

# Cloud & Big Data

## Elasticity in Cloud Computing

---

Simon Bihel, `simon.bihel@ens-rennes.fr`

Rémi Hutin, `remi.hutin@ens-rennes.fr`

April 5, 2017

University of Rennes I

École normale supérieure de Rennes

# Table of contents

1. Elasticity: Definition and Differentiation
2. When to Scale
3. How to Scale

# **Elasticity: Definition and Differentiation**

---

**Efficiency** Amount of resources consumed for a given amount of work.

## What It Is Not [3]

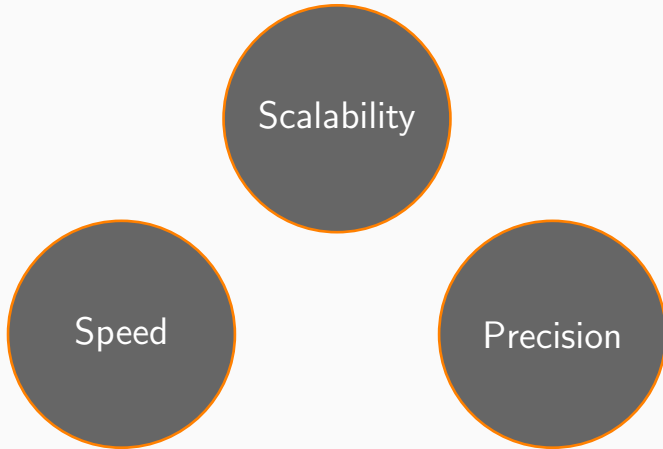
**Efficiency** Amount of resources consumed for a given amount of work.

**Scalability** Sustain increasing workloads with adequate performance.

**Elasticity  $\neq$  Scalability**

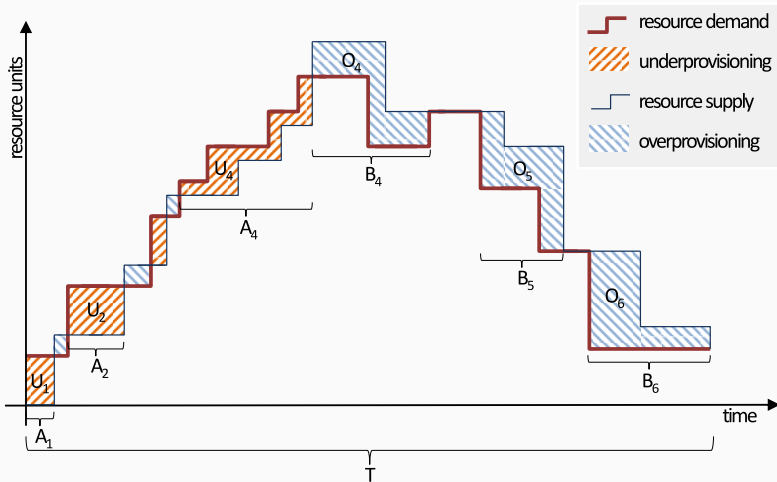
## **Elasticity $\neq$ Scalability**

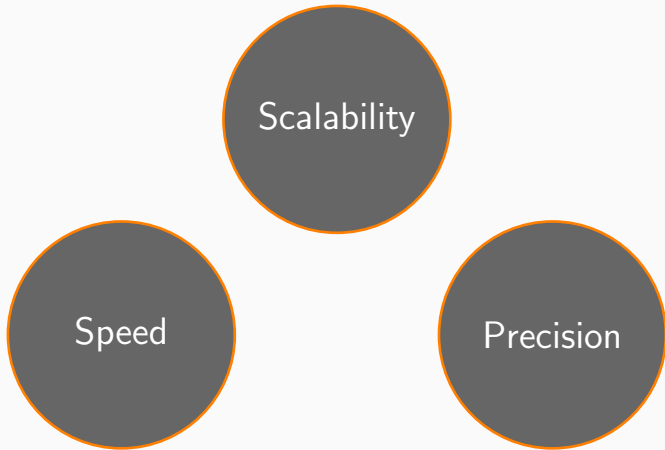
It's more than that. That's the selling point.

