MINES Saint-Étienne | Institut Mines-Télécom

Cloud Elasticity

Luis Gustavo Nardin
gnardin@emse.fr

Cloud and Edge Infrastructures

November 9, 2025

Outline

Introduction

Elasticity: Cloud Computing

- Elasticity can be loosely understood as the ability of a system to automatically provision and deprovision computing resources on demand as workloads change
- Essential characteristic of Cloud Computing
- Enabled by resources virtualization
- Main factor motivating the adoption of cloud computing
  - Adjust to the customer's resource needs
  - No commitment

Elasticity: Cloud Provider Perspective

- Elasticity is a non-functional requirement provided by Cloud Providers to their users
- Cloud Providers are responsible for satisfying this feature
- Cloud Providers must rely on efficient infrastructure management strategies to
  - Minimize energy consumption (e.g., server consolidation)
  - Allow rapid resource deployment
  - Provide VM placement and migration strategies
  - Provide adequate pricing and metering to respect the SLA

Elasticity: User Perspective

- No commitment
- Pay-as-you-go
- Start small, expand later
- Resources offered by the Cloud Provider appear to be unlimited

Definition

Elasticity: Definition

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

(Herbst et al., 2013)

$$\textit{Elasticity} = \underbrace{\textit{scalability} + \textit{automation}}_{\text{auto-scaling}} + \textit{optimization}$$

(Al-Dhuraibi et al., 2018)

Elasticity vs. Scalability

- Scalability is the property of a system to handle a growing amount of work by adding resources to the system (Bondi, 2000)
- Scalability is a **prerequisite** for elasticity, but it **does not consider temporal aspects** of how fast, how often, and at what granularity scaling actions can be performed

Elasticity	Scalability
Increase or reduce the capacity to fulfil an increase or decrease in workload	Increase the capacity to fulfil an increase of workload
Available resources matches current demands	Available resources may exceed current demands to fulfil future demands
Automatic adaptation to workload increase and decrease	Adapt only to workload increase by provisioning the resources incrementally
Short-term adjustments to demand needs	Long-term adjustment to demand needs

Over and Under Provisioning

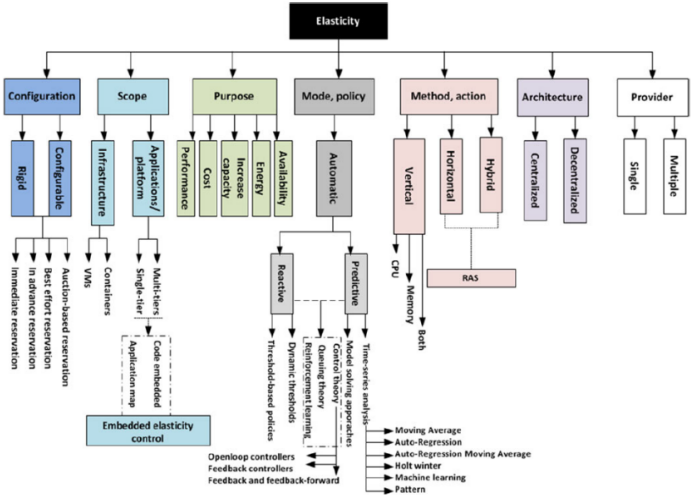
- The main reason for cloud elasticity is to avoid either **over-provisioning** and **under-provisioning** of resources
- **Under-provisioning**
  - Resources provided are smaller than the required resources
  - Service performance degradation
  - Possible violation of SLAs
- **Over-provisioning**
  - Resources provided are greater than the required resources
  - Extra and unnecessary costs
  - Idle resources

Elasticity Measurements

- **Speed**
  - The **speed of scaling up** is defined as the time it takes to switch from an under-provisioned state to an optimal or overprovisioned state
  - The **speed of scaling down** is defined as the time it takes to switch from an overprovisioned state to an optimal or under-provisioned state
  - The speed of scaling up/down **does not correspond directly to** the technical resource **provisioning/deprovisioning time**
- **Precision**
  - The absolute **deviation** of the **current amount of allocated resources** from the **actual resource demand**

