BUNGEE: An Elasticity Benchmark for Self-Adaptive IaaS Cloud Environments

Nikolas Herbst, Andreas Weber,
Henning Groenda, Samuel Kounev

Dept. of Computer Science,
University of Würzburg
FZI Research Center, Karlsruhe

SEAMS 2015, Firenze, Italy
May 18, 2015

http://descartes.tools/bungee

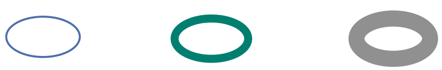
Characteristics of …

Rubber Bands	**Clouds**

Rubber Bands

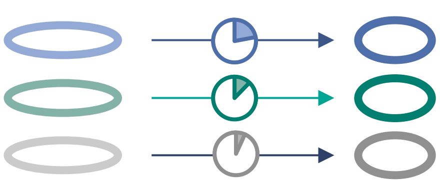
Base Length

Width/Thickness/Force



Strechability

Elasticity

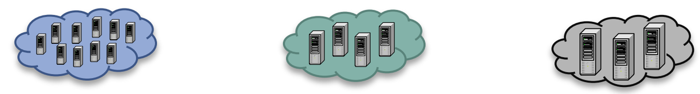


Price

$ $$ $$$

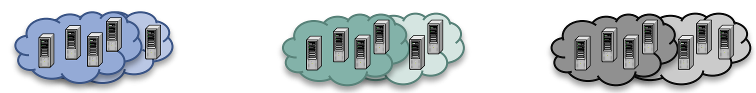
Clouds

Performance (1 resource unit)

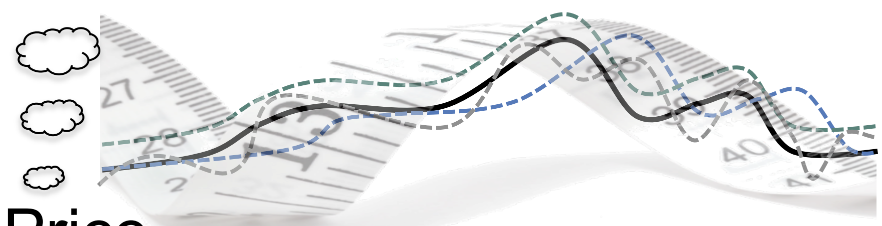


Quality Criteria / SLOs

Scalability



Elasticity

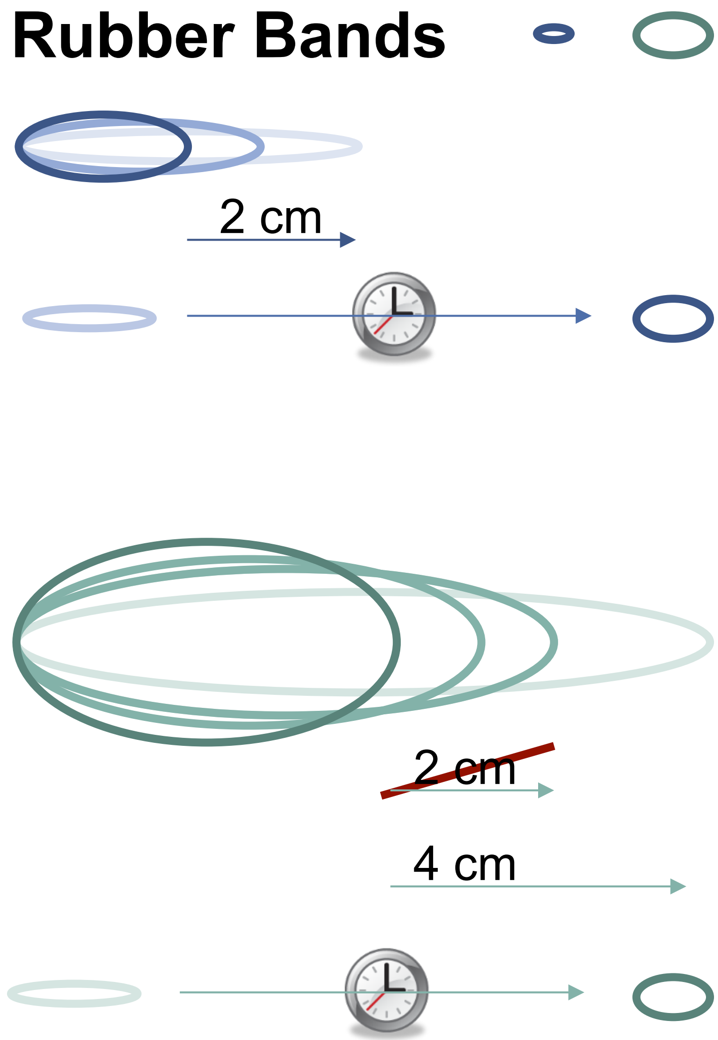


Price

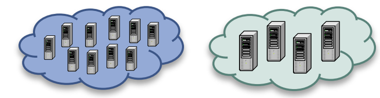
$ $$ $$$

Comparing Elastic Behavior of …

Rubber Bands



IaaS Clouds



Measure elasticity

independent of

performance
and
scalability

Agenda

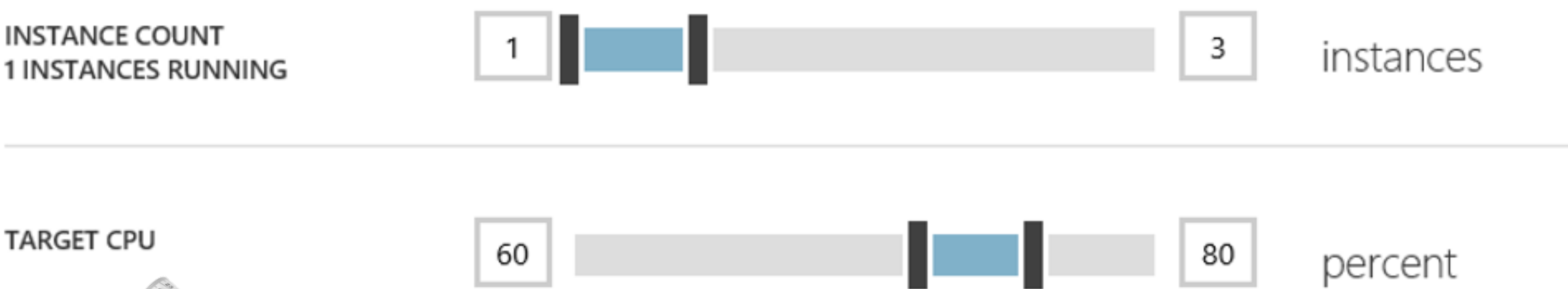
- Motivation
- Related Work
- Benchmark Concept & Implementation
- Evaluation & Case Study
- Conclusion

Motivation

Elasticity:

- Mayor quality attribute of clouds [Gartner09]
- Many strategies exist
  - Industry
  - Academia [Galante12, Jennings14]

amazon webservices™ EC2



→  Benchmark for comparability!

*"You can't **control** what you can't measure?"* (DeMarco)
*"If you cannot measure it, you cannot **improve** it"* (Lord Kelvin)

Related Work

- Specialized approaches
  - Measure technical provisioning time
  - Measure SLA compliance
  - Focus on scale up/out

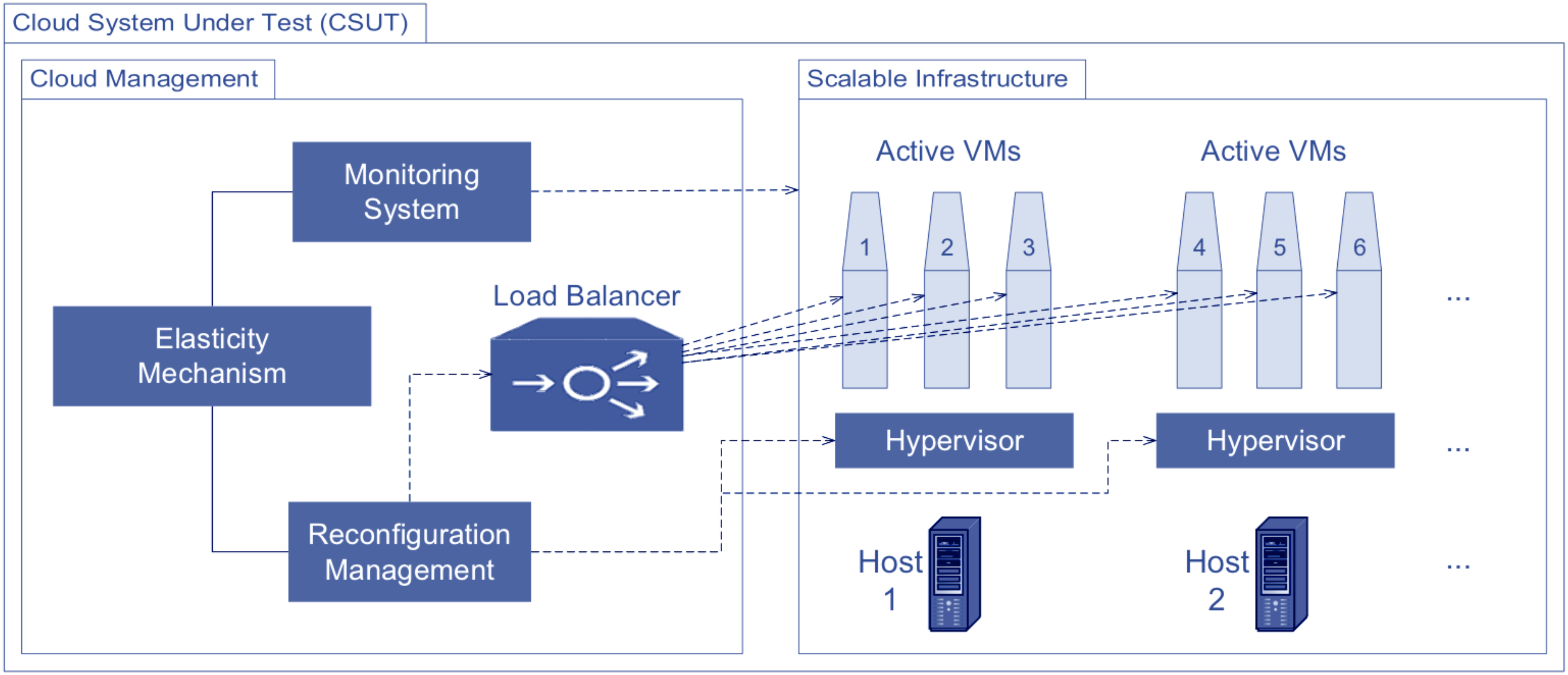
[Binning09, Li10, Dory11, Almeida13]

