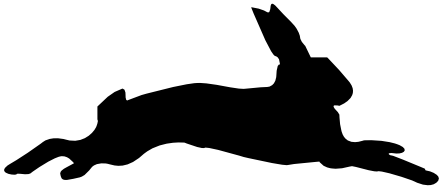
# Bursting with Possibilities

An Empirical Study of Credit-Based Bursting Cloud Instance Types

Dr. Philipp Leitner, Joel Scheuner
leitner@ifi.uzh.ch, joel.scheuner@uzh.ch

# A new Type of Cloud Instances



**Credit-Based Bursting Instances**

➔ Behave fundamentally different than any other existing instance type


Icons from the Noun Project: Rabbit by Hayden Kerrisk, Stopwatch by Nørgaard Andersen, Snail by Jems Mayor

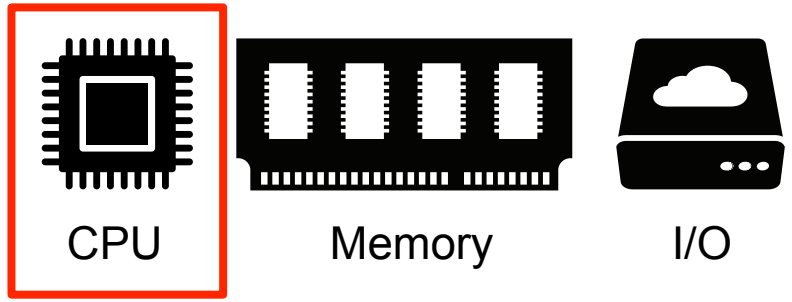
# Context

- Infrastructure-as-a-Service (IaaS)


Google Compute Engine

- Virtual Machines (VMs) on a pay-per-use basis

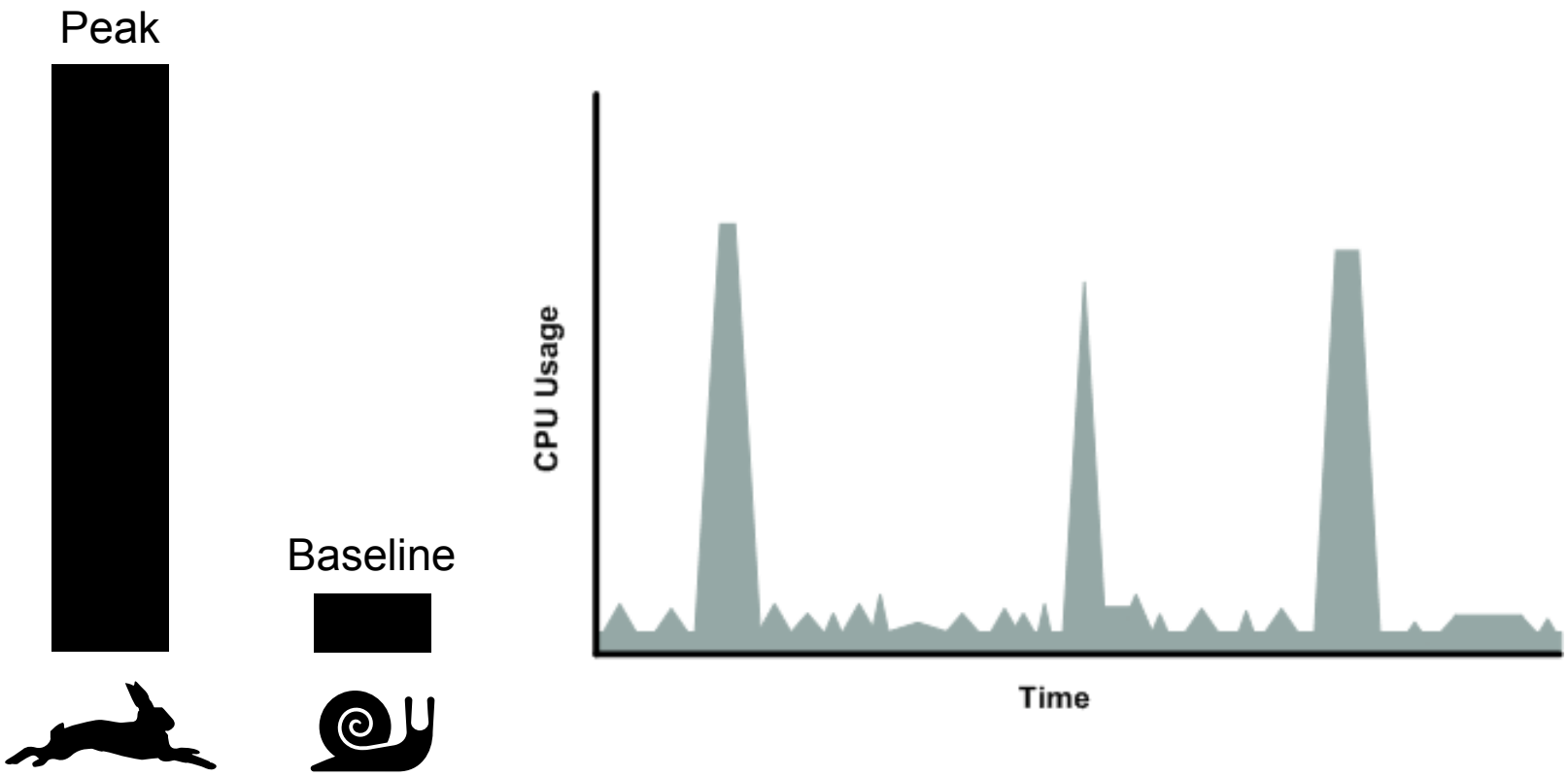
- Different performance characteristics





Icons from the Noun Project: CPU by iconsmind.com, ram by Bryn Bodayle, cloud-storage by Matthew Hawdon

# Credit-Based CPU Bursting





Icons from the Noun Project: Rabbit by Hayden Kerrisk, Snail by Jems Mayor

# Bursting Instance Types in Industry

amazon web services™

AWS Official Blog
Oct 2015

"The burstable model has proven to be **extremely popular** with our customers."

