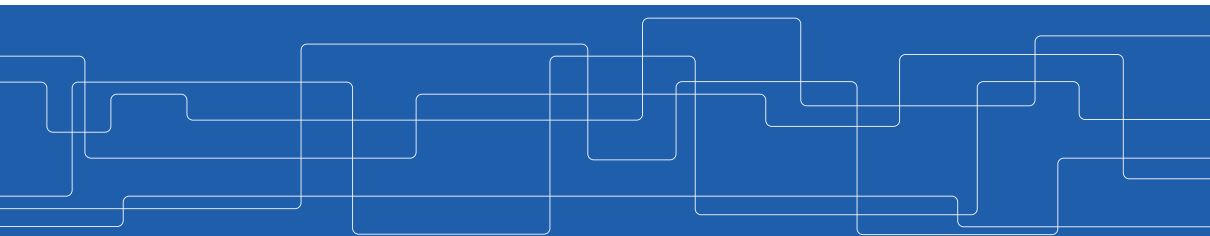




# An Introduction to Data Intensive Computing

Amir H. Payberah  
payberah@kth.se  
27/08/2019





# Course Information





## Course Objective

- ▶ Provide students with a solid foundation for **understanding** large scale distributed **systems** used for **storing and processing** massive data.
- ▶ Cover a wide variety of advanced topics in **data intensive computing platforms**, i.e., the frameworks to **store and process** big data.





## Intended Learning Outcomes (ILOs)

- ▶ **ILO1**: explaining **fundamental concepts** of **data-intensive computing platforms**, and also explain **how such platforms work**.



## Intended Learning Outcomes (ILOs)

- ▶ **ILO1:** explaining **fundamental concepts** of **data-intensive computing platforms**, and also explain **how such platforms work**.
- ▶ **ILO2:** **storing and retrieving** data in **distributed stores**, e.g., distributed file systems or NoSQL databases.



## Intended Learning Outcomes (ILOs)

- ▶ **ILO1:** explaining **fundamental concepts** of **data-intensive computing platforms**, and also explain **how such platforms work**.
- ▶ **ILO2:** **storing and retrieving** data in **distributed stores**, e.g., distributed file systems or NoSQL databases.
- ▶ **ILO3:** **processing** different types of data, e.g., structured, streaming and graph, using **data-intensive computing platforms**, such as Spark.



## Intended Learning Outcomes (ILOs)

- ▶ **ILO1:** explaining **fundamental concepts** of **data-intensive computing platforms**, and also explain **how such platforms work**.
- ▶ **ILO2:** **storing and retrieving** data in **distributed stores**, e.g., distributed file systems or NoSQL databases.
- ▶ **ILO3:** **processing** different types of data, e.g., structured, streaming and graph, using **data-intensive computing platforms**, such as Spark.
- ▶ **ILO4:** building **advanced** applications using data-intensive platforms, and make **scalable** applications on a **cluster** of computers.





# The Course Assessment

- ▶ **Task1**: the **reading** assignments (P/F)



# The Course Assessment

- ▶ **Task1:** the **reading** assignments (P/F)
- ▶ **Task2:** the **review** questions (P/F)



# The Course Assessment

- ▶ **Task1**: the **reading** assignments (P/F)
- ▶ **Task2**: the **review** questions (P/F)
- ▶ **Task3**: the **lab** assignments (A-F)





# The Course Assessment

- ▶ **Task1**: the **reading** assignments (P/F)
- ▶ **Task2**: the **review** questions (P/F)
- ▶ **Task3**: the **lab** assignments (A-F)
- ▶ **Task4**: the final **project** (A-F)



# The Course Assessment

- ▶ **Task1**: the **reading** assignments (P/F)
- ▶ **Task2**: the **review** questions (P/F)
- ▶ **Task3**: the **lab** assignments (A-F)
- ▶ **Task4**: the final **project** (A-F)
- ▶ **Task5**: the final **exam** (A-F)



## How Each ILO is Assessed?

	Task1	Task2	Task3	Task4	Task5
<b>ILO1</b>	x	x			x
<b>ILO2</b>			x		
<b>ILO3</b>			x		
<b>ILO4</b>			x	x	



## Task1: The Reading Assignments (P/F)

- ▶ To read and review scientific papers.



