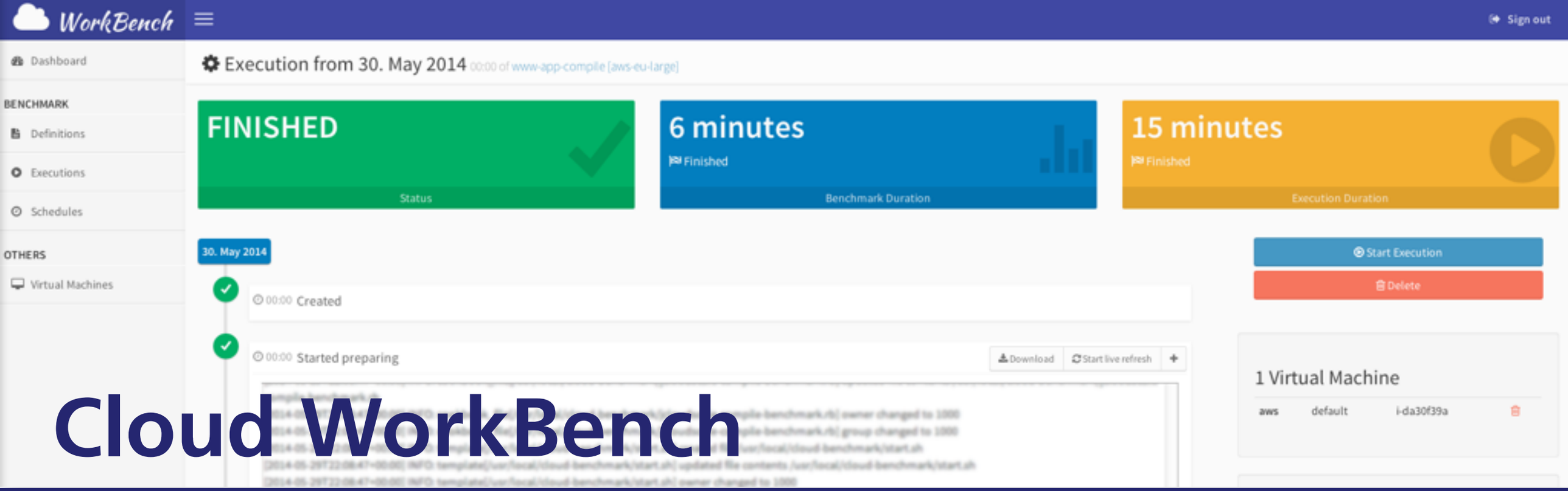


Cloud WorkBench

A Web-Based Framework for Benchmarking Cloud Services

Joel Scheuner

University of Zurich, Switzerland

Universität Zürich[UZH]

14.08.2014

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

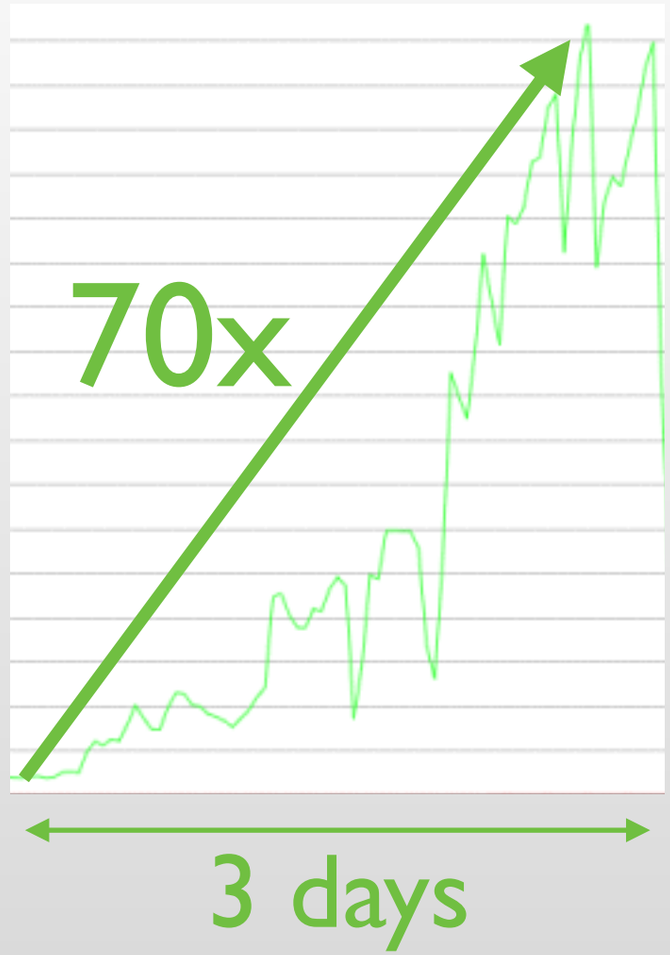
Cloud Computing - Essential Characteristics

1. On-demand Self-service


```
ec2-run-instances \\
  ami-896c96fe \\
  -t t1.micro \\
  -k my-key-pair
```

Cloud Computing - Essential Characteristics

1. On-demand Self-service
2. Rapid Elasticity



Cloud Computing - Essential Characteristics

1. On-demand Self-service
2. Rapid Elasticity
3. Utility-based Pricing

Cloud Computing - Essential Characteristics

1. On-demand Self-service
2. Rapid Elasticity
3. Utility-based Pricing
4. Resource Pooling

Cloud Computing - Essential Characteristics

1. On-demand Self-service
2. Rapid Elasticity
3. Utility-based Pricing
4. Resource Pooling
5. Broad Network Access

Infrastructure-as-a-Service (IaaS)

- Processing, storage, networks
- Virtual Machines (VMs)