AWS re:Invent
October 6 - 9, 2015 | Las Vegas

Announced a **new instance type** in the burstable T2 family


Google Compute Engine

"f1-micro machine types **offer bursting capabilities** that allow instances to use additional physical CPU for short periods of time"

# Related Work

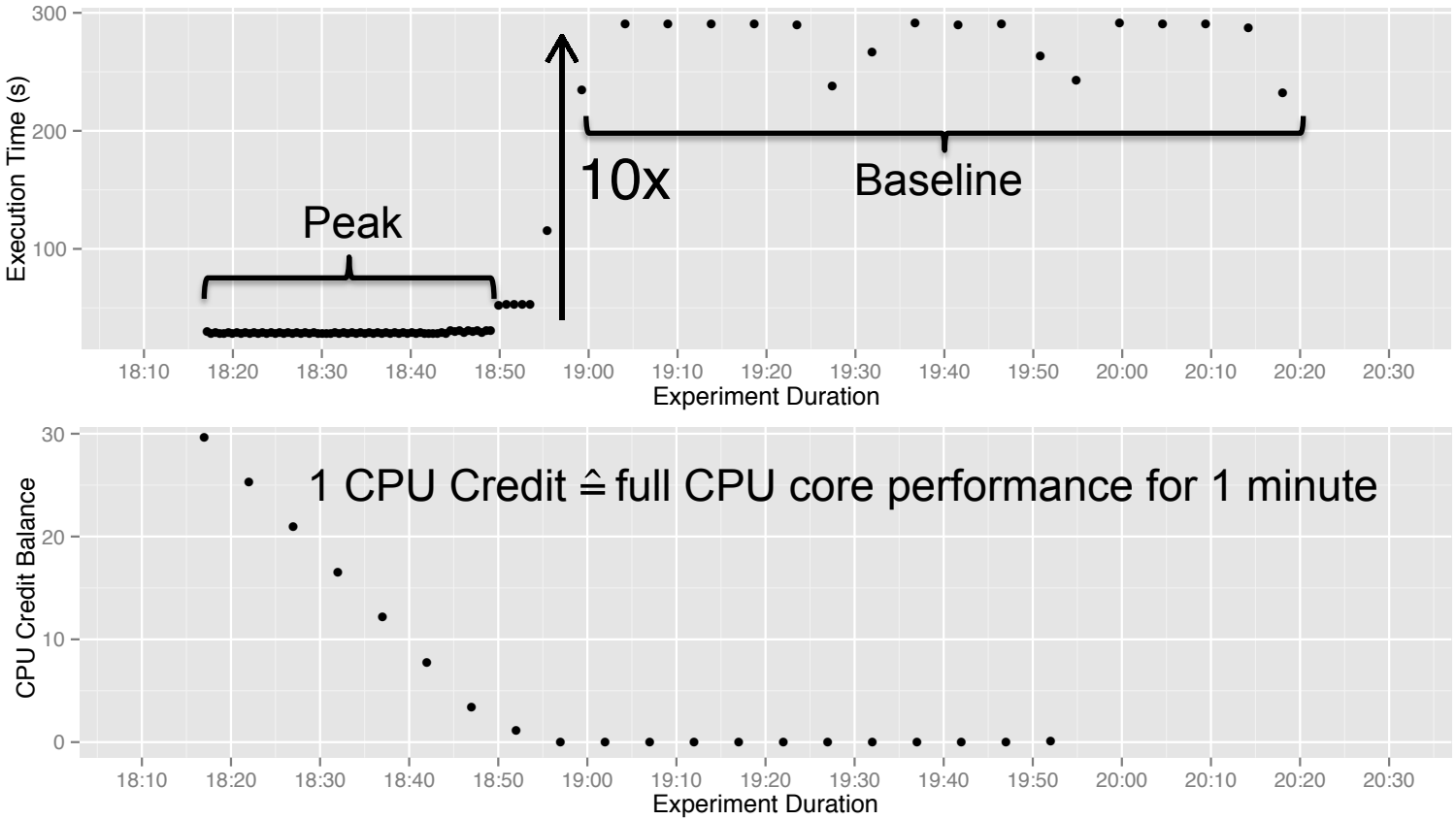
## Cloud Benchmarking

- S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "**A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing**," in *Cloud Computing*, ser. Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering. Springer, 2010, vol. 34, pp. 115–131.
- K. R. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, H. J. Wasserman, and N. J. Wright, "**Performance analysis of high performance computing applications on the amazon web services cloud**," in *Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science*, ser. CLOUDCOM '10, 2010, pp. 159–168.
- A. Iosup, S. Ostermann, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "**Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing**," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 6, pp. 931–945, Jun. 2011.