- Business perspective
  - What is the financial impact?
  - Disadvantage:

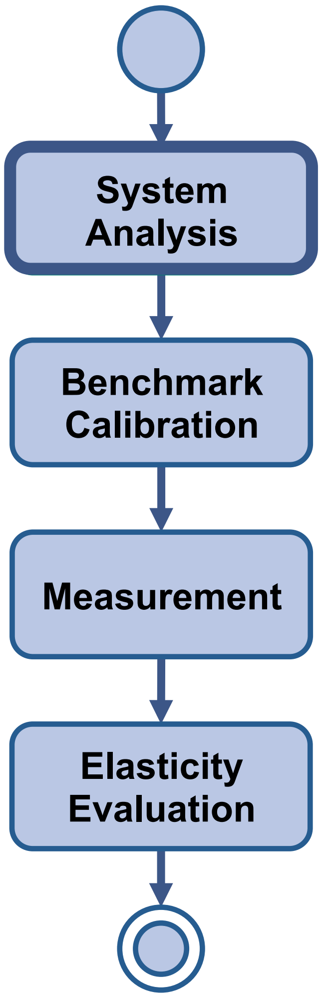
    Mix-up of elasticity technique and business model

[Weimann11, Folkerts12, Islam12, Moldovan13, Tinnefeld14]

Cloud System Under Test



Elasticity Benchmarking Concept

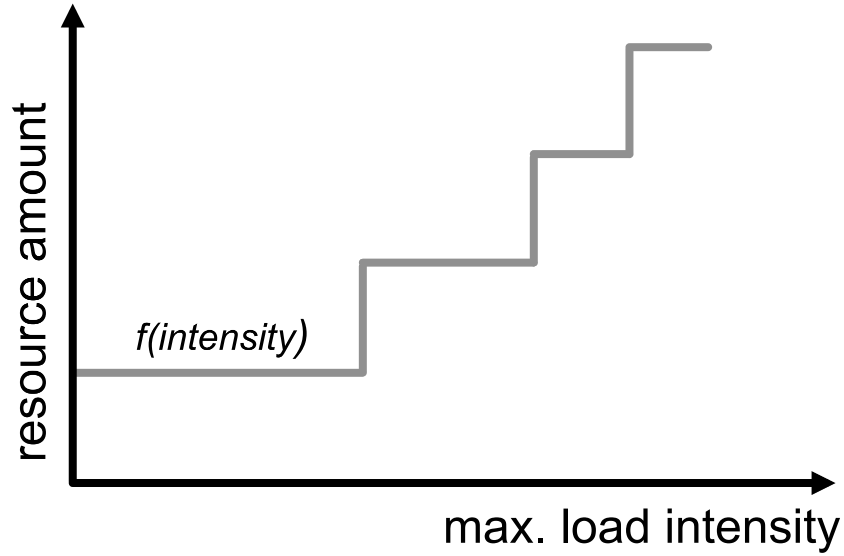


Analyze
performance of underlying resources &
scaling behavior

Analyze System Phase

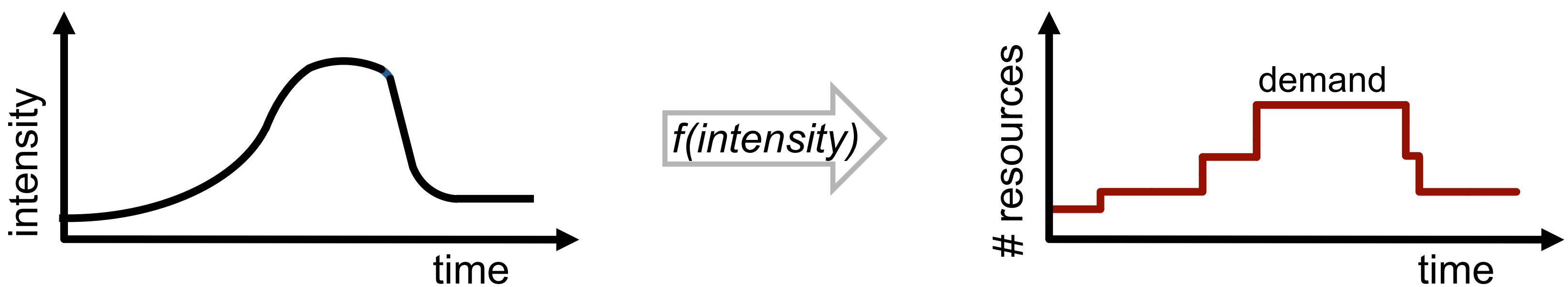
Approach:

- Evaluate system separately at each scale
- Find **maximal intensity** that the system can withstand **without violating SLO** (binary search)
- Derive demand step function: *resourceDemand = f(intensity)*

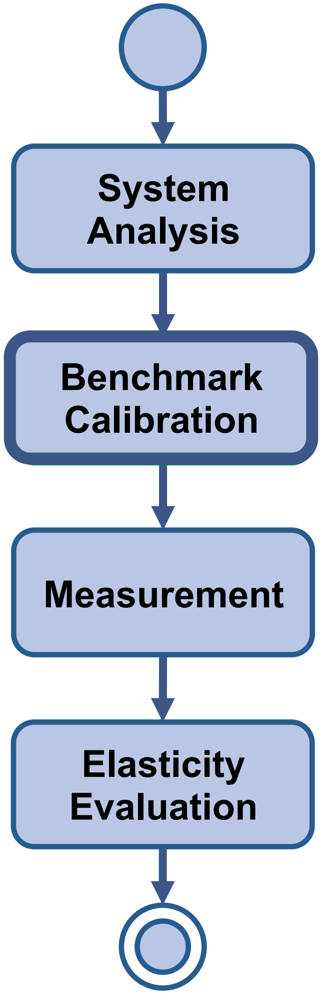


Benefit:

- Derive resource demand for arbitrary load intensity variations

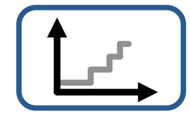


Elasticity Benchmarking Concept



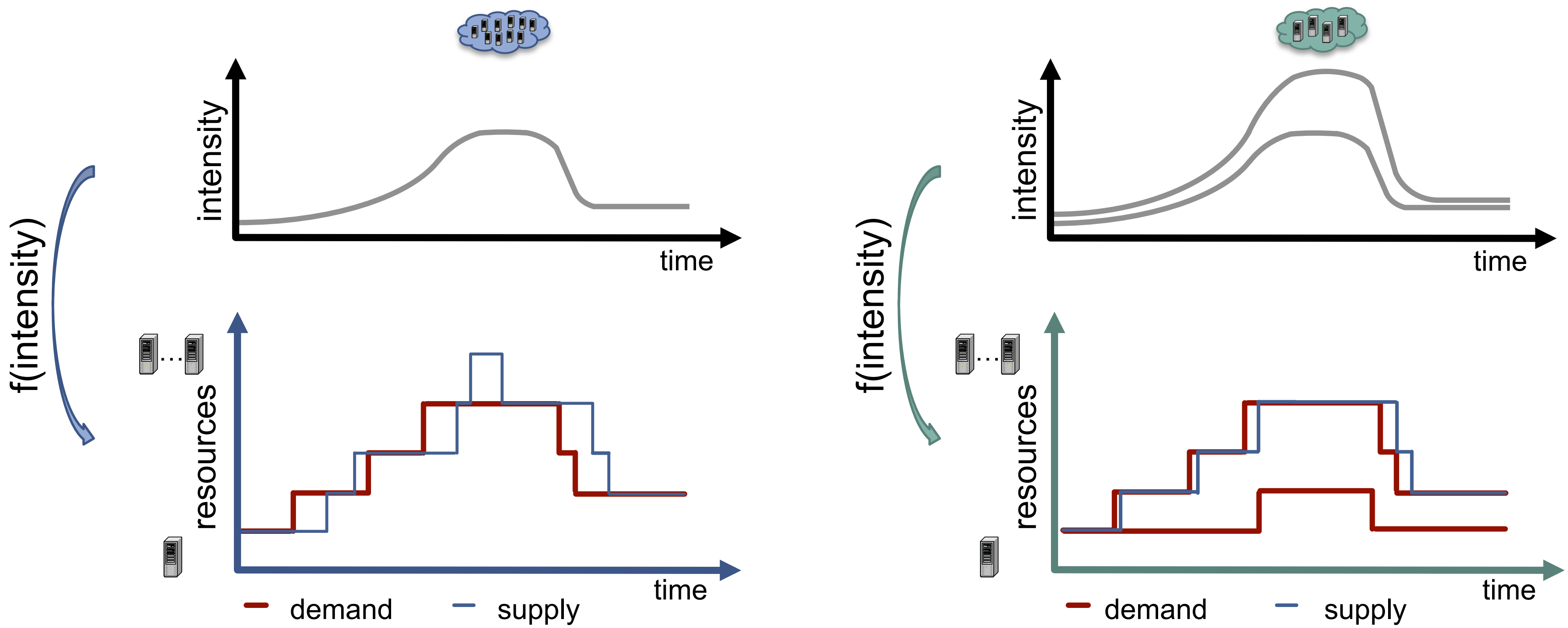
Analyze
performance of underlying resources &
scaling behavior

