An Approach and Case Study of Cloud Instance Type Selection for Multi-Tier Web Applications

Christian Davatz, Christian Inzinger, Joel Scheuner, Philipp Leitner

University of Zurich, Switzerland

Universität Zürich[UZH]

s.e.a.l. software evolution & architecture lab

Selecting IaaS instance types is hard!

amazon web services™ EC2

Instance Types Matrix

Instance Type	vCPU	Memory (GiB)	Storage (GB)	Networking Performance	Physical Processor	Clock Speed (GHz)	Intel® AES-NI	Intel® AVX†	Intel® Turbo	EBS OPT	Enhanced Networking
t2.micro	1	1	EBS Only	Low to Moderate	Intel Xeon family	2.5	Yes	Yes	Yes	-	-
t2.small	1	2	EBS Only	Low to Moderate	Intel Xeon family	2.5	Yes	Yes	Yes	-	-
t2.medium	2	4	EBS Only	Low to Moderate	Intel Xeon family	2.5	Yes	Yes	Yes	-	-
m3.medium	1	3.75	1 x 4 SSD	Moderate	Intel Xeon E5-2670 v2*	2.5	Yes	Yes	Yes	-	-
m3.large	2	7.5	1 x 32 SSD	Moderate	Intel Xeon E5-2670 v2*	2.5	Yes	Yes	Yes	-	-
m3.xlarge	4	15	2 x 40 SSD	High	Intel Xeon E5-2670 v2*	2.5	Yes	Yes	Yes	Yes	-
m3.2xlarge	8	30	2 x 80 SSD	High	Intel Xeon E5-2670 v2*	2.5	Yes	Yes	Yes	Yes	-

Common Questions

What cloud provider should I choose?

Should I go for many small or few large instances?

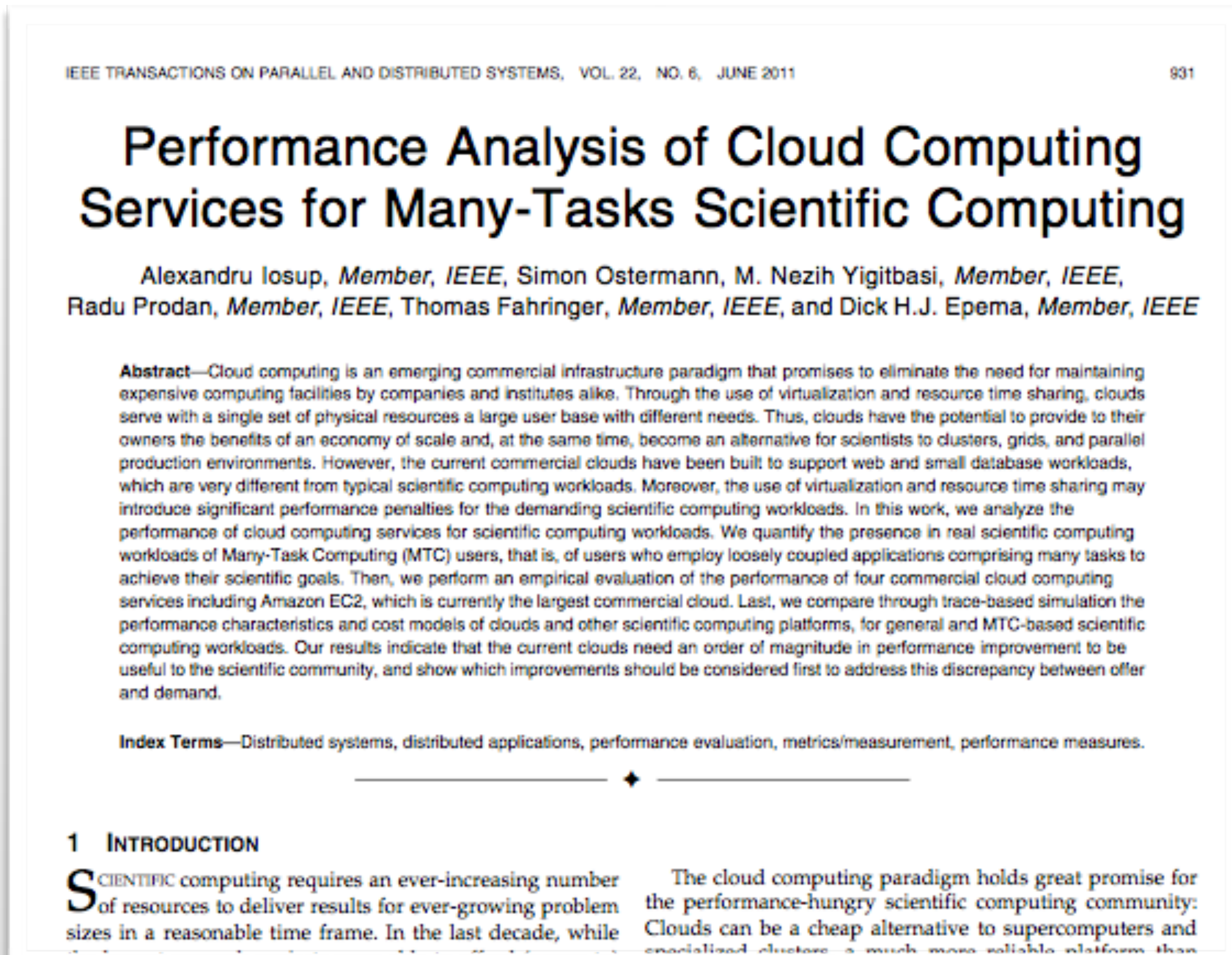
*General-purpose or *-optimized?*

Pay for better IOPS or not?

……………

➡ **Need for Benchmarking**

Existing Benchmarking Work



Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing

Alexandru Iosup, *Member, IEEE*, Simon Ostermann, M. Nezih Yigitbasi, *Member, IEEE*, Radu Prodan, *Member, IEEE*, Thomas Fahringer, *Member, IEEE*, and Dick H.J. Epema, *Member, IEEE*

Abstract—Cloud computing is an emerging commercial infrastructure paradigm that promises to eliminate the need for maintaining expensive computing facilities by companies and institutes alike. Through the use of virtualization and resource time sharing, clouds serve with a single set of physical resources a large user base with different needs. Thus, clouds have the potential to provide to their owners the benefits of an economy of scale and, at the same time, become an alternative for scientists to clusters, grids, and parallel production environments. However, the current commercial clouds have been built to support web and small database workloads, which are very different from typical scientific computing workloads. Moreover, the use of virtualization and resource time sharing may introduce significant performance penalties for the demanding scientific computing workloads. In this work, we analyze the performance of cloud computing services for scientific computing workloads. We quantify the presence in real scientific computing workloads of Many-Task Computing (MTC) users, that is, of users who employ loosely coupled applications comprising many tasks to achieve their scientific goals. Then, we perform an empirical evaluation of the performance of four commercial cloud computing services including Amazon EC2, which is currently the largest commercial cloud. Last, we compare through trace-based simulation the performance characteristics and cost models of clouds and other scientific computing platforms, for general and MTC-based scientific computing workloads. Our results indicate that the current clouds need an order of magnitude in performance improvement to be useful to the scientific community, and show which improvements should be considered first to address this discrepancy between offer and demand.

Index Terms—Distributed systems, distributed applications, performance evaluation, metrics/measurement, performance measures.

1 INTRODUCTION

SCIENTIFIC computing requires an ever-increasing number of resources to deliver results for ever-growing problem sizes in a reasonable time frame. In the last decade, while

The cloud computing paradigm holds great promise for the performance-hungry scientific computing community: Clouds can be a cheap alternative to supercomputers and

Existing Benchmarking Work

IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 22, NO. 6, JUNE 2011 931

Performance Analysis of Cloud Computing Services for Mar

Alexandru Iosup, *Member, IE*
Radu Prodan, *Member, IEEE*, Thoma

Abstract—Cloud computing is an emerging
expensive computing facilities by companies
serve with a single set of physical resources
owners the benefits of an economy of scale a
production environments. However, the curre
which are very different from typical scientific
introduce significant performance penalties fo
performance of cloud computing services for
workloads of Many-Task Computing (MTC) us
achieve their scientific goals. Then, we perfo
services including Amazon EC2, which is curr
performance characteristics and cost models
computing workloads. Our results indicate th
useful to the scientific community, and show w
and demand.

