Big data

Nowadays, the data growing very fast the amount of data generated exceed 30,000 GB every second from various sources. We have pushed the traditional database system to the limit and face failure to scale. So we found ourselves in need of new technology and here came big data. The pioneer company in the field were Google presenting distributed filesystem, MapReduce, and distributed locking services. Amazon with Dynamo, which is a distributed Key/value store. Then the open source community provide us with Hadoop, HBase, MongoDB, Cassandra, and Countless projects [Big Data, NATHAN MARZ, JAMES WARREN]

Big data is the description of a large amount of either organized or unorganized data that is analyzed  
To make an informed decision or evaluation. explain big data adequately, there are three main characteristics related to the big data: volume, Variety, and velocity

Volume: use to describe the enormous amount of data that used by big data, and to name a system as a big data system it should be able to handle this amount of data and its expected growth

Variety: the types of sources used by analytics big data systems with different storage formats wither it is structured or unstructured format.

Velocity: describe how fast the data had been generated. And the speed to processed this generated data. To ensure that data is current and updated in real-time

[Big Data Applications and Use Cases, Patrick C. K. Hung]

And we can add veracity as forth characteristic, which is the quality of data by cleaning the data to improve the accuracy. [BIG DATA FOR AGRICULTURE, Gerard Sylvester]

By 2017 there are roughly 100 projects related to big data or Hadoop. Apache produces a lot of open sources solution to solve a new challenge or to solve an old challenge in a new way. They introduce Hive as an extensive database, Kafka for messaging, and more [Big data demystified, David Stephenson]

As Forbes mention, we generate almost 2.5 quintillion byte of data every day, and it expected to increase dramatically with the rise of IoT. With all this amount of data, we have to ask our self, how we can make sense of these massive amounts of data, analyze, and get actionable intelligence out of it.

Hadoop

As a solution from Yahoo to improve its search platform, they introduce Hadoop in 2006, which become a defacto standard in the industry, helping a lot of enterprises to get benefit from their big data. When we say Hadoop, now we don't mean just core apache Hadoop project but also apache Hadoop technology along with an ecosystem that works with.

According to [Hadoop in Practice, Alex Holmes], we can define Hadoop as "Hadoop is a platform that provides both distributed storage and computational capabilities."