Elasticity Taxonomy

Elasticity Mechanisms Classification



(Al-Dhuraibi et al., 2018)

Elasticity: Configuration

- Elasticity Configuration represents the method of the first or initial reservation of resources with a Cloud Provider
- Rigid vs. Configurable
  - **Rigid**
    - ✓ Predefined resource limit
    - ✓ Resources rarely meets the demand
  - **Configurable**
    - ✓ Customer can choose the resources
- Reservation
  - On-demand
  - In advance
  - Best effort
  - Auction-based

(Al-Dhuraibi et al., 2018)

Elasticity: Scope

- The range of components controlled by the elasticity mechanism, i.e., infrastructure or application / platform level
- Elasticity solutions are mostly dedicated to the infrastructure level providing support to
  - Stateless applications
  - Client-server and Web-Queue-Worker applications
  - MapReduce applications
- **Embedded Elasticity**: Elasticity solution has an internal knowledge of the application
  - **Application Map**: Elasticity controller has a complete map of the application components and instances
  - **Code Embedded**: Elasticity controller is embedded in the application source code

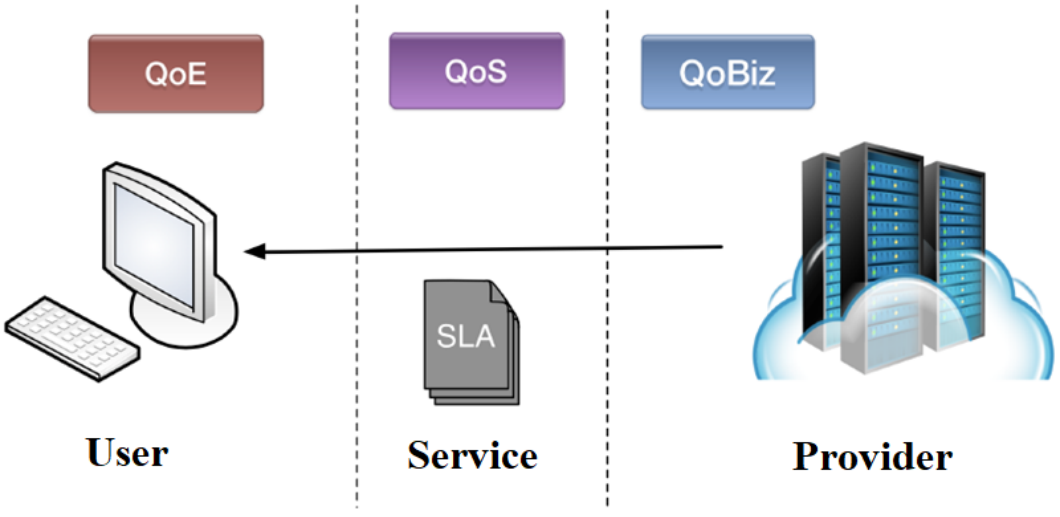
(Al-Dhuraibi et al., 2018)

Elasticity: Purpose

- Elasticity has different purposes
  - Improve performance
  - Increase resource capacity
  - Save energy
  - Reduce cost
  - Ensure availability
- Different perspectives elasticity objectives can be looked at
  - Quality of Service (QoS)
  - Quality of Experience (QoE)
  - Quality of Business (QoBiz)

(Al-Dhuraibi et al., 2018)

Quality Management in the Cloud



(Al-Dhuraibi et al., 2018)

Quality of Service (QoS)

- A technology-centric quality metrics
- Allocate resources to minimize Service Level Objectives (SLOs) violations
- Used QoS metrics
  - Response time
  - CPU utilization
  - Start-up time

Quality of Experience (QoE)

- According to subjective tests
- QoS is not sufficient to ensure user satisfaction
- Quality of experience (QoE) is a subjective metric that expresses quality of service as perceived by the user
- QoE is influenced by several factors
  - The context of use
  - The personality and preferences of the user
  - The system used to consume the service

Quality of Business (QoBiz)

- A provider-centric quality metric
- Express metrics associated to the profit of supplier
  - service price
  - revenue per user
  - revenue per transaction
  - budget
- A good QoBiz expresses the prosperity of the provider
- Decides the pricing policy of the suppliers: rate per hour, rate per month, etc.