## Task1: The Reading Assignments (P/F)

- ▶ To read and review scientific papers.
- ▶ Choose two papers from the given pool of papers (or propose yourself).



## Task1: The Reading Assignments (P/F)

- ▶ To read and review scientific papers.
- ▶ Choose two papers from the given pool of papers (or propose yourself).
- ▶ Review the papers, and write a report for each one.



## Task1: The Reading Assignments (P/F)

- ▶ To read and review scientific papers.
- ▶ Choose two papers from the given pool of papers (or propose yourself).
- ▶ Review the papers, and write a report for each one.
- ▶ Write a two-page report about the motivation, the contribution, and the solution of the paper and also write their strong/weak points.



## Task2: The Review Questions (P/F)

- ▶ One review question **per week**.
- ▶ Questions about the **lectures**.





## Task2: The Review Questions (P/F)

- ▶ One review question **per week**.
- ▶ Questions about the **lectures**.
- ▶ To pass each review question, you should answer correctly to **75% of questions**.



## Task3: The Lab Assignments (A-F)

- ▶ Two lab assignments.
- ▶ How to **store/retrieve** and **process** data using the platforms presented in the course.



## Task4: The Final Project (A-F)

- ▶ One final project.
- ▶ Implement an **advanced application** on processing massive data on a **cluster of computers**.
- ▶ **Proposed by students** and confirmed by the teacher.



## Task5: The Final Exam (A-F)

- ▶ A number of **questions** from different parts of the course.
- ▶ Assesses the **theoretical knowledge** of students about covered platforms in the course.

# How to Submit the Assignments?

- ▶ Through the [Canvas](#) site.
- ▶ Students will work in **groups of two** on all the [Tasks 1-4](#).





## The Final Grade

- ▶ The **final grade** is the **average** of the **two labs**, **the project**, and the **final exam**.



## The Final Grade

- ▶ The **final grade** is the **average** of the **two labs**, **the project**, and the **final exam**.
- ▶ To compute it, map **A-F** to **5-1**, and take the average.



## The Final Grade

- ▶ The **final grade** is the **average** of the **two labs**, **the project**, and the **final exam**.
- ▶ To compute it, map **A-F** to **5-1**, and take the average.
- ▶ The floating values are **rounded up**, if they are **more than half**, otherwise they are **rounded down**.
  - E.g., **3.6** will be rounded to **4**, and **4.2** will be rounded to **4**.



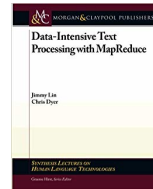


# The Final Grade

- ▶ The **final grade** is the **average** of the **two labs**, **the project**, and the **final exam**.
- ▶ To compute it, map **A-F** to **5-1**, and take the average.
- ▶ The floating values are **rounded up**, if they are **more than half**, otherwise they are **rounded down**.
  - E.g., **3.6** will be rounded to **4**, and **4.2** will be rounded to **4**.
- ▶ The half grades will be **rounded up**, if you submit the assignments **before their deadlines**, otherwise they will be **rounded down**.

# The Course Material

- ▶ Mainly based on [research papers](#).
- ▶ We also cover the following books.





## The Course Web Page

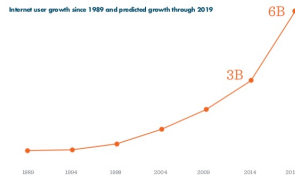
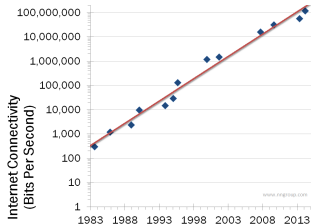
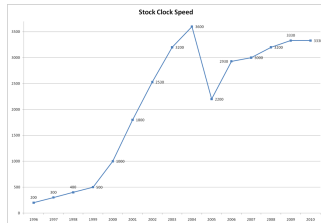
<https://id2221kth.github.io>



# The Course Overview

# Cloud Computing and Big Data

- ▶ The main trends:
  - Computers not getting any faster
  - Internet connections getting faster
  - More people connected to the Internet





# Cloud Computing and Big Data

## Conclusion

Move the **computation** and **storage** of **big data** to the **cloud**!