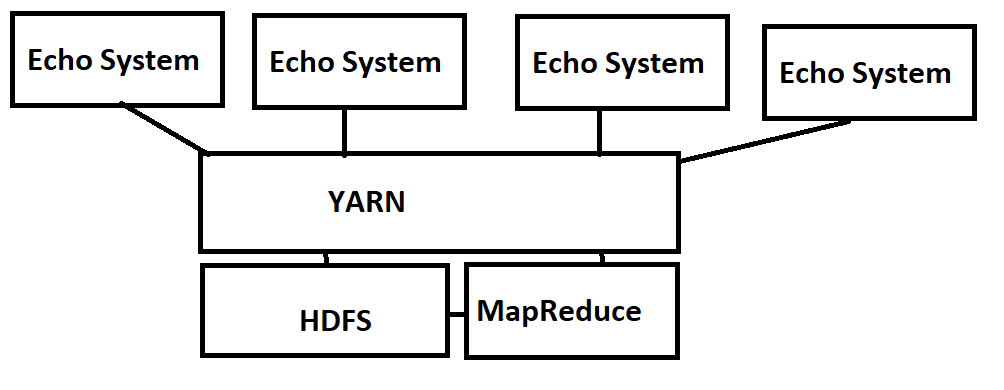
The component of Hadoop

1. Core Apache Hadoop  
   it is the platform to run and execute distributed computing tasks and content of three major components:
   1. HDFS (Hadoop Distributed File System): a distributed filesystem. Stores data in a distributed, scalable, fault-tolerant manner. Implemented as master-slave architecture where NameNode is the master and DataNode (one or more) are the slaves. The dataNode program manages each data node. In the name node, we found information about where to store and retrieve data besides the number of copies of the data. By default, HDFS stores three copies of the data across the cluster.
   2. YARN (Yet Another Resource Negotiator) resources manager to keep track of CPU, RAM, and Disk spaces, in order to smooth the execution of tasks.  
      it appears as apart of Hadoop in 2012, it is main rule is resource negotiation which before depends on the operating system for each node on the cluster
   3. MapReduce: A distributed data processing and analysis framework. To analyze data stored in HDFS. It consists of to processing phase: map and reduces. In the map phase, it performs record by record; in a reduce phase, it makes aggregation operation on a group of records. MapReduce gets the data from the map to reduce functions by shuffle and sort. If you find MapReduce too abstracted you can use Crunch, Cascading, or Scalding tools and also there is a high-level language like Apache Hive or Apache Big
2. The Hadoop Ecosystem  
   Hadoop ecosystem is a collection of tools and systems that run with the Hadoop core. It's changed over time and along with distribution. The famous distribution includes Cloudera, Hortonworks, and MapR. Those tools and systems distributed between data ingestion, workflow, message manager, analyzing, and other tasks. Tools and systems like HBase, Spark, Hive, Kafka, Storm, Flume, ZooKeeper, and uncountable other. [Hadoop: What You Need to Know, Donald Miner]  
   Hadoop came with the term Schema-on-read which mean you would load your data first then ask question later in contrast for schema-on-write in a relational database when the schema encoded when the data stored in analytic platform.  
   in this research, we will use the following systems/tools
   1. HDFS
   2. Hive
   3. NiFi
   4. Spark

[Hadoop: What You Need to Know, Donald Miner]

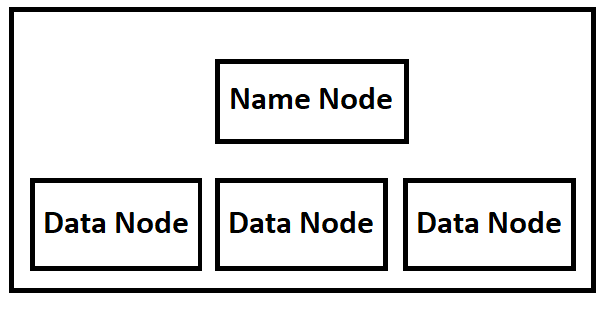
Hadoop

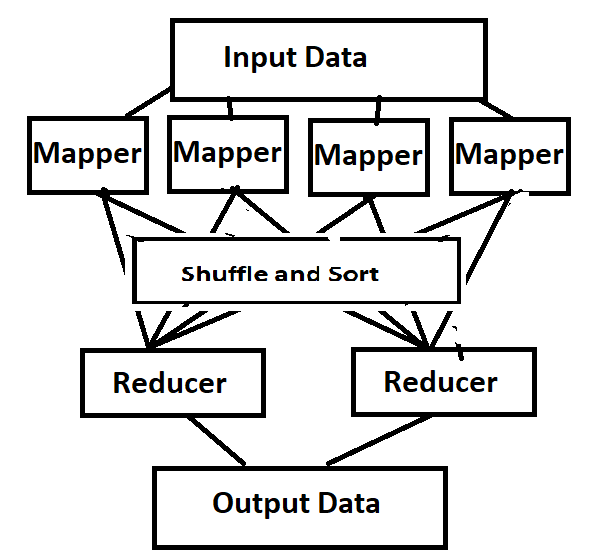
According to [Hadoop in Practice, Alex Holmes], we can define Hadoop as "Hadoop is a platform that provides both distributed storage and computational capabilities."



So at the high level, we can divide Hadoop to tow main components:

1. Core Hadoop Components
   1. HDFS  
      It's modeled after Google File System (GFS) paper. The key traits for it are scalability and availability, with data replicant to achieve fault tolerance



* 1. Yarn: is a distributed resource scheduler. It introduced with Hadoop 2. It enables us to execute a new set of workloads as natively supported. Its contents of a client, which uses a resource manager that connects with Node manager and application master this gives us to allocate containers in various configurations.
  2. MapReduce: it's a batch-based framework to distributed computing and also after Google's paper on MapReduce. With MapReduce, we can parallelize work over a numerous amount of row data  
     

1. The Hadoop ecosystem: they aim to increase the accessibility of Hadoop and distributed between high-level languages(ex. Pig, Crunch), SQL-on-Hadoop (ex. Hive, Impala), Predictive analytics (ex. Mahout, R), alternative processing (ex. Storm, Spark), and miscellaneous (ex. Sqoop, Oozie)

[Hadoop in Practice, Alex Holmes]

