 59, pp. 62–69, 2016.

A Quest for Reproducibility in Computer Science

Several conferences are introducing Artifact Evaluation

- ACM SIGCOMM
- ACM SIGMOD
- ACM SIGPLAN
- ACM/SPEC ICPE
- SuperComputing
- ECRTS
- RTSS
- RTAS
- …



Study on Cloud Computing Benchmarking

RQ1 What methodological principles are needed for sound experimental evaluation of cloud performance?

RQ2 Can the methodological principles be applied in common practice?

RQ3 How are cloud performance results currently obtained and reported?

Cloud Experiment Methodology RQ 1: Basic Principles 1 - 4

1 – Repeated experiments (statistical)

After identifying the sources of variability, decide **how many repetitions with the same configuration** of the experiment should be run, and then quantify the confidence in the final result.

2 – Independent experiments

Experiments should be conducted in **different (possibly randomized) configurations** of relevant parameters, especially parameters that are not completely under control or those that may interact with the platform in unexpected ways, e.g., the workload…

3 – Experimental setup description

Description of the **hardware and software setup** used to carry out the experiments, and of other **relevant environmental parameters**, must be provided…

4 – Open access artifact

At least a representative subset of the developed software and data (e.g., workload traces, configuration files, experimental protocol, evaluation scripts) used for the experiment should be made **available to the scientific community**…

Cloud Experiment Methodology RQ 1: Basic Principles 5 - 8

5 – Probabilistic result description of measured performance

Report a **characterization of the empirical distribution** of the measured performance, including **aggregated values and variations** around the aggregation, with the **confidence** that the results tend to these values.

6 – Statistical evaluation

When comparing different approaches, provide a **statistical evaluation of the significance** of the obtained results.

7 – Measurement units

For all the reported quantities, report the corresponding **unit of measurement**.

8 – Cost

Every cloud experiment should include
(i) cost model used or assumed for exp.;
(ii) **accounted resource usage** (per second), independently of the model; and
(iii) **charged cost** according to the model

Principle Use in Industry Standard Benchmarks

P1 Most benchmarks define minimum number of runs,

e.g. SPEC IaaS Cloud: 5 runs.

P2 Benchmarks run a set of workloads/worklets,

e.g., SPEC IaaS Cloud: Cassandra & k-means, SPEC CPU > 20 worklets

P3 Run & reporting rules are strictly defined and reviewed

P4 Measurement methodology and execution is documented in high-detail, benchmark harnesses often open-source

P5 Average or median values reported,

SPEC Sert 2 suite: coef. var., SPEC IaaS Cloud: 99 percentile

P6 Statistical testing mostly out of scope

P7 Reported units are well defined

P8 Costs are partially reported for cloud-focused benchmarks

RQ 2 - Cloud Experiment: Example

Hypothesis: *"The scaling behavior of a standard, reactive, CPU utilization-rule-based auto-scaler depends on its environment."*

Environments:

- Cloudstack-based private cloud (CSPC)
- AWS EC2
- DAS-4 IaaS cloud of a medium-scale multi-cluster experimental environment (MMEE)

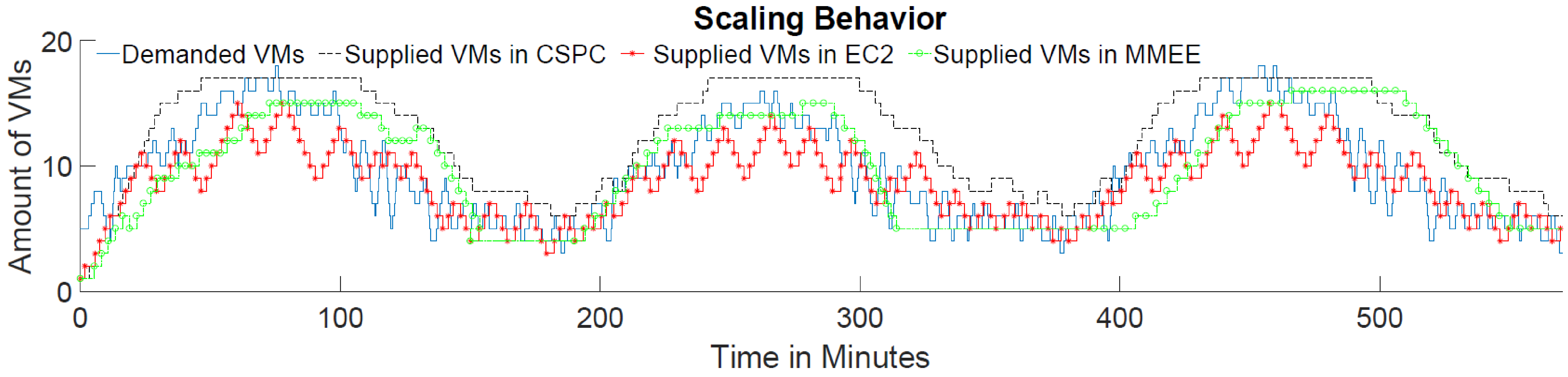
Workload:

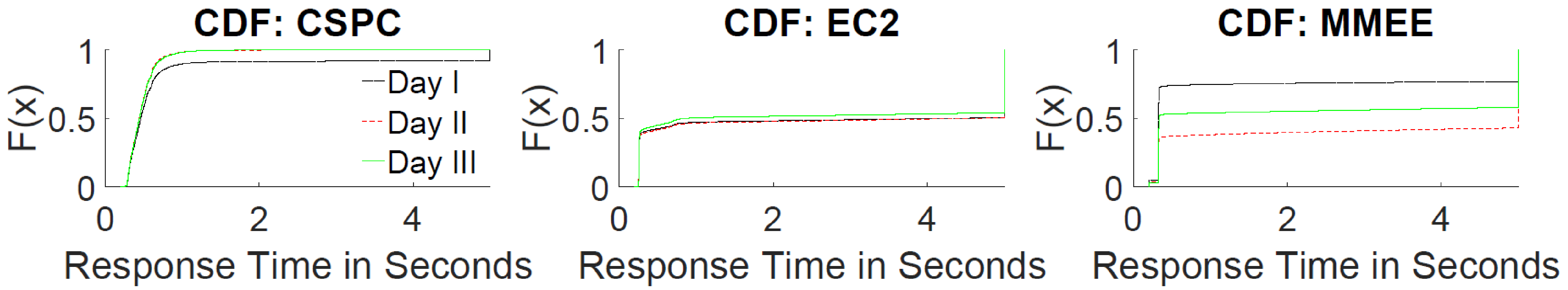
- 3 days from FIFA'98 workload as repetitions with load variation
- LU decomposition worklet from SPEC SERT 2 suite, CPU-bound, low I/O

Auto-Scaler Config:

- CPU utilization collected via TOP command and averaged across running VMs
- Scale UP 1VM, if CPU util. >90% for 1 min,
Scale DOWN 1VM, if CPU util. <60% for 1 min
- Auto-scaler and experiment data: https://doi.org/10.5281/zenodo.1169900

RQ 2 - Cloud Experiment: Example Results





RQ 2 - Cloud Experiment: Example Results

Average metric (and standard deviation) for a day in each scenario.

Metric	CSPC	EC2	MMEE
θ_U (accuracy$_U$)[%]	2.39 (1.54)	14.05 (1.82)	19.42 (5.04)
θ_O (accuracy$_O$)[%]	43.22 (4.38)	10.09 (1.75)	54.98 (11.87)
τ_U (time share$_U$)[%]	9.76 (4.77)	57.20 (2.60)	42.16 (1.76)
τ_O (time share$_O$)[%]	82.95 (5.46)	27.53 (4.42)	53.06 (3.08)
ψ (SLO violations)[%]	2.70 (3.68)	49.30 (1.71)	53.02 (7.11)
Avg. response time [s]	0.60 (0.17)	2.68 (0.08)	2.32 (0.68)
#Adaptations	25.67 (1.88)	80.66 (3.40)	39.67 (7.54)
Avg. #VMs [VMs]	10.53 (0.44)	8.84 (0.07)	11.01 (0.12)