Adjust load profile



Benchmark Calibration Phase

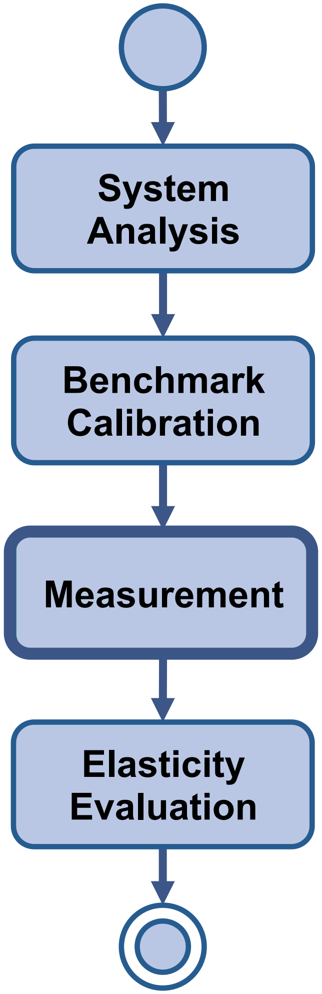
Goal: Induce same resource demand on all systems



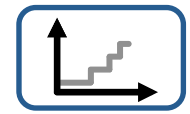
Approach: Adjust load intensity profile to overcome

- Different performance of underlying resources
- Different scalability

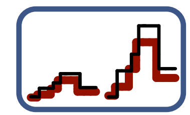
Elasticity Benchmarking Concept



Analyze
performance of underlying resources &
scaling behavior



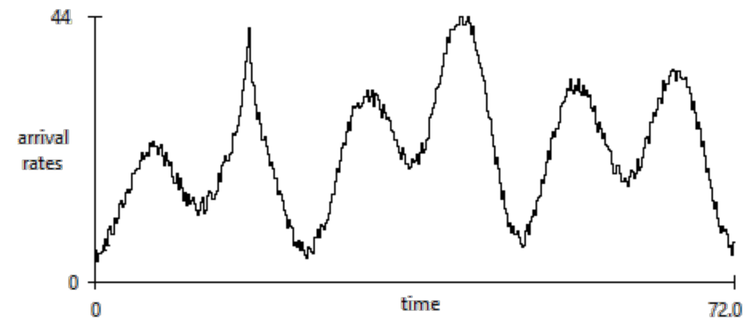
Adjust load profile



Expose CSUT to varying load
&
monitor resource supply & demand

Measurement Phase

- Requirements: Stress SUT in a representative manner
  - Realistic variability of load intensity
  - Adaptability of load profiles to suit different domains



http://descartes.tools/limbo

- Approach:
  - Open workload model [Schroeder06]
  - Model load variations with the LIMBO toolkit [SEAMS15Kistowski]

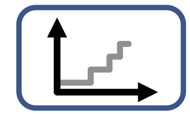
    Facilitates creation of new load profiles
    - Derived from existing traces
    - With desired properties (e.g. seasonal pattern, bursts)
  - Execute load profile using JMeter

    A JMeter Timer-Plugin delays requests according to timestamp file created by LIMBO

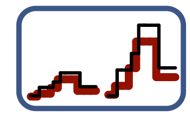
https://github.com/andreaswe/JMeterTimestampTimer

Elasticity Benchmarking Concept

System Analysis — Analyze performance of underlying resources & scaling behavior



Benchmark Calibration — Adjust load profile



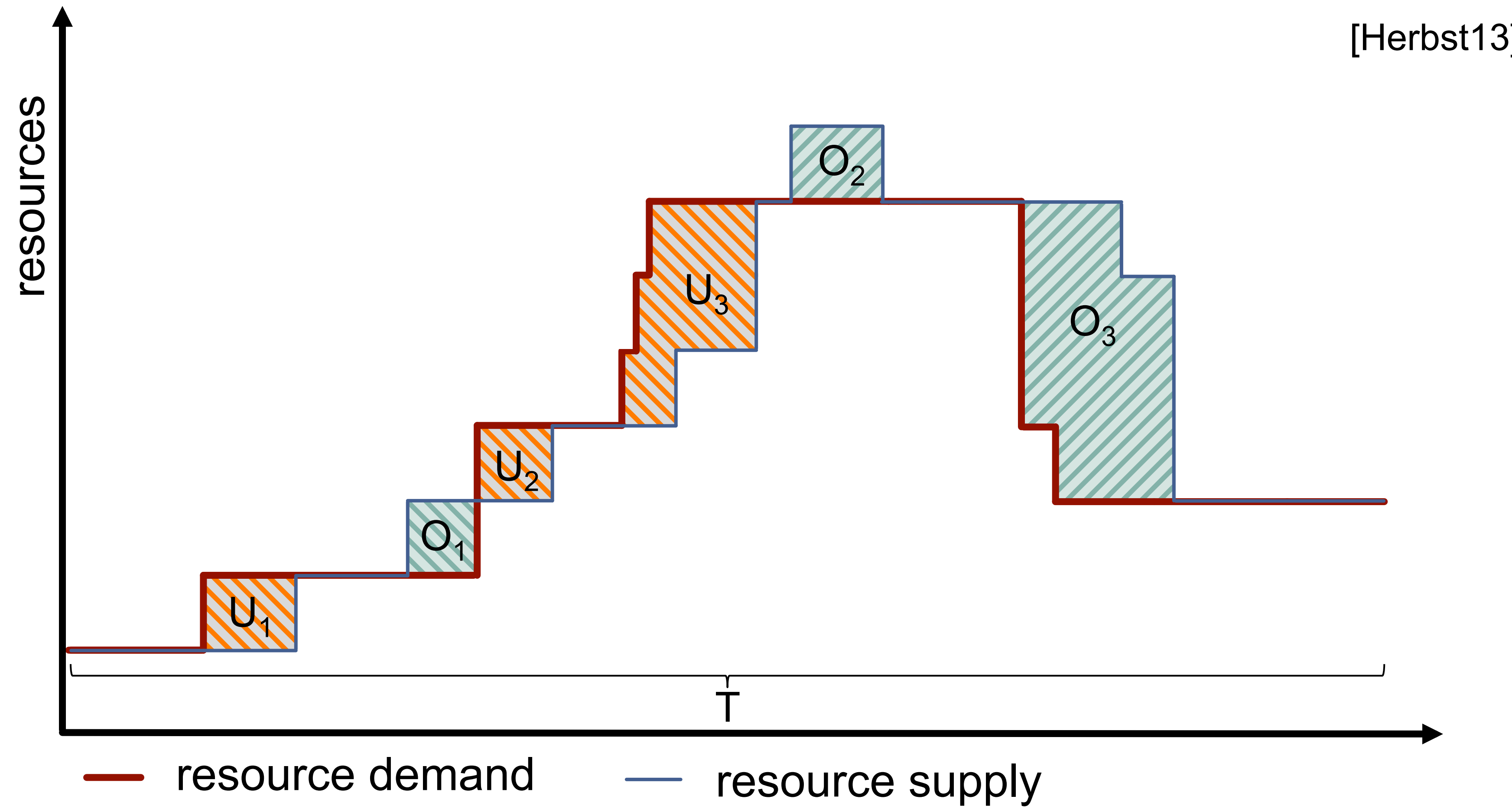
Measurement — Expose CSUT to varying load & monitor resource supply & demand

f(intensity)

LIMBO
Apache JMeter
amazon web services
CloudStack
EC2

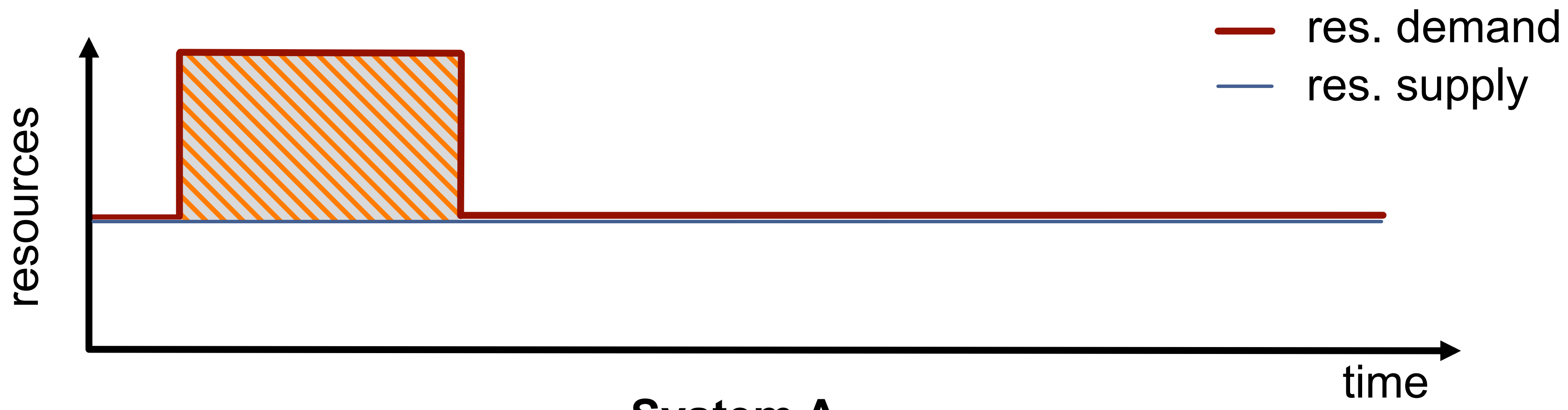
Elasticity Evaluation — Evaluate elasticity aspects accuracy & timing with metrics

Metrics: Accuracy (1/3)