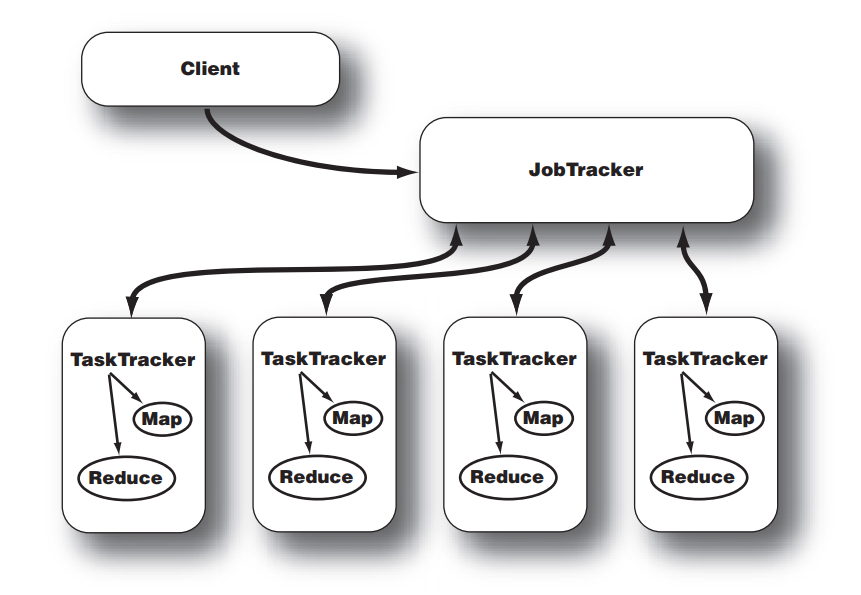
Hadoop

Benefit of Hadoop

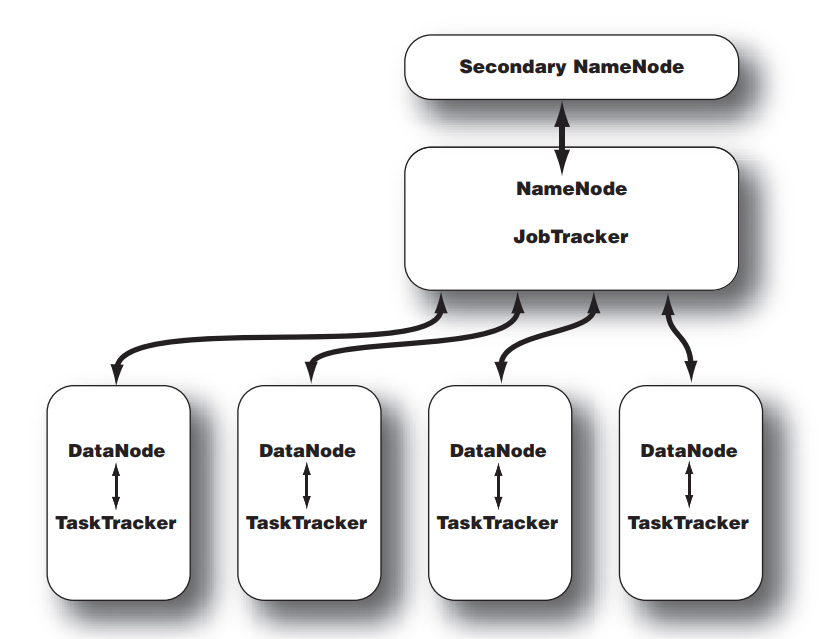
1. Accessibility: it runs on large clusters either on a local machine or on cloud
2. Robust: because its aim to run on a different platform
3. Scalable: we can add more than node to the cluster
4. Simple: easy to use where user can writ efficiently parallel code quickly

Cluster

The building blocks of Hadoop is cluster when we configure cluster which running a set of daemons (resident programs) on the different servers. The daemons have a rule; some on a server only and others may exist across multiple servers, the daemons like:

* NameNode (NN): it directs the slave DataNode daemons to perform the low-level I/O tasks. It keeps track of the file blocks in DataNode. Each cluster had one NN
* DataNode: it is the actual files, whereas reading and writing the file we distributed on HDFS blocks on those DataNode and keep tracking by using NameNode
* Secondary NameNode (SNN): is an assistant daemon for monitoring the state of the cluster HDFS. And each cluster had only on SNN.
* JobTracker: this daemon work as a connection between the application and Hadoop. Monitors task and assign them node. And also every cluster have one JobTracker
* TaskTraker: this daemon does a lot; it responsible: supervision the MapReduce job, manage TaskTrackers, manage task execution on each slave node.  
  

