Week 4: Distributed and Cloud Computing

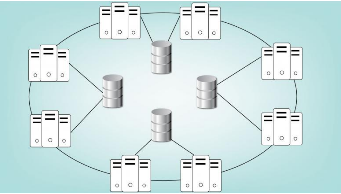
Tien-Nam Le

tien-nam.le@ens-lyon.fr
perso.ens-lyon.fr/tien-nam.le/su

Sciences U Lyon

Outline

1. **Distributed Computing**



2. **Cloud Computing**

Part I. Distributed Computing

Two distributed computing frameworks

hadoop

VS

APACHE Spark™

Problems

Problem 1 – SUM() SQL query:
An international fashion company has many shops, each shop keep data of every sale. Objective: query total revenue from sale of each product.
Naive solution: Run through the whole database one-by-one record.

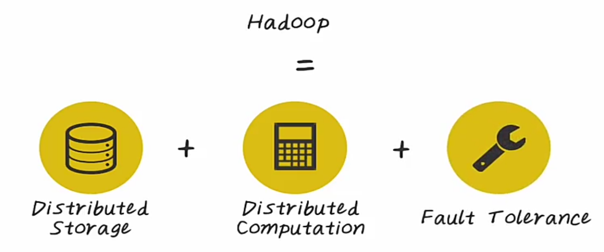
Problem 2 – Machine learning problems
Eg: Linear regression, k-nearest-neighbor of a large dataset.
Naive solution: Run through the every data point one-by-one.

Question: Can we do faster?

Another Question. What happens if there are errors, failures during the computation process?

Hadoop

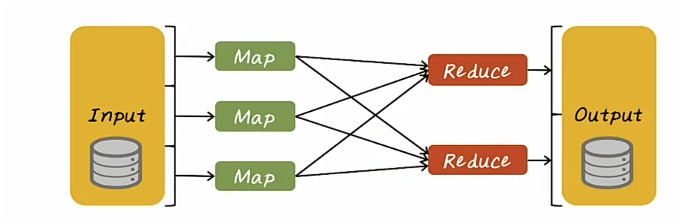
- Created in 2004 by Google, open source



Two key concepts in Hadoop

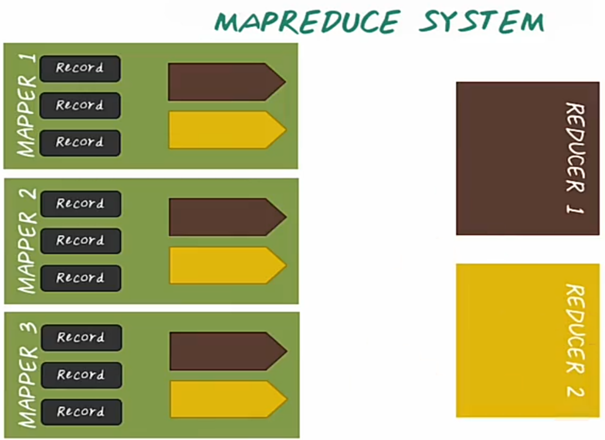
- **Map-Reduce** to process data
- **HDFS** to store data

Map-Reduce System

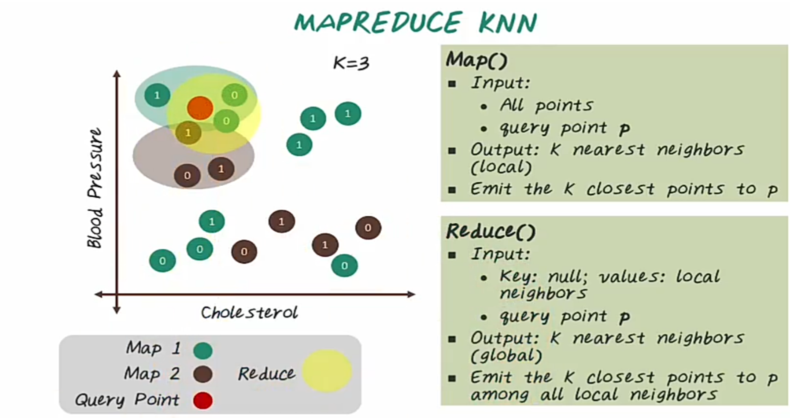


- Data flow in one direction (acyclic graph).
- Data flow from one disk to another disk inside the cluster.

