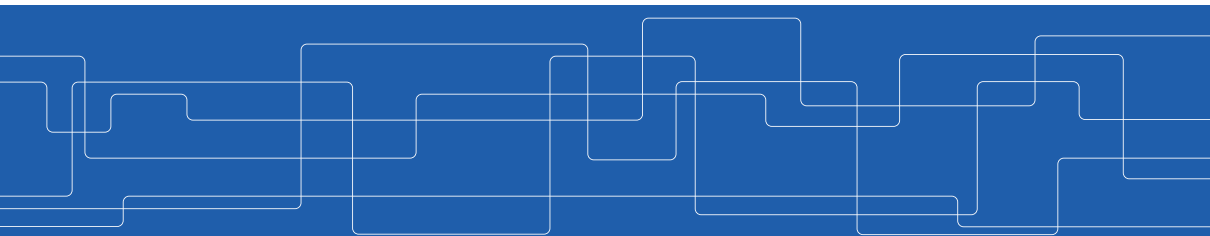




An Introduction to Data Intensive Computing

Amir H. Payberah
payberah@kth.se
2021-08-31





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 **review** questions.



The Course Assessment

- ▶ **Task1**: the **review** questions.
- ▶ **Task2**: the **lab** assignments.



The Course Assessment

- ▶ **Task1**: the **review** questions.
- ▶ **Task2**: the **lab** assignments.
- ▶ **Task3**: the final **project**.



How Each ILO is Assessed?

	Task1	Task2	Task3
ILO1	x	x	x
ILO2		x	x
ILO3		x	x
ILO4		x	x



Task1: The Review Questions (A-F)

- ▶ One review question **per week**.
- ▶ Questions about the **lectures**.
- ▶ The review questions are **graded (A-F)**.



Task2: The Lab Assignments (A-F)

- ▶ Two lab assignments: **source code** and **oral presentation**.



Task2: The Lab Assignments (A-F)

- ▶ Two lab assignments: **source code** and **oral presentation**.
- ▶ E: source code



Task2: The Lab Assignments (A-F)

- ▶ Two lab assignments: **source code** and **oral presentation**.
- ▶ E: source code
- ▶ D: source code + half questions (basic)



Task2: The Lab Assignments (A-F)

- ▶ Two lab assignments: **source code** and **oral presentation**.
- ▶ E: source code
- ▶ D: source code + half questions (basic)
- ▶ C: source code + all questions (basic)



Task2: The Lab Assignments (A-F)

- ▶ Two lab assignments: **source code** and **oral presentation**.
- ▶ E: source code
- ▶ D: source code + half questions (basic)
- ▶ C: source code + all questions (basic)
- ▶ B: source code + half questions (basic and advanced)



Task2: The Lab Assignments (A-F)

- ▶ Two lab assignments: **source code** and **oral presentation**.
- ▶ E: source code
- ▶ D: source code + half questions (basic)
- ▶ C: source code + all questions (basic)
- ▶ B: source code + half questions (basic and advanced)
- ▶ A: source code + all questions (basic and advanced)



Task3: The Final Project (A-F)

- ▶ One final project: **source code** and **oral presentation**.
- ▶ Proposed by students and confirmed by the teacher: **A-level** or **C-level** proposals.



Task3: The Final Project (A-F)

- ▶ One final project: **source code** and **oral presentation**.
- ▶ **Proposed by students** and confirmed by the teacher: **A-level** or **C-level** proposals.
- ▶ **E**: C-level source code



Task3: The Final Project (A-F)

- ▶ One final project: **source code** and **oral presentation**.
- ▶ **Proposed by students** and confirmed by the teacher: **A-level** or **C-level** proposals.
- ▶ **E**: C-level source code
- ▶ **D**: C-level source code + half questions (basic and advanced)



Task3: The Final Project (A-F)

- ▶ One final project: **source code** and **oral presentation**.
- ▶ **Proposed by students** and confirmed by the teacher: **A-level** or **C-level** proposals.
- ▶ **E**: C-level source code
- ▶ **D**: C-level source code + half questions (basic and advanced)
- ▶ **C**: C-level source code + all questions (basic and advanced) or A-level source code + all questions (basic)