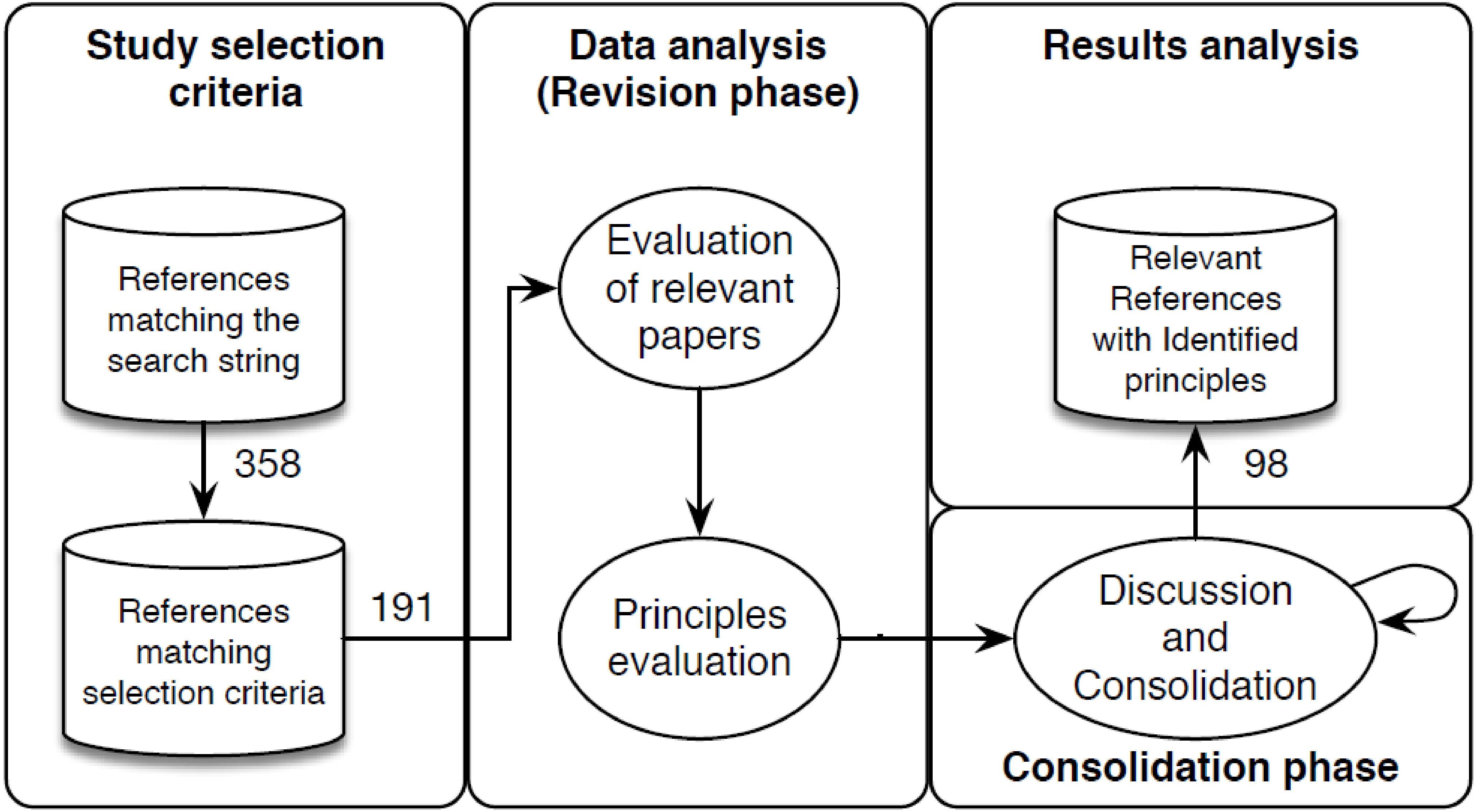
Cost overview of the experiments.

Instance hours	CSPC	EC2	MMEE
Used [h]	121.0	83.4	93.8
Charged [h]	121.0	131.0	93.8

ANOVA results per metric.