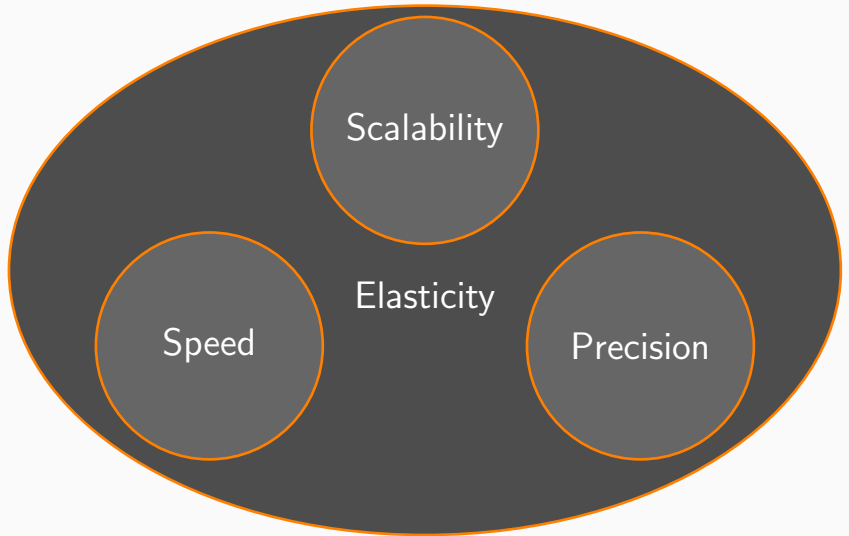




## Example of matching demand [3]



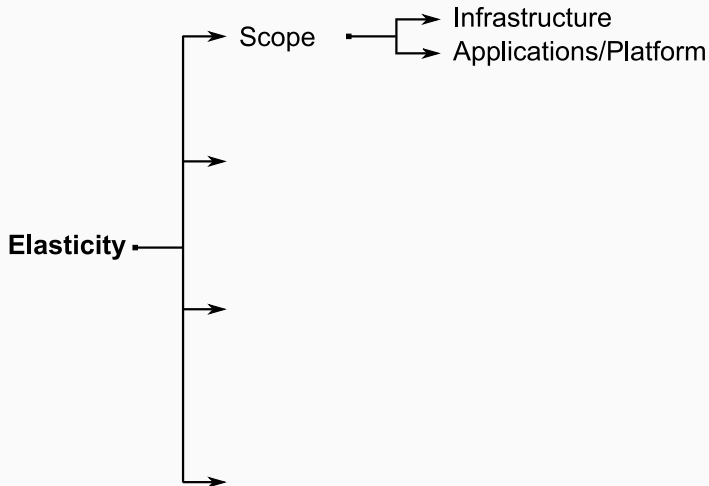




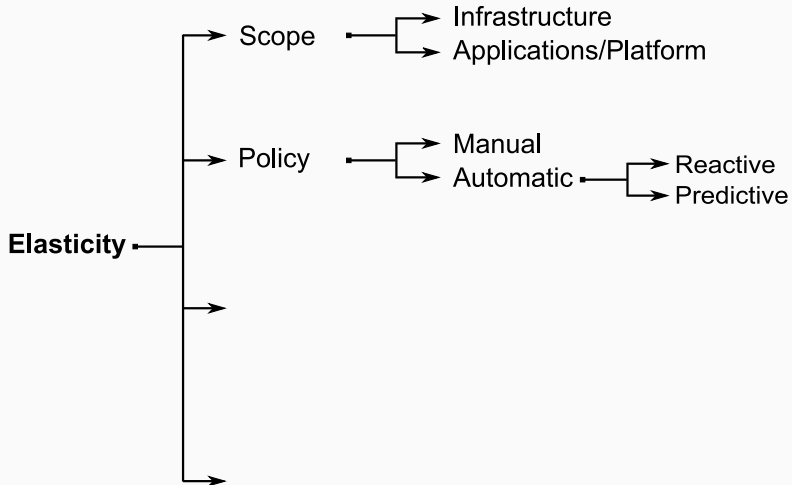
### Definition

**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.

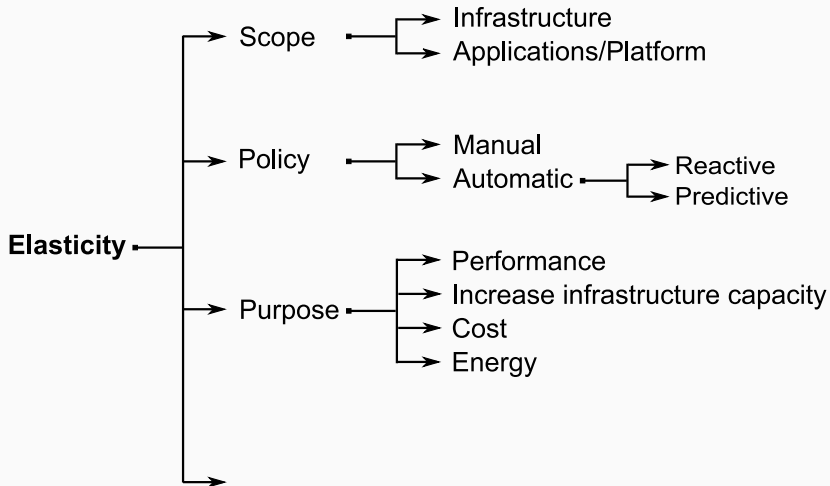
## Different parameters involved in elasticity [1]



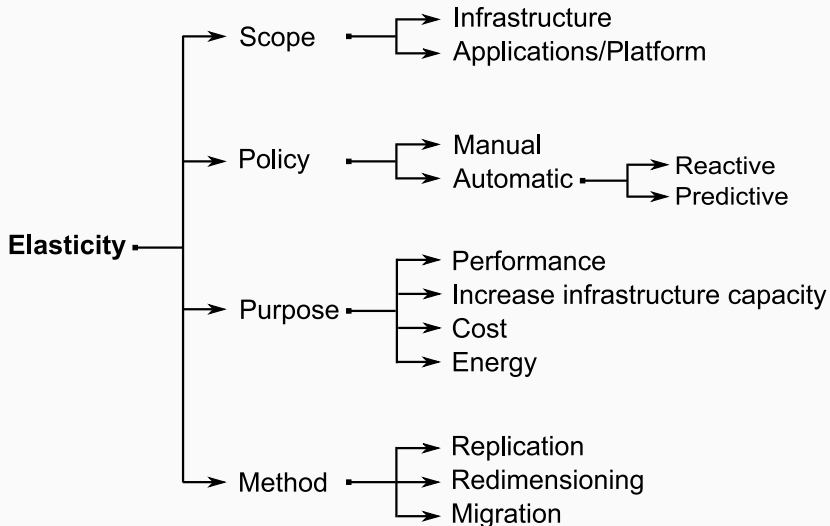
## Different parameters involved in elasticity [1]



## Different parameters involved in elasticity [1]



# Different parameters involved in elasticity [1]





## When to Scale

---

Set of rules (*thresholds*).

Set of rules (*thresholds*).

Different resource managements. [2]

Set of rules (*thresholds*).

Different resource managements. [2]

**Hierarchical** Management systems built on top of each other.  
E.g. cluster level with layer on top.