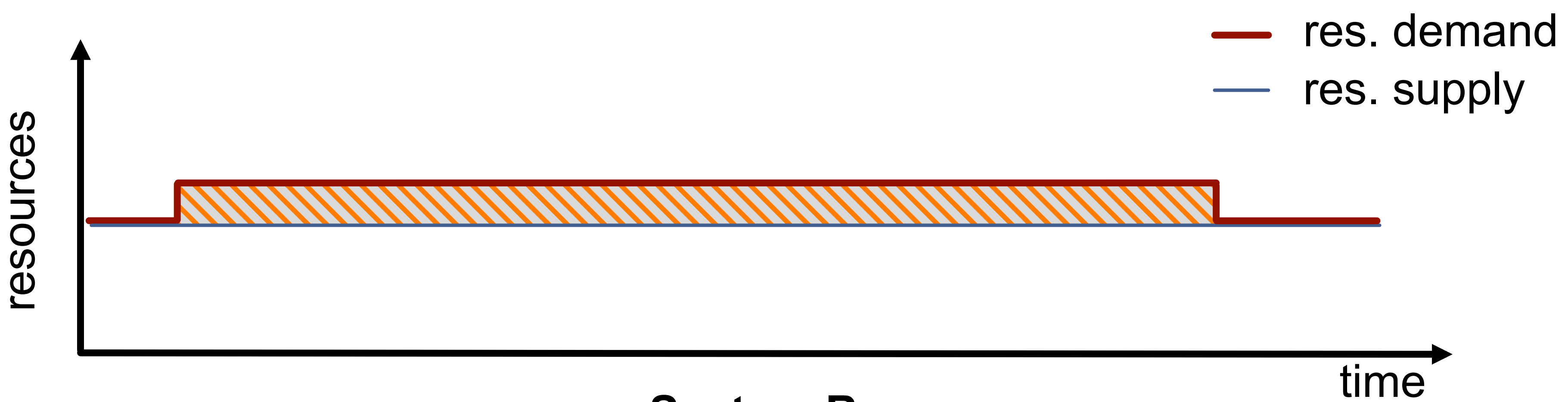


$$\text{accuracy}_U \quad \frac{\sum U}{T} \qquad\qquad \text{accuracy}_O \quad \frac{\sum O}{T}$$

Same Value – Different Behavior

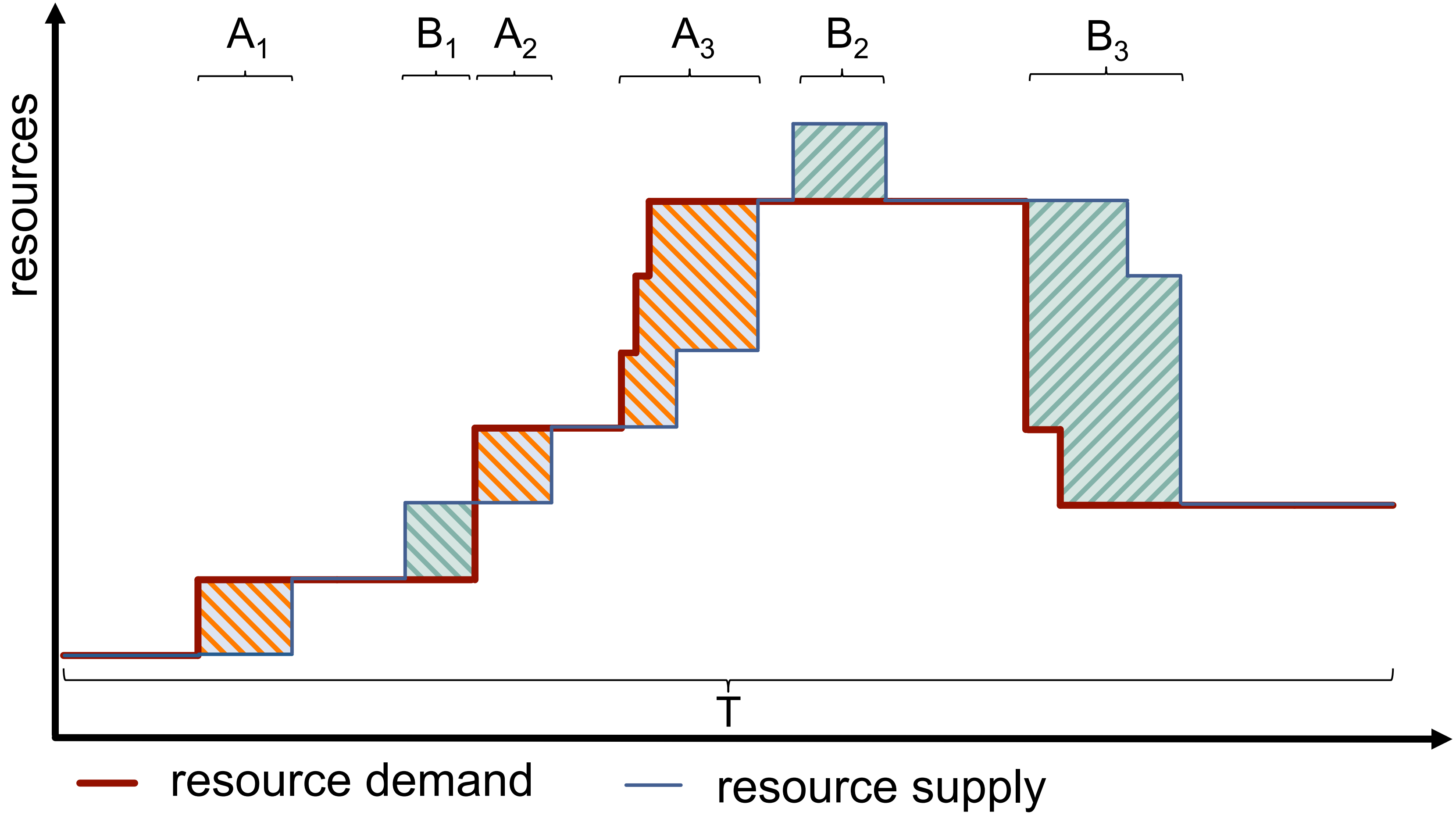


System A



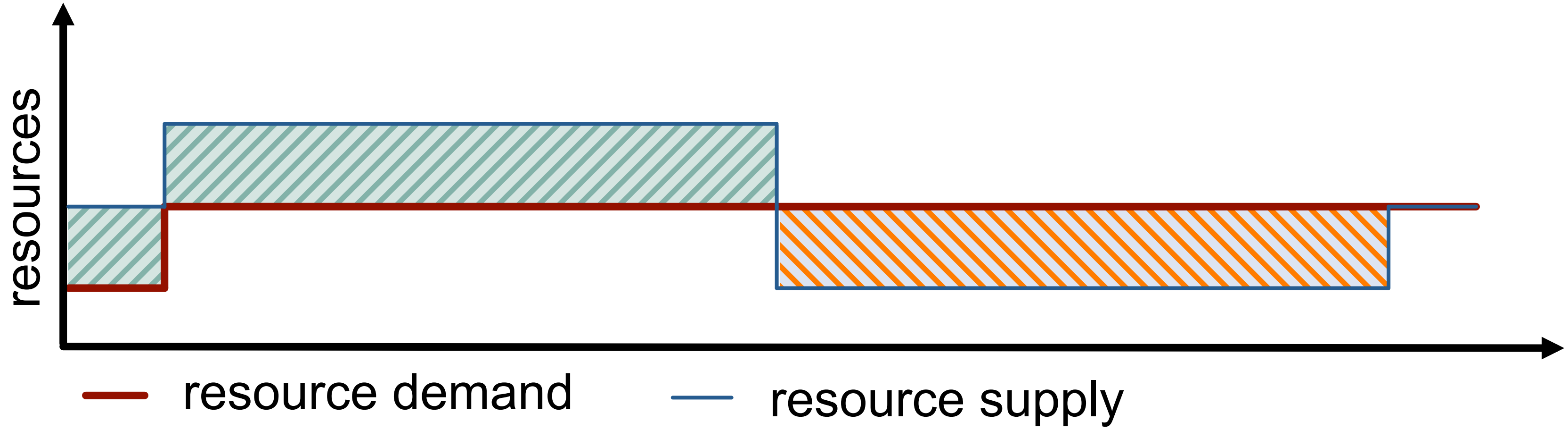
System B

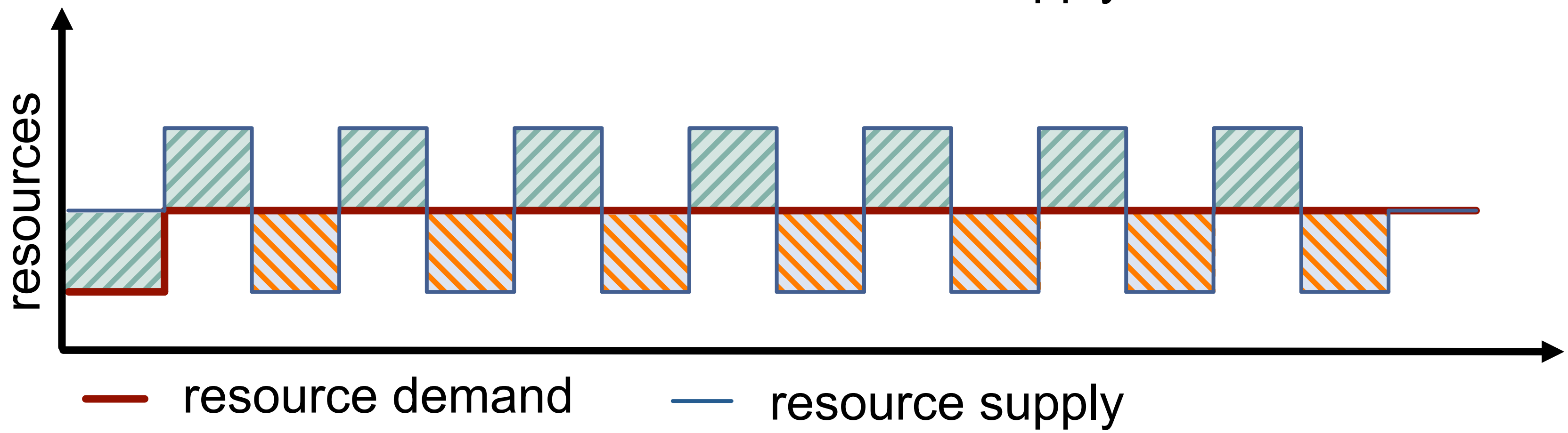
Metrics: Timeshare (2/3)



$$\text{timeshare}_U \quad \frac{\sum A}{T} \qquad\qquad \text{timeshare}_O \quad \frac{\sum B}{T}$$

Metrics: Jitter (3/3)



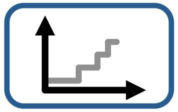


jitter $\frac{E_S - E_D}{T}$ E_D: # demand adaptations, E_S: # supply adaptations

Elasticity Benchmarking Concept

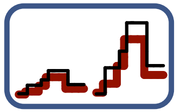
System Analysis

Analyze
performance of underlying resources &
scaling behavior



Benchmark Calibration

Adjust load profile



Measurement

Expose CSUT to varying load
&
monitor resource supply & demand

Elasticity Evaluation

Evaluate elasticity aspects
accuracy & timing
with metrics

BUNGEE Implementation

- Java-based elasticity benchmarking framework
- Components
  - Harness (Benchmark Node)
  - Cloud-side load generation application (CSUT)
- Automates the four benchmarking activities