Index Terms—Distributed systems, distribute

1 Introduction

Scientific computing requires an ever-inc
of resources to deliver results for ever-g
sizes in a reasonable time frame. In the las

Patterns in the Chaos—A Study of Performance Variation and Predictability in Public IaaS Clouds

PHILIPP LEITNER and JÜRGEN CITO, Department of Informatics, University of Zurich

Benchmarking the performance of public cloud providers is a common research topic. Previous work has already extensively evaluated the performance of different cloud platforms for different use cases, and under different constraints and experiment setups. In this article, we present a principled, large-scale literature review to collect and codify existing research regarding the predictability of performance in public Infrastructure-as-a-Service (IaaS) clouds. We formulate 15 hypotheses relating to the nature of performance variations in IaaS systems, to the factors of influence of performance variations, and how to compare different instance types. In a second step, we conduct extensive real-life experimentation on four cloud providers to empirically validate those hypotheses. We show that there are substantial differences between providers. Hardware heterogeneity is today less prevalent than reported in earlier research, while multitenancy has a dramatic impact on performance and predictability, but only for some cloud providers. We were unable to discover a clear impact of the time of the day or the day of the week on cloud performance.

Categories and Subject Descriptors: H.3.4 [**Systems and Software**]: Distributed Systems

General Terms: Experimentation, Measurement, Performance

Additional Key Words and Phrases: Infrastructure-as-a-service, public cloud, benchmarking

ACM Reference Format:
Philipp Leitner and Jürgen Cito. 2016. Patterns in the chaos—A study of performance variation and predictability in public IaaS clouds. ACM Trans. Internet Technol. 16, 3, Article 15 (April 2016), 23 pages.
DOI: http://dx.doi.org/10.1145/2885497

1. INTRODUCTION

In an Infrastructure-as-a-Service (IaaS) cloud [Armbrust et al. 2010], computing resources are acquired and released as a service, typically in the form of virtual machines with attached virtual disks [Buyya et al. 2009]. Cloud benchmarking, that is, the pro-

Existing Benchmarking Work

15

IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 22, NO. 6, JUNE 2011 931

Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing

Alexandru Iosup, *Member, IEEE*, Simon Ostermann, M. Nezih
Radu Prodan, *Member, IEEE*, Thomas Fahringer, *Member, IEEE*, and

Abstract—Cloud computing is an emerging commercial
expensive computing facilities by companies and institute
serve with a single set of physical resources a large user base with different needs. Thus, c
owners the benefits of an economy of scale and, at the same time, become an alternative fo
production environments. However, the current commercial clouds have been built to supp
which are very different from typical scientific computing workloads. Moreover, the use of vir
introduce significant performance penalties for the demanding scientific computing workloa
performance of cloud computing services for scientific computing workloads. We quantify t
workloads of Many-Task Computing (MTC) users, that is,
achieve their scientific goals. Then, we perform an empirical
services including Amazon EC2, which is currently the larg
performance characteristics and cost models of clouds and
computing workloads. Our results indicate that the current clouds need an order of magnit
useful to the scientific community, and show which improvements should be considered first
and demand.

Index Terms—Distributed systems, distributed applications, performance evaluation, metric

1 Introduction

Scientific computing requires an ever-increasing number
of resources to deliver results for ever-growing pro
sizes in a reasonable time frame. In the last decade,

Patterns in the Cha
and Predictability i

PHILIPP LEITNER and J

Benchmarking the performan
already extensively evaluate
under different constraints a
literature review to collect an
Infrastructure-as-a-Service (I
variations in IaaS systems, to
instance types. In a second st
empirically validate those hy
Hardware heterogeneity is to
a dramatic impact on perform
discover a clear impact of the

Categories and Subject Descr

General Terms: Experimenta

Additional Key Words and Ph

ACM Reference Format:
Philipp Leitner and Jürgen C
dictability in public IaaS clou
DOI: http://dx.doi.org/10.1145

1. INTRODUCTION

In an Infrastructure-as
sources are acquired an
with attached virtual di

Benchmarking Cloud Serving Systems with YCSB

Brian F. Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, Russell Sears

Yahoo! Research
Santa Clara, CA, USA
{cooperb,silberst,etam,ramakris,sears}@yahoo-inc.com

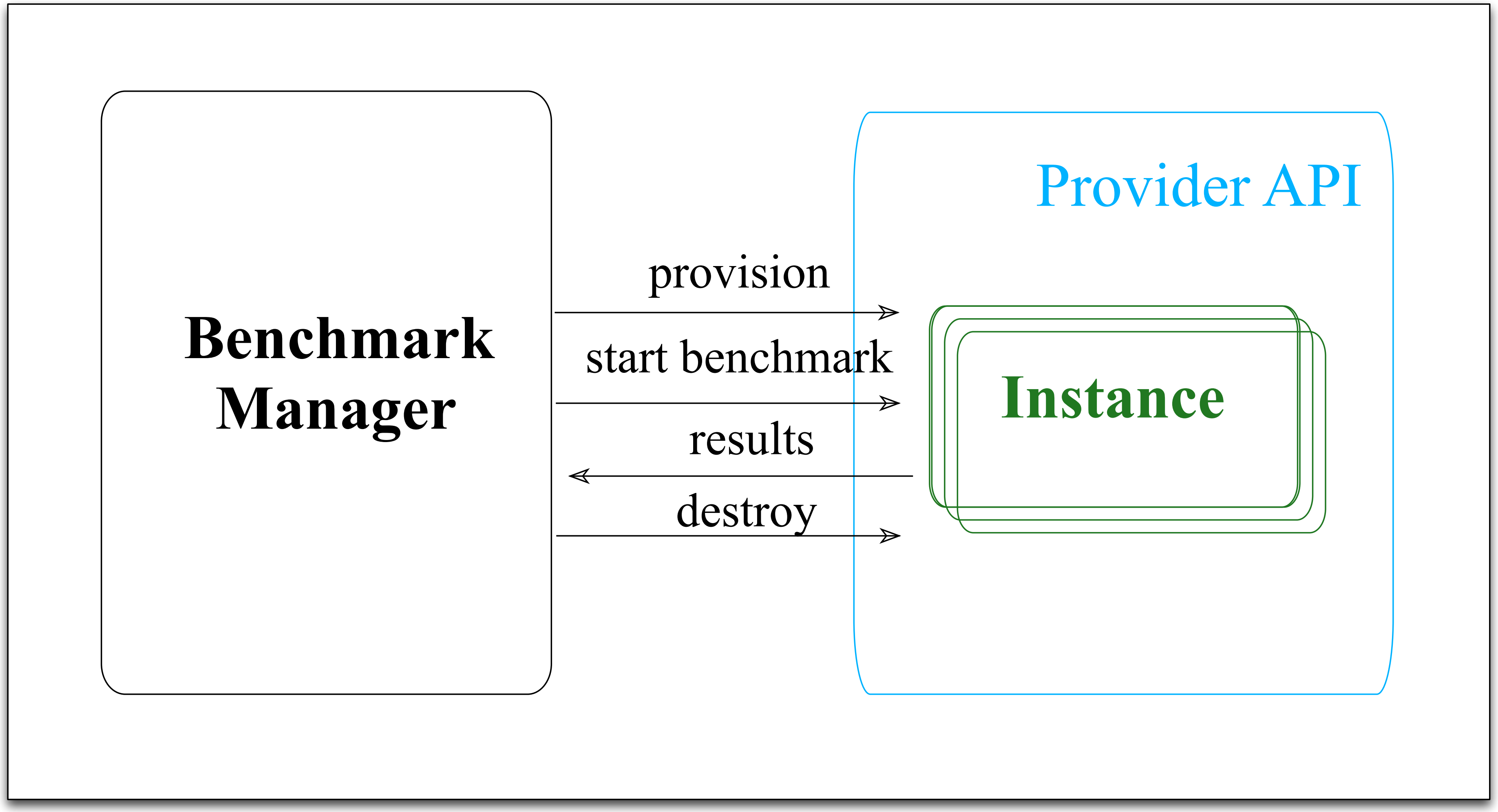
ABSTRACT

While the use of MapReduce systems (such as Hadoop) for large scale data analysis has been widely recognized and studied, we have recently seen an explosion in the number of systems developed for cloud data serving. These newer systems address "cloud OLTP" applications, though they typically do not support ACID transactions. Examples of systems proposed for cloud serving use include BigTable, PNUTS, Cassandra, HBase, Azure, CouchDB, SimpleDB, Voldemort, and many others. Further, they are being applied to a diverse range of applications that differ considerably from traditional (e.g., TPC-C like) serving workloads. The number of emerging cloud serving systems and the wide range of proposed applications, coupled with a lack of apples-to-apples performance comparisons, makes it difficult to understand the tradeoffs between systems and the workloads for which they are suited. We present the *Yahoo! Cloud Serving Benchmark* (YCSB) framework, with the goal of facilitating performance comparisons of the new generation of cloud data serving systems. We define a core set of benchmarks and report results for four widely used systems: Cassandra, HBase, Yahoo!'s PNUTS, and a simple sharded