Elasticity: Mode or Policy

- Refer to needed interactions to perform elasticity actions
- **Reactive Mode**
  - Static thresholds (or role-condition-actions)
    - ✓ e.g., If CPU utilization greater 80% for more than 5 minutes
  - Dynamic thresholds
- **Proactive Mode**
  - Time series analysis
  - Model solving mechanism
- **Hybrid**
  - Reinforcement Learning
  - Control Theory
  - Queue Theory

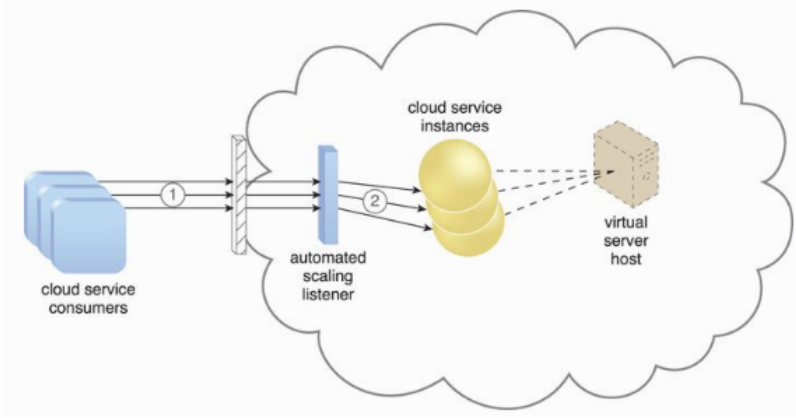
(Al-Dhuraibi et al., 2018)

Elasticity: Method

- The method elasticity is deployed
- **Horizontal Scaling**
  - Add/remove instances fluctuating workload
  - Load balancers are used to distribute the load among different instances
- **Vertical Scaling**
  - Modify resource sizes for an instance at runtime
  - Migration may be required if
    - ✓ there is no enough resources on the host machine
    - ✓ host machine is too crowded

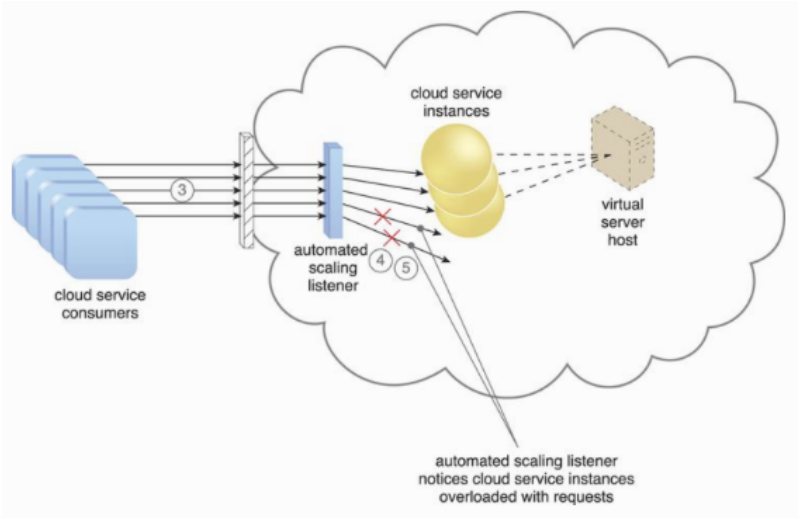
(Al-Dhuraibi et al., 2018)