System Analysis | Benchmark Calibration | Measurement | Elasticity Evaluation

- Currently: Analysis of horizontally scaling clouds based on
  - CloudStack
  - AWS
- Extensible with respect to
  - new cloud management software
  - new resource types
  - new metrics

Sources soon available at
http://descartes.tools/bungee

Evaluation & Case Study

- Evaluation (private cloud)
  - Reproducibility of system analysis

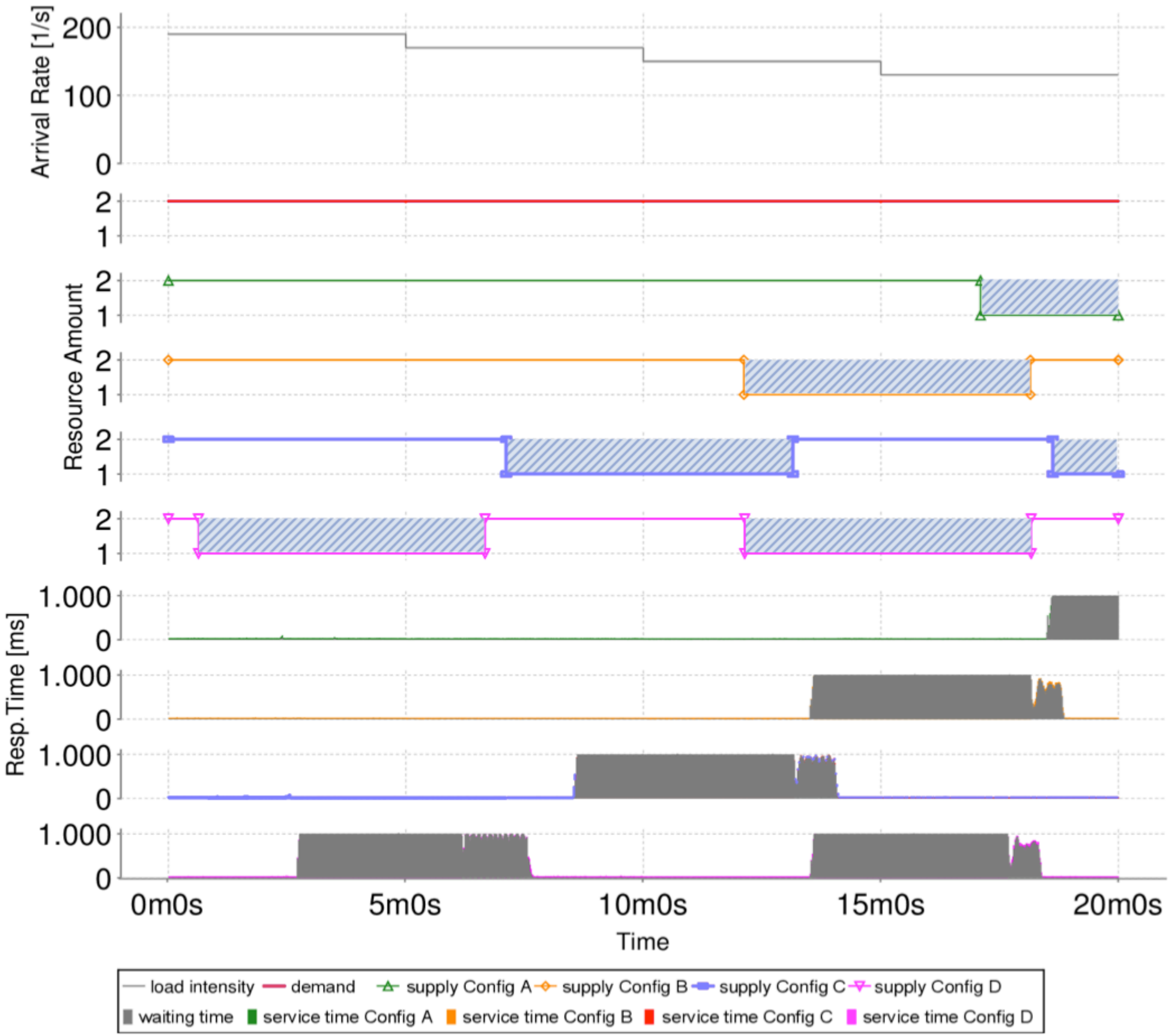
    $Err_{rel} < 5\%$, confidence 95% for first scaling stage
  - Simplified system analysis

    Linearity assumption holds for test system
  - Consistent ranking by metrics

    Separate evaluation for each metric, min. 4 configurations per metric

- Case Study (private & public cloud)
  - Applicability in real scenario
  - Different performance of underlying resources
  - Metric Aggregation