Map-Reduce System



Example of Map-Reduce



Hadoop distributed file system HDFS

(blackboard)

Main concepts:

- data node
- name node

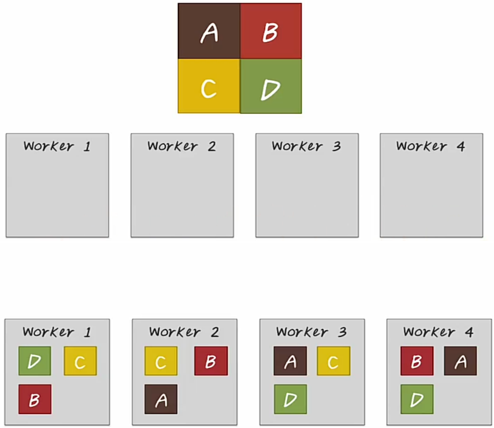
Question: Is there any problem in the case

- network failure
- disk failure on data node
- not all data node are used
- block sizes differ
- disk failure on name node

Fault tolerance

Question. How to fix failure data node?

Answer. Duplicate every block to 3 nodes at random. Name node will control all the location of blocks.



Fault tollerance

Question. What happens if name node dies?

Answer. All data are lost.

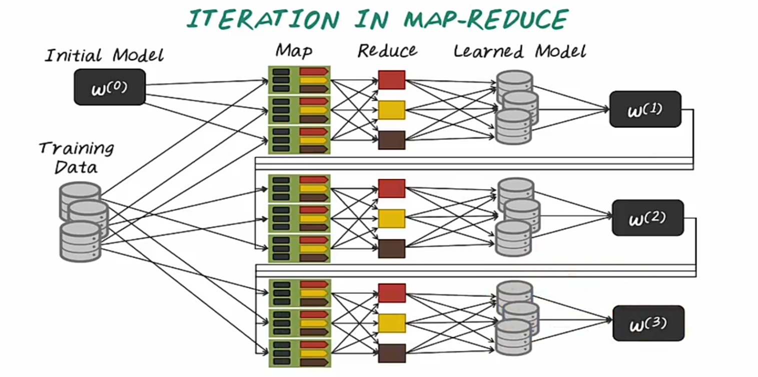
Solution. Create two name nodes: one *active* and one *standby*. If active dies, standby will become active and another name node is added.

Benefits of Hadoop

1. Free (open-source)
2. Distributed Processing: data is stored in many different nodes
3. Fault Tolerance: default 3 replicas of each block are stored across the cluster.
4. Reliability: Automatically recover from failure.
5. High Availability: Data is available and accessible due to multiple copies of data.
6. Scalability: new hardware and new data can be easily added.
7. Data Locality: all computation is done and data only move locally in the cluster.

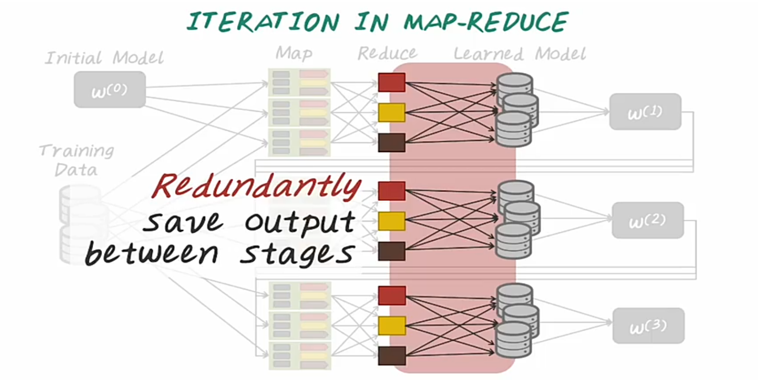
Disadvantages of Hadoop

Inefficient for applications that repeatedly reuse data. Especially iterative algorithms (Machine learning, Graph Analysis).
Example: k-means clustering algorithm.



Disadvantages of Hadoop

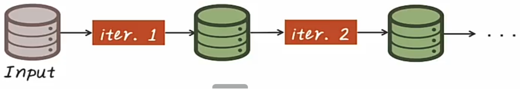
Example: *k*-means clustering algorithm.



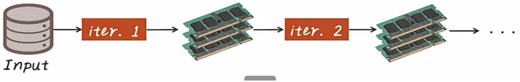
Recall previous week: The most time-consuming steps are reading/writing to hard disk, not computation.