Task3: The Final Project (A-F)

- ▶ One final project: **source code** and **oral presentation**.
- ▶ **Proposed by students** and confirmed by the teacher: **A-level** or **C-level** proposals.
- ▶ **E**: C-level source code
- ▶ **D**: C-level source code + half questions (basic and advanced)
- ▶ **C**: C-level source code + all questions (basic and advanced) or A-level source code + all questions (basic)
- ▶ **B**: A-level source code + half questions (basic and advanced)



Task3: The Final Project (A-F)

- ▶ One final project: **source code** and **oral presentation**.
- ▶ **Proposed by students** and confirmed by the teacher: **A-level** or **C-level** proposals.
- ▶ **E**: C-level source code
- ▶ **D**: C-level source code + half questions (basic and advanced)
- ▶ **C**: C-level source code + all questions (basic and advanced) or A-level source code + all questions (basic)
- ▶ **B**: A-level source code + half questions (basic and advanced)
- ▶ **A**: A-level source code + all questions (basic and advanced)



The Final Grade

- ▶ The **final grade** is the **weighted average** of the **review questions** (0.2), **two labs** (0.25 each), and the **final project** (0.3).



The Final Grade

- ▶ The **final grade** is the **weighted average** of the **review questions** (0.2), **two labs** (0.25 each), and the **final project** (0.3).
- ▶ To compute it, map **A-E to 5-1**, and take the average.



The Final Grade

- ▶ The **final grade** is the **weighted average** of the **review questions** (0.2), **two labs** (0.25 each), and the **final project** (0.3).
- ▶ To compute it, map **A-E 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 **4.5** will be rounded to **4**.



The Final Grade

- ▶ The **final grade** is the **weighted average** of the **review questions** (0.2), **two labs** (0.25 each), and the **final project** (0.3).
- ▶ To compute it, map **A-E 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 **4.5** will be rounded to **4**.
- ▶ A **late** submission will **reduce you grade level by one**. That is, A will become B, B will become C, and so on.



The Final Grade

- ▶ The **final grade** is the **weighted average** of the **review questions** (0.2), **two labs** (0.25 each), and the **final project** (0.3).
- ▶ To compute it, map **A-E 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 **4.5** will be rounded to **4**.
- ▶ A **late** submission will **reduce you grade level by one**. That is, A will become B, B will become C, and so on.
- ▶ To pass the course, you need to take **at least E** in **all the assignments**.

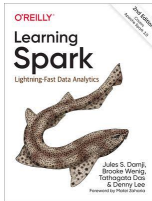
How to Submit the Assignments?

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



The Course Material

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





The Course Web Page

`https://id2221kth.github.io`



The Questions-Answers Page

<https://tinyurl.com/f6x544h>



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

- ▶ The main trends:
 - Computers not getting any faster
 - Internet connections getting faster
 - More people connected to the Internet
- ▶ **Conclusion:** move the computation and storage of big data to the cloud!

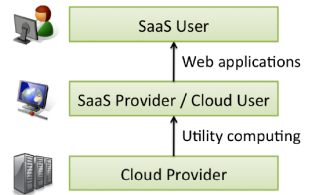


Cloud Computing

Cloud Computing Definition

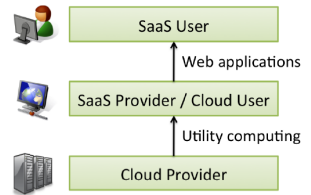
► **Cloud Computing** refers to both:

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



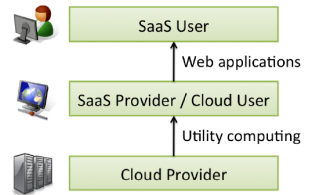
Cloud Computing Definition

- ▶ **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

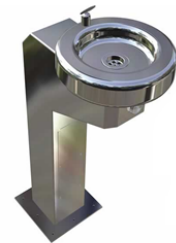


Measured
service with
pay per use

[<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

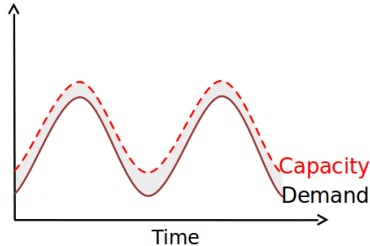
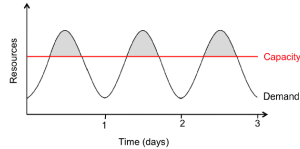
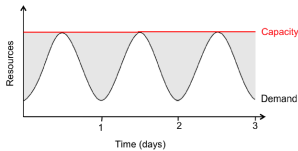
- ▶ Provider's computing resources are pooled to serve consumers
- ▶ Location transparent



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.