ers [3, 5, 7, 8]. Some systems are offered only as cloud services, either directly in the case of Amazon SimpleDB [1] and Microsoft Azure SQL Services [11], or as part of a programming environment like Google's AppEngine [6] or Yahoo!'s YQL [13]. Still other systems are used only within a particular company, such as Yahoo!'s PNUTS [17], Google's BigTable [16], and Amazon's Dynamo [18]. Many of these "cloud" systems are also referred to as "key-value stores" or "NoSQL systems," but regardless of the moniker, they share the goals of massive scaling "on demand" (elasticity) and simplified application development and deployment.

The large variety has made it difficult for developers to choose the appropriate system. The most obvious differences are between the various data models, such as the column-group oriented BigTable model used in Cassandra and HBase versus the simple hashtable model of Voldemort or the document model of CouchDB. However, the data models can be documented and compared qualitatively. Comparing the performance of various systems is a harder problem. Some systems have made the decision to optimize for writes by using on-disk structures that can be maintained using sequential I/O (as in the case of Cassandra and HBase),

Basic Approach to Benchmarking Clouds

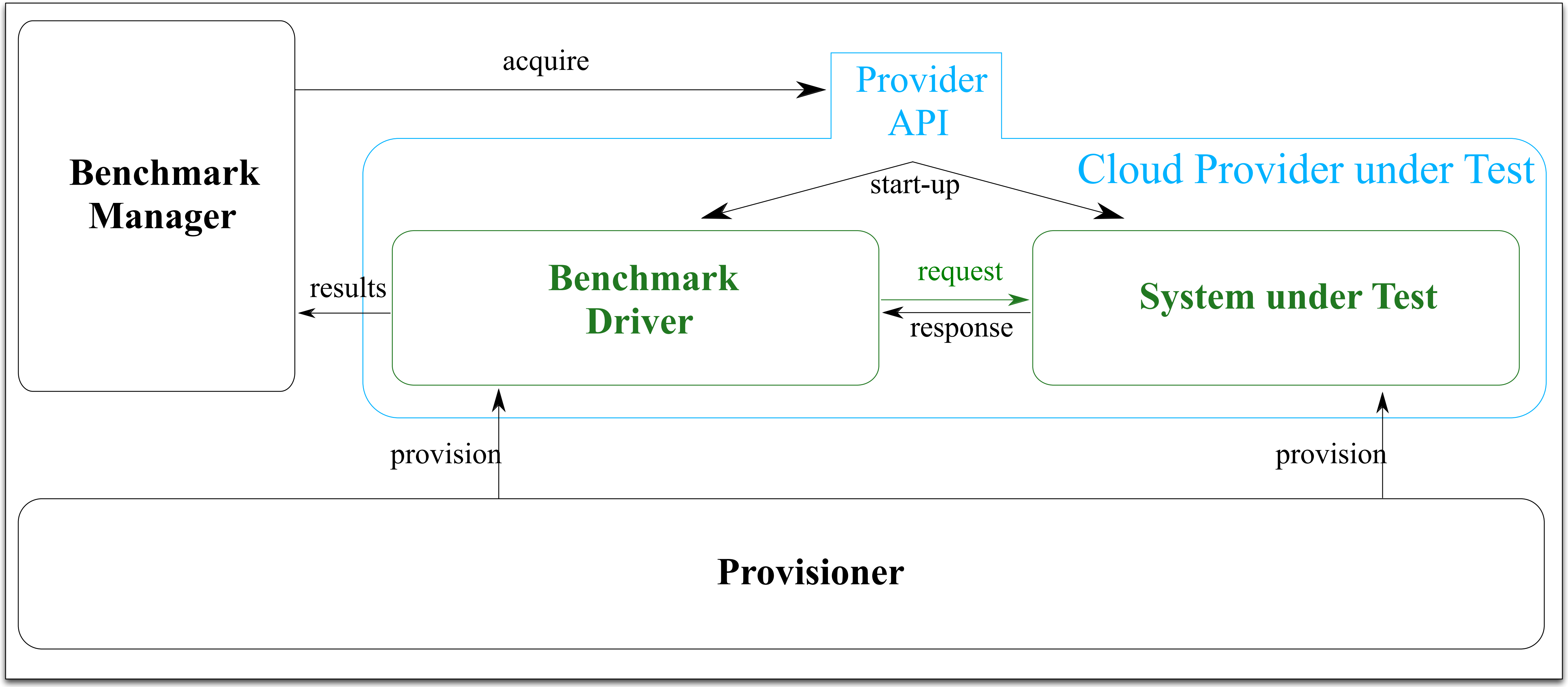


Used for instance in:

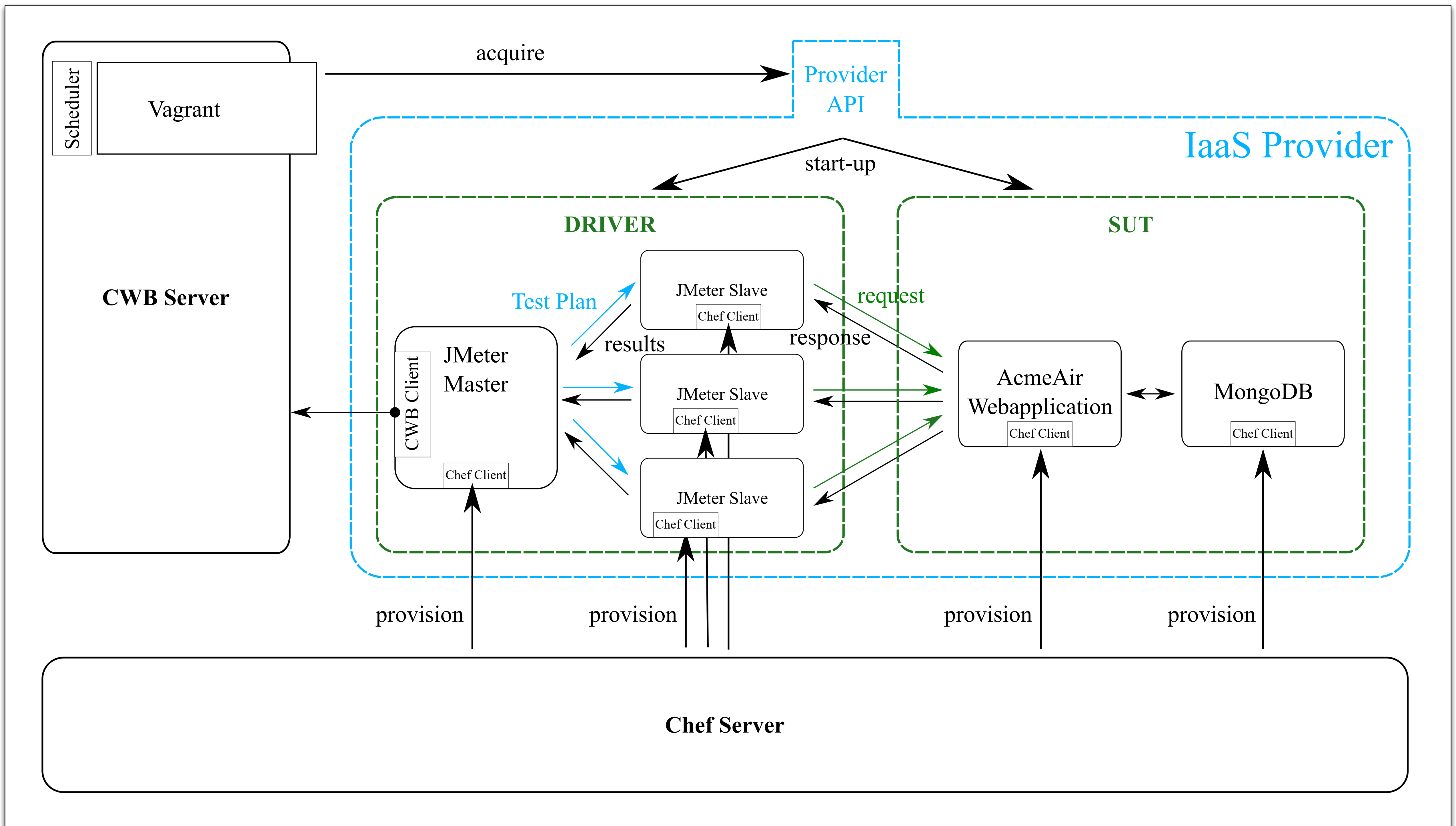
Philipp Leitner and Jürgen Cito. 2016. **Patterns in the Chaos — A Study of Performance Variation and Predictability in Public IaaS Clouds**. ACM Trans. Internet Technol. 16, 3, Article 15 (April 2016), 23 pages. DOI: http://dx.doi.org/10.1145/2885497

