Cloud

工海系 b06501018 朱紹勳

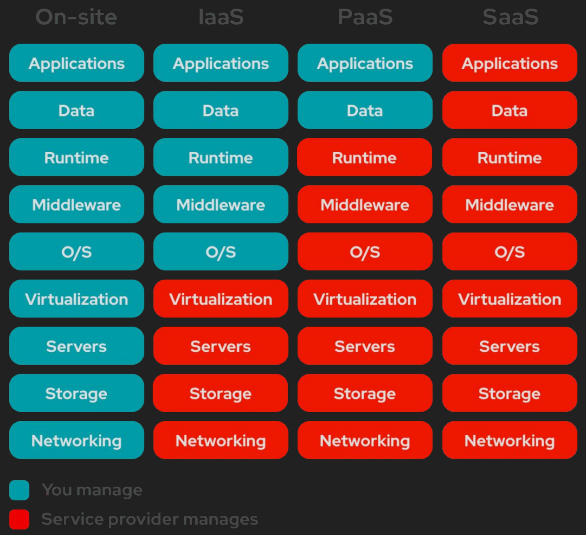
Intro of Cloud Service(I)

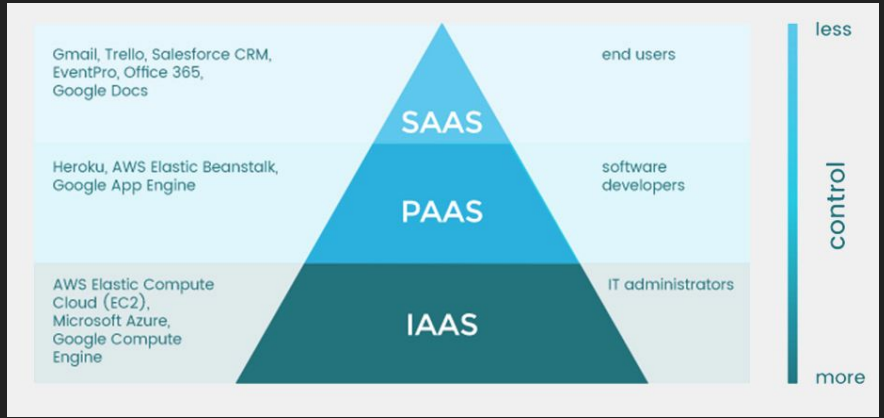
Scalability, On Demand, Reliability

- Software as a service(SaaS)
  - Application
- Platform as a service (PaaS)
  - Dev Tools, Interface
- Infrastructure as a service (IaaS)
  - Physical Asset

MiddleWare
- OS vs. Middleware

Public, Private, Hybrid Cloud





Virtualization (I)

Focus on IaaS (Infrastructure)

To deal with different requirment on hardware & OS & middleware

(Abstraction of physical hardware resource,
then combine to different enviroment)

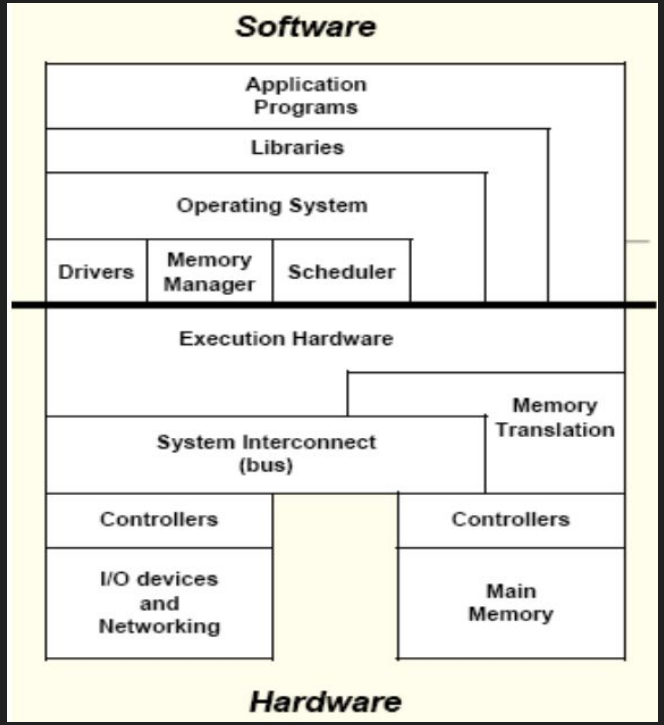
Divide to different level:

1. System Level
2. Machine Level
3. Os Level
4. Library Level

Virtual Machine(computation)