Solution: Spark

Instead of writing to hard drives at every step,

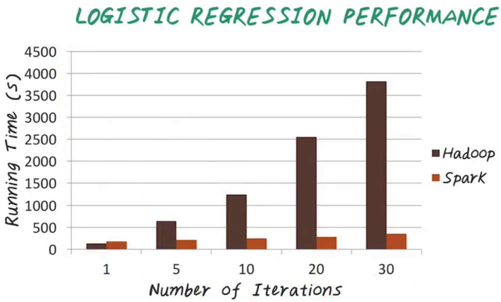


Spark keeps data in main memory



Spark

- Created in 2012 by Berkeley University, open source
- Significantly faster than Hadoop



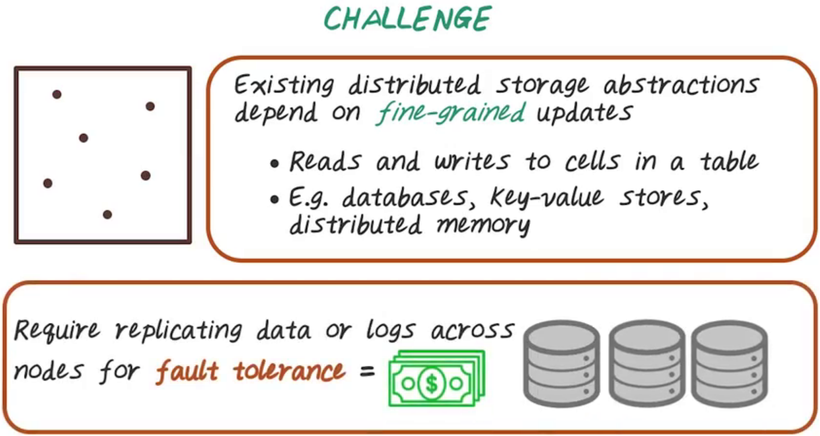
Problems of Spark

Problem 1. RAMs are much more expensive than hard drives

Problem 2. RAMs are efficient but much *less stable* than hard drives $\rightarrow$ *less fault-tolerant*.

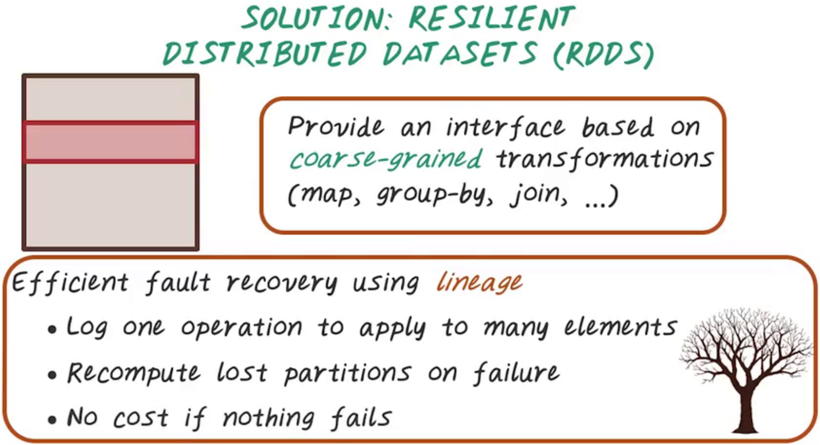
Question. How to make Spark fault-tollerant?

Challenge


CHALLENGE
Existing distributed storage abstractions depend on fine-grained updates
• Reads and writes to cells in a table
• E.g. databases, key-value stores, distributed memory
Require replicating data or logs across nodes for fault tolerance =
$

Resilient Distributed Datasets (RDD)

RDD is the key concept in Spark.