Elasticity: Method



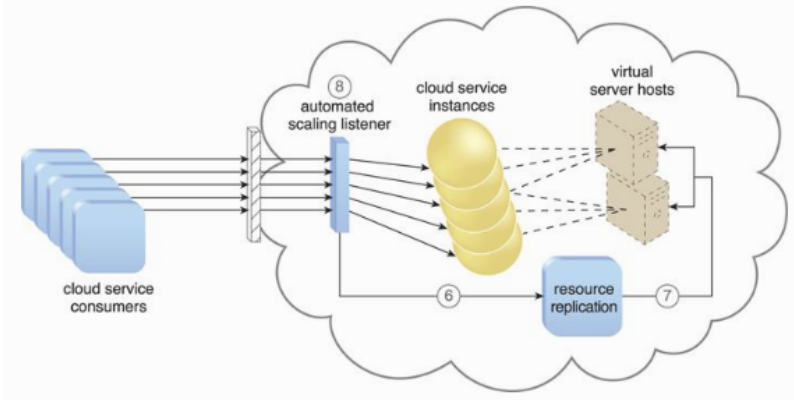
(Erl et al., 2013)

Elasticity: Horizontal Scalability



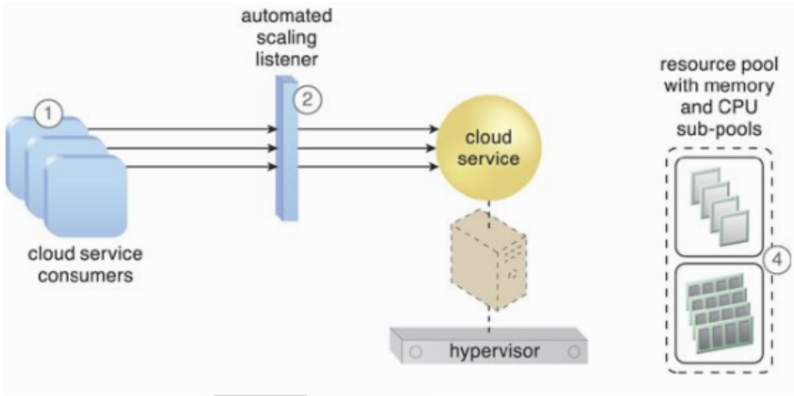
(Erl et al., 2013)

Elasticity: Horizontal Scalability



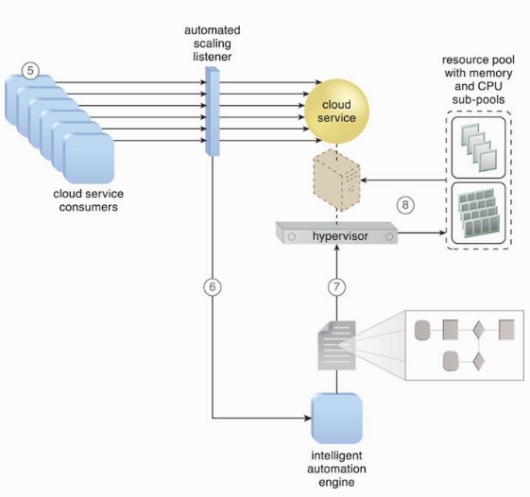
(Erl et al., 2013)

Elasticity: Vertical Scalability



(Erl et al., 2013)

Elasticity: Vertical Scalability



(Erl et al., 2013)

Elasticity: Architecture

- The architecture of the elasticity management can be either
  - **Centralized**
    - ✓ Single elasticity controller
  - **Decentralized**
    - ✓ Many elasticity controllers or application managers
    - ✓ Key master component

Elasticity: Provider

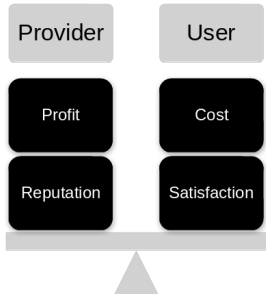
- Elastic solutions can be applied to a single or multiple cloud providers
  - **Single**
    - ✓ Public or private with one or multiple regions or data centers
  - **Multiple**
    - ✓ Public, private or hybrid clouds from multiple providers

