Clouds computing: Some introductory concepts

Renaud Lachaize & Thomas Ropars
Univ. Grenoble Alpes
M2 GI
January 2024

Cloud computing – The roots

- Cluster and Grid computing
- Utility computing (“pay and use”)
- Service-oriented architectures (SOA)
- Virtualization technologies
- Autonomic computing
- DevOps (“you build it, you run it”)

Cloud computing – A definition from NIST (1/9)

- NIST: National Institute of Standards and Technology (USA)
- Definition coined in 2011. Probably the most commonly used definition (although some of its parts have become incomplete today).
- Available from: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf
- We will cite it extensively. (emphasis added)

> "Cloud computing is a model for enabling ubiquitous, convenient, **on-demand network access to**
> a **shared pool of configurable computing resources** (e.g., networks, servers, storage, applications, and services)
> that can be **rapidly provisioned and released** with minimal management effort or service provider interaction.
> This cloud model is composed of **five essential characteristics**, three service models, and four deployment models."

Cloud computing – A definition from NIST (2/9)

5 essential characteristics (1/3)

1) On-demand self-service:

"A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically **without requiring human interaction with each service provider**."

2) Broad network access:

"**Capabilities are available over the network** and accessed **through standard mechanisms** that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations)."

Cloud computing – A definition from NIST (3/9)

5 essential characteristics (2/3)

3) Resource pooling:

"The provider's computing resources are pooled to serve multiple consumers using **a multi-tenant model**, with **different physical and virtual resources dynamically assigned** and reassigned according to consumer demand.

There is a sense of location independence in that the customer generally has **no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter).**

Examples of resources include storage, processing, memory, and network bandwidth."

Cloud computing – A definition from NIST (4/9)

5 essential characteristics (3/3)

4) Rapid elasticity:
"Capabilities can be elastically provisioned and released, in some cases automatically, to **scale rapidly outward and inward** commensurate with demand.
To the consumer, the **capabilities available for provisioning often appear to be unlimited** and can be appropriated in any quantity at any time."

5) Measured service:
"Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts).
Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service."

Cloud computing – A definition from NIST (5/9)

3 service models (1/3)

1) Software as a service (SaaS):

"The **capability provided to the consumer is to use the provider's applications running on a cloud infrastructure**.

The applications are **accessible from various client devices** through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface.

The **consumer does not manage or control the underlying cloud infrastructure** including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings. "

Cloud computing – A definition from NIST (6/9)

3 service models (2/3)

2) Platform as a Service (PaaS):

"**The capability provided to the consumer is to deploy** onto the cloud infrastructure **consumer-created or acquired applications** created **using programming languages, libraries, services, and tools supported by the provider.**

The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment."

Cloud computing – A definition from NIST (7/9)

3 service models (3/3)

3) Infrastructure as a Service (IaaS):

"The capability provided to the consumer is to **provision processing, storage, networks, and other fundamental computing resources**
where **the consumer is able to deploy and run arbitrary software**, which can include operating systems and applications.

The consumer does not manage or control the underlying cloud infrastructure
but **has control over operating systems, storage, and deployed applications**;
and possibly limited control of select networking components (e.g., host firewalls)."

Cloud computing – A definition from NIST (8/9)

4 deployment models (1/2)

1) Private cloud:

"The cloud infrastructure is provisioned for **exclusive use by a single organization comprising multiple consumers** (e.g., business units).

It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises."

2) Community cloud:

"The cloud infrastructure is provisioned for **exclusive use by a specific community of consumers from organizations that have shared concerns**

(e.g., mission, security requirements, policy, and compliance considerations).

It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises."

Cloud computing – A definition from NIST (9/9)

4 deployment models (2/2)

3) Public cloud:

"The cloud infrastructure is **provisioned for open use by the general public.**
It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. **It exists on the premises of the cloud provider**."

4) Hybrid cloud:

"The cloud infrastructure is a **composition of two or more distinct cloud infrastructures (private, community, or public)** that remain unique entities, but are bound together by standardized or proprietary technology that **enables data and application portability** (e.g., cloud bursting for load balancing between clouds)."

Beyond IaaS, PaaS and SaaS: Other service models

- “Anything/Everything as a Service” (also sometimes named “XaaS”).