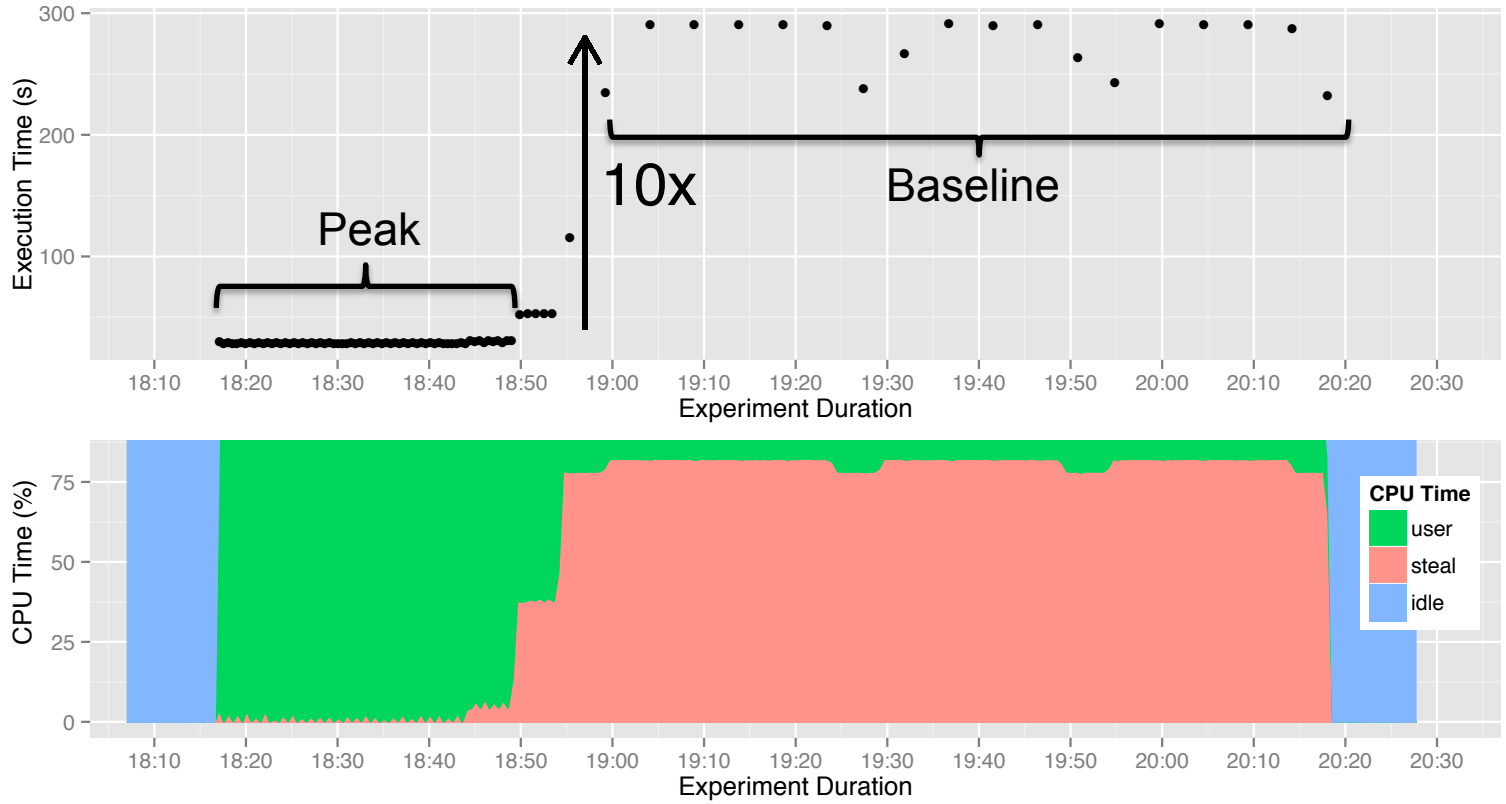
## Burstable Instances

- J. Wen, L. Lu, G. Casale, and E. Smirni, "**Less can be More: micro-Managing VMs in Amazon EC2**," in *Proceedings of the 2015 IEEE International Conference on Cloud Computing (CLOUD'15)*, 2015.

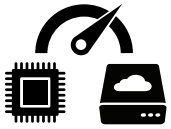
# Credit-Based CPU Bursting – Explained (1)



# Credit-Based CPU Bursting – Explained (2)



# Research Questions



1. How do *t2* bursting instance types perform in terms of CPU and IO speed in comparison to other instances?

2. When are *t2* bursting instance types more cost-efficient than other instance types?

3. How do *t2* instance types perform in comparison to the previous generation (*t1*) types?



Icons from the Noun Project: CPU by iconsmind.com, cloud-storage by Matthew Hawdon, Dashboard by Björn Andersson, Coins by hunotika, History by Joe Mortell

# Empirical Study Setup

amazon web services EC2 Region Ireland (eu-west-1)

All T2 bursting instance types in May 2015
(t2.micro, t2.small, t2.medium)