Elasticity Management

Elasticity Management: Definition

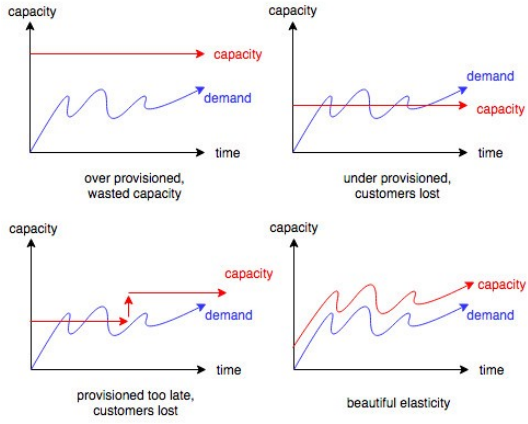
Elasticity Management is an ***user-centric problem****. It concerns finding an optimal* ***tradeoff*** *between the* ***satisfaction of the user and business goals****. It may be achieved by using* ***resource provisioning*** *alone or in conjunction with* ***scheduling****.*

(Najjar et al., 2015)



Elasticity Management: Processes

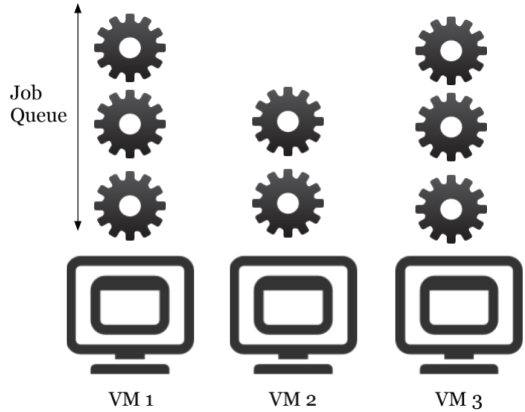
- Two related process
  - Scheduling
  - Resource Provisioning



Source: https://pablo-iorio.medium.com/elasticity-does-not-equal-scalability-246bd9b3c128

Scheduling

- Prediction model that performs the load forecasting based on historical data
- Take into account
  - Different VM sizes
  - VM job queue
  - The requirement of each job
  - Billing mechanism



Resource Provisioning

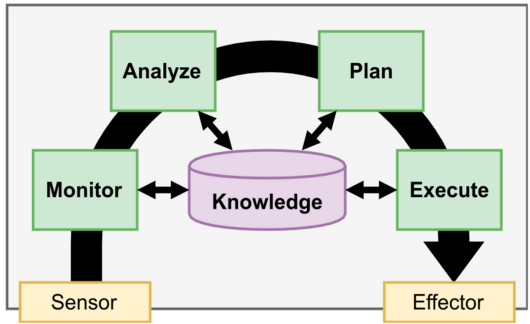
- A mechanism of allocating or releasing resources from the cloud as required
- Communication with the Cloud Provide via API
- Good resource provisioning algorithm should
  - Take the demand into account
  - Choose the best combination of resources

Resource Provisioning Methods

- **Demand-driven**: This method adds or removes computing instances based on current utilization level of the allocated resources.
- **Event-driven**: This method adds or removes machine instances on a specific time event.
  - Works better for seasonal or predicted events such as Christmas
  - Results in minimal loss of the QoS, if the event is predicted correctly
- **Popularity-driven**: Searches the Internet for popularity of certain applications and creates instances by popularity demand.
  - Anticipates increase traffic by popularity
  - Results in minimal loss of the QoS, if the event is predicted correctly

Elasticity Manager Model

- Elasticity Manager is usually seen as a autonomic system, with sensors, actuators, and reasoning unit
- Design based on the MAPE-K control loop proposed by IBM (Monitor, Analyze, Plan & Execute)