Interaction between JobTracker and TaskTracker



The topology of a Hadoop cluster

[Hadoop in Action, Chuck Lam]

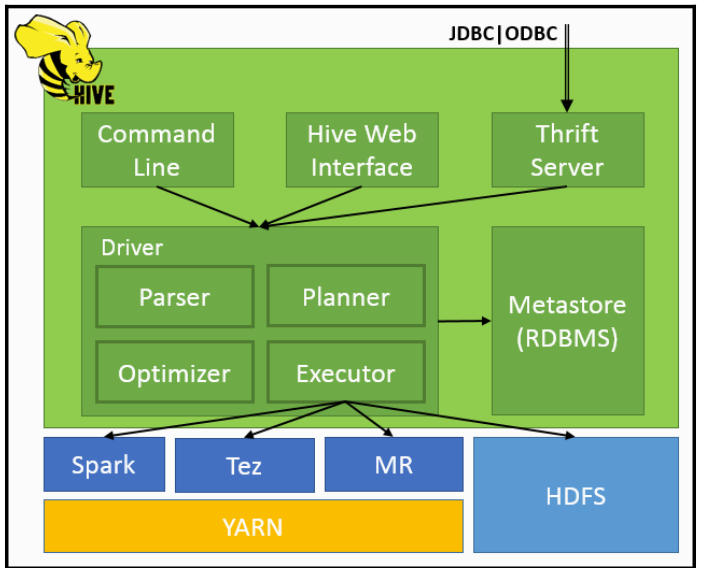
* + **Hive**

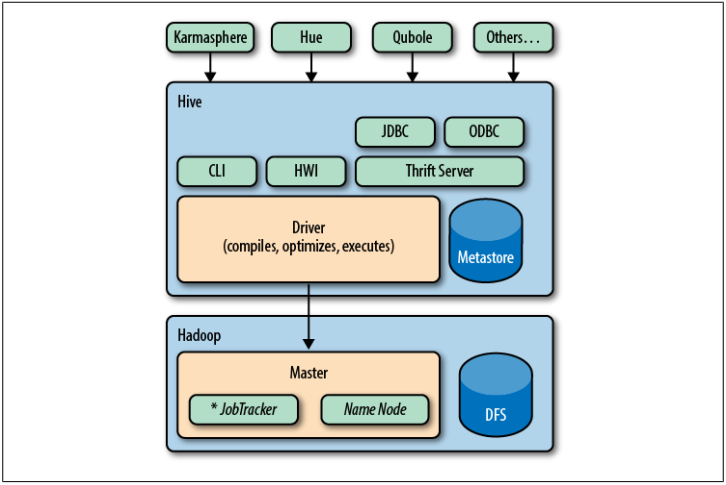
Apache Hive is a data warehouse framework built on Hadoop. It provides us with the ability to read, write, and manage large datasets. Apache Hive allows you to write SQL queries to retrieve data stored in different databases and file systems. Apache Hive enables data storage tasks such as extract/transform/load (ETL), Hive provides an *SQL* language, called *Hive Query Language* (*HiveQL* or just *HQL*) for querying data stored in a Hadoop cluster. It also offers direct access to files stored in Apache HDFS or other storage media such as Apache HBase. Apache Hive was initially used by Facebook but has also been certified by other companies. In data warehouse applications it is appropriate to use Hive, where the data that is slowly changing is analyzed, and a rapid response is not required. Because Hive is not a full database, it had its limitations. In Hive, we can not perform a record-level update, insert, or delete. Hive does not provide the fundamental capabilities required for OLTP and online transaction processing. It is closer to being an OLAP tool, online analytical processing. [Programming Hive, Edward Capriolo, Dean Wampler, and Jason Rutherglen] [**Apache Hive Essentials, Dayong Du]**.