>23 Instance Types

Google Compute Engine

Differences between IaaS Services

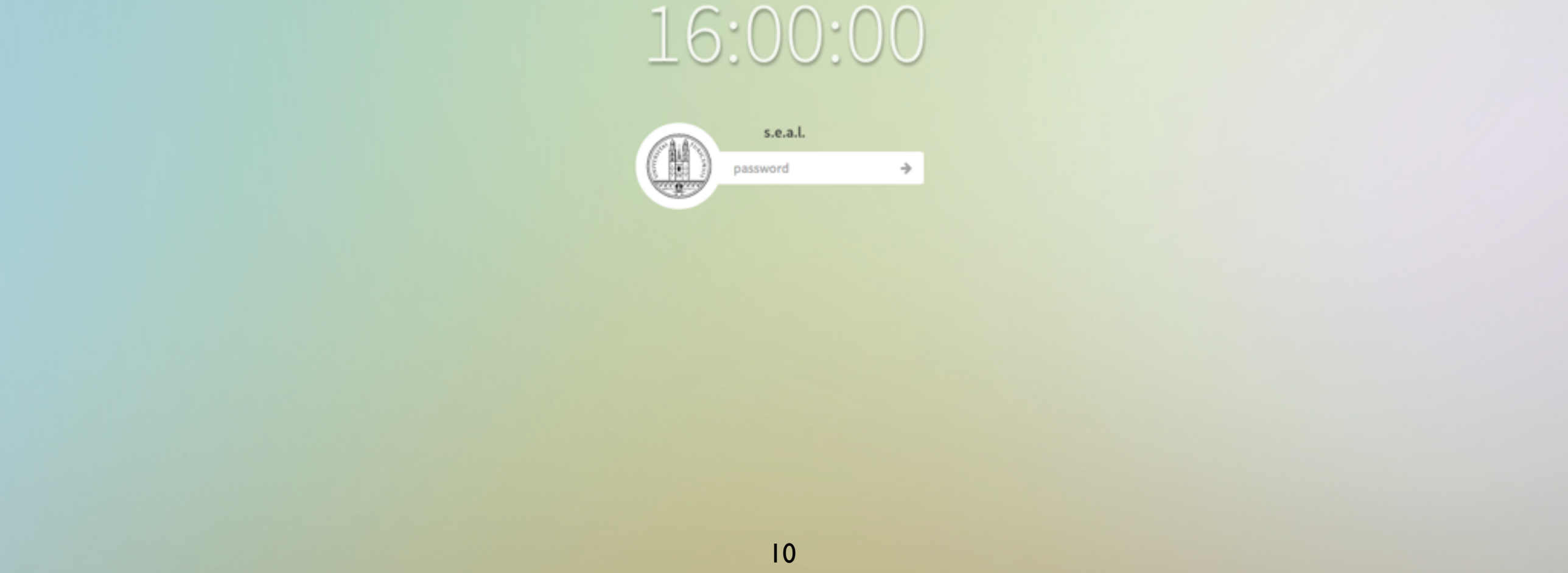
- Performance
- Hardware being served
- Reliability
- Costs

Even for services with the same specification!