Cisco predicts that by 2020, **92%** of IT market workloads will be processed by **cloud data centers**, while only **8%** will be processed by **traditional data centers**.

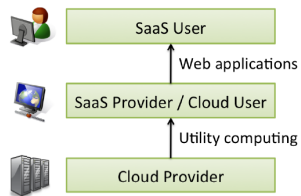


# Cloud Computing

# Cloud Computing Definition

► **Cloud Computing** refers to both:

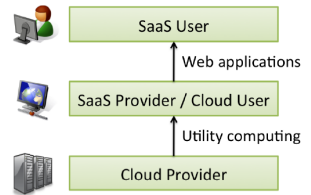
1. The **applications** delivered as **services** over the Internet
2. The **hardware and systems software** in the datacenters that provide those **services**





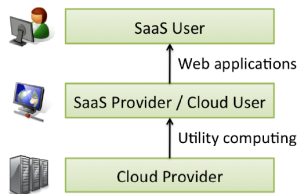
# Cloud Computing Definition

- ▶ **Cloud Computing** refers to both:
  1. The **applications** delivered as **services** over the Internet
  2. The **hardware and systems software** in the datacenters that provide those **services**
- ▶ The **services**: called **Software as a Service (SaaS)**



# Cloud Computing Definition

- ▶ **Cloud Computing** refers to both:
  1. The **applications** delivered as **services** over the Internet
  2. The **hardware and systems software** in the datacenters that provide those **services**
- ▶ The **services**: called **Software as a Service (SaaS)**
- ▶ The datacenter **hardware and software** is called **cloud**





▶ The **NIST** definition:

- Five **characteristics**
- Three **service models**
- Four **deployment models**

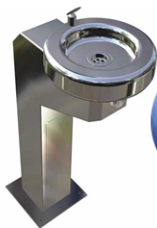
**NIST**

**National Institute of Standards and Technology**  
Technology Administration, U.S. Department of Commerce



# Cloud Characteristics

# Cloud Characteristics



On-demand  
self-service



Ubiquitous  
network  
access



Location  
transparent  
resource  
pooling



Rapid  
elasticity

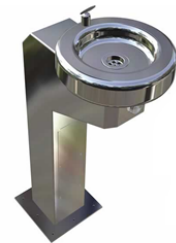


Measured  
service with  
pay per use

[<http://aka.ms/532>]

## Cloud Characteristics - On-demand Self-Service

- ▶ A consumer can **independently** provision **computing capabilities** without **human interaction** with the service provider.



On-demand  
self-service

## Cloud Characteristics - Ubiquitous Network Access

- ▶ Available over the **network**
- ▶ Accessed through mobile phones, laptops, ...



Ubiquitous  
network  
access

## Cloud Characteristics - Resource Pooling

- ▶ Provider's computing resources are pooled to serve consumers
- ▶ Location transparent

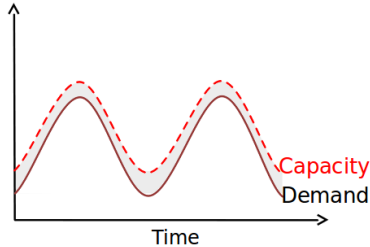
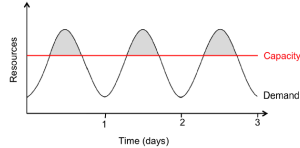
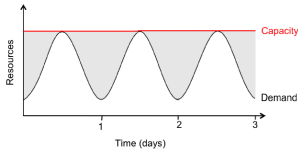


Location  
transparent  
resource  
pooling



# Cloud Characteristics - Rapid Elasticity

- ▶ **Capabilities** can be rapidly and **elastically provisioned**, in some cases automatically.



Rapid elasticity

## Cloud Characteristics - Measured Service

- ▶ **Resource usage** can be monitored, controlled, and reported providing transparency for both the **provider** and **consumer**.



Measured  
service with  
pay per use



# Cloud Service Models

## Cloud Service Models



SaaS



PaaS



IaaS

[<http://aka.ms/532>]

- ▶ Assume, you just moved to a city and you are looking for a place to live.



► What is your choice?



- ▶ What is your choice?
  - Build a **new house**?
  - Buy an **empty house**?
  - Live in a **hotel**?