The major components of the Apache Hive architecture can be described as follow:

**Hive metastore** or referred to as HCatalog also, stores metadata such as column names, data types, comments, etc. in table-like structure. Hive provides three main data structures: tables, partitions, and buckets. Hive stores the data in a traditional Relational Database Management System (RDBMS) format.

The tables correspond to HDFS directories and may be divided into partitions, where data files can be divided into buckets. Hive's metadata structure is the schema of the Schema-on-Read concept on Hadoop, which implies you are doing not should define the schema in Hive before you store data in HDFS. Applying Hive metadata after storing data brings more flexibility and efficiency to your data work. The recognition of the Hive's metadata makes it the standard way to describe big data and is employed by many tools within the big data ecosystem. [**Apache Hive Essentials, Dayong Du]**





**Driver:** in Apache Hive driver acts sort of a controller. Which is to blame for receiving the queries submitted by thrift, Java Database Connectivity(JDBC), Open Database Connectivity(ODBC), command Interface(CLI), Web User Interface(Web UI )which are a part of the Hive client. The driving force stores the mandatory metadata generated during the execution of a HiveQL statement. The driving force interacts with the compiler to urge a thought to perform the execution of a question and collects its related metadata information.

**Compiler**: it is responsible for compiling the HiveQL query by creating a plan for a query to be executed. The plan shaped by the compiler contains the tasks and steps required to be performed by Map Reduce. The compiler communicates with the Meta store to get the metadata request.

**Execution Engine** is the component responsible for executing the execution plan after compilation and optimization. It schedules how tasks are to be run by interacting with the Hadoop Job Tracker.

**Hive UI** enables external users to interact with Hive by running queries to fetch the result, submit instructions, and performing other processes. However, the Hive Web Interface is a Graphical User Interface, which is an alternative to the Hive Command Line Interface (CLI).

* + **Nifi**

For developing the data integration layer, Apache Nifi had been used. Apache NiFi is designed to automate data migrate from one external system to another. In Apache Nifi, "data flow" is used to mean the automated and managed flow of information between systems [27].

Apache Nifi is a web-based tool that offers an experience between the design, control, feedback, and monitoring of a data ingestion platform. We can say it is easy to configure and confirms data ingestion, low latency, high throughput, and enables tasks to be dynamically prioritized. Flows can be modified at runtime without any trouble in Apache Nifi. Data provenance service is provided which allows tracking dataflow from the beginning to the end. It facilitates the creation of processors to ingest data from external sources for developers and performs effective testing with this tool. It is also secure as it supports SSL, SSH, HTTPS, encrypted content, and provides authorization for multiple tenants, internal authorization/policy management. [27].

**Flow** **File**: Represents each object moving through the system. File flow is a single piece of data. It is made up of two components: flow file attributes and flows file content. The flow file content is the data, and the flow file attribute provides information about the data.

**Processors**: The Nifi component responsible for listening for incoming data, pull data from external sources, publish data to external sources, and route, transform or extract information from flow files.

**Relationships**: An association between processors. A processor transfers the flow file to one of the relationships after it has finished processing it. The Data Flow manager then connects each of these relationships to other components to specify where the flow file should go next.

**Controller Service:** the controller service automatically starts when NiFi starts up. It also provides information necessary for other components to use. They are being added and configured by the Data Flow Manager.

**Process Group**: are used to group multiple components. It is preferable to use in cases when a data flow becomes too complicated.

**Port**: to connect two or more process groups, we need to use ports. Incoming flow files into a process group use input ports, and outgoing flow files from a process group use output ports.

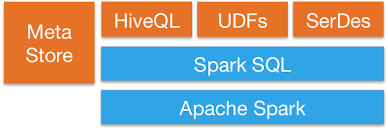
* + Apache Spark SQL

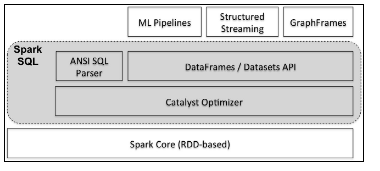
Spark SQL is one of the most powerful components of the Apache Spark that has been included since the release of Spark 1.0. It provides support for Java, Scala, Python and the R API. In the figure ??, Spark SQL components are shown that support for the application of machine learning, graph, streaming and other applications

Spark SQL is one of the most superior elements of Apache Spark. It has been a part of the core distribution since Spark 1.0 and supports Python, Scala, Java, and R programming APIs. As illustrated in the figure ?? Spark SQL components supply the foundation for Spark machine learning applications, streaming applications, graph applications, and many other types of software architectures.