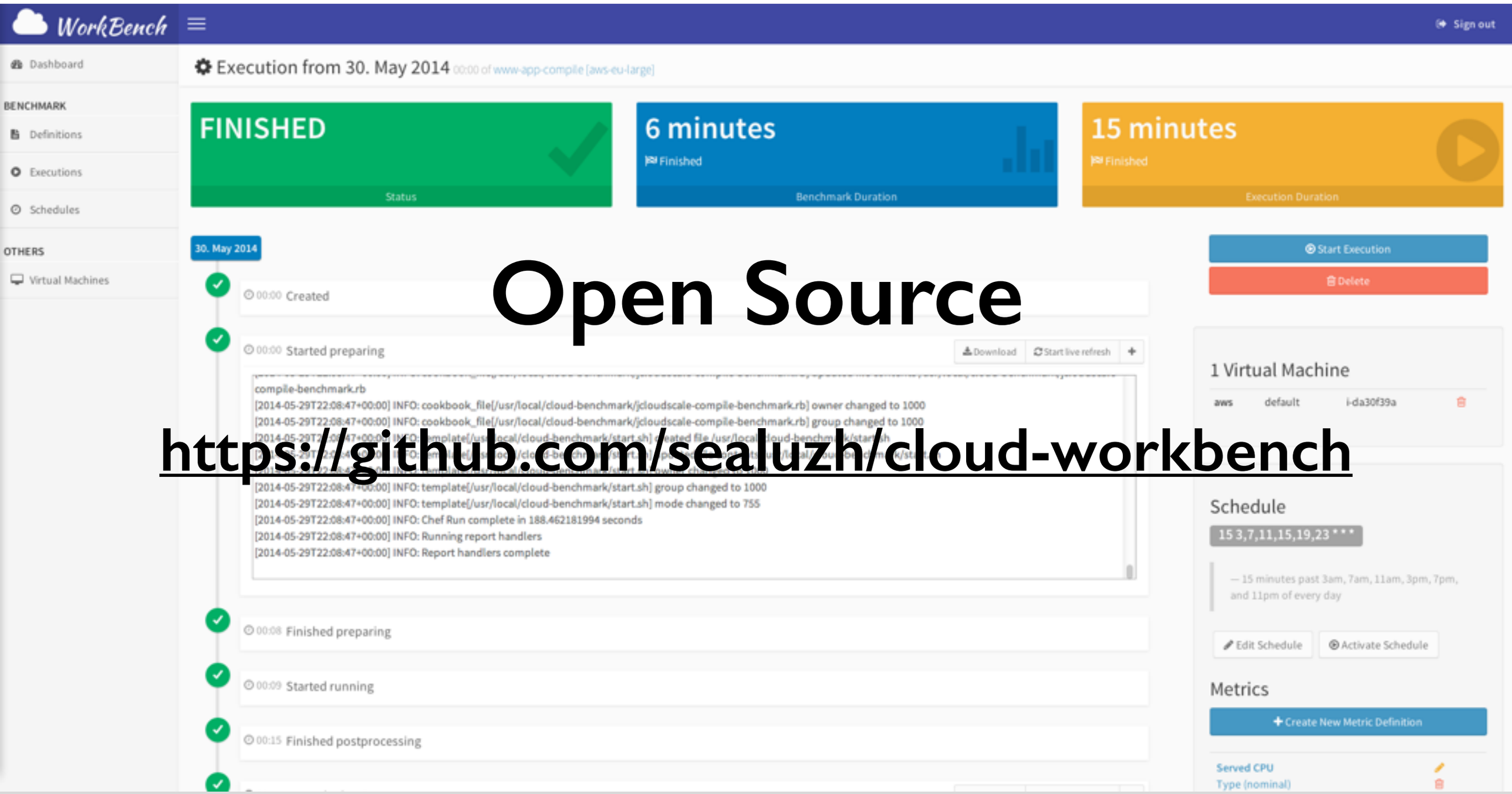
Benchmark

- Performance test
- Types
  - Micro-Benchmarks
  - Application Benchmarks

Demo



Cloud WorkBench



Open Source

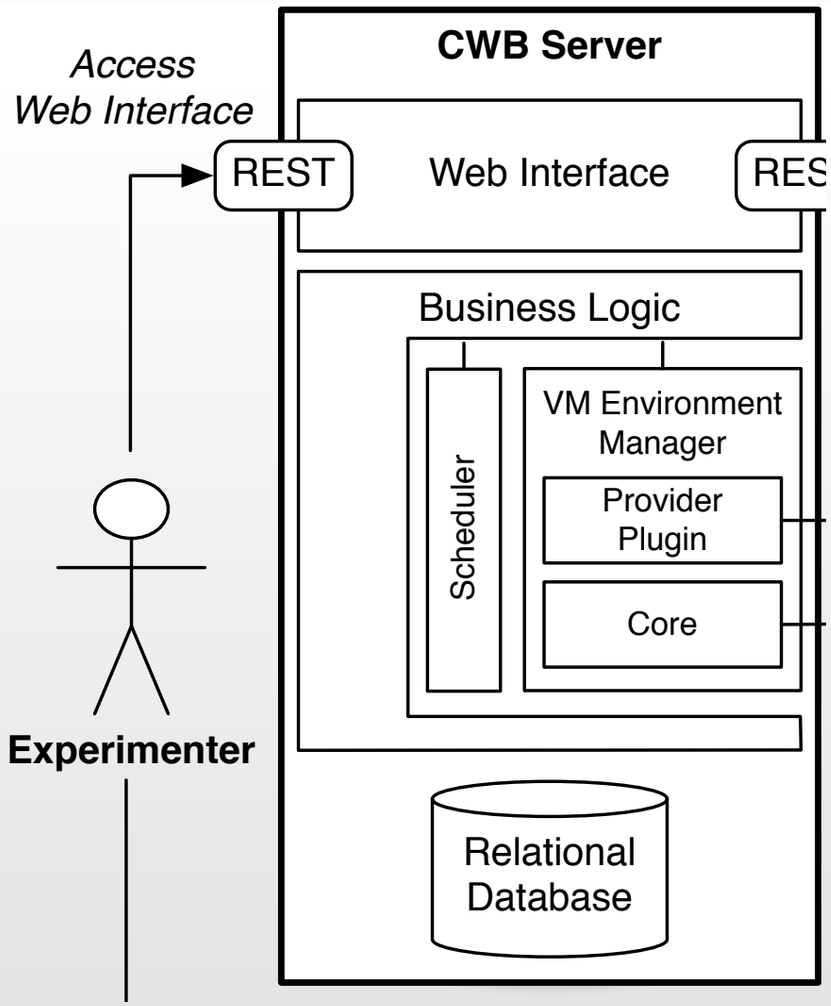
https://github.com/sealuzh/cloud-workbench

Research Questions

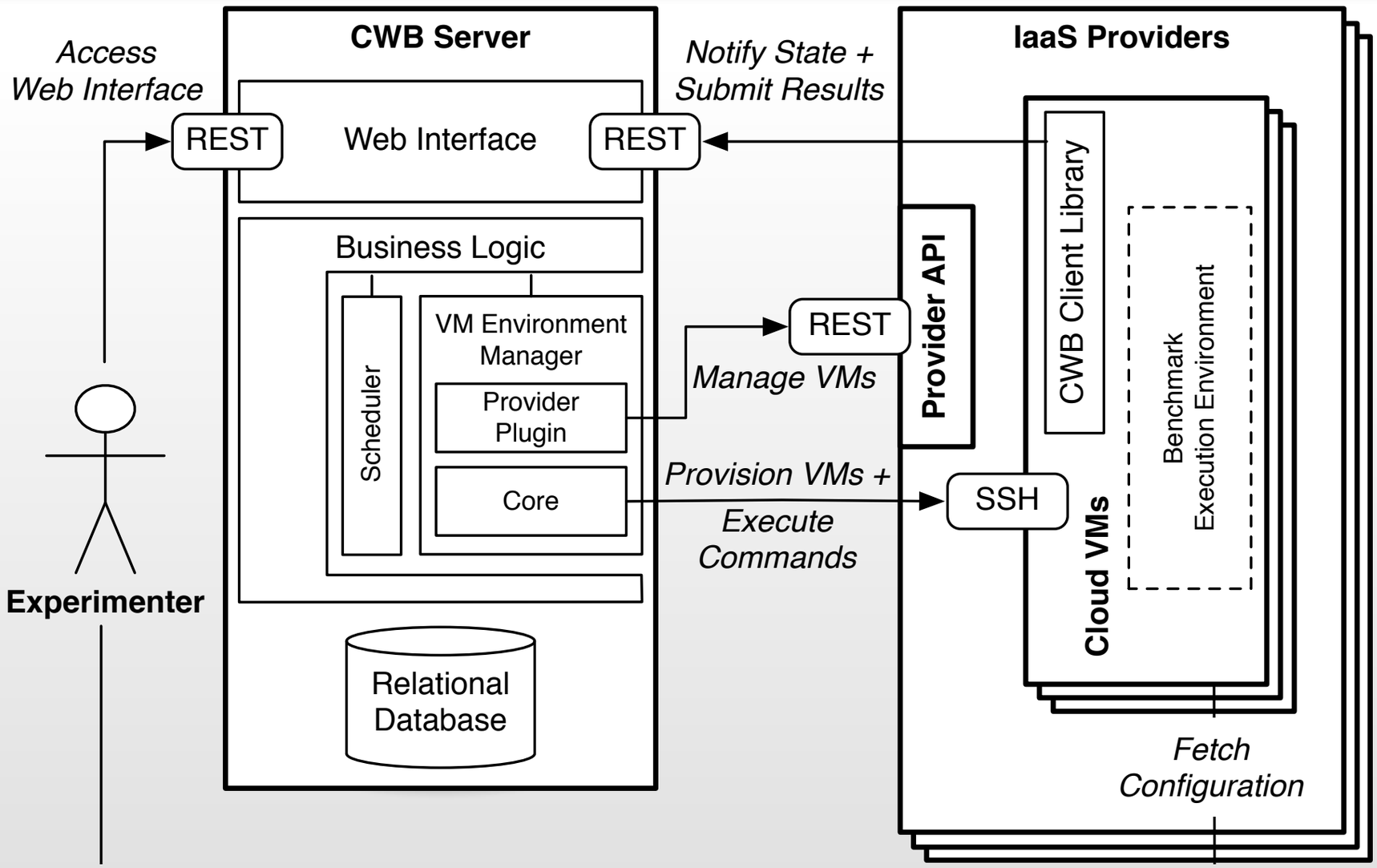
I *How can common IaaS cloud benchmarks from literature be defined in a modular and portable manner?*