- A few examples:
  - Hardware as a Service
  - Monitoring as a Service
  - Various compute-level abstractions that sit between IaaS and PaaS. For example (discussed later) :
    - Containers as a Service (CaaS)
    - Functions as a Service (FaaS)

Scalability

- (Also known as **"scaling"**)
- *A generic term related to **the expected behavior of a system/application when the input load and/or the available resources evolve.***
- Two types of considerations: "weak scaling" and "strong scaling"
- **"Weak scaling":**
  - "**Can I achieve more work per unit of time** if I increase the quantity of resources in my system?" *(And if so in which proportions?)*
  - Here, "more work" can mean "more requests" or "more/larger data items" or "more complex tasks" …
- **"Strong scaling':**
  - "**Can I complete the same work in a shorter amount of time** if I increase the quantity of resources in my system?" *(And if so in which proportions?)*

Elasticity

- A notion related to scaling (mainly “weak scaling”) and to the flexible resource reservation/billing model of cloud computing.
- Sometimes also called **“autoscaling”**
- Corresponds to **the ability of a system to dynamically adjust (i.e., grow or shrink) the allocated resources according to the current input load.**
- Example:
  - “My Web-based e-commerce application should handle any request in less than 500ms, regardless of the total number of requests that it is currently handling.”
  - However, you do not want to dimension your system for the worst case: wasted resources would lead to high costs.
- Ideally, an elasticity manager must be:
  - Quick to react (can be predictive and/or reactive)
  - Accurate
  - Fully automated

Vertical vs. horizontal scaling

- In practice, there are two main approaches to achieve (weak or strong) scaling: vertical vs. horizontal scaling.

- **Vertical scaling**, also known as **"scaling up"**:
  - In essence, consists in **using more powerful machines**.
- **Horizontal scaling**, also known as **"scaling out"**:
  - In essence, consists in **using a larger number of machines**.

- Both approaches can be combined.
  - But in the design of a large-scale system, one of the two will generally dominate.
  - Resorting to horizontal scaling is almost unavoidable for some requirements because it introduces distribution, which enables replication & geo-placement.

Vertical scaling

- **Idea:** **Replace existing machine(s) with more powerful one(s)**
- **Examples:**
  - Faster CPUs (better microarchitecture, higher clock frequency)
  - More CPU cores
  - Faster RAM
  - Greater RAM capacity
  - Faster disks
  - Faster networking
  - Better hardware accelerators
- **Pros:**
  - Relatively simple
  - May work well in some cases
- **Cons:**
  - Limited scalability potential
  - Non-linear increase of hardware costs

Horizontal scaling

- **Idea: Increase the number of machines in the system**
  - Typically using commodity servers (i.e., no specific/expensive hardware required)
  - Coordination between machines is performed at the software level, using a conventional network
  - Warning: the network must be dimensioned accordingly to avoid bottlenecks
- **Pros:**
  - **Better scalability potential** (esp. for very large scales)
  - **Better cost-effectiveness** (esp. for very large scales)
  - Distribution **enables fault-tolerance (replication) and performance optimizations**
- **Cons:**
  - Using a large number of machines **increases the probability of failures.**
  - **More efforts required** from humans to achieve good performance and reliability:
    - Middleware/infrastructure designers
    - System administrators
    - Application developers (often)

Vertical vs. horizontal scaling in the cloud

- Historically, over the last 20 years, cloud computing platforms have mainly been been built according to the horizontal scaling approach.
- **Our lectures will mostly focus on techniques related to horizontal scaling.**
- **However:**
  - **Today, many cloud computing platforms (especially public cloud providers) provide solutions for both models.**
    - Greater diversity of available hardware (CPUs, RAM, disks, networking, accelerators).
    - Possibility to reserve dedicated machines.
  - **In some circumstances, vertical scaling is just simpler and more cost-effective.**
    - See some of the next slides for examples.
  - For some types of cloud services, the internal design is completely abstracted for the end-user.

Distributed systems – Main techniques

- Software infrastructures for the Cloud are **inherently distributed systems**.
- Distributed systems extensively rely on **two techniques**: replication and partitioning.
  - These two approaches are often combined.
  - These two approaches can be applied to processing and storage.
- **Replication**
  - Keeping a copy of the same service/data on multiple nodes, potentially in different geographical locations.
  - Provides redundancy, which is useful for fault-tolerance and can also help improve performance.
- **Partitioning** (also known as "**sharding**")
  - Splitting a data set into multiple partitions, and routing a given request to the service in charge of the appropriate partition.
  - Helpful for performance (load balancing), and possibly for fault tolerance (partial availability).