It enables us to query structured data inside Spark programs. It provides a programming abstraction called DataFrame, and it can provide an SQL engine to ensure distributed processing. Apache SparkSQL had the ability to perform in-memory processing, which makes it very suitable to perform analysis. It is faster than HiveQL by 100 times.

Apache Spark SQL is integrated, provides unified access, easily integrates with HIVE, provides standard connectivity, and scalable.





[**Learning Spark SQL,** Aurobindo Sarkar] [**Spark: The Definitive Guide,** **Bill Chambers, and Matei Zaharia][** **https://spark.apache.org/sql/]**

* + Apache Zeppelin

**Apache Zeppelin** is a web-based notebook that enables its users to create interactive data analytics. Apache Zeppelin also allows data ingestion, discovery, visualization, sharing, and collaboration with Apache Spark, SQL, Scala, etc.

With Zeppelin, we can make beautiful data-driven, interactive, and collaborative documents with a rich set of pre-built languages: Python, SparkSQL, Hive, Markdown, Angular, and Shell.

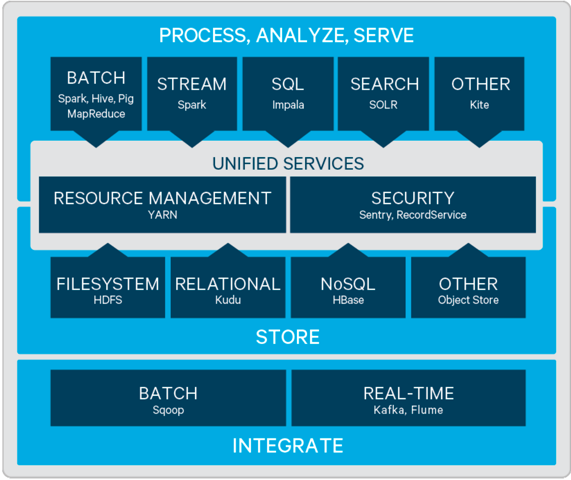
Apache Zeppelin provides a JDBC interpreter, which enables you to create a JDBC connection to any data sources smoothly and perform SQL queries such as Insert, Updates, etc. It has been tested with PostgreSQL, MySQL, Maria DB, Redshift, Apache Hive, Apache Phoenix, Apache Drill, and Apache Tajo. Apache Zeppelin provides a whole range of a group of interpreters for Apache Spark interpreter. It also supports interpreters for Python.

[https://zeppelin.apache.org/docs/0.6.0/]

**Cloudera Distributed Hadoop (CDH)**

CDH is Cloudera's one hundred percent open source platform distribution, which includes Apache Hadoop and constructed specifically to meet organization demands. CDH provides the entirety you want for company use proper out of the box. By integrating Hadoop with greater than a dozen different imperative open source projects, Cloudera has created a functionally advanced device that helps you perform end-to-end Big Data workflows. Here we can install Apache Hadoop core, Apache Accumulo, Apache HBase, Apache Hive, Apache Flume, HUE, Apache Impala, Apache Kafka, Apache Pig, Apache Sentry, Apache Sqoop, and Apache Spark. in addition, you can install the component of Cloudera Flow Management (CFM)

CDH is the most complete, tested, and famous distribution of Apache Hadoop and associated projects. CDH gives you the core factors of Hadoop – scalable storage and allotted computing – along with a Web-based user interface and necessary agency capabilities. CDH is Apache-licensed open supply and is the only Hadoop solution to offer unified batch processing, interactive SQL and interactive search, and role-based get entry to controls.



[https://www.cloudera.com/products/open-source/apache-hadoop/key-cdh-components.html]

[https://www.cloudera.com/downloads/cdh.html]

[https://docs.cloudera.com/documentation/enterprise/5-9-x/topics/cdh\_intro.html]

Hadoop for agriculture

With all those benefits Hadoop provides, it becomes widely used in many sectors and agriculture is not far from that.

Agriculture must now feed 8.5 billion people by 2030 with many environmental and ecological challenges. Digital agriculture was introduced to increase yields and reduce losses and wastage. With improved supply chains, it relies on high-quality data to improve decision-making. Providing this data on various aspects of the agricultural process and analyzing it will help support this solution to help provide food.