- ▶ Let's build a **new house!**





- ▶ Let's build a **new house!**
- ▶ You can **fully control** everything you like your new house to have.
- ▶ But that is a **hard work.**



- ▶ What if you buy an **empty house**?



- ▶ What if you buy an **empty house**?
- ▶ You can **customize** some part of your house.
- ▶ But never change the original architecture.



- ▶ How about living in a [hotel](#)?



- ▶ How about living in a **hotel**?
- ▶ Living in a hotel will be a good idea if the only thing you care is about enjoying your life.
- ▶ There is **nothing you can** do with the house except living in it.



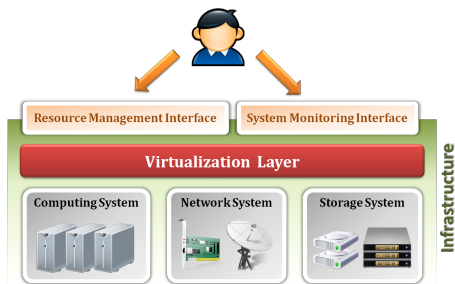
Let's translate it to Cloud Computing



## Service Models

- ▶ Infrastructure as a Service (**IaaS**): similar to **building a new house**.
- ▶ Platform as a Service (**PaaS**): similar to **buying an empty house**.
- ▶ Software as a Service (**SaaS**): similar to **living in a hotel**.

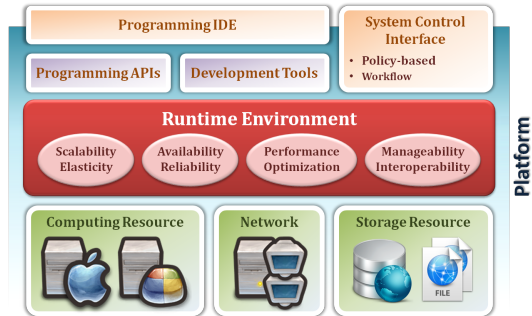
- ▶ Vendor provides **resources**, e.g., processing, storage, network, ...
- ▶ Consumer is provided customized **virtual machines**.
- ▶ **Example: Amazon Web Services (EC2 instances and S3 storage)**





# PaaS

- ▶ Vendor provides hardware and **development environment**.
- ▶ **Example: Google app engine**

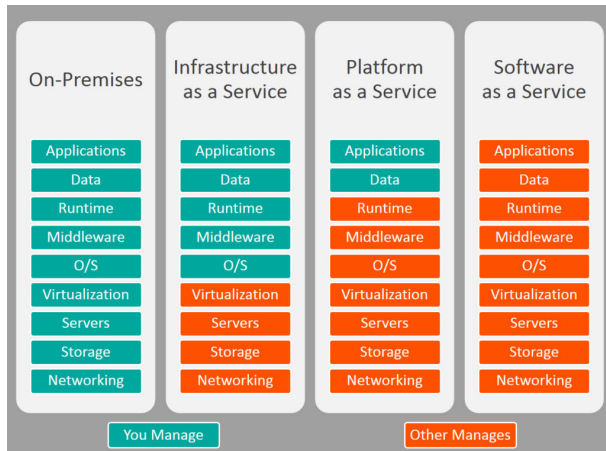


# SaaS

- ▶ Vendor provides **applications** accessed over the network.
- ▶ Example: Gmail, Github



# IaaS - PaaS - SaaS



[<https://goo.gl/xMko1z>]



# Deployment Models

# Deployment Models




**VS**






 Publically Shared  
Virtualised Resources

 Privately Shared  
Virtualised Resources

Supports multiple  
customers 

Cluster of dedicated  
customers 

 Supports connectivity  
over the internet

 Connectivity over  
internet, fibre and private network 

Suited for less  
confidential information 

 Suited for secured  
confidential information  
& core systems 

[<https://goo.gl/fWmcGK>]

# Public Cloud Infrastructure Vendors

- ▶ Amazon Web Services (AWS)
- ▶ Microsoft Azure
- ▶ Google Cloud Platform
- ▶ IBM Bluemix
- ▶ ...





## Main Services

- ▶ Computing
- ▶ Storage
- ▶ Database
- ▶ Big data analytics
- ▶ ...