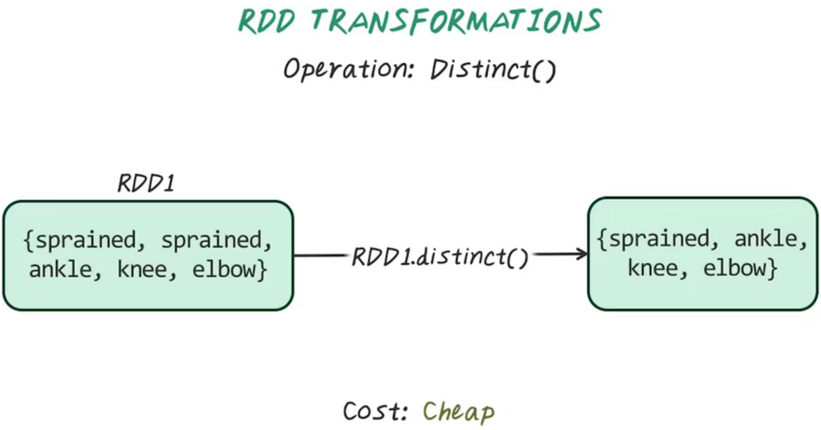


RDD Operations

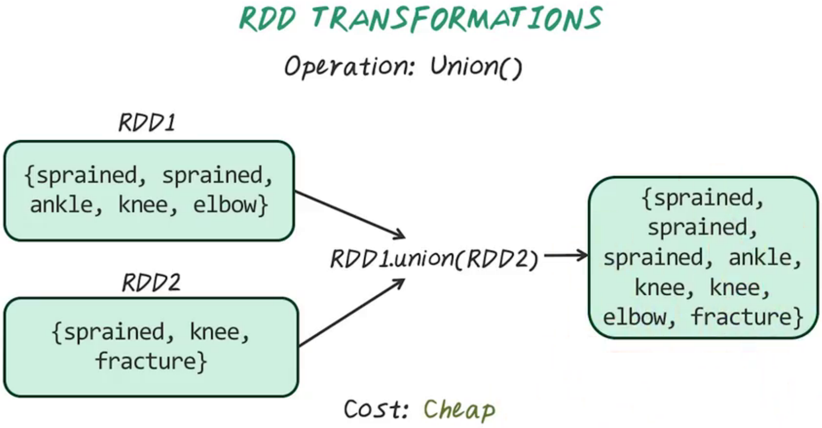
SPARK OPERATIONS

Transformations (define a new RDD)	map filter sample groupByKey reduceByKey sortByKey	flatMap union join cogroup Cross mapValues
Actions (return a result to driver program)	collect reduce Count save lookupKey	