Joel Scheuner, Jürgen Cito, Philipp Leitner, Harald C. Gall (2015). **Cloud WorkBench: Benchmarking IaaS Providers Based on Infrastructure-as-Code**. In Proceedings of the 24th International Conference on World Wide Web, pp. 239–242, New York, NY, USA.

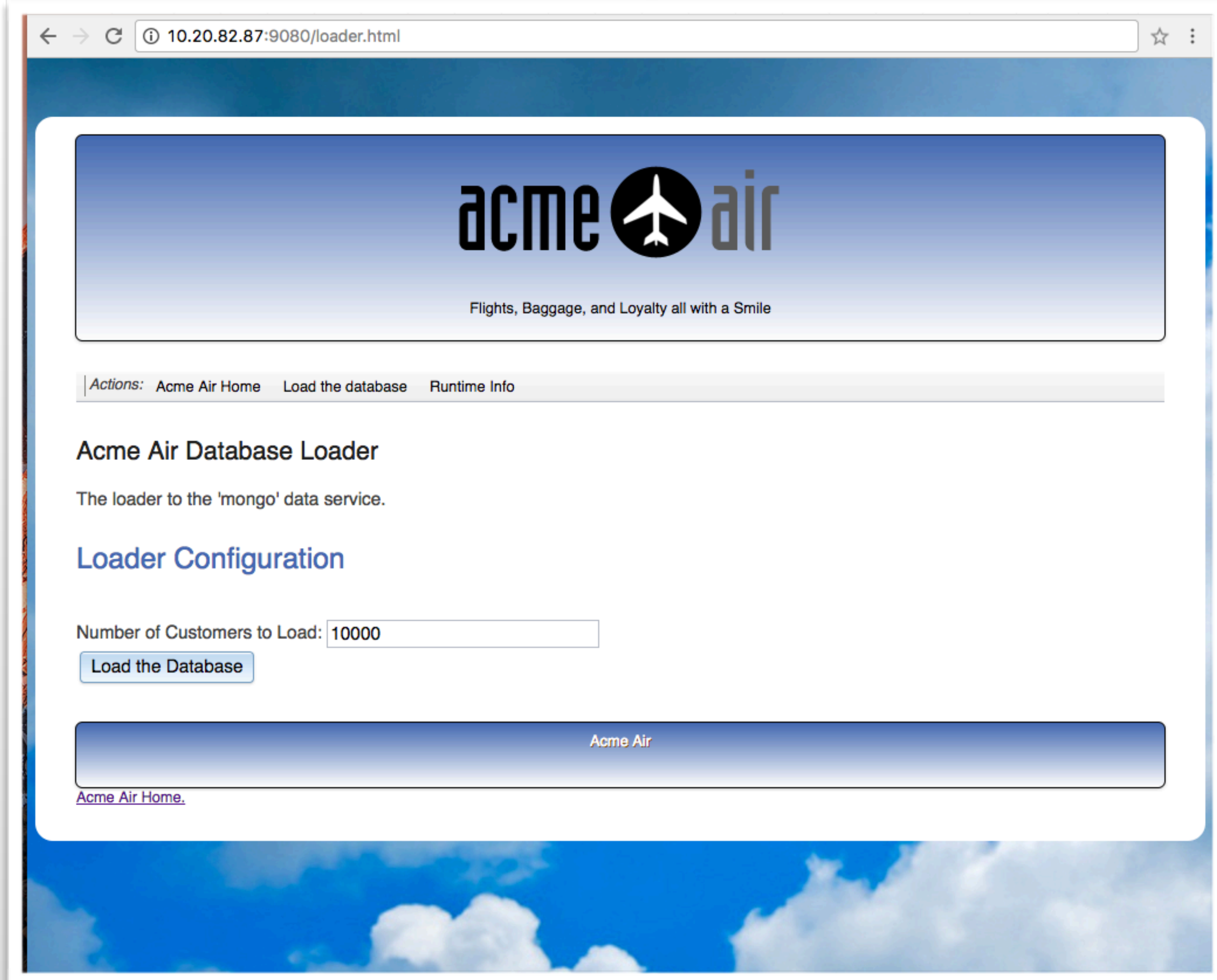
For Multi-Tier (Application) Benchmarks



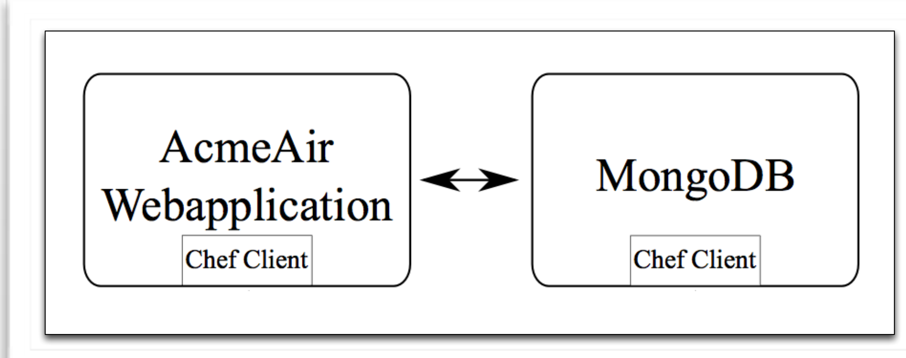
A Concrete Instantiation



AcmeAir

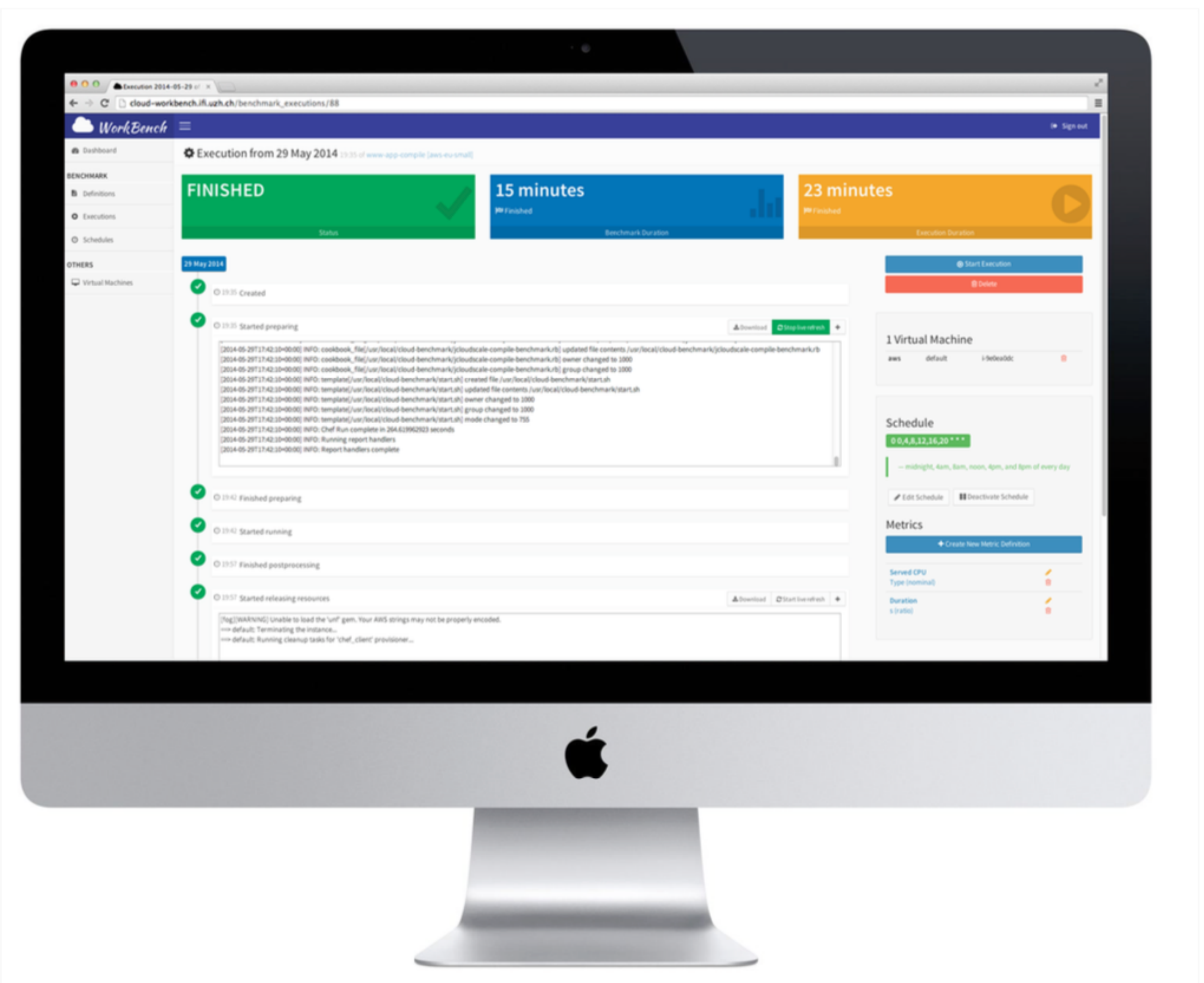


Two-Tier App



OSS "Microservice"-Based App: https://github.com/acmeair/acmeair

CBW

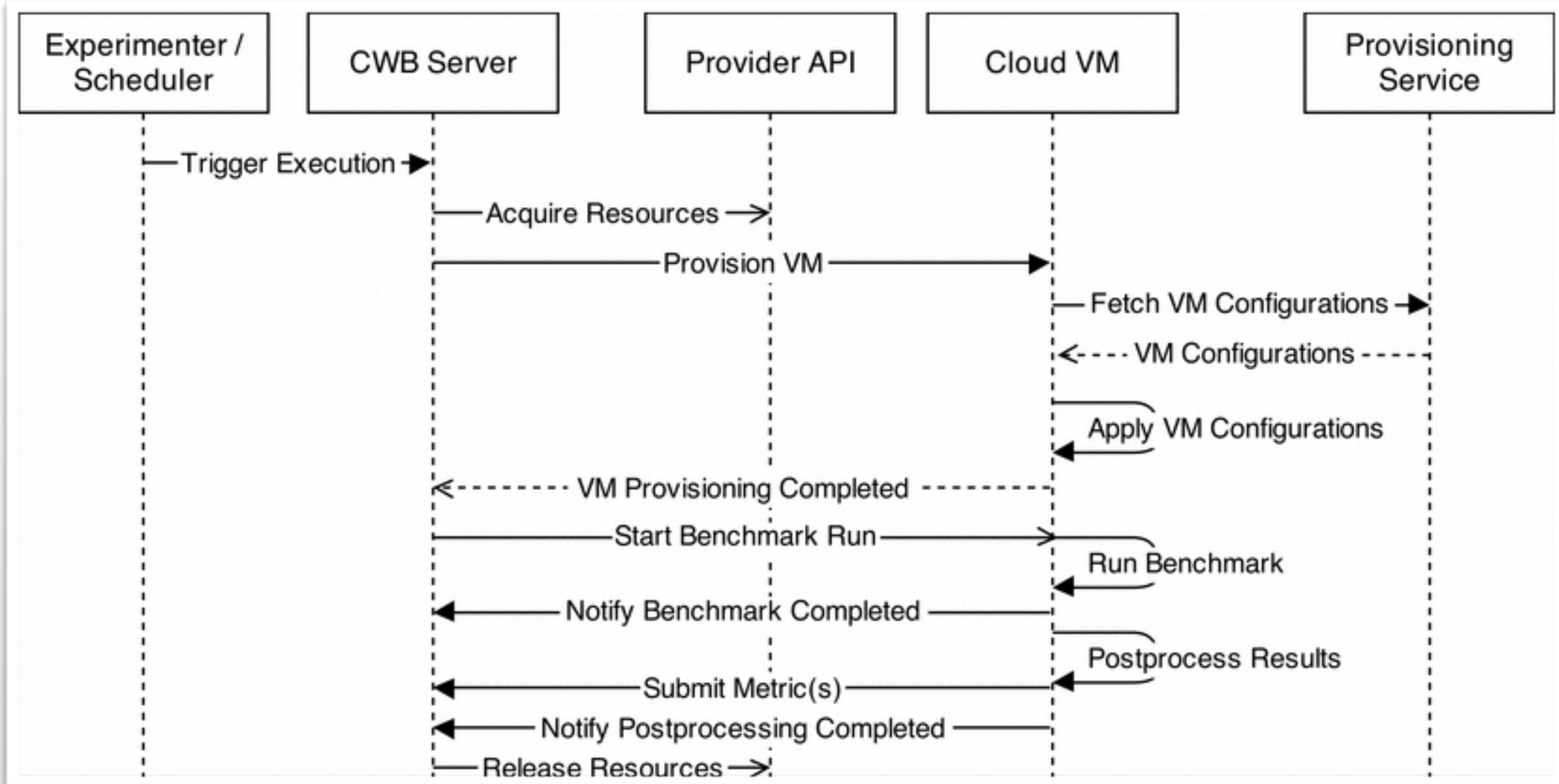


Code:
https://github.com/sealuzh/cloud-workbench

Demo:
https://www.youtube.com/watch?v=0yGFGvHvobk

J. Scheuner, P. Leitner, J. Cito and H.C. Gall: **Cloud Work Bench - Infrastructure-as-Code Based Cloud Benchmarking** 2014 IEEE 6th International Conference on Cloud Computing Technology and Science, Singapore, 2014, pp. 246-253. doi: 10.1109/CloudCom.2014.98

CBW



J. Scheuner, P. Leitner, J. Cito and H.C. Gall: **Cloud Work Bench - Infrastructure-as-Code Based Cloud Benchmarking** 2014 IEEE 6th International Conference on Cloud Computing Technology and Science, Singapore, 2014, pp. 246-253. doi: 10.1109/CloudCom.2014.98

Research Questions

RQ1: *What sustained performance, measured in throughput of successful requests per second, can we achieve with each configuration?*

RQ2: *Can we observe statistically significantly different performance for each configuration?*