# Computing Services

- ▶ Virtual machines
- ▶ Container services
- ▶ Serverless compute



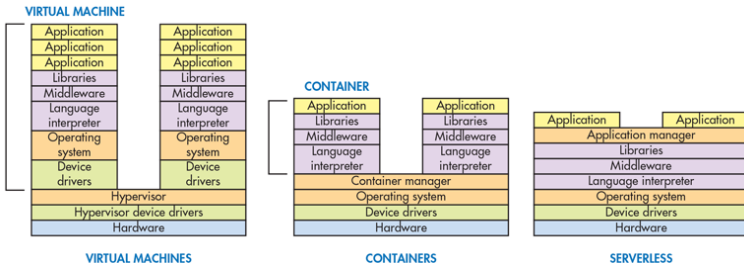
VM



Container



Serverless





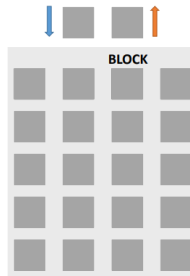
# Storage Services

- ▶ File storage
- ▶ Block storage
- ▶ Object storage

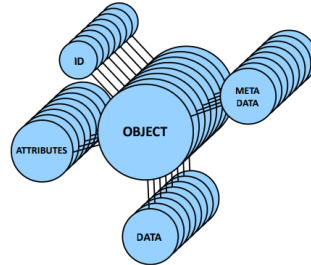
**File Storage**



**Block Storage**

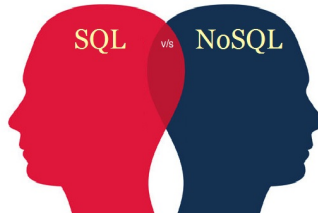


**Object Storage**



# Database Services

- ▶ Relational Database Management Services (RDBMS)
- ▶ NoSQL databases
- ▶ In-Memory data services



# Big Data Analytics

- ▶ Big Data Managed Cluster-as-a-Service
- ▶ Data warehouse
- ▶ Data streaming
- ▶ Data queuing





# Big Data



**“THAT’S your Ark for the Big Data flood? Noah, you will need a lot more storage space!”**

[<https://www.kdnuggets.com/2012/12/cartoon-preparing-for-big-data-flood.html>]

# What is Big Data?



[<https://www.sue-anderson.com.au/index.php/2017/08/18/cursing-curious-work>]



# Big Data

Big data is the data characterized by 4 key attributes: volume, variety, velocity and value.

**ORACLE®**



# Big Data

Big data is the data characterized by 4 key attributes: volume, variety, velocity and value.

**Buzzwords**

**ORACLE®**





# Big Data in Simple Words



**DevOps Borat**

@DEVOPS\_BORAT

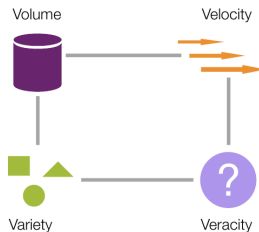
Small Data is when is fit in RAM.  
Big Data is when is crash because  
is not fit in RAM.

2/6/13, 8:22 AM

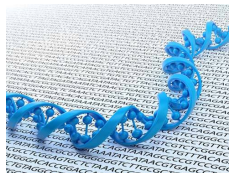


# The Four Dimensions of Big Data

- ▶ **Volume:** data size
- ▶ **Velocity:** data generation rate
- ▶ **Variety:** data heterogeneity
- ▶ This 4th **V** is for **V**acillation:  
Veracity/**V**ariability/**V**alue



# Big Data Sources



# How Much Data?

## 2019 *This Is What Happens In An Internet Minute*



# How To Store and Process Big Data?



# Problem

- ▶ Traditional platforms **fail** to show the expected performance.
- ▶ Need **new systems** to **store and process** large-scale data

## Scale Up vs. Scale Out (1/2)

- ▶ Scale **up** or scale **vertically**: adding **resources** to a **single** node in a system.
- ▶ Scale **out** or scale **horizontally**: adding **more nodes** to a system.



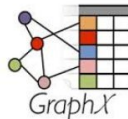


## Scale Up vs. Scale Out (2/2)

- ▶ Scale **up**: more **expensive** than scaling out.
- ▶ Scale **out**: more challenging for **fault tolerance** and **software development**.