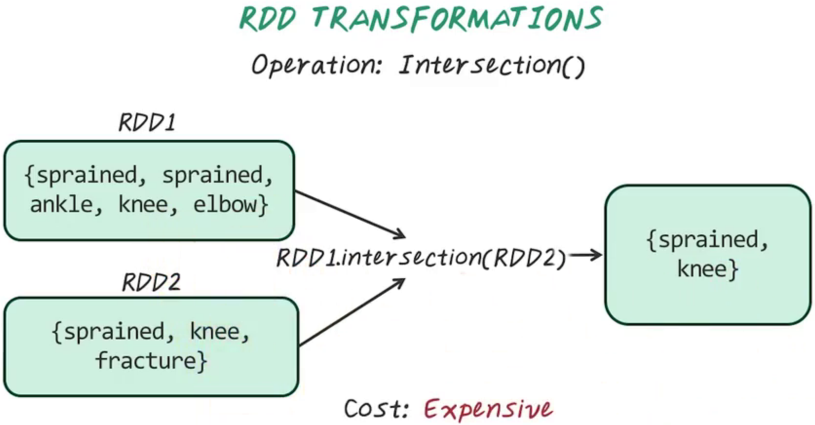
Some RDD Transformations



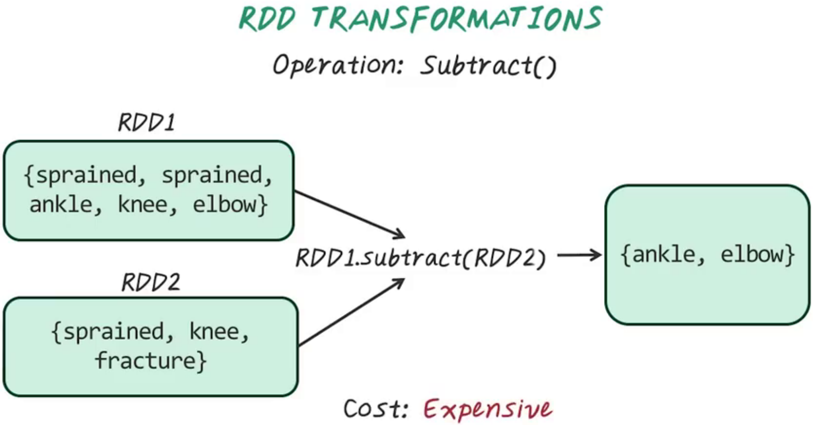
Some RDD Transformations



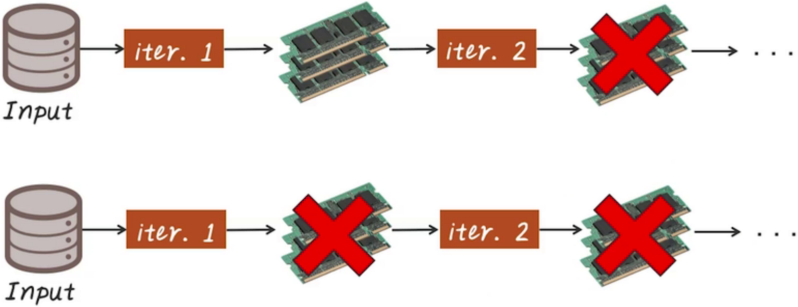
Some RDD Transformations



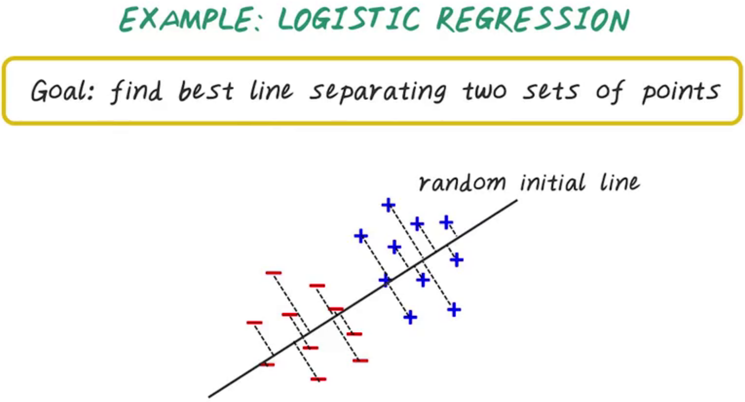
Some RDD Transformations



RDD recovery



Example of using Spark



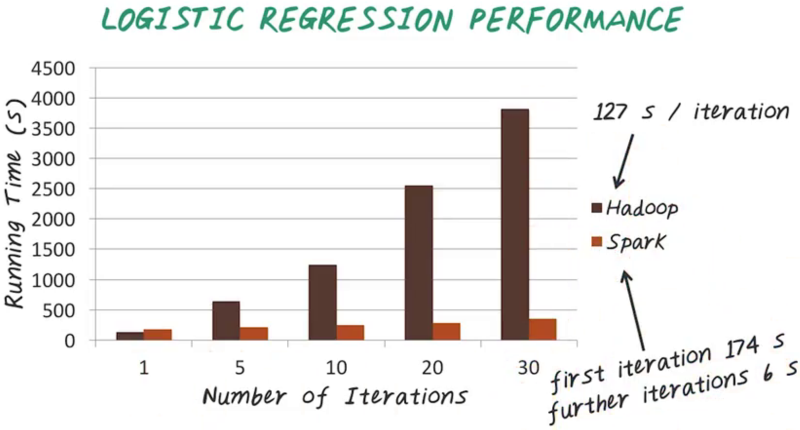
Example of using Spark

EXAMPLE: LOGISTIC REGRESSION

```
val data = spark.textFile(...).map(readPoint).cache()

var w = Vector.random(D)

for (i <- 1 to ITERATIONS) {
  val gradient = data.map(p =>
    (1 / (1 + exp(-p.y*(w dot p.x))) - 1) * p.y * p.x
  ).reduce(_ + _)
  w -= gradient
}

println("Final w: " + w)
```

Repeated MapReduce steps to do gradient descent

Example of using Spark



Conclusion

- Use Hadoop when
  - computation is one-way
  - budget is important
- Use Spark when
  - computation is iterative
  - speed is important

Part II. Cloud Computing

Contents

1. **Cloud Computing**
2. **Types of cloud services:** Public Cloud, Private Cloud, and Hybrid Cloud.
3. **Levels of cloud services:** IaaS, PaaS, and SaaS.
4. **Study cases**

Why the cloud?