Evaluation: Accuracy_U



threshold Down [%]	accuarcy_U [res. units]
55	0.145
65	0.302
75	0.371
85	0.603

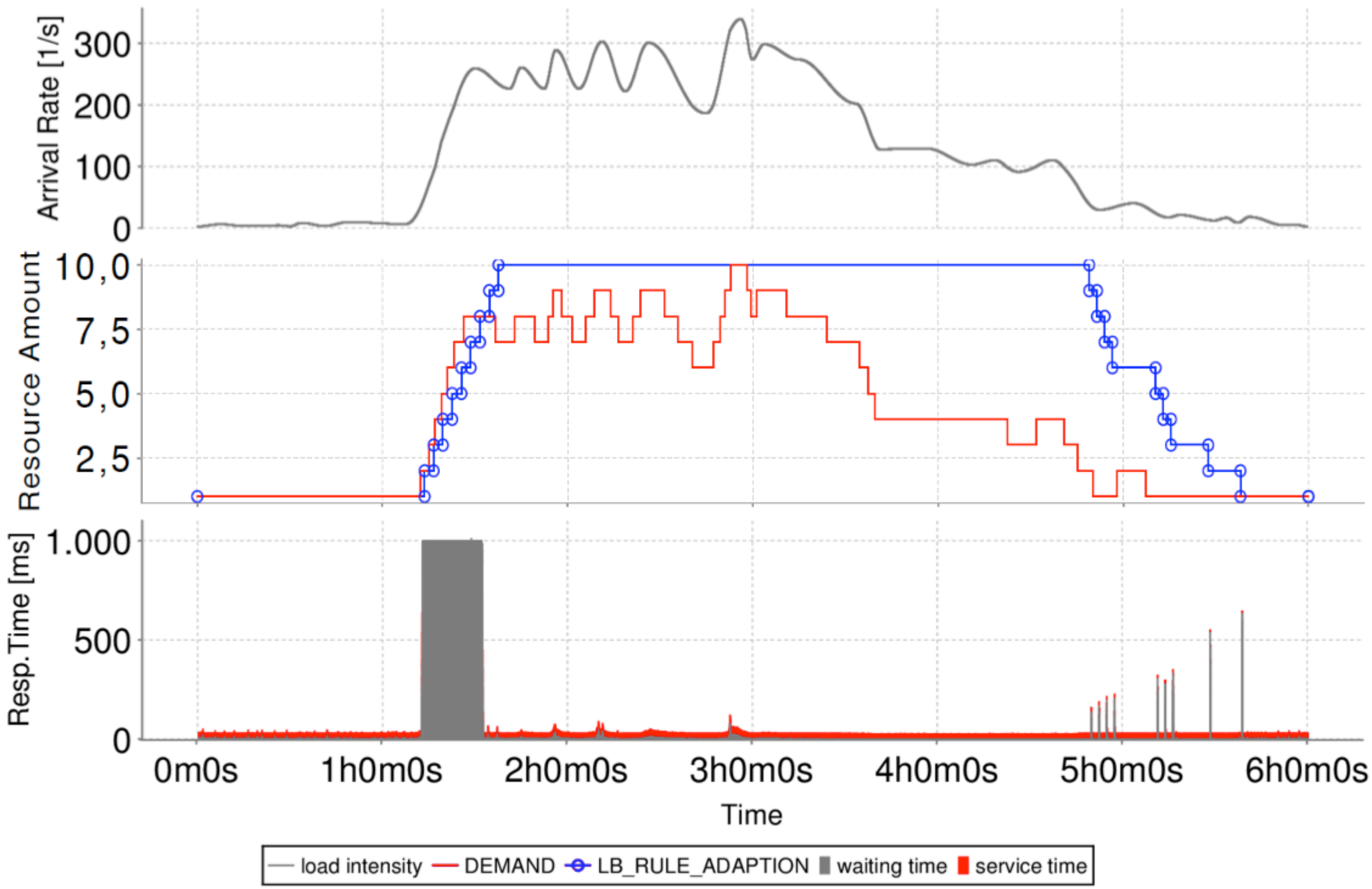
accuracy_U

allows to **rank** different

elastic behaviors on

an **ordinal scale**

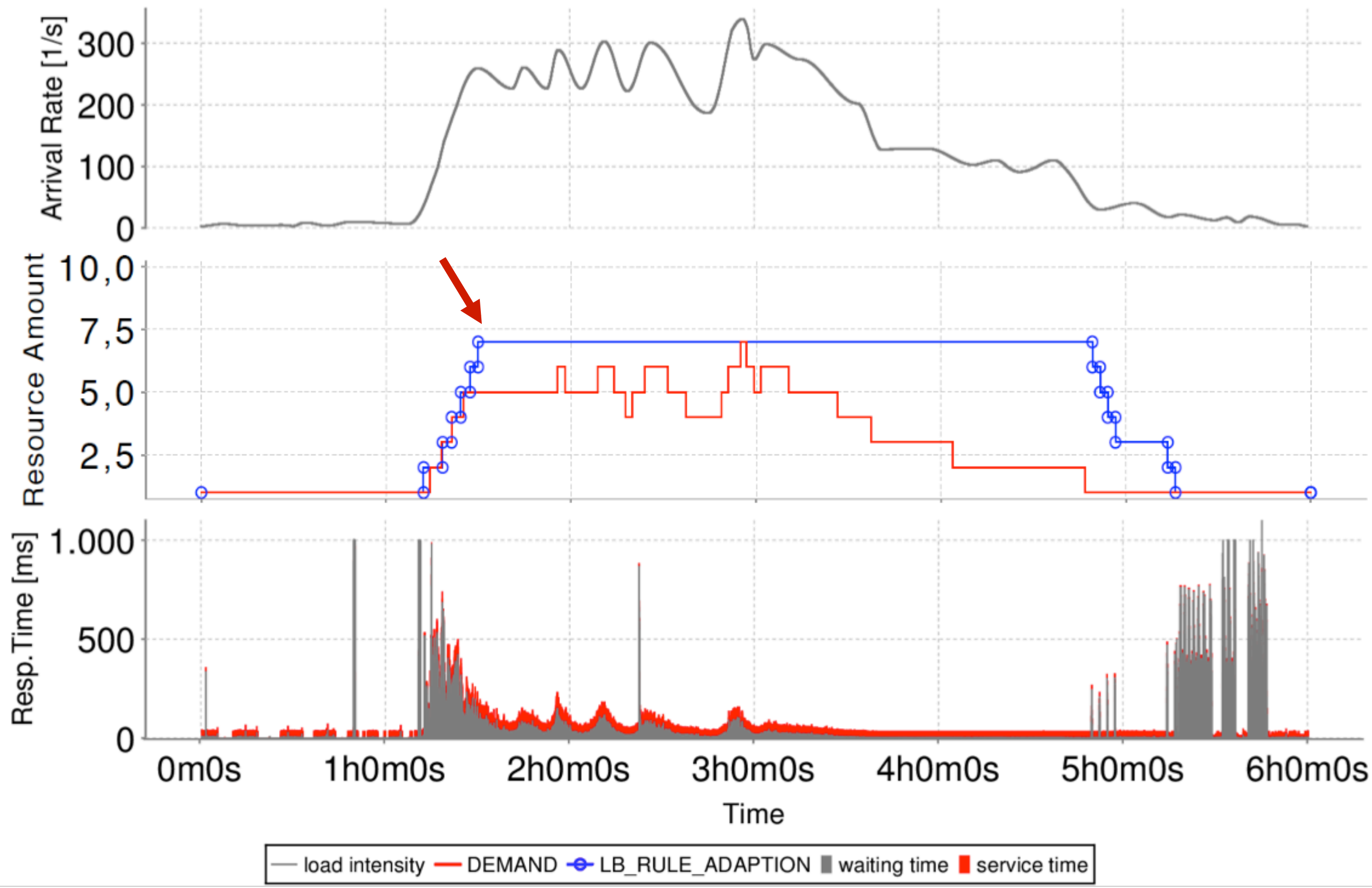
Case Study: Configuration F - 1Core



F - 1Core
quietTime
120s
condTrueDur
30s
threshUp
65%
threshDown
10%

Configuration	accuarcy$_O$ [res. units]	accuracy$_U$ [res. units]	timeshare$_O$ [%]	timeshare$_U$ [%]	jitter [adap/min.]	elastic speedup	violations [%]
F – 1Core	2.423	0.067	66.1	4.8	-0.067	1.046	7.6

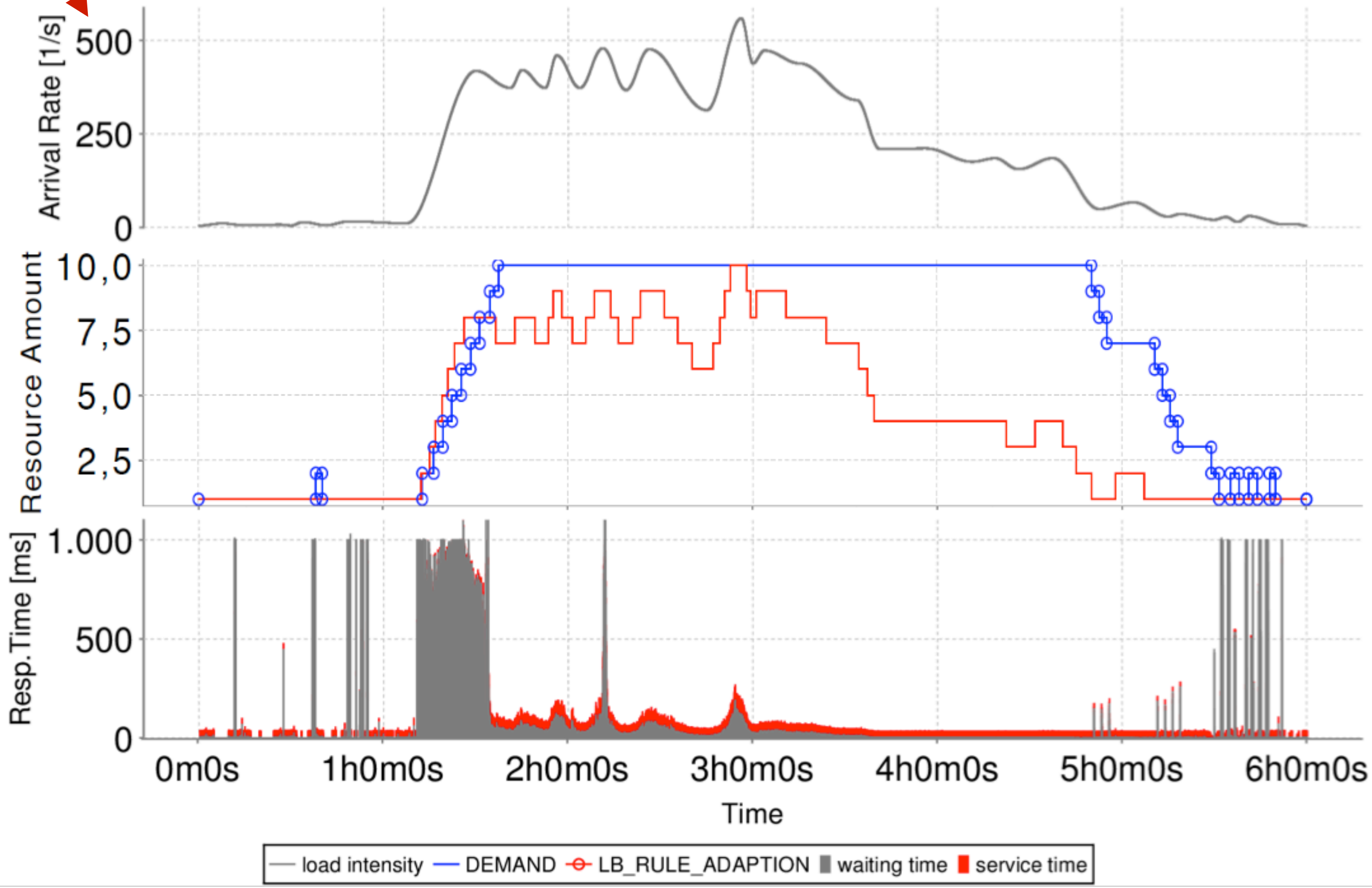
Case Study: Config. F - 2Core not adjusted



F - 2Core no adjustment
quietTime
120s
condTrueDur
30s
threshUp
65%
threshDown
10%

Configuration	accuarcy$_O$ [res. units]	accuracy$_U$ [res. units]	timeshare$_O$ [%]	timeshare$_U$ [%]	jitter [adap/min.]	elastic speedup	violations [%]
F – 1Core	2.423	0.067	66.1	4.8	-0.067	1.046	7.6
F – 2Core no adjustment	1.811	0.001	63.8	0.1	-0.033	1.291	2.1