Set of rules (*thresholds*).

Different resource managements. [2]

**Hierarchical** Management systems built on top of each other.  
E.g. cluster level with layer on top.

**Flat** Completely decentralized management.

Models built through analysis, learning, queueing theory. . .

Models built through analysis, learning, queueing theory. . .

Another resource managements.

Models built through analysis, learning, queueing theory. . .

Another resource managements.

**Statistical** Small scale with dynamic clusters. Repeated small scale optimizations attain large scale load and optimal placement.

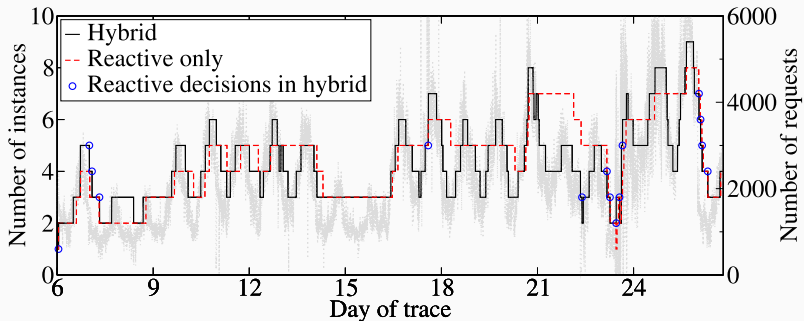


## Example of an elasticity controller [4]

Mixing reactive and predictive methods.