Statistic	θ_U	θ_O	τ_U	τ_O
$Pr(> F)$	0.006	0.001	0.003	0.003
Prop. of Var. due to Env. [%]	82	84	98	97

Systematic Review Process



Included Cloud Experimental Research Works

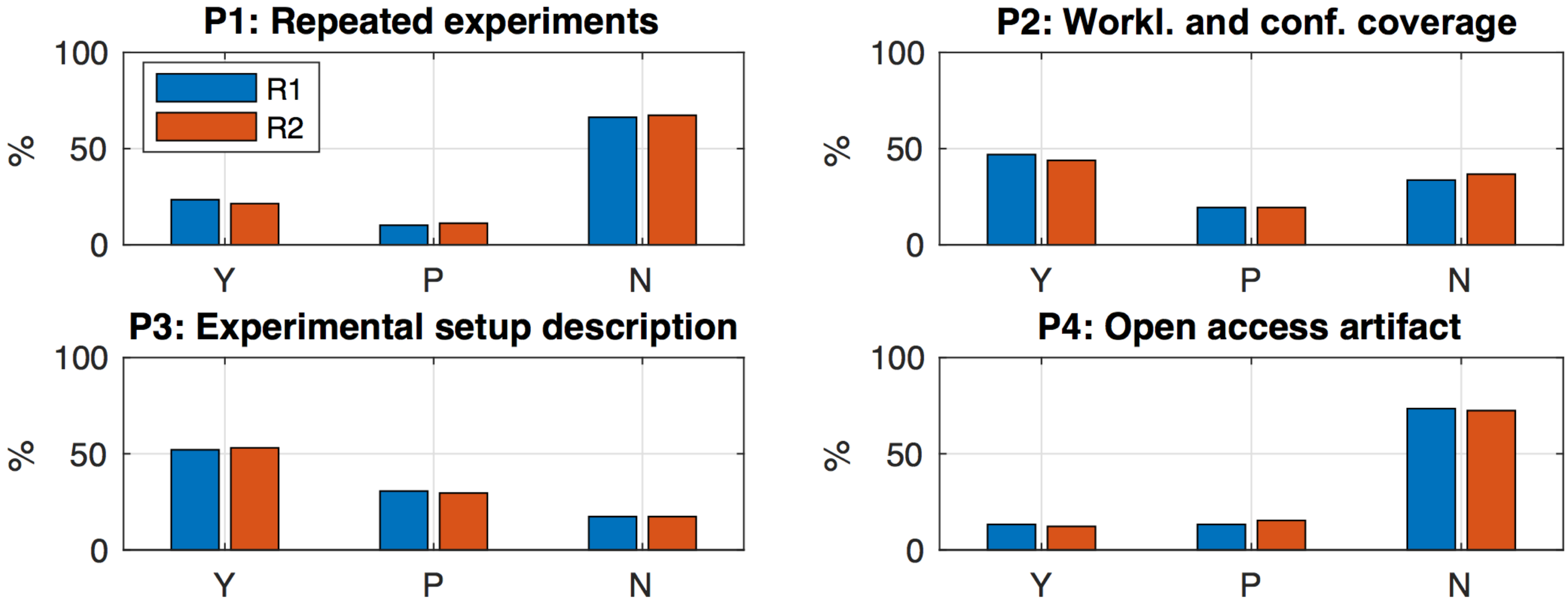
Venue	Total	2017	2016	2015	2014	2013	2012
IEEECloud	96	0	24	24	14	15	19
UCC	68	0	6	14	30	13	5
CCGrid	31	10	4	0	11	0	6
TPDS	31	8	6	6	4	5	2
IC2E	22	0	5	0	12	5	0
CloudCom	20	2	7	5	6	0	0
ICAC	18	3	3	6	4	1	1
ICPE	18	4	1	3	4	5	1
TCC	15	4	6	3	1	1	0
SoCC	11	0	0	3	4	3	1
HPDC	9	0	3	1	0	2	3
SIGMETRICS	6	1	0	0	2	1	2
FGCS	5	1	1	0	3	0	0
EuroSys	3	0	1	1	0	0	1
SC	3	0	0	0	1	2	0
NSDI	2	0	0	0	1	1	0
Total	358	33	67	66	97	54	41