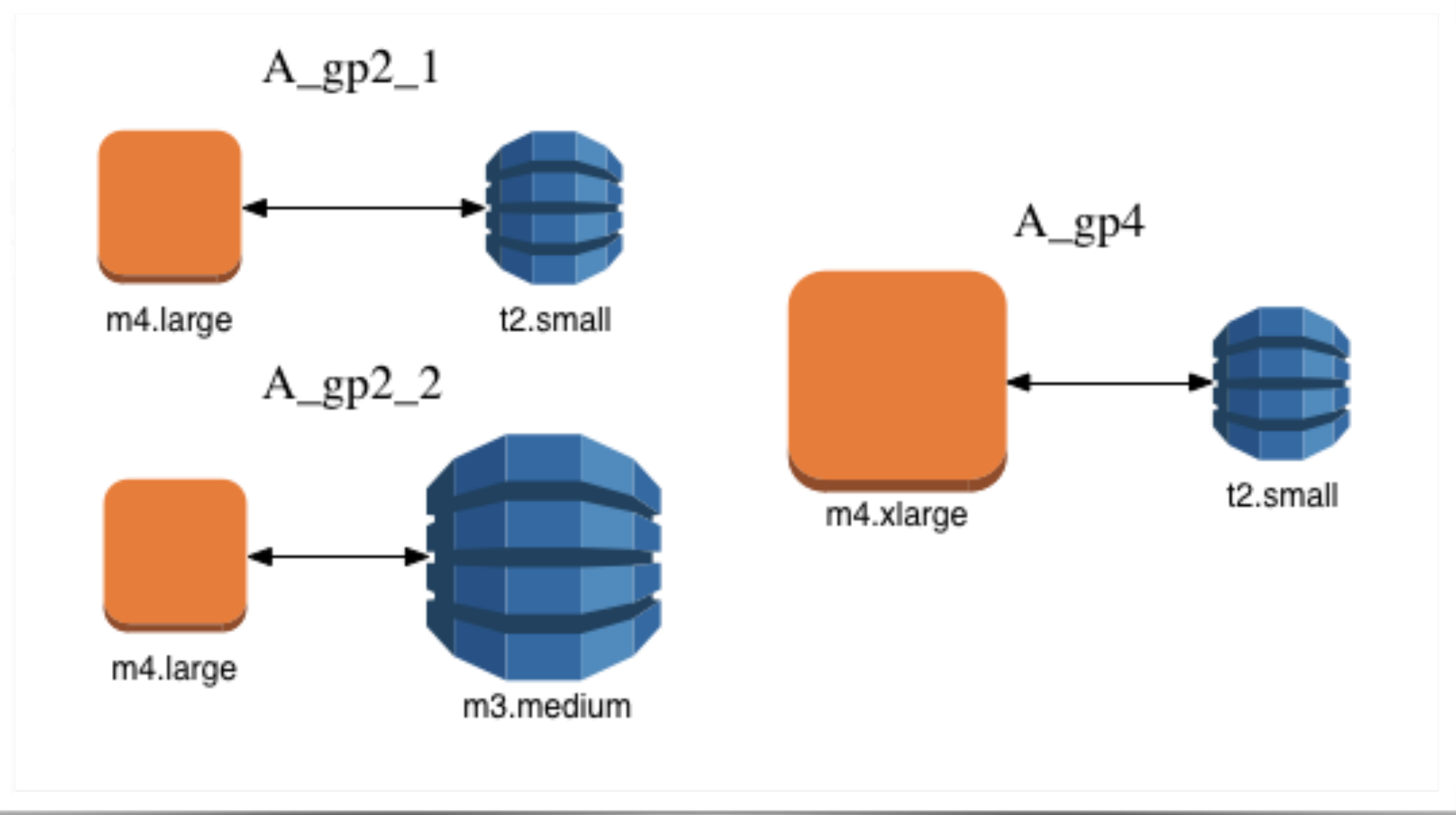
RQ3: *Which configuration is the most cost-effective way to host AcmeAir for the defined workload?*

Used Configs

Configuration $c \in C$	Webapp	DB	Costs m_c	# of Runs $\lvert c_r \rvert$
EC2				
A_gp2_1	m4.large	t2.small	$0.173	37
A_gp2_2	m4.large	m3.medium	$0.222	27
A_gp4	m4.xlarge	t2.small	$0.315	23
A_co2_1	c4.large	t2.small	$0.164	35
A_co2_2	c4.large	m3.medium	$0.213	26
A_co4	c4.xlarge	t2.small	$0.297	19
GCE				
G_gp1	n1-standard-1	n1-standard-1	$0.110	26
G_gp2	n1-standard-2	n1-standard-1	$0.165	26
G_gp4	n1-standard-4	n1-standard-1	$0.270	24
G_co2	n1-highcpu-2	n1-highcpu-2	$0.168	18
G_co4	n1-highcpu-4	n1-standard-1	$0.223	23

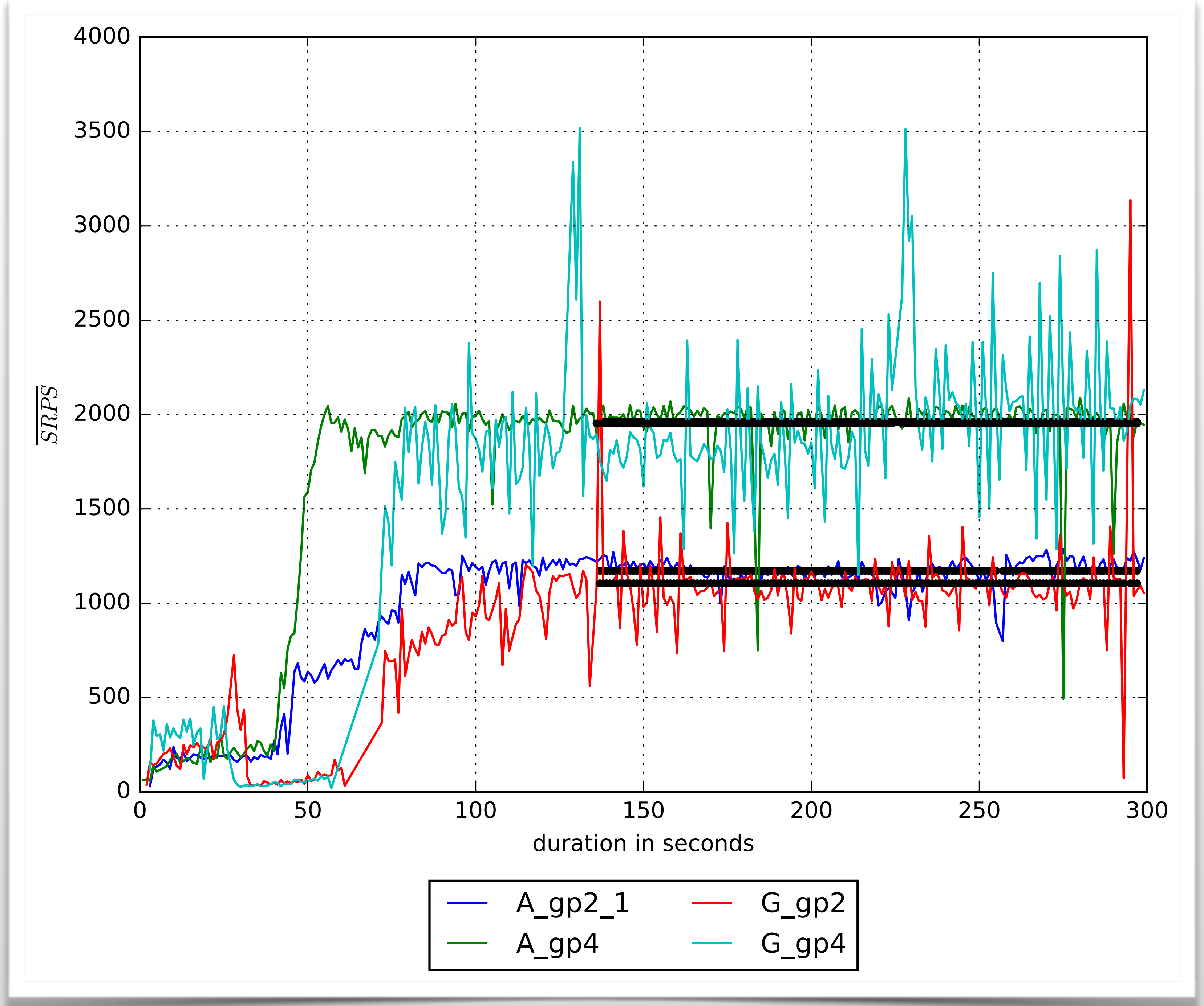
Used Configs

Configuration $c \in C$	Webapp	DB	Costs m_c	# of Runs $\lvert c_r \rvert$
EC2				
A_gp2_1	m4.large	t2.small	$0.173	37
A_gp2_2	m4.large	m3.medium	$0.222	27
A_gp4	m4.xlarge	t2.small	$0.315	23
A_co2_1	c4.large	t2.small	$0.164	35
A_co2_2	c4.large	m3.medium	$0.213	26
A_co4	c4.xlarge	t2.small	$0.297	19
GCE				
G_gp1	n1-standard-1	n1-standard-1	$0.110	26
G_gp2	n1-standard-2	n1-standar		
G_gp4	n1-standard-4	n1-standar		
G_co2	n1-highcpu-2	n1-highcpu		
G_co4	n1-highcpu-4	n1-standar		