Virtual Storage

Virtual Network(communication)



Virtualzation (II)

	System Virtual Machine	Process Virtual Machine
Emulation	*Transmeta Crusoe* *(Emulate x86 on VLIW cpu)*	*Multi-processing system*
Virtualization	*XEN, KVM, VMWare* *(x86 virtualization software)*	*JVM, Microsoft CLI* *(High level language virtualization)*

Virtual Machine :

- Emulator vs. Virtualization
- Process VM vs. System VM (lower level)
- Full-Virtualization vs. Para-Virtualization (modify guest OS)

Container :

- Light weight virtualization
- Not need to install another OS, the container only works for application
- Consists of : Image, Container, Repository
- Example : Docker

Kubernetes : Automating deployment & managing containeriaed application

Serverless Computing

Tradtional Model → VM → Container → Serverless

Even more LIghtweight and Portable compare to Container

FaaS (Function), between the layer of Platform and Application Service

(User does not need to manage server, and only need to focus on writing code)

The operations are event driven (send asychonous invocation record to trigger)

Example : AWS Lambda, Google Cloud Function, Microsoft Azure Function

Distributed Computing

MapReduce : Google

Hadoop : Yahoo (Open Source)
Implement in Java but provide Hadoop Streaming Interface for other language

The Mapreduce can be:

1. A Programming Model, 2. An Implementation, 3. System Architecture

For Programming Model, it implement Divide & Conquer technic

Iterate → Extract → Shuffle & Sort → Aggregate → Output
(Map) (Reduce)

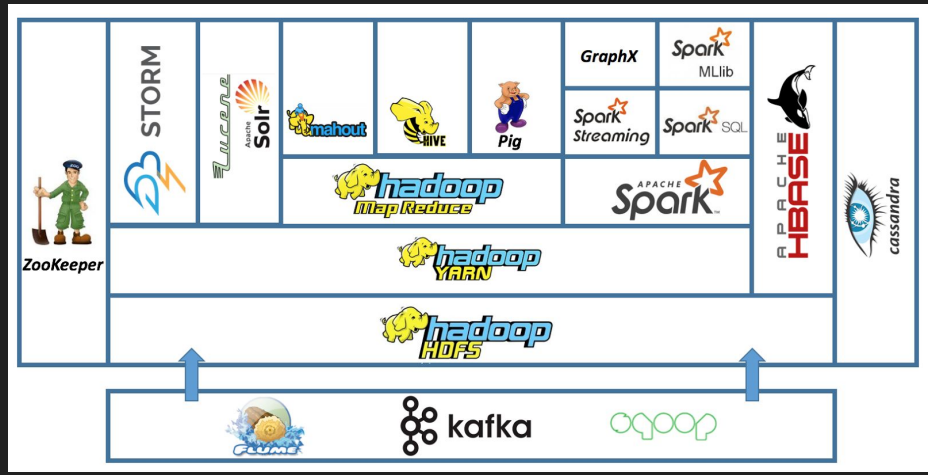
That is, we only have to write Mapping and Reducing part

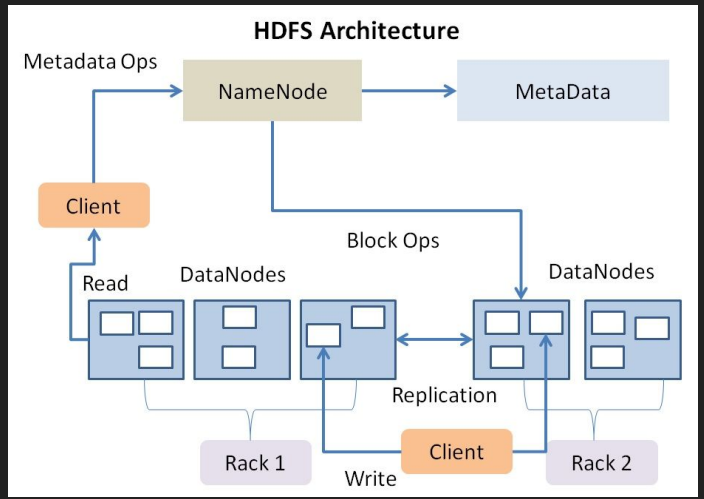
Hadoop & its Ecosystem

Consist of three parts:

1. Computation: MapReducce
2. Storage : HDFS (hadoop distributed)
3. Resource : YARN (container)

hive : Query Language, table store on HDFS
pig : Dataflow (script) Language





MapReduce in Action