II *How can benchmarks from RQ I be periodically scheduled and reproducibly executed in cloud environments without manual interaction?*

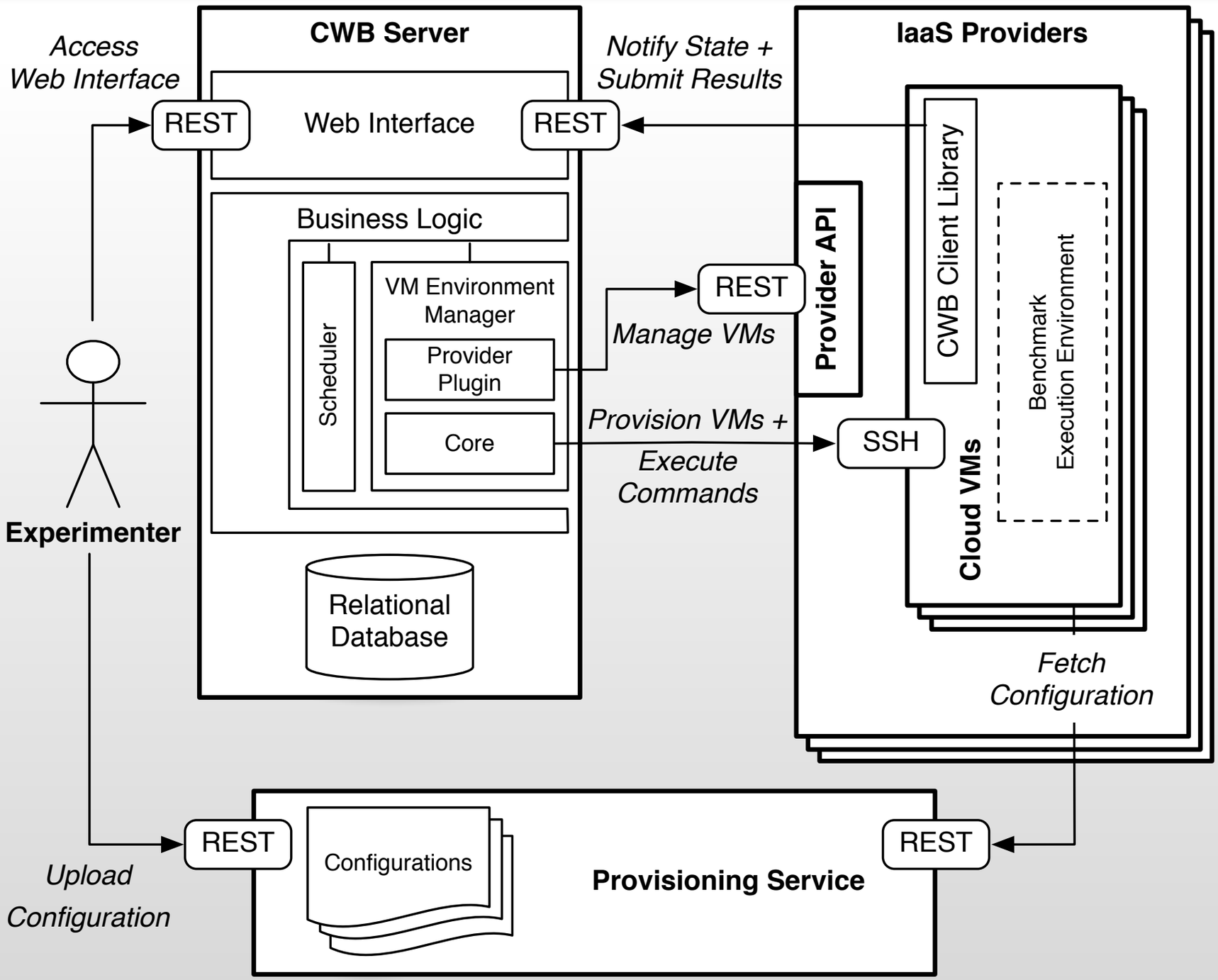
Overall Architecture



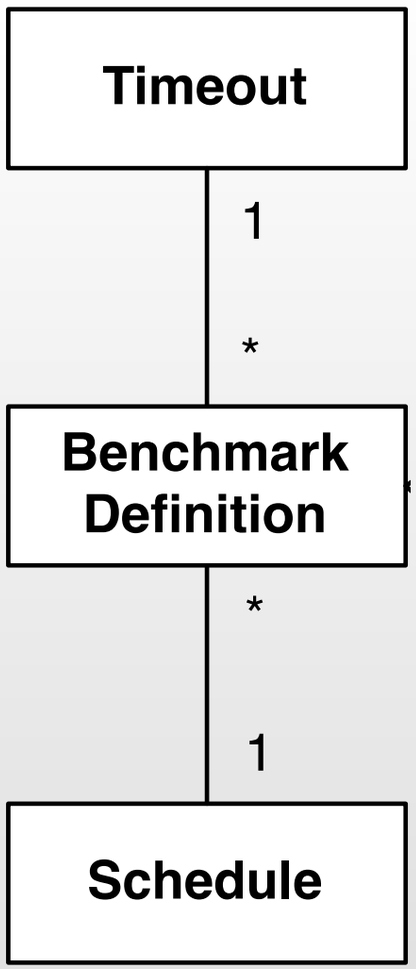
Overall Architecture