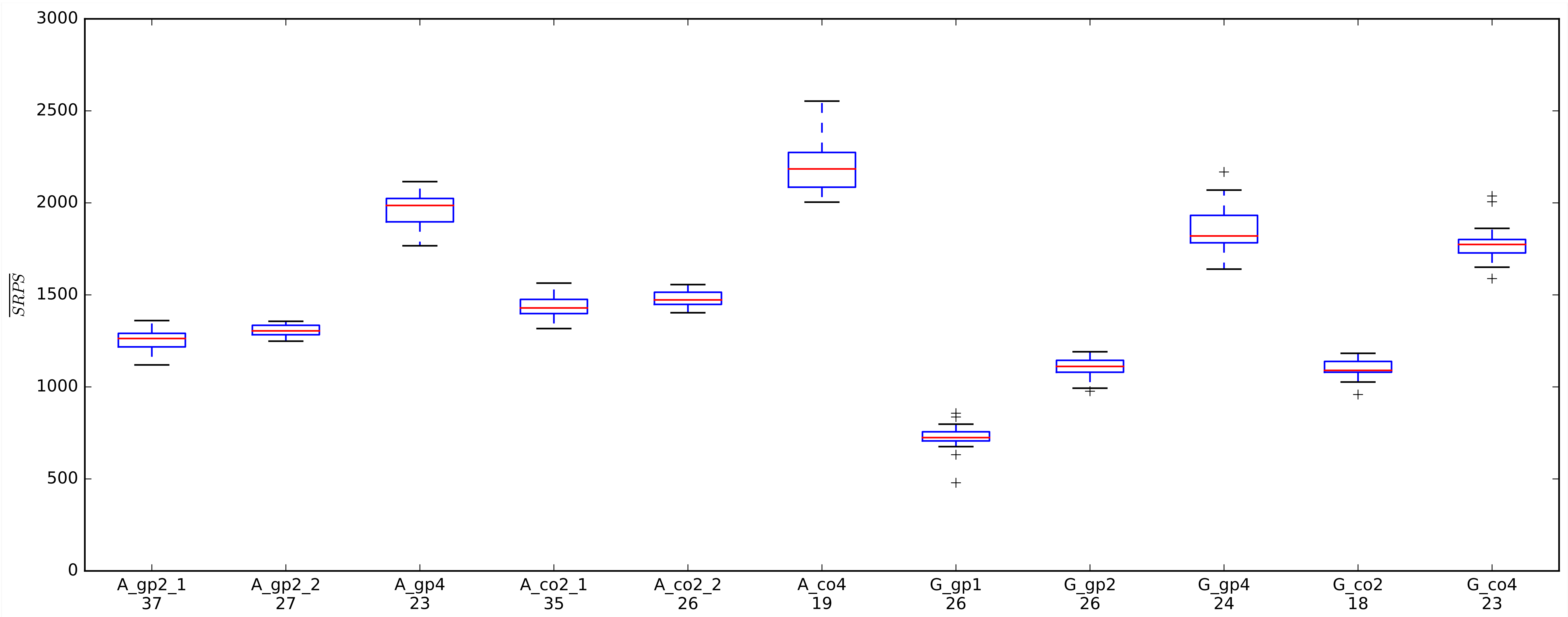
Used Metric

“Sustainable Throughput”



Results

RQ1 + RQ2



Results

RQ3

Metric: Mio. Requests per $

Configuration $c \in C$	Avg. Throughput $SRPS_c$	Costs m_c	Mio. Requests per $ pci_c	Rank
A_co2_1	1417.09	$0.164	31.107	1
G_co4	1791.98	$0.223	28.929	2
A_co4	2192.07	$0.297	26.571	3
A_gp2_1	1247.37	$0.173	25.957	4
G_gp4	1888.37	$0.270	25.178	5
A_co2_2	1472.01	$0.213	24.879	6
G_gp2	1102.49	$0.165	24.054	7
G_gp1	722.21	$0.110	23.636	8
G_co2	1095.28	$0.168	23.470	9
A_gp4	1939.74	$0.315	22.168	10
A_gp2_2	1302.83	$0.222	21.127	11

Lessons Learned

Importance of Benchmarking

Least cost-effective instance type only about 67% of perf / $ of best configuration