In a nutshell: With Netflix, who buys DVD anymore?

Problems:

- Companies need to plan and build their new system in detail.
- Include: buildings and rooms, servers, networking devices, storage devices, extra security, cooling systems, power supplies.
- Usually overestimate or underestimate the computing needs for their new systems → waste of money.

Ideal solution:

- Don't need to estimate cost and gambling financially on building the system
- Quickly grow and shrink your computing infrastructure based on your actual needs.
- Don't need to deal with security, cooling, software...

Cloud Computing

Cloud computing: a pool of services such as storage, networking, and computers remotely via Internet.

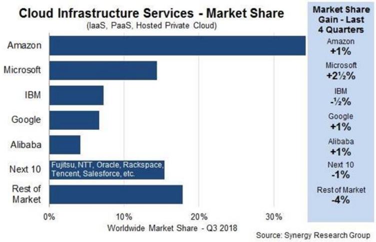
Benefit:

- Immediate access to various computing resources (networks, servers, storage, GPU, ...)
- Out of the box integration
- Elastic, on-demand, pay-per-use $\rightarrow$ reduce cost
- Minimal management effort: no security, no cooling...

Disadvantage:

- Need to trust cloud provider for your data.

Cloud computing services



Cloud Data Center

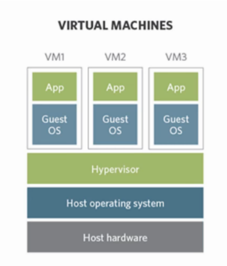
An Amazon cloud data center

Technologies

Two key technologies: **virtualization** and high broadband **Internet**

Virtualization: a technique to share a single physical resources dynamically among multiple customers. Example:

- A big hard drive can be virtualized into several virtual small hard drives for different customers
- Run operation systems (Windows, Linux) on virtual machines (vmware, virtualbox) instead of installing directly on hardware.



Types of cloud services

1. Public cloud

2. Private cloud

3. Hybrid cloud

Public Cloud

Public cloud is the most popular type of cloud model.

- Hardware is located at cloud provider, deliver via Internet
- Hardware is shared by several customers i.e. you and other customers may run on the same hardware.

Benefit

- *Low costs:* pay only for the service they use.
- *No maintenance:* provider provides the maintenance.
- *Near-unlimited scalability:* On-demand resources.
- *High reliability:* A vast network of servers ensures against failure.

Disadvantages

- *noisy neighbor effect:* performance fluctuations (CPU, RAM,...) depends on the use of other customers on the same machine.

Private Cloud

Private cloud: To avoid noisy neighbor effect, customer want to get exclusive access to hardware.

Location: can be physically located at customer's or provider's site.

Used by government agencies, financial institutions, any other mid-to large-size business want to have better control.

Benefits

- *Flexibility.* The organization can customize its cloud environment to meet specific business needs.
- *High security.* Resources are not shared with others, so higher levels of control and security are possible.
- *Good scalability.* Private clouds still afford the scalability and efficiency of a public cloud.

Hybrid cloud

Hybrid cloud: data and applications can move between private and public clouds for greater flexibility and more deployment options.

Example:

- Use the public cloud for high-volume, lower-security needs such as web-based email
- Use private cloud for sensitive, business-critical operations such as financial reporting.

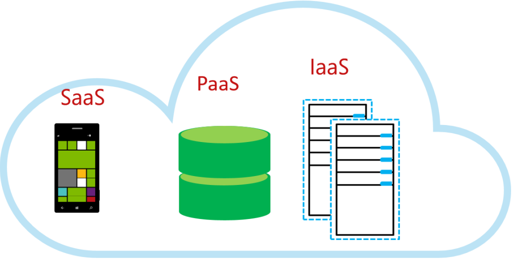
Advantage:

- *Control.* Still control sensitive data.
- *Flexibility.* Access additional resources in the public cloud when needed.
- *Ease-of-use.* Transitioning between private and public cloud are easy.

3 levels of cloud services

Cloud Service Types

- **SaaS – a complete software solution**
- **PaaS – a platform of services for hosting a custom solution**
- **IaaS – a way to run virtual servers in the cloud with full control**



3 levels of cloud services

Level 1. Software as a Service (SaaS)

- Fully-formed software applications, delivered as cloud-based services.
- Customer subscribe to the service and use the application, normally through a web browser or by installing a client-side application.