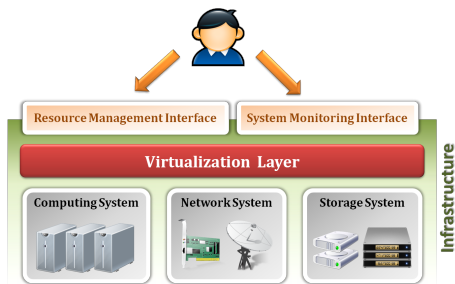
Let's translate it to Cloud Computing



Service Models

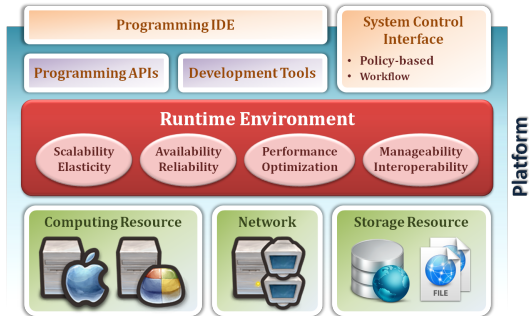
- ▶ Infrastructure as a Service (**IaaS**): similar to **building a new house**.
- ▶ Platform as a Service (**PaaS**): similar to **buying an empty house**.
- ▶ Software as a Service (**SaaS**): similar to **living in a hotel**.

- ▶ 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**

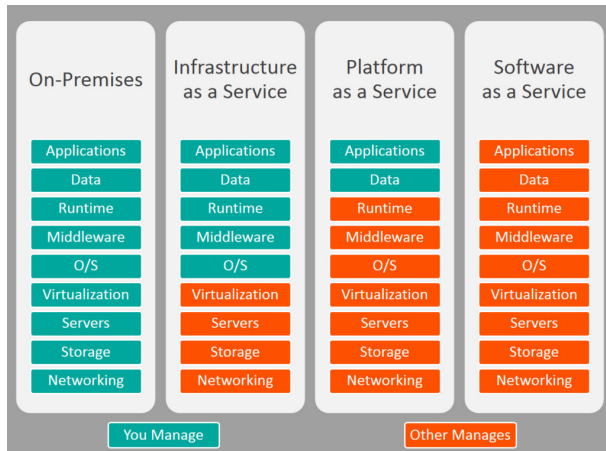


SaaS

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



IaaS - PaaS - SaaS



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

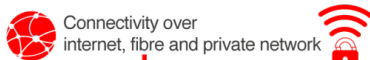
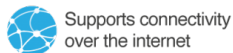
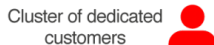