Sysbench measures CPU and I/O performance

1.-15. May 2015

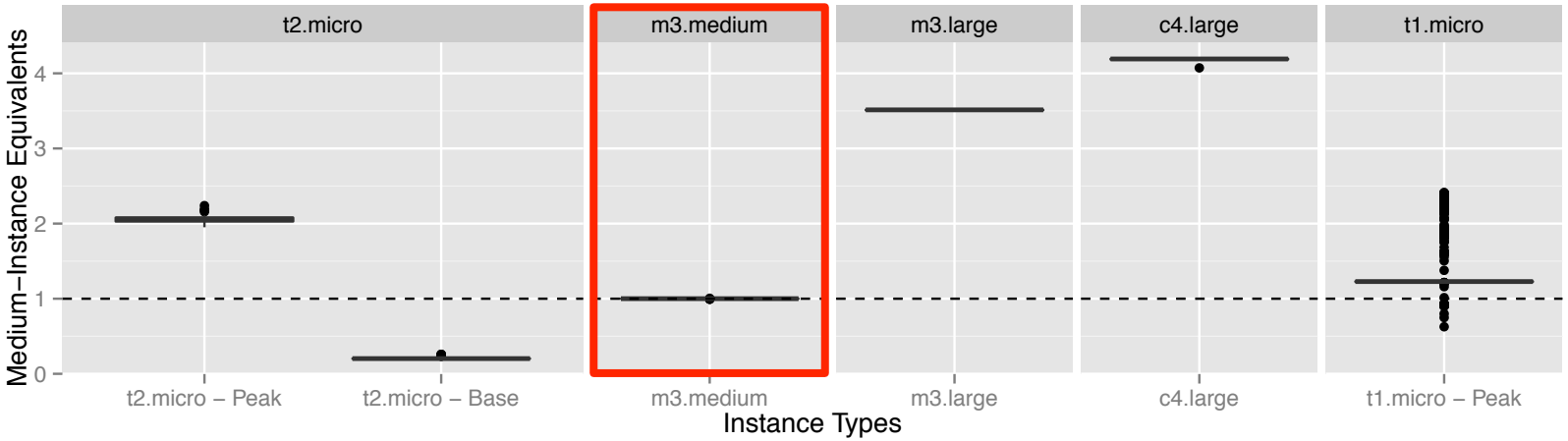
50 data points for each configuration (~1000 in total)

Automated execution with Cloud WorkBench (CWB) [1]

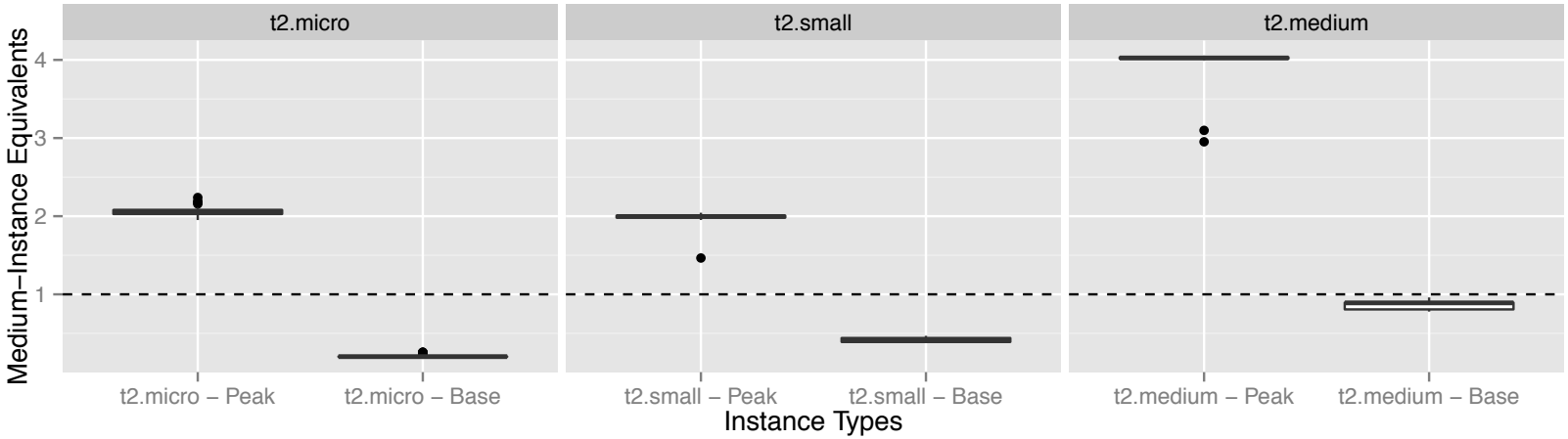
Benchmark definitions and data publicly available:
https://github.com/sealuzh/bursting-cloud-instances

[1] Scheuner, Leitner, Cito, Gall: Cloud WorkBench - Infrastructure-as-Code Based Cloud Benchmarking. CloudCom'14