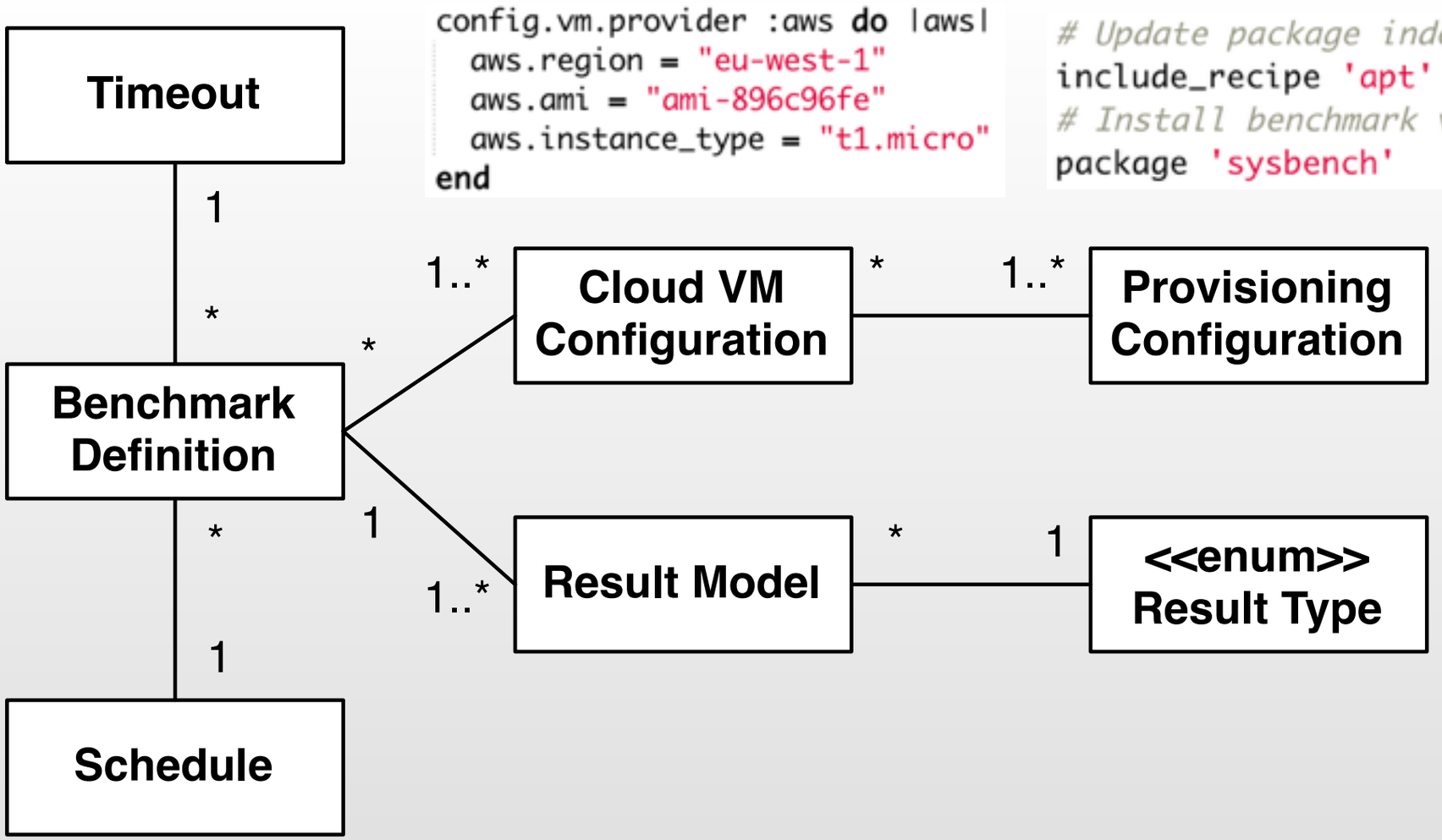
Overall Architecture



Benchmark Anatomy



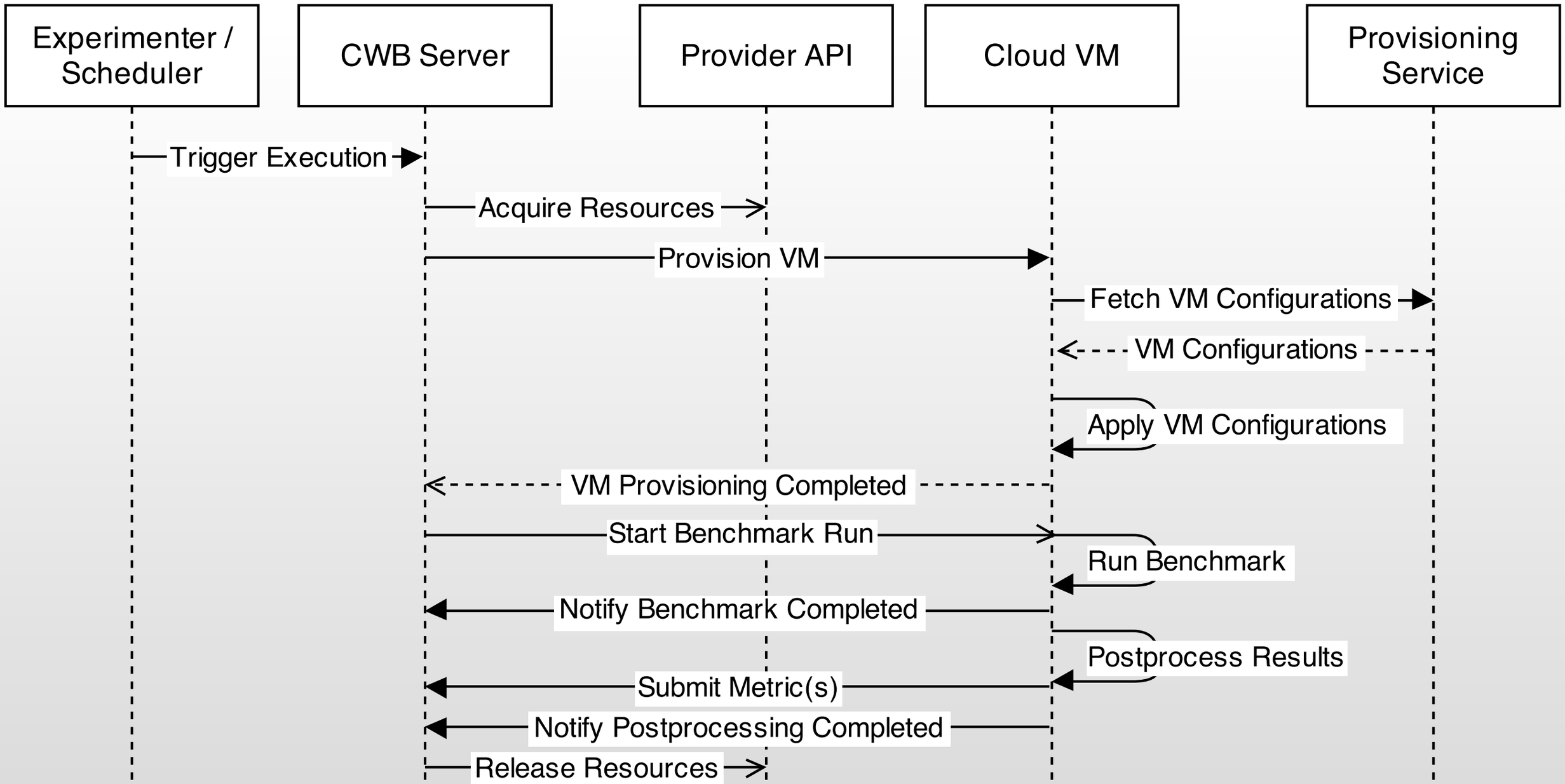
Benchmark Anatomy




```
config.vm.provider :aws do |aws|
  aws.region = "eu-west-1"
  aws.ami = "ami-896c96fe"
  aws.instance_type = "t1.micro"
end
```