Deployment Models

Deployment Models



VS



[<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



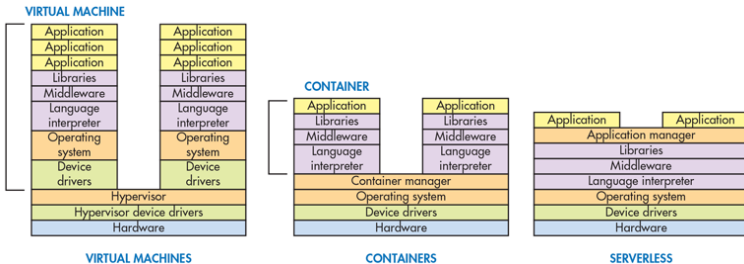
VM



Container



Serverless



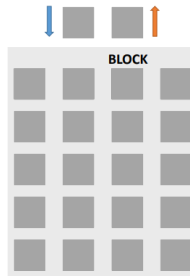
Storage Services

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

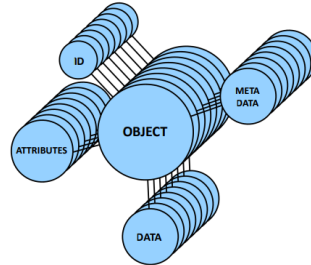
File Storage



Block Storage

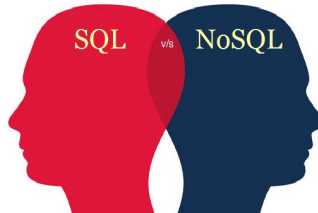


Object Storage



Database Services

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





Big Data



“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

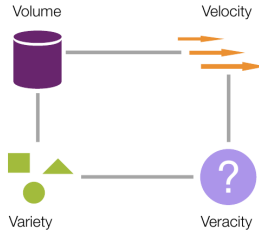
Small Data is when is fit in RAM.
Big Data is when is crash because
is not fit in RAM.

2/6/13, 8:22 AM

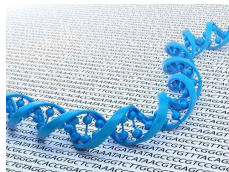


The Four Dimensions of Big Data

- ▶ **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?

2021 *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.

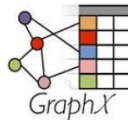


Scale Up vs. Scale Out (2/2)

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



APACHE
HBASE



Storm



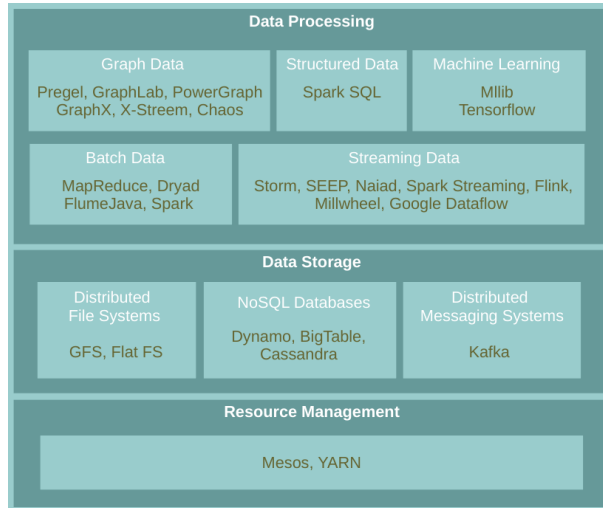
S4 distributed stream
computing platform