APACHE  
**HBASE**



Storm



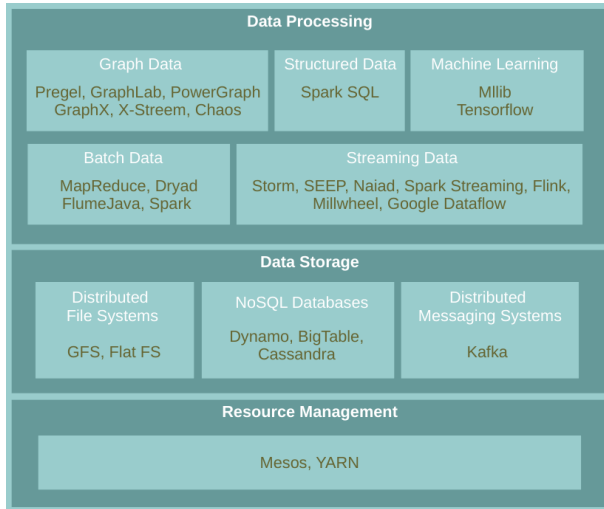
**HIVE**

**S4** distributed stream  
computing platform



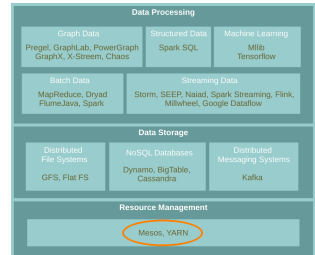
Google Cloud Platform

# Big Data Stack



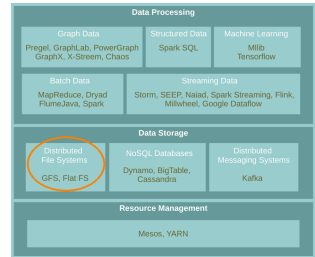
# Resource Management

- ▶ Manage resources of a cluster
- ▶ Share them among the platforms
- ▶ Mesos, YARN, Borg, ...



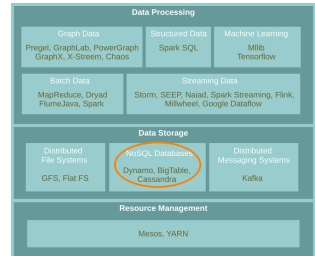
# Data Storage - Distributed File Systems

- ▶ Store and retrieve **files** on/from distributed disks
- ▶ **GFS, HDFS, FlatFS, ...**



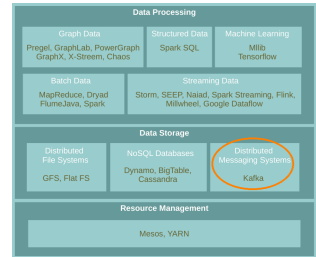
# Data Storage - NoSQL Databases

- ▶ BASE instead of ACID
- ▶ BigTable, Dyanamo, Cassandra, ...



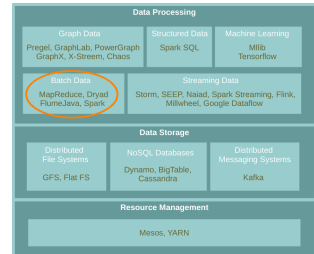
# Data Storage - Messaging Systems

- ▶ Store streaming data
- ▶ Kafka, Flume, ActiveMQ, ...



# Data Processing - Batch Data

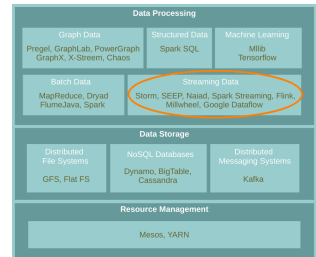
- ▶ Process data-at-rest
- ▶ Data-parallel processing model
- ▶ MapReduce, FlumeJava, Spark, ...





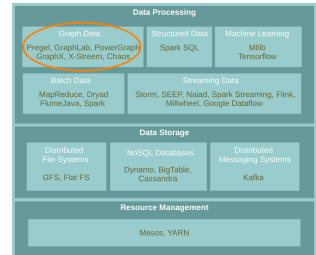
# Data Processing - Streaming Data

- ▶ Process data-in-motion
- ▶ Storm, Flink, Spark Streaming, ...