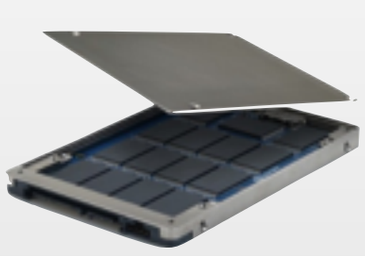
```
# Update package index
include_recipe 'apt'
# Install benchmark via package manager
package 'sysbench'
```

Benchmark Execution



Case Study

- Raw sequential write speed
- HDD vs. SSD storage
- Different instance types



Questions

1. *When do larger instance types perform better than smaller instance types?*

2. *When should larger instance types be preferred over the better block storage type?*

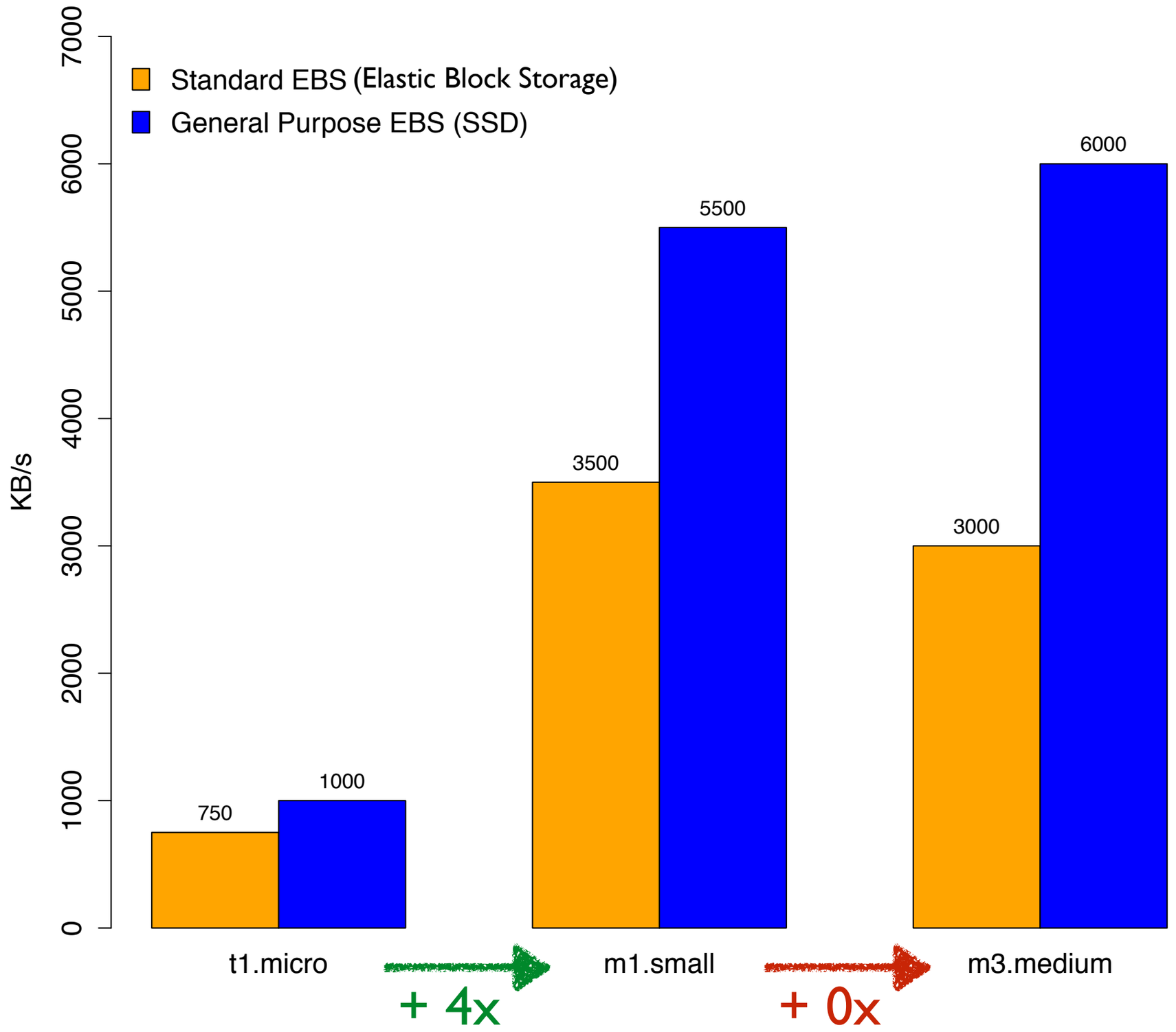
3. *How do instance types and block storage types influence performance variability?*

Experiment Setup

- Amazon EC2 Ireland (eu-west-1)
- Ubuntu 14.04 (trusty)
- FIO benchmark
- 20. - 23. June 2014
- 8 - 12 repetitions