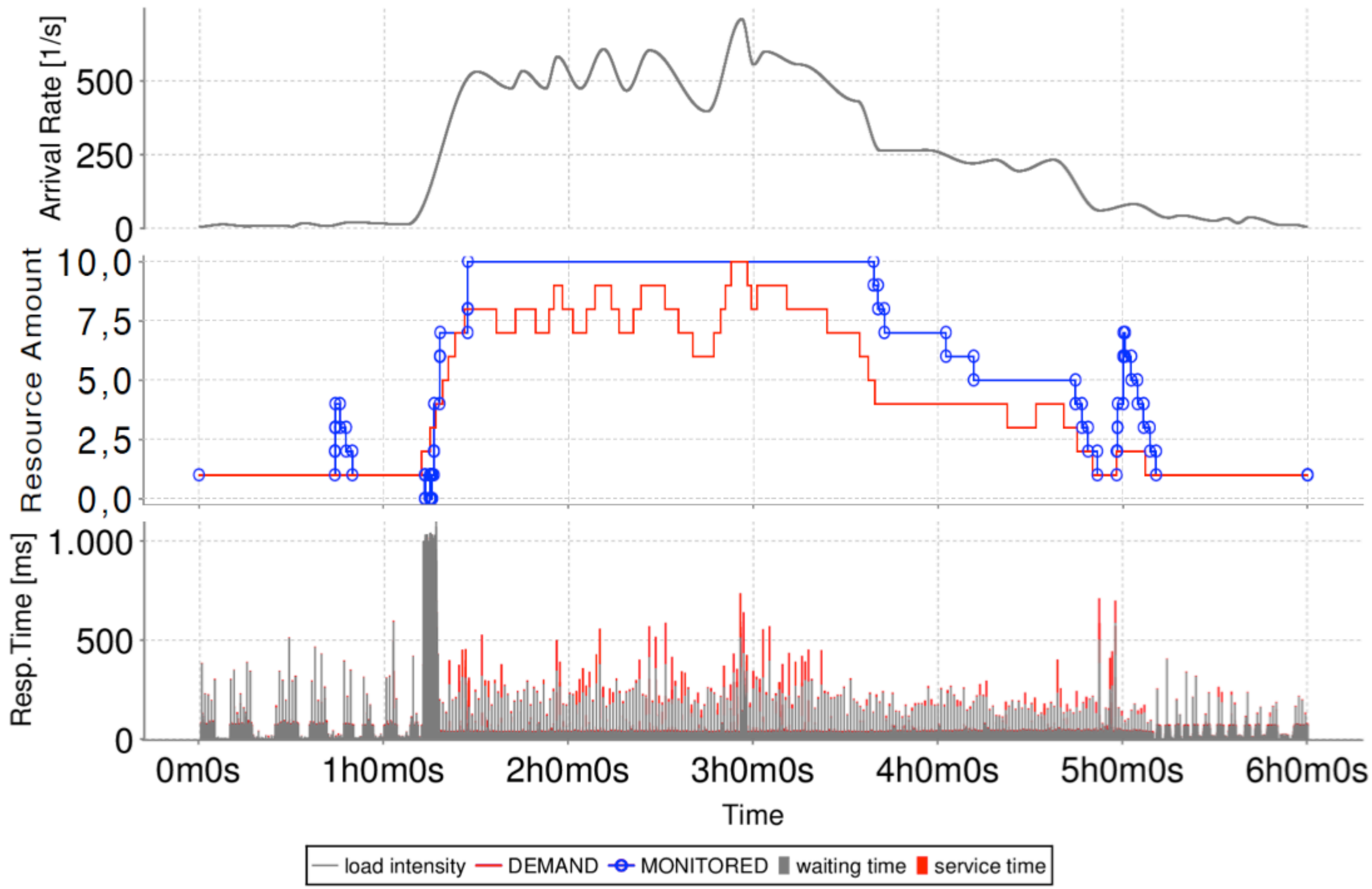
Case Study: Config. F - 2Core adjusted



F - 2Core adjusted
quietTime
120s
condTrueDur
30s
threshUp
65%
threshDown
10%

Configuration	accuarcy$_O$ [res. units]	accuracy$_U$ [res. units]	timeshare$_O$ [%]	timeshare$_U$ [%]	jitter [adap/min.]	elastic speedup	violations [%]
F – 1Core	**2.423**	**0.067**	**66.1**	**4.8**	**-0.067**	**1.046**	**7.6**
F – 2Core no adjustment	1.811	0.001	63.8	0.1	-0.033	1.291	2.1
F – 2Core adjusted	**2.508**	**0.061**	**67.1**	**4.5**	**-0.044**	**1.025**	**8.2**

Case Study: Config. K – AWS m1.small



K - AWS m1.small
quietTime
60s
condTrueDur
60s
threshUp
80%
threshDown
50%
instUp/Down
3/1

Configuration	accuarcy$_O$ [res. units]	accuracy$_U$ [res. units]	timeshare$_O$ [%]	timeshare$_U$ [%]	jitter [adap/min.]	elastic speedup	violations [%]
F – 1Core	2.423	0.067	66.1	4.8	-0.067	1.046	7.6
F – 2Core adjusted	2.508	0.061	67.1	4.5	-0.044	1.025	8.2
K – AWS m1.small	1.340	0.019	61.6	1.4	0.000	**1.502**	2.5

Conclusion

Goal	▪ Evaluate elastic behavior independent of ▪ Performance of underlying resources and scaling behavior ▪ Business model
Contribution	▪ Elasticity benchmark concept for IaaS cloud platforms ▪ Refined set of elasticity metrics ▪ Concept implementation: BUNGEE - framework for elasticity benchmarking
Evaluation	▪ Consistent ranking of elastic behavior by metrics ▪ Case study on AWS and CloudStack
Future Work	▪ BUNGEE: Distributed load generation, scale vertically, dif. resource types ▪ Experiments: Tuning of elasticity parameters, evaluate proactive controllers

Literature (1/2)

Gartner09: D.C. Plume, D. M. Smith, T.J. Bittman, D.W. Cearley, D.J. Cappuccio, D. Scott, R. Kumar, and B. Robertson. Study: “Five Refining Attributes of Public and Private Cloud Computing", Tech. rep., Gartner, 2009.

Galante12: G. Galante and L. C. E. d. Bona, “A Survey on Cloud Computing Elasticity" in Proceedings of the 2012 IEEE/ACM Fifth International Conference on Utility and Cloud Computing, Washington, 2012

Jennings14: B. Jennings and R. Stadler, “Resource management in clouds: Survey and research challenges“, Journal of Network and Systems Management, pp. 1-53, 2014

Binning09: C. Binnig, D. Kossmann, T. Kraska, and S. Loesing, “How is the weather tomorrow?: towards a benchmark for the cloud" in Proceedings of the Second International Workshop on Testing Database Systems, 2009

Li10: A. Li, X. Yang, S. Kandula, and M. Zhang, “CloudCmp: Comparing Public Cloud Providers" in Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, 2010

Dory11: T. Dory, B. Mejías, P. V. Roy, and N.-L. Tran, “Measuring Elasticity for Cloud Databases" in Proceedings of the The Second International Conference on Cloud Computing, GRIDs, and Virtualization, 2011

Almeida13: R.F. Almeida, F.R.C. Sousa, S. Lifschitz, and J.C. Machado: “On defining metrics for elasticity of cloud databases“, Simpósio Brasileiro de Banco de Dados - SBBD 2013, http://www.lbd.dcc.ufmg.br/colecoes/sbbd/2013/0012.pdf, last consulted July 2014

Weimann11: J. Weinman, “Time is Money: The Value of “On-Demand”,” 2011, http://www.joeweinman.com/resources/Joe_Weinman_Time_Is_Money.pdf, last consulted July 2014

Literature (2/2)

Islam12: S. Islam, K. Lee, A. Fekete, and A. Liu, “How a consumer can measure elasticity for cloud platforms" in Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering, New York, 2012

Folkerts12: E. Folkerts, A. Alexandrov, K. Sachs, A. Iosup, V. Markl, and C. Tosun, “Benchmarking in the Cloud: What It Should, Can, and Cannot Be“ in Selected Topics in Performance Evaluation and Benchmarking, Berlin Heidelberg, 2012

Moldovan13: D. Moldovan, G. Copil, H.-L. Truong, and S. Dustdar, “MELA: Monitoring and Analyzing Elasticity of Cloud Services,” in IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom), 2013

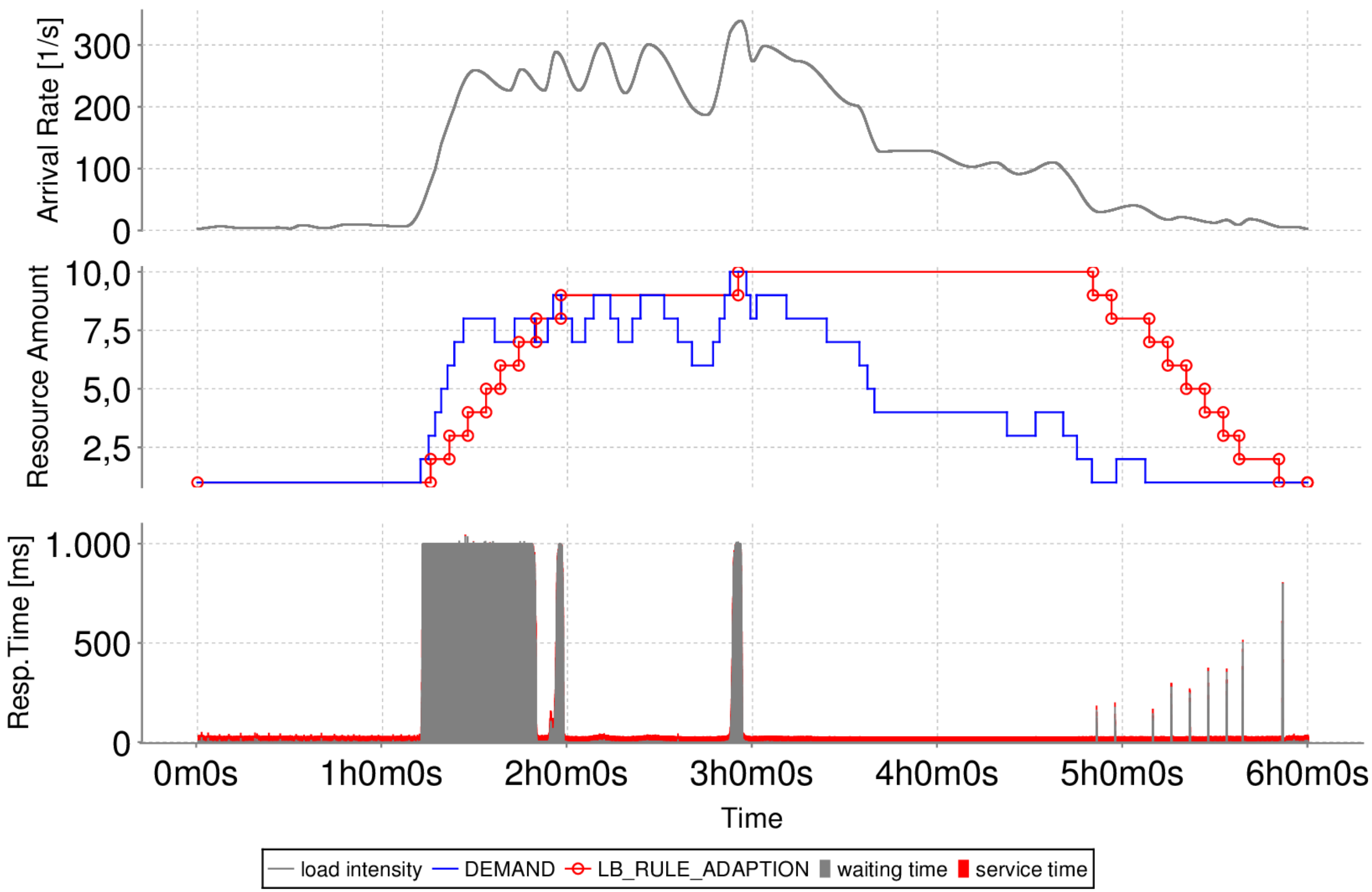
Tinnefeld14: C. Tinnefeld, D. Taschik, and H. Plattner, “Quantifying the Elasticity of a Database Management System,” in DBKDA 2014, The Sixth International Conference on Advances in Databases, Knowledge, and Data Applications, 2014

Schroeder06: B. Schroeder, A. Wierman, and M. Harchol-Balter, Open Versus Closed: A Cautionary Tale," in Proceedings of the 3rd Conference on Networked Systems Design & Implementation - Volume 3, ser. NSDI'06. Berkeley, CA, USA: USENIX Association, 2006

SEAMS15Kistowski: Jóakim von Kistowski, Nikolas Roman Herbst, Daniel Zoller, Samuel Kounev, and Andreas Hotho. Modeling and Extracting Load Intensity Profiles. In Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2015), Firenze, Italy, May 18-19, 2015.