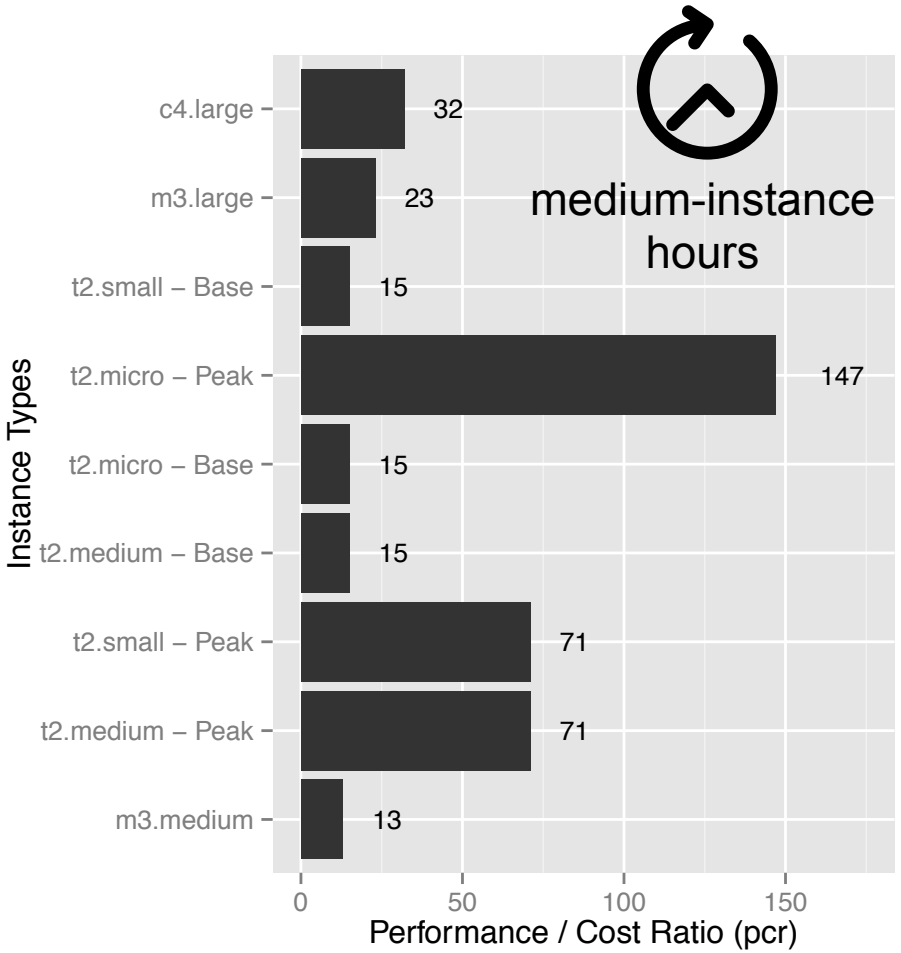
# Results – T2 vs. Other Instance Types



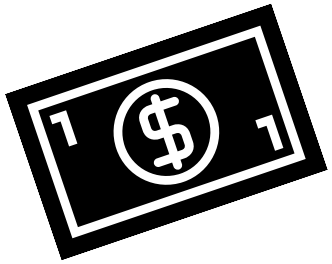
# Results – T2 Bursting Instances

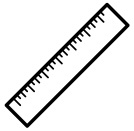


# Results – Performance-Cost Ratio (1)




≙

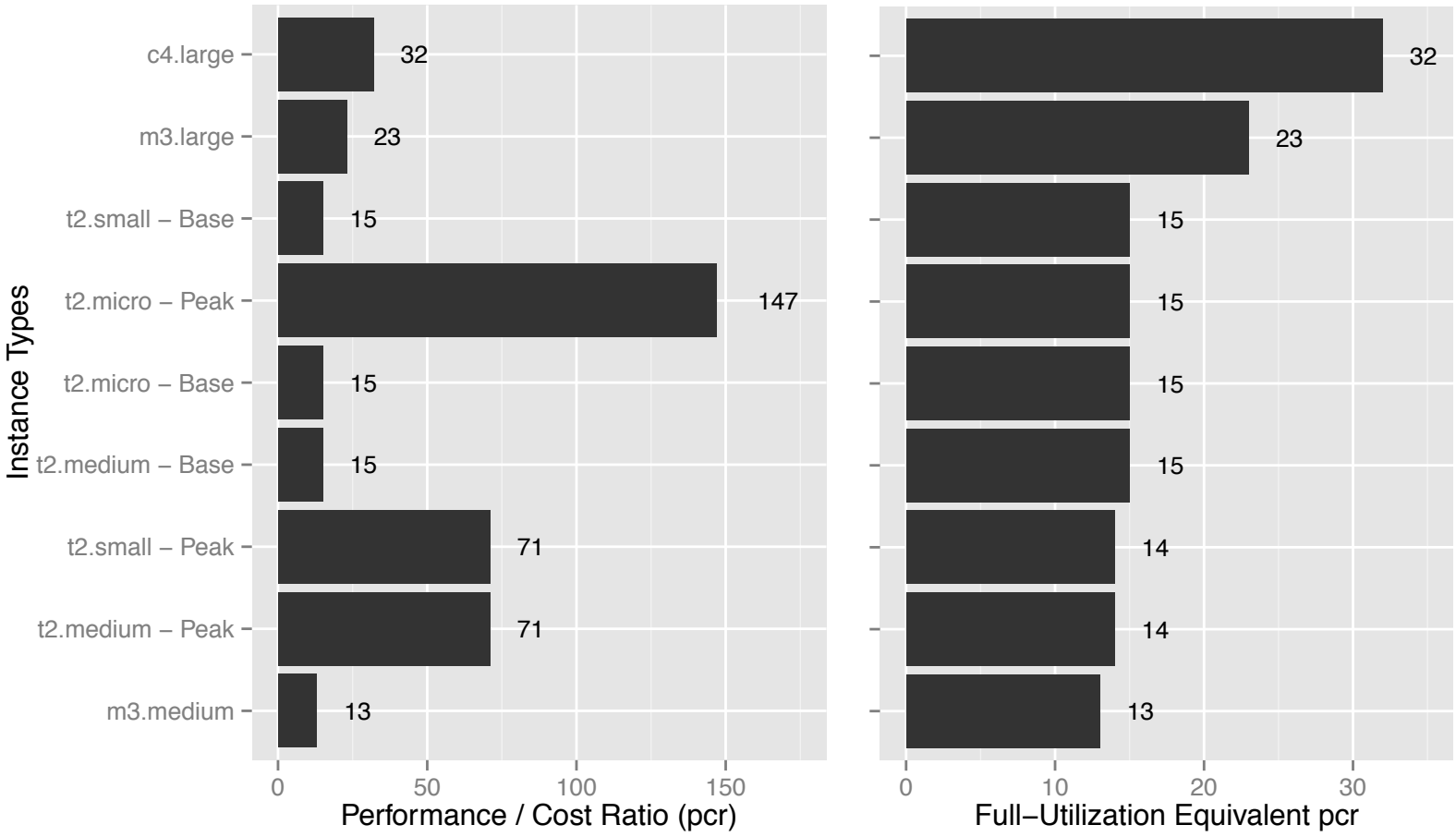




medium-instance equivalents per USD and hour

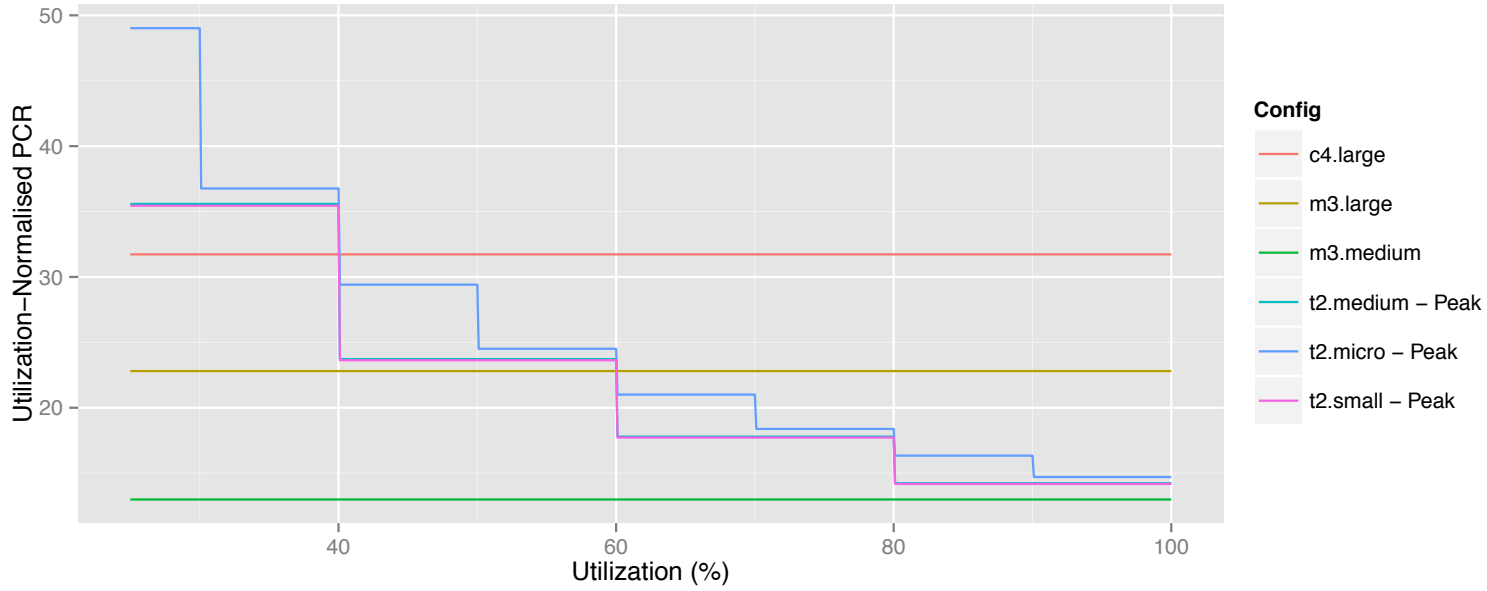

Icons from the Noun Project: 24 hour by iconsmind.com, Ruler by Arthur Shlain, dollar by Simple Icons

# Results – Performance-Cost Ratio (2)



# Usage Scenarios – Low or Irregular Load

- Identify the cutoff point for each T2 instance
  - Where does higher avg. utilization (u) make them less cost efficient
- Assumptions: Service is CPU-bound + always requires peak performance