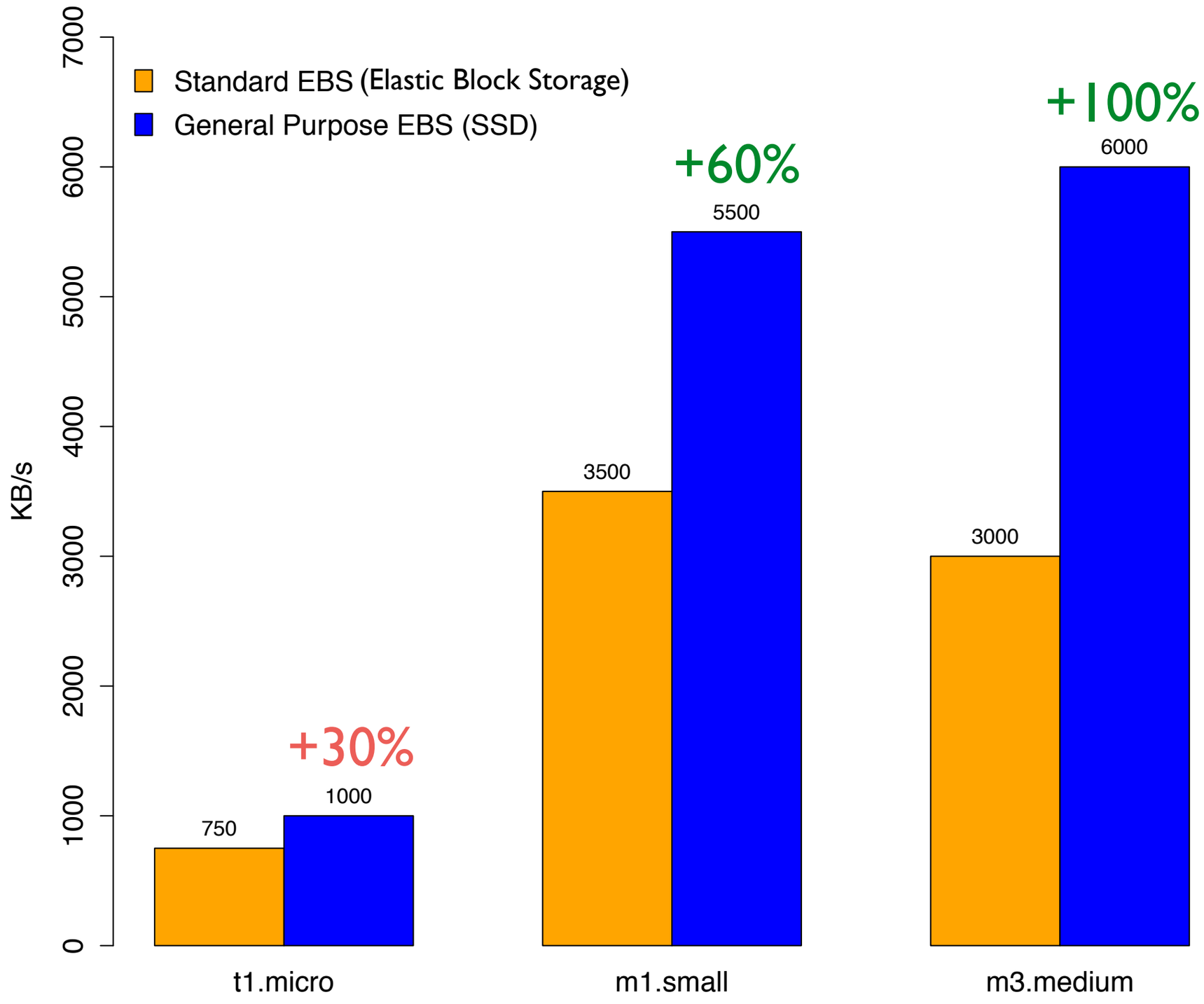
Instance Type	Price per Hour
t1.micro	$ 0.020
m1.small	$ 0.047
m3.medium	$ 0.077

When do larger instance types perform better than smaller instance types?



2

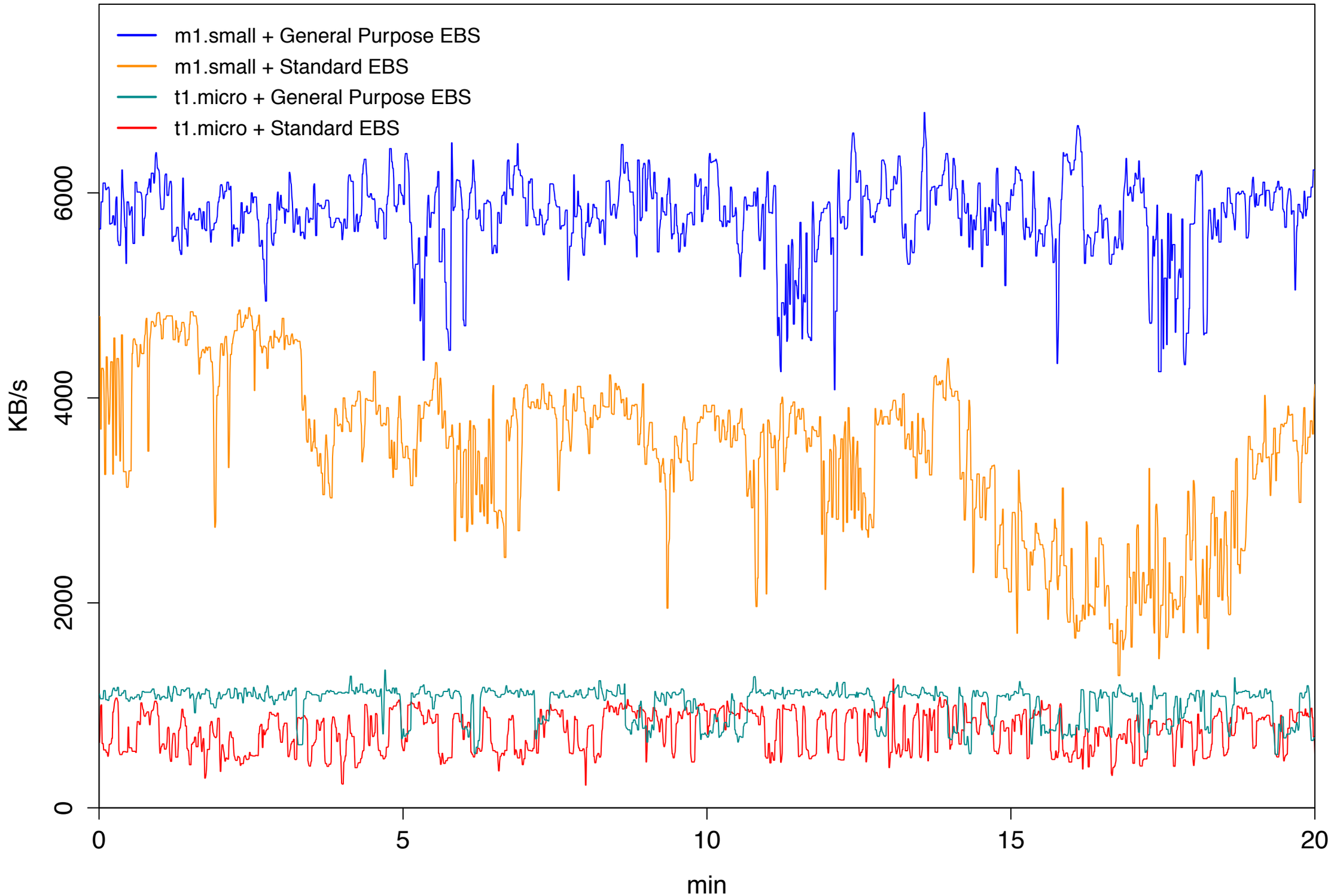
When should larger instance types be preferred over the better block storage type?