Elasticity Manger Components

1. **Monitors**
   - Sensors of the system
   - Collect information about key performance and workload indicators
     - ✓ e.g., CPU utilization, average queue length, etc.
2. **Resource Allocator**
   - Executes the provisioning actions decided by the elasticity manger
     - ✓ e.g., Uses the API to request more resources from the Cloud Provider

Elasticity Manger Components

3. **Load Balancer**
   - Distribute requests among the existing VMs
   - If the elasticity management is of complex type, the load-balancer is replaced by a cloud-oriented scheduler
4. **Manager**
   - Compile the information from the sensors
   - Analyze these data
   - Take decisions
   - The decision is then sent to the resource allocator

Example : Auto Scaling Feature of Amazon EC2

- Auto Scaling allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define
- Ensure that the number of Amazon EC2 instances used increases seamlessly during demand spikes to maintain performance, and decreases automatically during demand lulls to minimize costs
- Auto Scaling is particularly well suited for applications that experience hourly, daily, or weekly variability in usage
- Auto Scaling is enabled by Amazon CloudWatch and available at no additional charge beyond Amazon CloudWatch fees

Example : Amazon CloudWatch

- A web service that provides real-time monitoring to Amazon EC2
  - CPU Virtualization
  - No information about memory, disk, etc.
- Accessible via Command-Line tools and APIs
- Charge by the number of monitoring instances
- Enable the Auto Scaling feature to dynamically add or remove Amazon EC2 instances

Service Level Agreement

Service Level Agreement (SLA)

An explicit statement of expectations and obligations that exist in a business relationship between two organizations: the service provider and the customer

- A Service Level Agreement is a document that defines the quality of service required between a provider and a client
- There are several languages of SLA
  - WS-Agreement by OGF (Open Grid Forum)
  - WSLA by IBM
- An SLA is a document of a legal nature
  - if the SLA is not satisfied, there will be penalties to pay

SLA in the Cloud

- Cloud providers often offer several SLAs because involve several actors
- (Ideally) SLA between a cloud provider and users must be negotiated
- (Usually) Service providers offer a single contract for everyone

Service Level Objective (SLO)

- An SLA is composed of a set of SLOs
- An SLO specifies the limits of the acceptable
  - Availability > 95%
  - response time t < 5 millisecond
- A SLA violation is generated when an SLO is not respected
  - The penalty is often proportional to the violation
- Often SLOs are about non-functional requirements

Cloud Non-Functional Requirements

- **Availability**: an essential requirement, the service must be always present even in case of breakdown
  - e.g., 99.99% during work days, 99.9% for nights/weekends)
- **Reliability**: disaster Recovery expectations
  - System operation must be consistent
  - Data must be stored in redundancy (e.g., RAID systems)
  - Multiple VMs must provide the same tasks
- **Performance**
  - Maximum response times
- **Location of the data**: Consistent with local legislation
- **Portability of the data** (e.g., ability to move data to a different provider)

References

- Al-Dhuraibi, Y., Paraiso, F., Djarallah, N., & Merle, P. (2018). Elasticity in Cloud Computing: State of the art and research challenges. IEEE Transactions on Services Computing, 11(2), pp. 430-447.
- Bondi, A. B. (2000). Characteristics of scalability and their impact on performance. In Proceedings of the 2nd international workshop on Software and performance (WOSP '00). New York, NY, USA, pp. 195–203. DOI: 10.1145/350391.350432.
- Erl, T., Mahmood, Z., & Puttini, R. (2013). Cloud Computing: Concepts, Technology & Architecture. Upper Saddle River, NJ: Prentice Hall.
- Herbst, N., Kounev, S., & Reussner, R. (2013). Elasticity in cloud computing: What it is, and what it is not. In Proceedings of the International Conference on Autonomic Computing. pp. 23-27.
- Najjar, A., Serpaggi, X., Gravier, C., & Boissier, O. (2014). Survey of elasticity management solutions in cloud computing. In Continued Rise of the Cloud. Berlin: Springer, pp. 235–263.