Google Cloud Platform

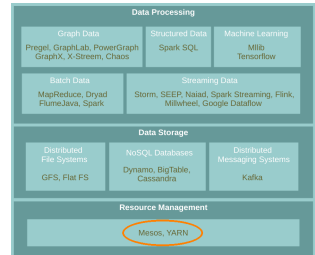


Big Data Stack



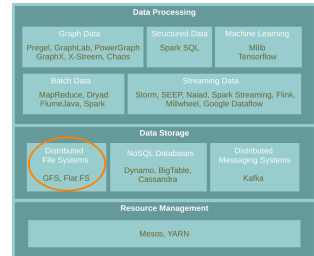
Resource Management

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



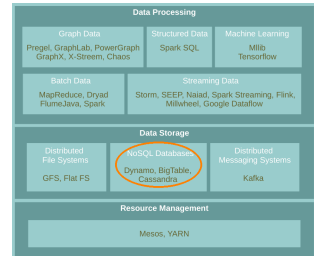
Data Storage - Distributed File Systems

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



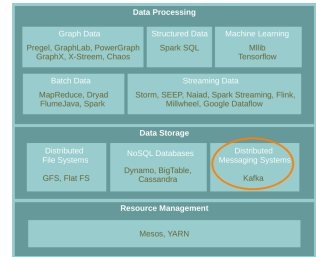
Data Storage - NoSQL Databases

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



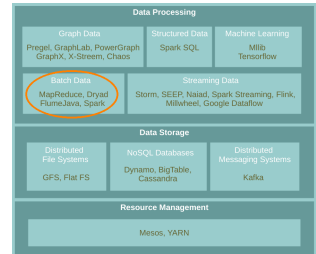
Data Storage - Messaging Systems

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



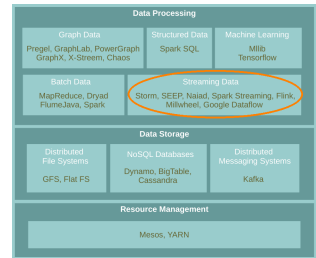
Data Processing - Batch Data

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



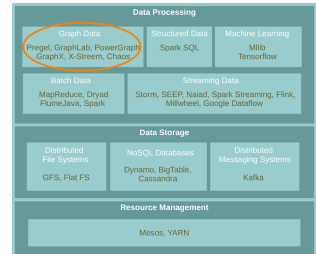
Data Processing - Streaming Data

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



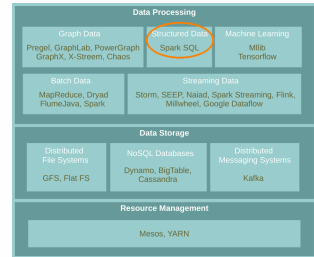
Data Processing - Linked Data (Graph)

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



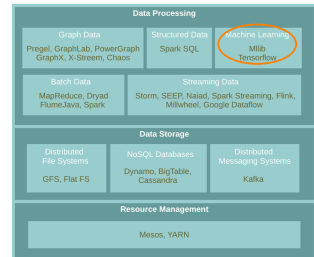
Data Processing - Structured Data

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



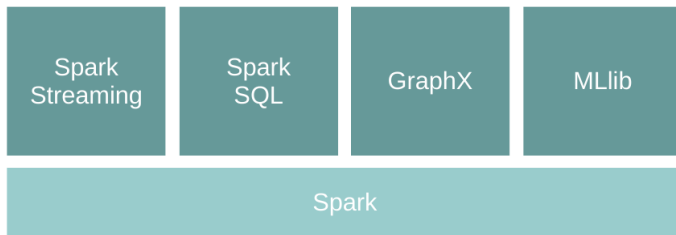
Data Processing - Machine Learning

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



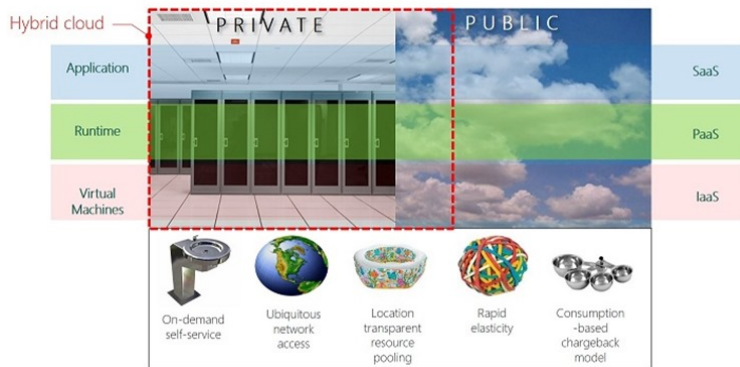


Spark Processing Engine



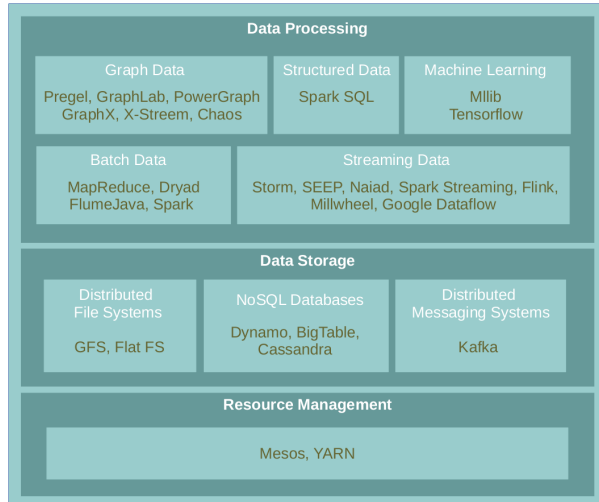
Summary

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?