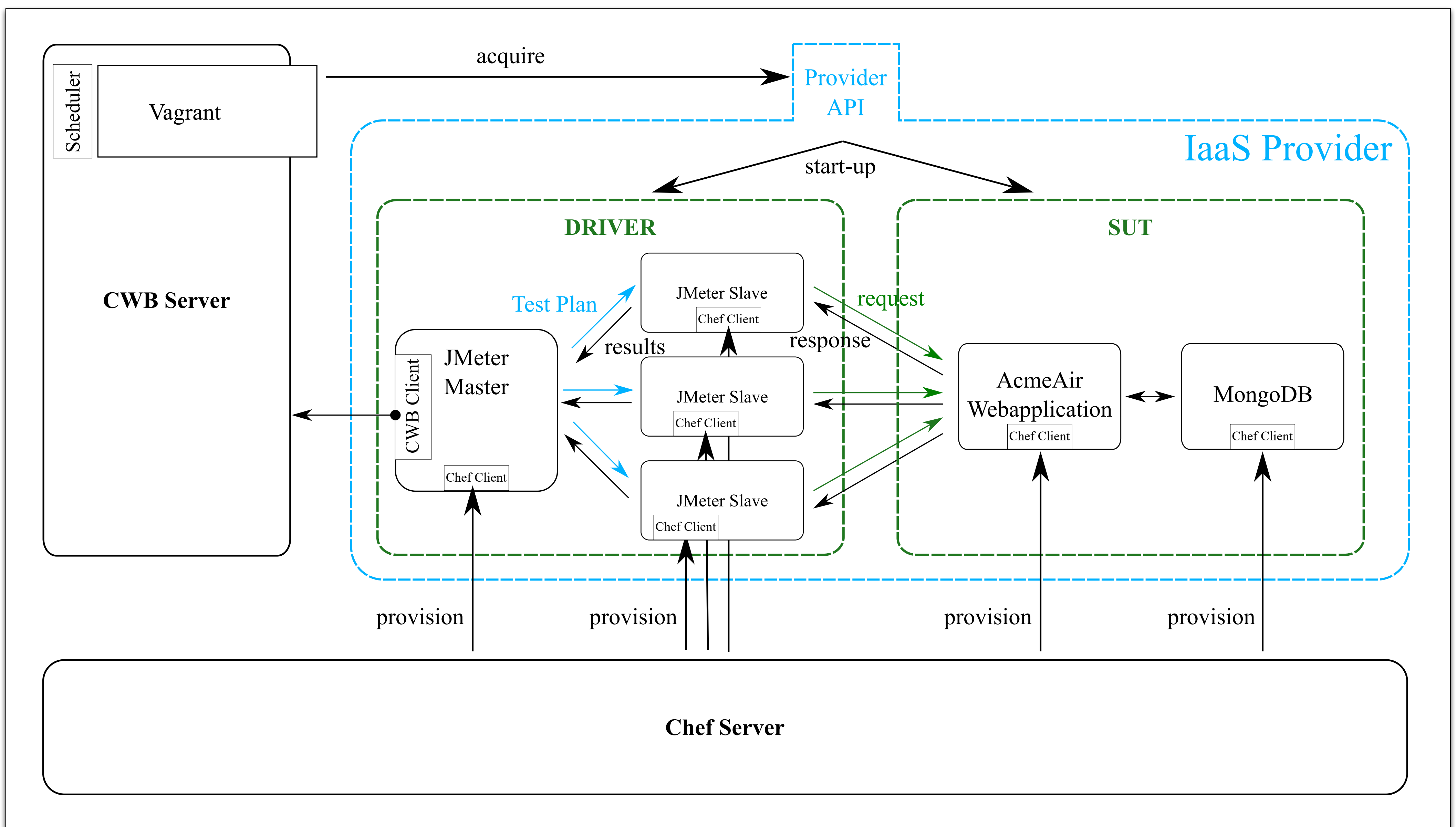
No clear "cheap" cloud provider

Comparable offerings from different providers are similarly cost-effective

No easy rules of thumb

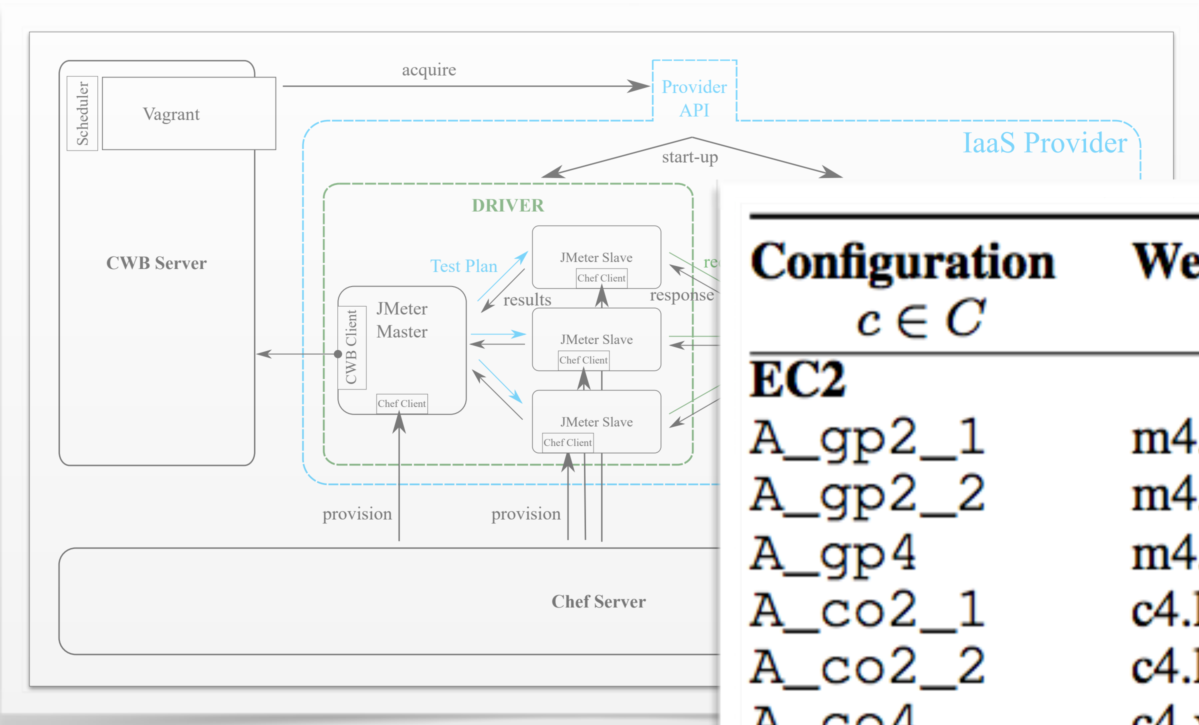
Compute-optimized instances may be better for our workload, but results vary

Summary



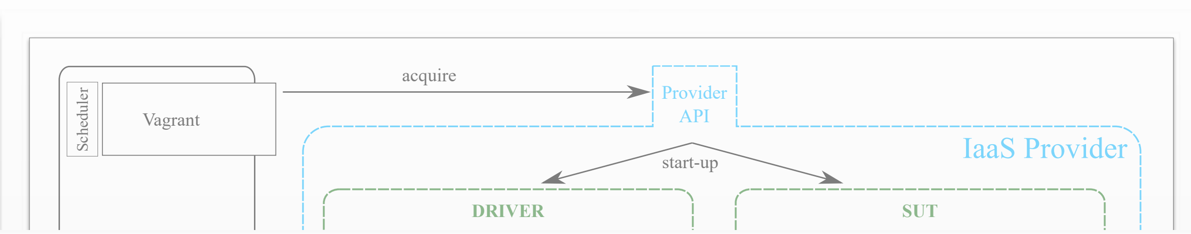
Scheduler
Vagrant
acquire
Provider API
IaaS Provider
start-up
CWB Server
DRIVER
SUT
Test Plan
results
JMeter Slave
Chef Client
request
response
JMeter Master
CWB Client
AcmeAir Webapplication
MongoDB
provision
Chef Server

Summary

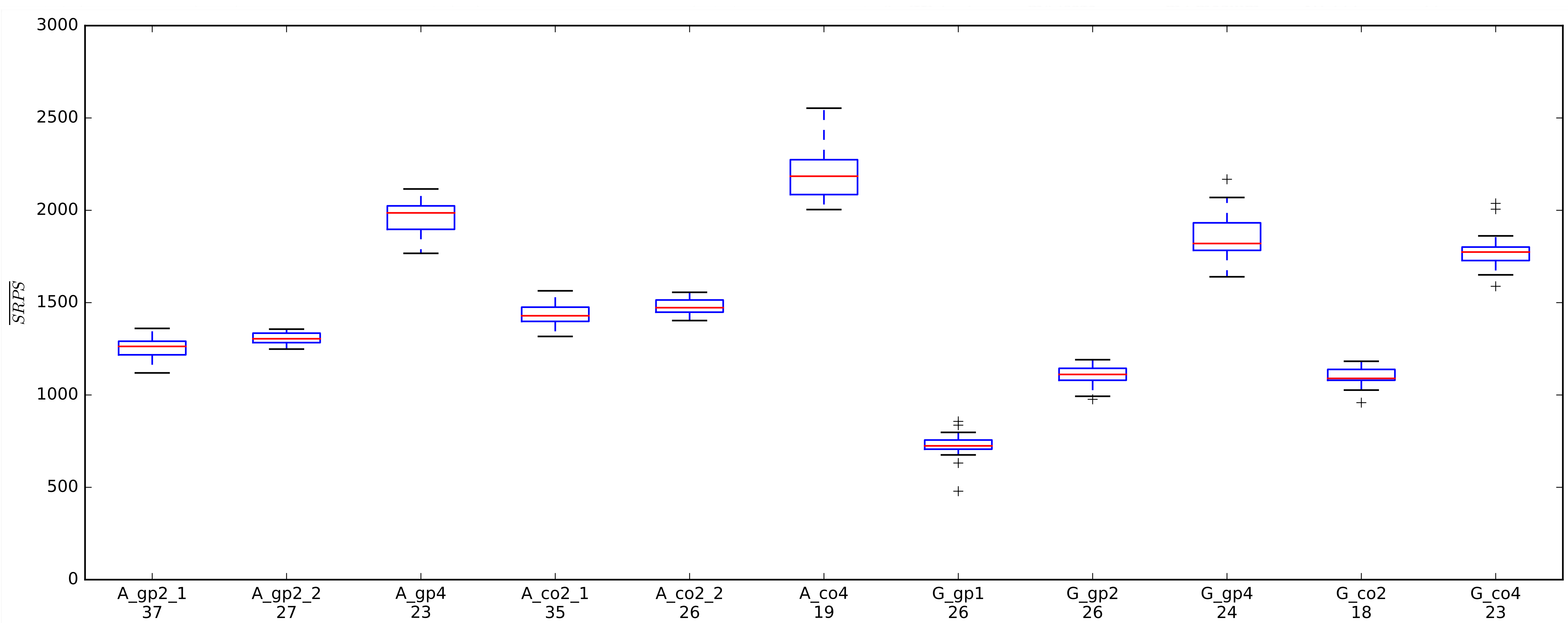


Configuration $c \in C$	Webapp	DB	Costs m_c	# of Runs $\lvert c_r \rvert$
EC2				
A_gp2_1	m4.large	t2.small	$0.173	37
A_gp2_2	m4.large	m3.medium	$0.222	27
A_gp4	m4.xlarge	t2.small	$0.315	23
A_co2_1	c4.large	t2.small	$0.164	35
A_co2_2	c4.large	m3.medium	$0.213	26
A_co4	c4.xlarge	t2.small	$0.297	19
GCE				
G_gp1	n1-standard-1	n1-standard-1	$0.110	26
G_gp2	n1-standard-2	n1-standard-1	$0.165	26
G_gp4	n1-standard-4	n1-standard-1	$0.270	24
G_co2	n1-highcpu-2	n1-highcpu-2	$0.168	18
G_co4	n1-highcpu-4	n1-standard-1	$0.223	23

Summary



Configuration $c \in C$	Webapp	DB	Costs m_c	# of Runs $\lvert c_r \rvert$
EC2				
A_gp2_1	m4.large	t2.small	$0.173	37



I'm Hiring!

Talk to me if you are interested in doing a PhD in Gothenburg, Sweden!

AVANCEZ

1829

CHALMERS

UNIVERSITY OF TECHNOLOGY