Example: Office 365, Google Apps, Dropbox, Netflix

Advantage:

- Easy access to applications without the need to install.
- No worry about issues such as updating applications and maintaining

3 levels of cloud services

Level 2: Platform as a Service (PaaS)

- Provide resources and platform software on which developers can build their own applications.
- Include: operating system (OS), storage and compute capacity, and functional services for custom applications.

Example: Google App Engine, AWS Elastic Beanstalk, Windows Azure

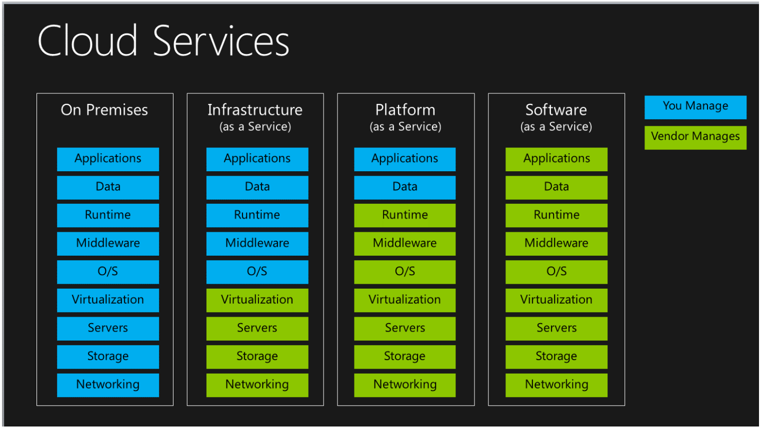
3 levels of cloud services

Level 3: Infrastructure as a Service (IaaS)

- Provide virtualized server, network, and storage infrastructure components
- The maintenance of the OS and all software are up to customer.

Example: Amazon Web Services, Microsoft Azure

3 levels of cloud services



3 basic levels of cloud services: differ by which controlled by **cloud provider**, and which by the **customer**.

Quick questions

Q1. You want to develop a PHP Web application using the PaaS cloud service. Who should be responsible for providing a web server with PHP interpreter and libraries installed?

- Provider
- Customer

Q2. You are using an IaaS cloud service with Windows virtual servers, and Microsoft has released a critical security patch. Who should install the patch on your server?

- Provider
- Customer

Q3. You are using a SaaS cloud service providing access to a Customer Relationship Management (CRM) package. Who is responsible for installing a new version of the package?

- Provider
- Customer

Study case 1.

Suppose you want to have a set of

- 10 Linux systems with 4GB RAM each
- 2 Windows systems with 8GB each

to deploy your software.

Generally, a cloud provider

- creates the respective VMs in the background
- puts them in the same internal network
- provide you account, password $\rightarrow$ allowing you to access them

Lead service: Amazon EC2 package. Some benefits:

- secure and robust functionality.
- 99.99% uptime.
- specialized hardware for workloads: high graphics capability, high input/output (I/O), High Performance Computing (HPC).

Study case 2.

Suppose you want to train your neural networks with deep learning

- Using high-end GPU
- only several hours per day.

Lead service: Amazon SageMaker package. Some benefits:

- up to 8 NVIDIA V100 GPU in parallel
- Most deep learning libraries available (TensorFlow, Keras, PyTorch...)
- Pay-per-minute use
- label data by human in demand (Amazon Mechanical Turk)

Study case 3.

Suppose you want to have large storage ($\geq$ 20 TB) for your company (web server, big data, content distribution...)

Lead service: Amazon S3 package. Some benefits:

- fast access and high throughput
- boost access in Availability Zones: period with high frequent access
- several types: frequent access, infrequent access, archive (rarely access) with different prices

Study Case 4.

Suppose you want to do computation on big data efficiently $\rightarrow$ using Hadoop or Spark.

Lead service: Amazon EMR package. Some benefits:

- Process data directly from Amazon S3
- Support Hadoop and Spark
- Amazon EMR Notebooks, based on Jupyter Notebooks, provide a managed environment for data scientists, analysts, and developers.

Summary Cloud Computing

1. **Cloud Computing**
2. **Types of cloud services:** Public Cloud, Private Cloud, and Hybrid Cloud.
3. **Levels of cloud services:** IaaS, PaaS, and SaaS.
4. **Study cases**