# Usage Scenarios – Boosting Performance-Cost Ratio

**Idea**

Exploit initial CPU credit balance on VM startup

**Implementation**

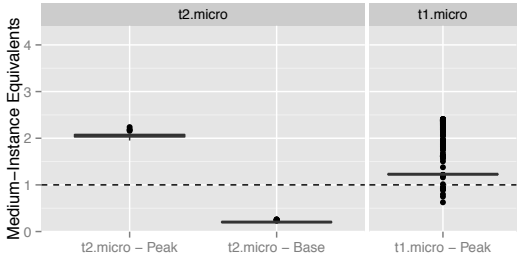
Systematically restart VM instances when they run out of CPU credits

**Effect**

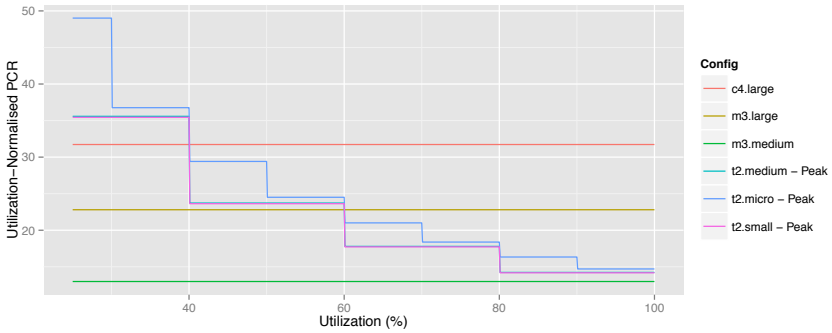
Improved (utilization normalized) performance cost ratio up to 4x

# Conclusions

T2 instance types **perform highly predictable** unlike the previous T1 generation of bursting instances.




T2 instance types provide **superior performance-cost ratio below 40% average utilization**




SEVENTH FRAMEWORK PROGRAMME
CloudWave
This research has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 610802 (CloudWave).