## Example of an elasticity controller [4]

Mixing reactive and predictive methods.



# How to Scale

---

**Horizontal** (*Replication*) Adding/Removing instances (e.g. VMs, modules. . . ).

**Horizontal** (*Replication*) Adding/Removing instances (e.g. VMs, modules. . . ).

**Vertical** (*Resizing*) Adding resources (e.g. processing, memory. . . ). *Not always available.*

# Kinds of scaling / Mechanisms

**Horizontal** (*Replication*) Adding/Removing instances (e.g. VMs, modules. . . ).

**Vertical** (*Resizing*) Adding resources (e.g. processing, memory. . . ). *Not always available.*

**Migration** (*Scaling Out*) Transferring a VM from one physical server to another one.

## Configurations & Transitions

Amazon EC2 Cloud Platform			
Server size	Configuration	Cost/hr	\$/core
Small	1 ECU, 1.7GB RAM, 160GB disk	\$0.085	\$0.085
Large	4 ECUs, 7.5GB RAM, 850GB disk	\$0.34	\$0.085
Med-Fast	5 ECUs, 1.7GB RAM, 350GB disk	\$0.17	\$0.034
XLarge	8 ECUs, 15GB RAM, 1.7TB disk	\$0.68	\$0.085
XLarge-Fast	20 ECUs, 7GB RAM, 1.7TB disk	\$0.68	\$0.034
New Server's NS Cloud Platform			
Small	1-core 2.8GHz, 1 GB RAM, 36GB disk	\$0.11	\$0.11
Medium	2-core 3.2 GHz, 2 GB RAM, 146GB disk	\$0.17	\$0.085
Large	4-core 2.0GHz, 4GB RAM, 250 GB disk	\$0.25	\$0.063
Fast	4 core 3.0 GHz, 4 GB RAM, 600GB disk	\$0.53	\$0.133
Jumbo	8 core 2.0GHz, 8GB RAM, 1TB disk	\$0.60	\$0.075

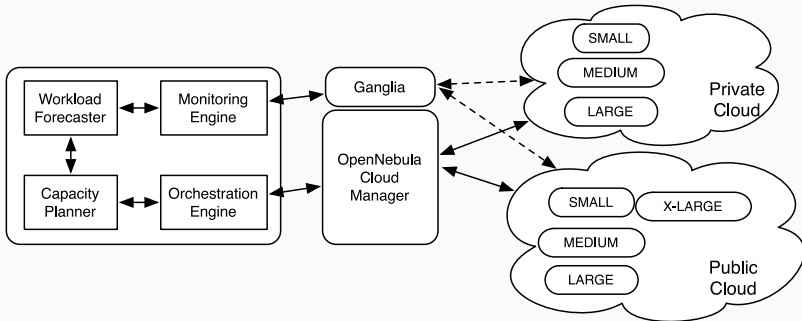
There is also cost for transition.

## Example of a provisioning system [5]

Integer Linear Program to take into account multiple parameters.



## Example of architecture [5]



## Conclusion

---

Not there yet.

- Elements tightly coupled but studied independently.
- Mechanisms conceived with assuming other elements in the workflow to be perfect.
- Overhead can be a problem (frequency, decomposition, failures. . . ).

## References

---

# References I



G. Galante and L. C. E. de Bona.

**A survey on cloud computing elasticity.**

In *Utility and Cloud Computing (UCC), 2012 IEEE Fifth International Conference on*, pages 263–270. IEEE, 2012.



A. Gulati, G. Shanmuganathan, A. M. Holler, and I. Ahmad.

**Cloud scale resource management: Challenges and techniques.**

*HotCloud*, 11:3–3, 2011.



N. R. Herbst, S. Kounev, and R. H. Reussner.

**Elasticity in cloud computing: What it is, and what it is not.**

In *ICAC*, pages 23–27, 2013.

## References II



L. R. Moore, K. Bean, and T. Ellahi.

**A coordinated reactive and predictive approach to cloud elasticity.**

2013.



U. Sharma, P. Shenoy, S. Sahu, and A. Shaikh.

**A cost-aware elasticity provisioning system for the cloud.**

In *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, pages 559–570. IEEE, 2011.