Larger Instance Type Better Block Storage Type

3

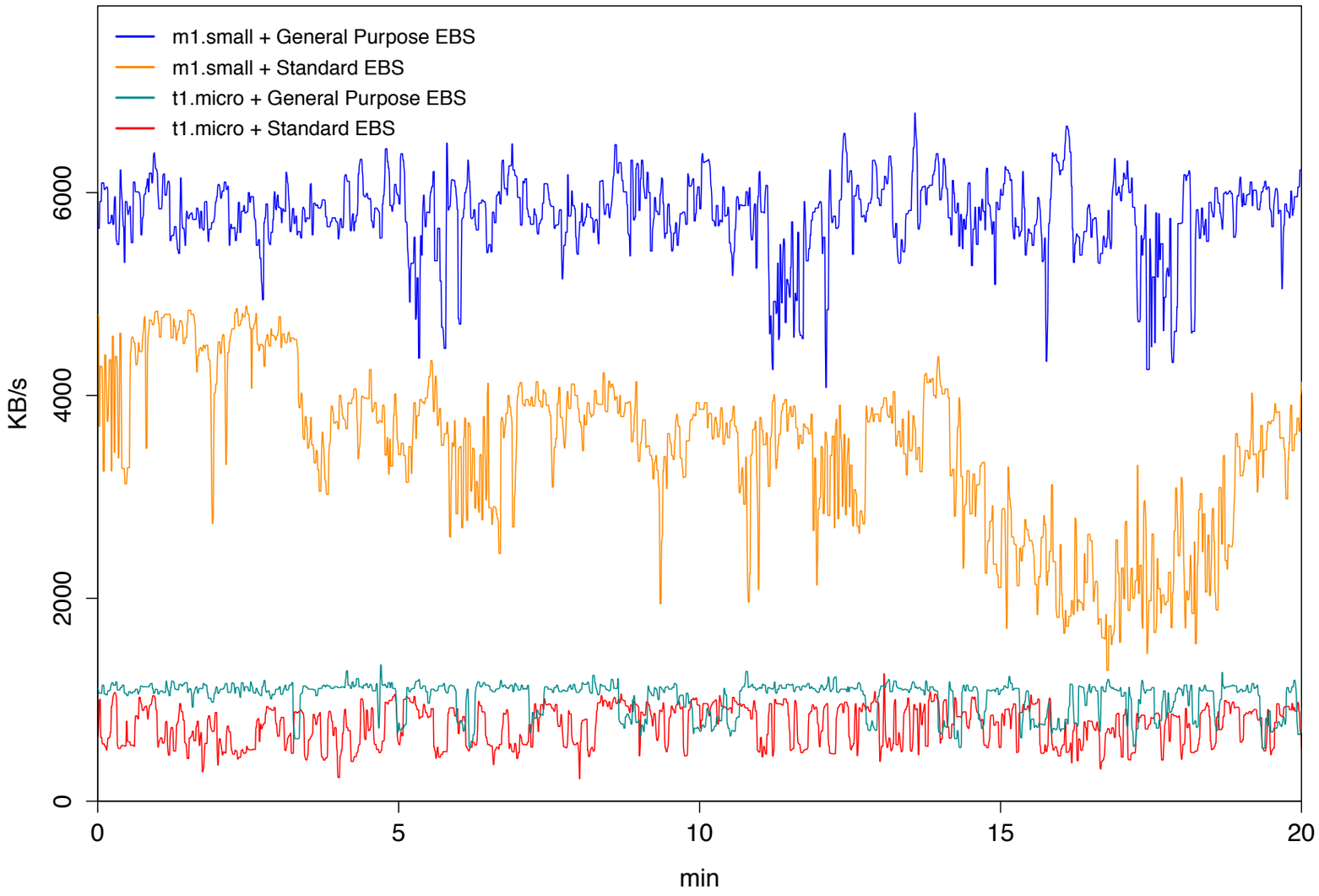
How do instance types and block storage types influence performance variability?



3

How do instance types and block storage types influence performance variability?

s in % of $\bar{x}$	t1.micro	m1.small	m3.medium
Standard EBS	**20%** (20-50%)	**20%** (10-20%)	**30%** (15-60%)
General Purpose EBS	**10 %** (20-40%)	**10%** (5-15%)	**10%** (5-10%)



Conclusion I

How can common IaaS cloud benchmarks from literature be defined in a modular and portable manner?

- Entirely define benchmarks by means of code
- Apply common software engineering techniques
- Make components and benchmarks configurable

Conclusion II

II

How can benchmarks from RQ1 be periodically scheduled and reproducibly executed in cloud environments without manual interaction?

- Use system utilities and existing tools to build a fully automated benchmark execution environment
- Eliminate any error-prone human interactions threatening reproducibility

Outlook

- Large-scale evaluation for a World Wide Web Conference paper
- Extend CWB in a master project
  - Web-based benchmarking studio
  - Support entire lifecycle