Icons from the Noun Project: Dice by chris dawson, Coins by hunotika

# APPENDIX

# Future Work

- Limited to micro-benchmarks

  → Validate results using application benchmarks / actual applications

- Limited to CPU credit bursting

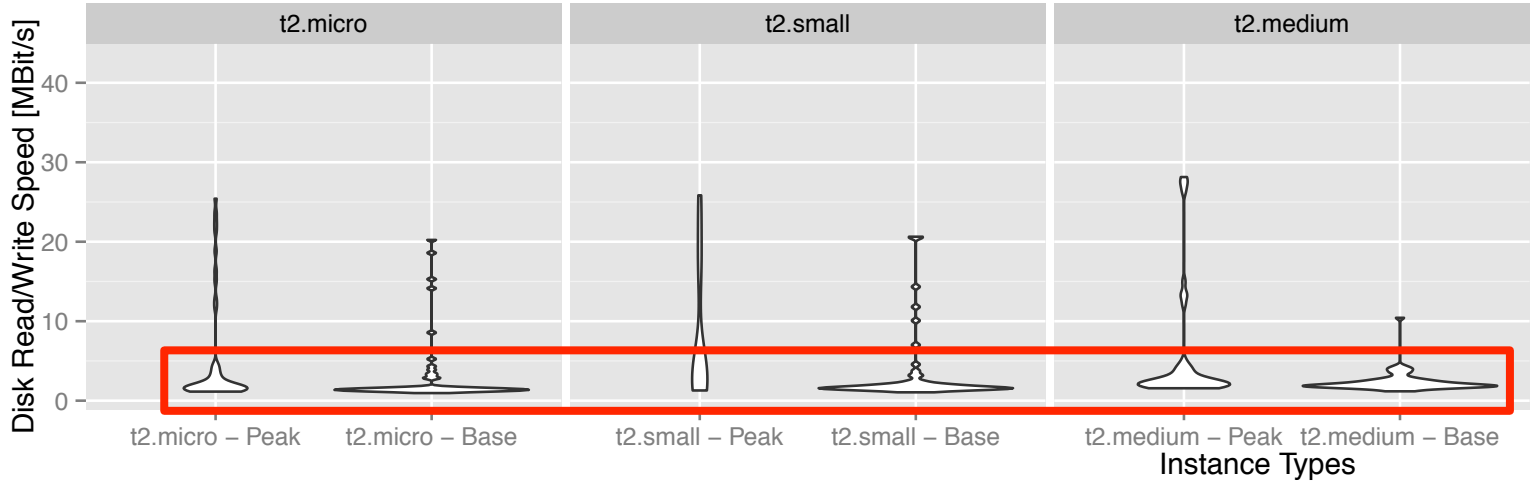
  → Analyze the same bursting model for IOPS[1]

[1] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html#IOcredit

# Usage Scenarios – Non-Critical IO

- Trend towards more homogenous IO performance
- No substantial IO performance degradation even at baseline performance

→ Use cost-efficient CPU instances for IO-bound applications



# Generations – T1 (previous) vs T2 (current)

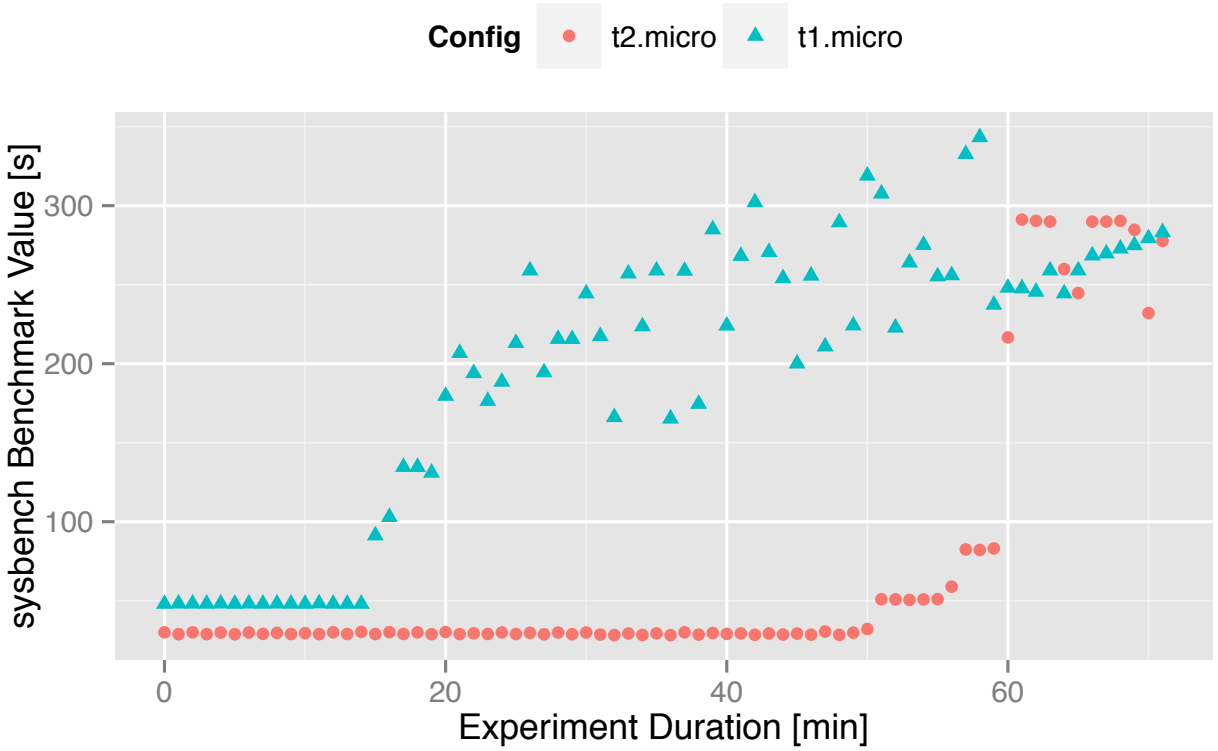



Figure 5: Comparison of performance development of stressed `t1` and `t2` instances.