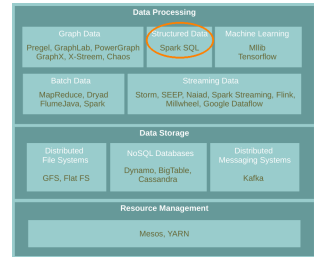
# Data Processing - Linked Data (Graph)

- ▶ Graph-parallel processing model
- ▶ Vertex-centric and Edge-centric programming model
- ▶ Pregel, GraphLab, GraphX, ...



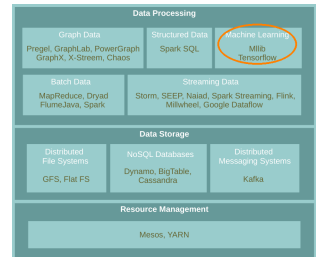
# Data Processing - Structured Data

- ▶ Take advantage of **schemas** in data to process
- ▶ **Hive, Spark SQL, ...**



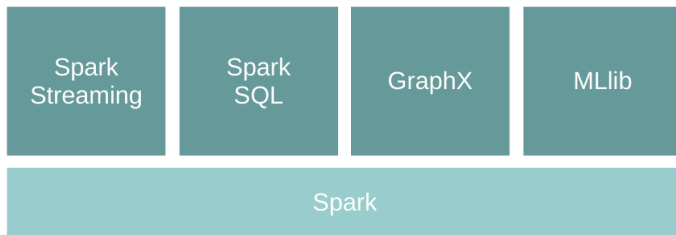
# Data Processing - Machine Learning

- ▶ Data analysis, e.g., supervised and unsupervised learning
- ▶ Mahout, TensorFlow, MLlib, ...



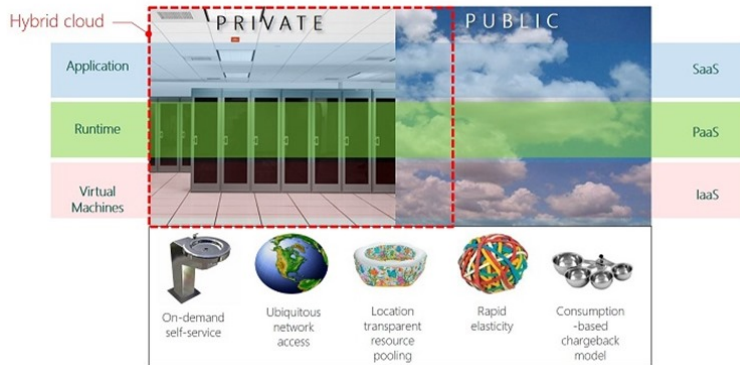


# Spark Processing Engine



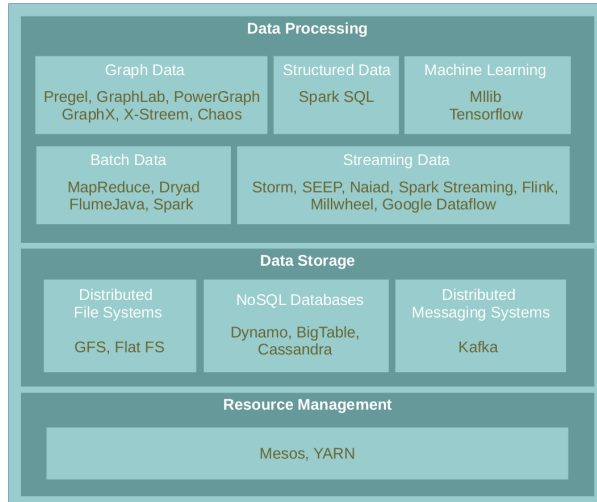
# Summary

# Summary



[<http://aka.ms/532>]

# Summary







## References

- ▶ D. Sikeridis et al., A Comparative Taxonomy and Survey of Public Cloud Infrastructure Vendors, arXiv preprint arXiv:1710.01476, 2017.
- ▶ A. Fox et al., Above the clouds: A berkeley view of cloud computing, UCB/EECS 28.13 (2009): 2009.
- ▶ P. Mell et al., The NIST definition of cloud computing, 2011.

Questions?