Search in late 2017 for

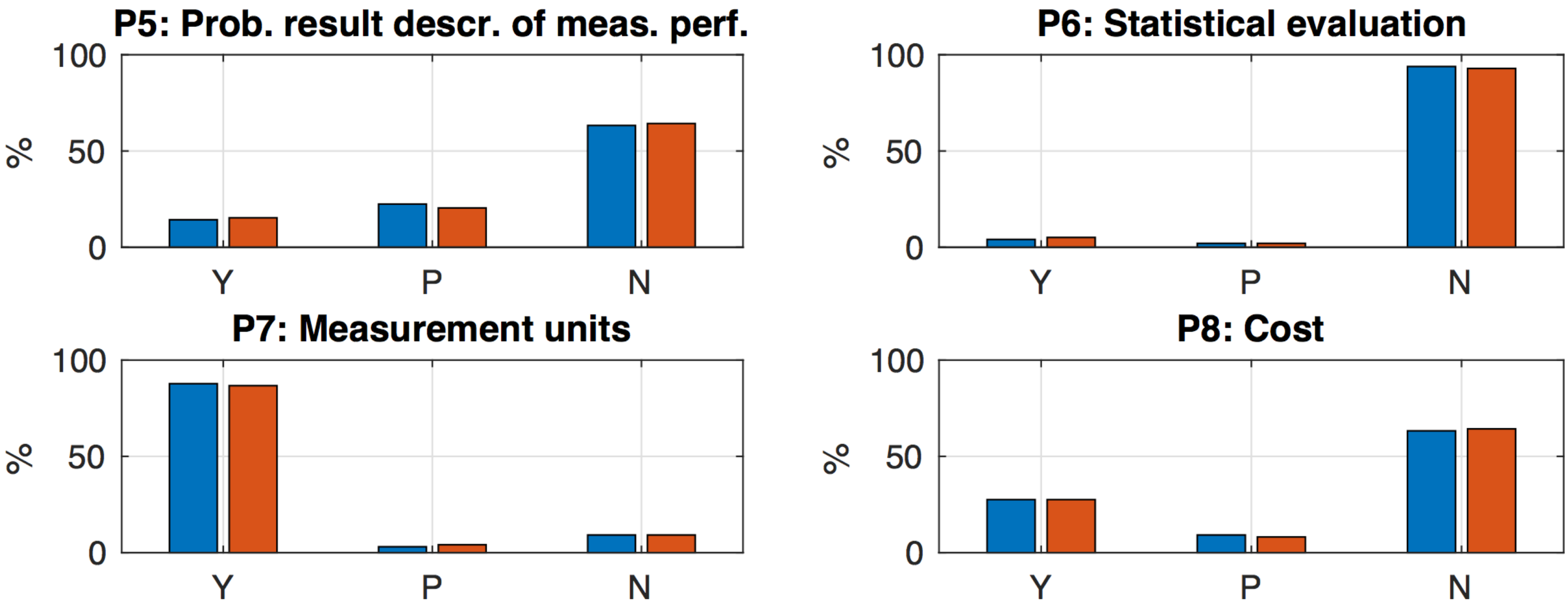
Including:
“cloud”
“experiment”
“management”

Excluding:
“security”

RQ3 How are cloud performance results currently obtained and reported?



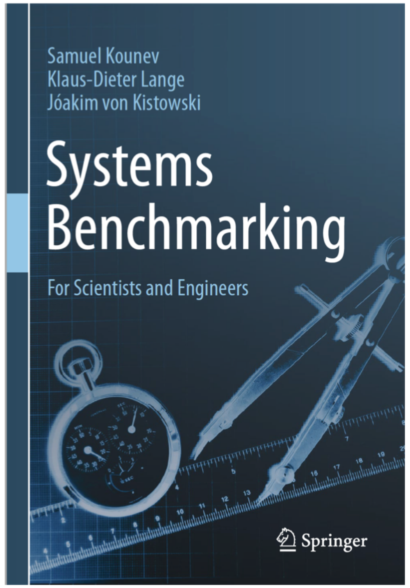
RQ3 How are cloud performance results currently obtained and reported? (cont.)



Conclusions

- Systems benchmarking has grown into a **scientific discipline**
- Building fair and reliable benchmarks poses many challenges
  - Representative **metrics** are needed to understand system behavior
  - Choice of **workloads** is critical for fair comparisons
  - A solid measurement **methodology** is essential
- **Standardization** is important to avoid biased designs
- We are still far from seeing a broad adoption of even basic **measurement principles** in performance evaluations
- **Education on benchmarking** is urgently needed
  - Both for industry and academia

Questions?



http://benchmarking-book.com

http://research.spec.org/