Herbst13: N. R. 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, San Jose, 2013

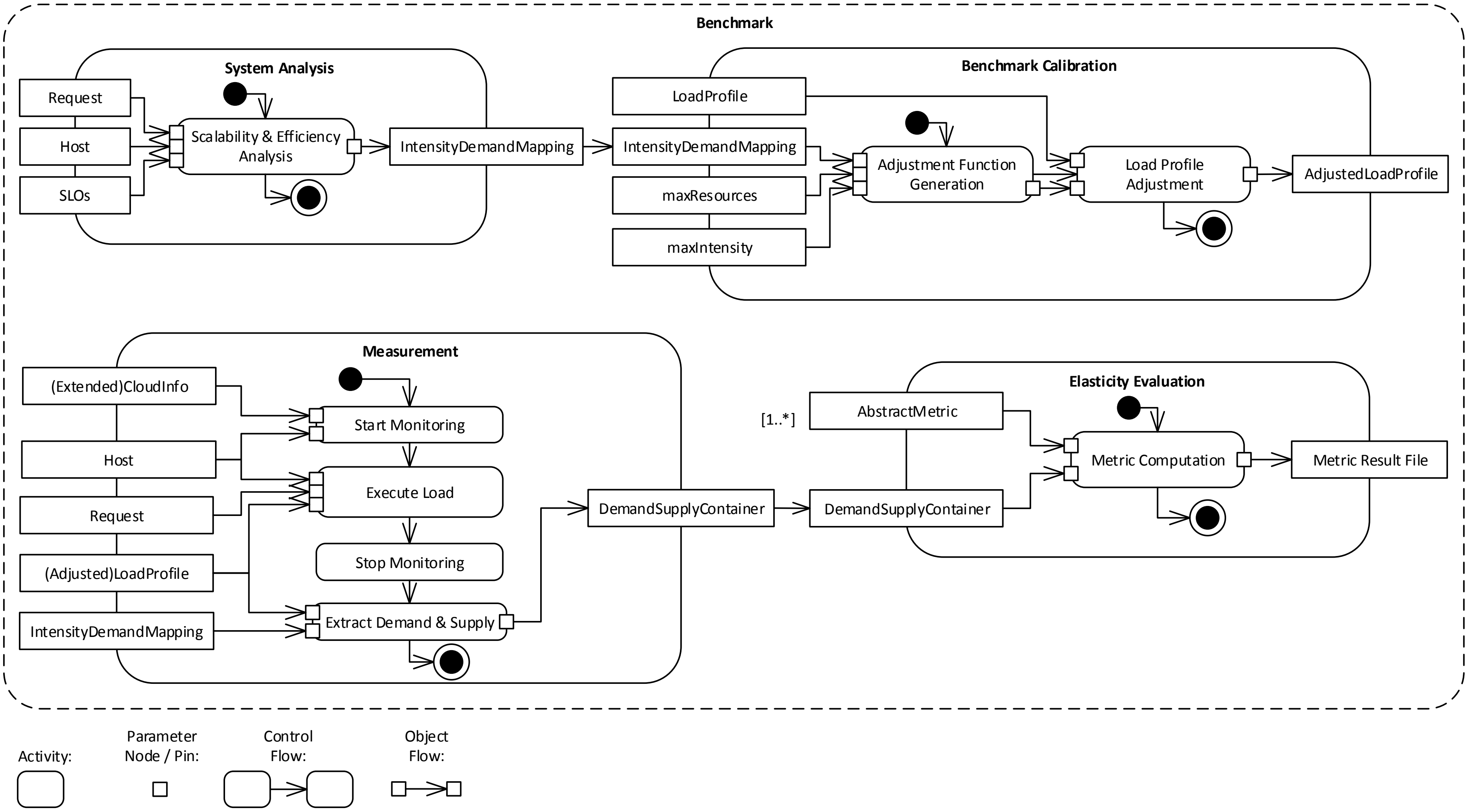
Case Study: A - Baseline Configuration



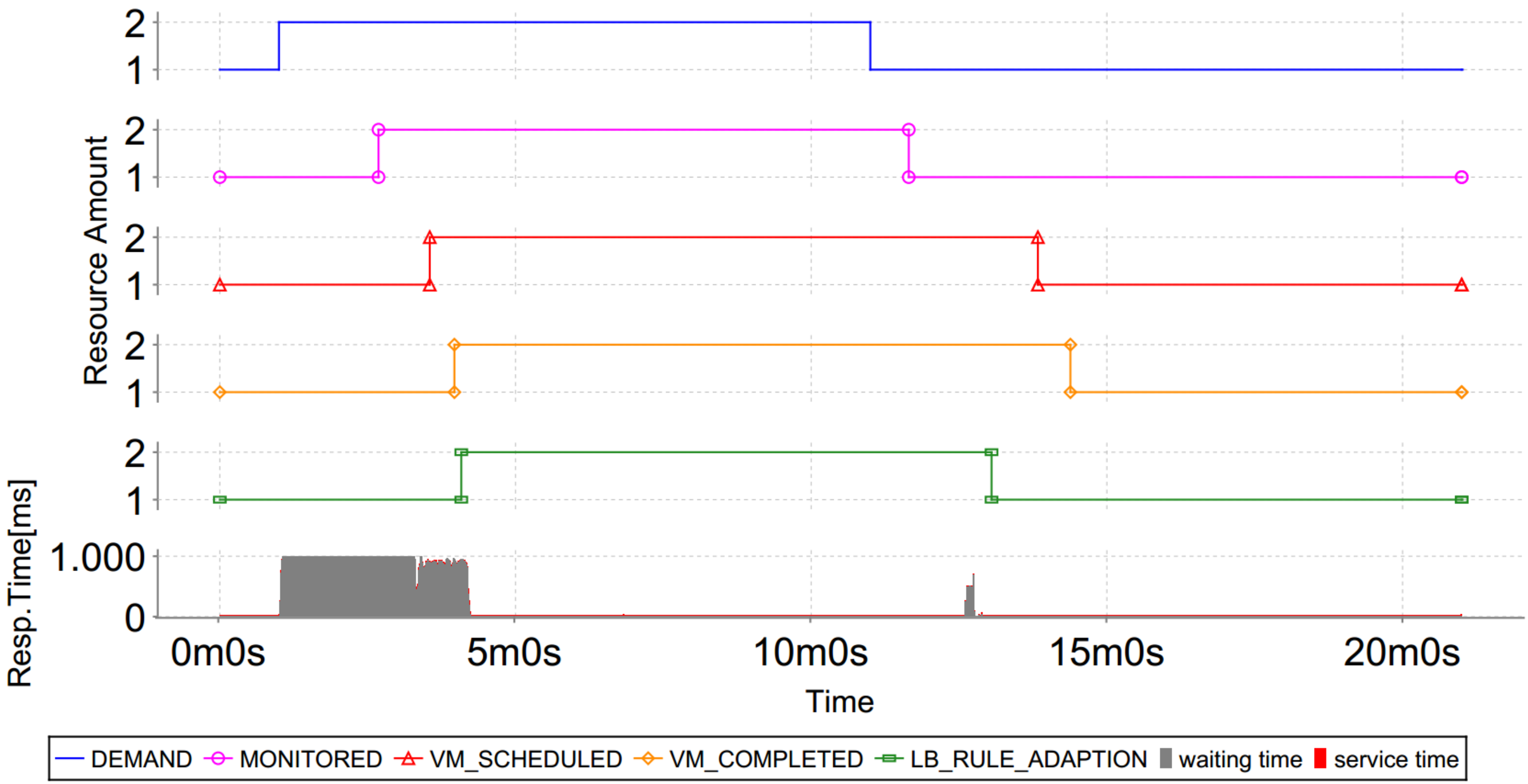
A 1Core Baseline
quietTime
240s
condTrueDur
120s
threshUp
90%
threshDown
10%

Configuration	accuarcy$_O$ [res. units]	accuarcy$_U$ [res. units]	timeshare$_O$ [%]	timeshare$_U$ [%]	jitter [adap/min.]	elastic speedup	violations [%]
A – 1Core Baseline	2.425	0.264	60.1	11.7	-0.067	1.000	20.3

Implementation – Activity Diagram



CloudStack Supply Events



Elasticity Definition

[Herbst13]

Elasticity

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

Definitions

ODCA, Compute Infrastructure-as-a-Service:

*"[...] defines elasticity as the configurability and expandability of the solution[...] Centrally, it is the ability to **scale up** and **scale down** capacity **based on subscriber workload**."* [OCDA12]

NIST Definition of Cloud Computing

*"**Rapid** elasticity: Capabilities can be elastically **provisioned and released**, in **some cases automatically**, to scale rapidly **outward** and **inward commensurate with demand**. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at anytime."* [Mell11]

IBM, Thoughts on Cloud, Edwin Schouten:

*"Elasticity is basically a 'rename' of scalability [...]" and "**removes any manual labor** needed to **increase or reduce** capacity."* [Shouten12]

Rich Wolski, CTO, Eucalyptus:

*"Elasticity **measures** the ability of the cloud to map a single user request to different resources."* [Wolski11]

Reuven Cohen:

*Elasticity is "the **quantifiable** ability to manage, measure, predict and adaptive responsiveness of an application **based on real time demands** placed on an infrastructure using a combination of local and remote computing resources."* [Cohen09]

Prerequisites

- Autonomic Scaling
  - Ensures repeatability

- Comparability with respect to
  - Resource Types (cpu, memory, vm)
  - Resource Scaling Units (cpu cycles, processors, vm)
  - Scaling Method (up/down, in/out)
  - Scalability Bounds (max. amount of resources)

Different scaling ranges:

- 4 Providers:
  - Provider A: 5 vms
  - Provider B: 7 vms
  - Provider C: 10 vms
  - Provider D: 15 vms

- Compare within a range that is supported by all providers
  - Option 1: Benchmark only first 5 resources
  - Option 2: Build groups (A,B: 5 C,D:10)