# Formal Model – Concise

- $pcr(t) = \frac{\bar{m}}{c(t)}$ performance-cost ratio (at a given time t)
  - Unit: medium instance equivalents per USD and hour
  - $\bar{m}$ Arithmetic mean of all 50 benchmark observations [seconds]
  - $c(t) \in \mathbb{R}^+$ Hourly costs per started billing time unit [USD]

- $unpcr(t,u) = \frac{pcr(t)}{\lceil \frac{u}{t_{\bar{u}}} \rceil}$ utilization-normalized cost-performance ratio
  - Intuitively: costs of operating a cluster of bursting instances, so that one instance can always be operated at peak performance under the assumed utilization level (e.g., need 10x t2.micro for u=100)
  - $u \in [0; 100]$ Utilization level
  - $t_{\bar{u}}$ Standard instance utilization (i.e., utilization rate that keeps CPU credit balance constant)

# Formal Model – Basic Definitions (1)

- Bursting instances *(I)* $i \in I$
- Credit-based bursting instance types *(T)* $i^t \in T$

- Peak performance level $s_p(t) \in \mathbb{R}^+$
- Baseline performance level $s_b(t) \in \mathbb{R}^+$

Lower number represents better performance

- CPU credits available $i_c \in \mathbb{R}^+$
- Replenishment rate per hour (when CPU idle) $t_r \in \mathbb{N}^+$
- Depletion rate per hour (when CPU non-idle) $t_d \in \mathbb{N}^+$
- Startup credits (initial credits on VM startup) $t_s \in \mathbb{N}^+$
- Credit limit (max amount of credits) $t_m \in \mathbb{N}^+$

# Formal Model – Basic Definitions (2)

- Standard Instance Utilization *($\bar{u}$)* (i.e., utilization rate that keeps the instance credit account constant) $t_{\bar{u}}$
- Utilization level (i.e., percentage of time a user wants to operate a bursting instance at peak performance level) $u \in [0; 100]$
- Hourly costs (US $ per started billing time unit) $c(t) \in \mathbb{R}^+$
- Arithmetic mean of all 50 benchmark observations $\bar{m}$
- Relative standard deviation in percent $m_\sigma$

# Formal Model – Performance-Cost Metrics

- Performance-cost ratio

$$pcr(t) = \frac{\bar{m}}{c(t)}$$

Unit: medium-instance equivalents per US dollar and hour

- Utilization-normalized performance-cost ratio

$$unpcr(t,u) = \frac{pcr(t)}{\left\lceil \frac{u}{t\,\bar{u}} \right\rceil}$$

Number of required instances to operate at peak performance, given u (e.g., 10x t2.micro for u=100)

Intuitively: costs of operating a cluster of bursting instances, so that one instance can always be operated at peak performance under the assumed utilization level u

# Contributions

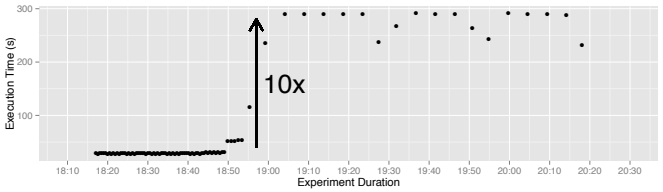
1. Basic formal model for credit-based bursting behavior

$$\textit{peak performance level } s_p(t) \in \mathbb{R}^+ \qquad pcr(t) = \frac{\bar{m}}{c(t)}$$

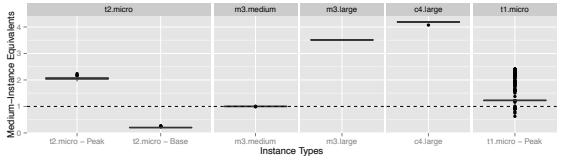
$$\textit{baseline performance level } s_b(t) \in \mathbb{R}^+ \qquad unpcr(t,u) = \frac{pcr(t)}{\lceil \frac{u}{t_{\bar{u}}} \rceil} \quad u \in [0;100]$$

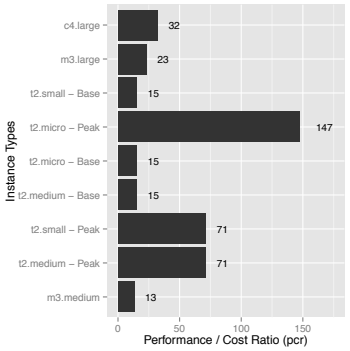
2. Empirical study of performance behavior

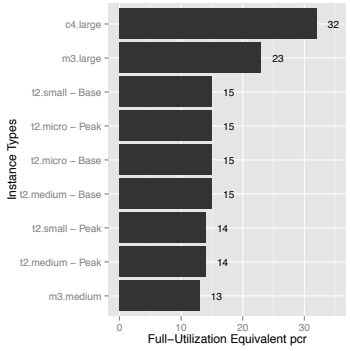
| | t2.micro (0.014 $ / hour) | | t2.small (0.028 $ / hour) | | t2.medium (0.056 $ / hour) | |
|---|---|---|---|---|---|---|
| | $s_p$ | $s_b$ | $s_p$ | $s_b$ | $s_p$ | $s_b$ |
| $\bar{m}$ (CPU) | 2.06 | 0.21 | 1.98 | 0.41 | 3.99 | 0.87 |
| $m_\sigma$ (CPU) | 3% | 8% | 4% | 6% | 5% | 6% |




3. Comparison with current/previous generation instances (performance/cost)




4. Potential uses cases for practitioners