The big data benefit in Agriculture sector

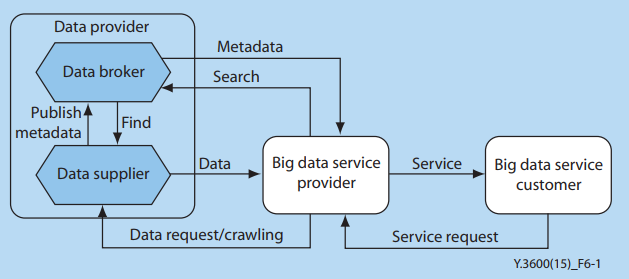
Big data will provide many benefits for digital farming. Using it, we can benefit from artificial intelligence and machine learning, in addition to using the Internet of Things, which helps to:

Make farms smarter (driverless tractors, drones, smart irrigation, smart greenhouses)

Interpretation of previous hardware data and building models and prediction

Using machine vision to diagnose pests and soil defects

Big data ecosystem (Figure ??) includes data providers( data brokers and suppliers), big data service providers and big data service customers



In this diagram, the suppliers provide data from various sources to the brokers. Then the brokers work as a connector between suppliers and bigdata service providers. The big data services provider is the big data platform that gives us the power to analyze data besides the infrastructure. The last part is customers or the end-user, which consumes the services from the big data services provider.  
Some Use Case

1. OFIS

Agriculture communities benefit big data in many ways. Like Olam Farmer Information System (OFIS), which use for improving smallholder productivity livelihoods through allowing field staff to survey and record on the spot. That let them get better advice on interventions and compare progress on these as well as identify hotspot for risks, OFIS also gives insights to tackle issues ranging from poor yields to climate change.

OFIS start in 2014 at Cote d'Ivoire to understand the farm landscape of Olam's cocoa suppliers. But since then has been rolled out to eight product categories: coffee, cashew, cotton, and rice, in 27 countries.

It collects data about farm size, location, age of tree stock, economic, social, and health infrastructure, and eco-support system using a handheld device.

1. mSTAR  
   data scientists are piloting new methods to estimate yields, growing areas, and even poverty, and research institutions are demonstrating what can be gained by analyzing large amounts of data on crop management, yield, soils, and weather conditions in local farming systems.

mSTAR project supported the Digital Development for Feed the Future  
(D2FTF) a team within USAID's Center for Digital Development, USAID/Cambodia, and USAID/  
Nepal over the course of 14 months to identify ways to improve the structure, storage, and  
governance of FTF data and facilitate analysis across the portfolios.

[BIG DATA FOR AGRICULTURE, Gerard Sylvester]

1. CGIAR

Their goal, according to them, Using big data approaches to solve agricultural development problems faster, better, and at a greater scale than before.

The CGIAR platform for the Big Data is implemented on farms with the support of the Development Fund with a bilateral financing agreement. The aim of the platform is to take advantage of the data capabilities available to accelerate and enhance agricultural research worldwide. The platform duration is five years 2017 – 2021 and it provides open data and organizes it. What is required from the partners is to provide innovative ideas and verify the power of analyzes provided by big data.

<https://bigdata.cgiar.org/about-the-platform/>

There are many success cases in developing countries, for example in Kenya, which is a distinct center of information technology innovations for farmers, where they can access the extension records provided by iCow, which change the concepts and methods of agricultural extension. Also in Mexico, MASAGRO, which the Mexican government operates with CIMMYT, has developed an application to provide satellite images to help farmers calculate the amount of nitrogen their farms need.

[Leveraging CGIAR data, CGIAR]

Big data in agriculture can offer:

Business solutions

The big data in agriculture thus collected from such interactions regarding crops, weather, terrain, geographic conditions, water, and more are stored and processed. This leads to the analytics part of our solution. By processing this data, the application will be able to assist:

* Agriculture companies regarding the prospective success of products in different markets
* Farmers about the success of various crops, the predictive impact of natural conditions, etc.
* Consultants will learn more about the most affected geographic area where farmers could be assisted to deliver higher productivity.

Technology solutions

Such applications developed with Hadoop and any front-end technology using deep learning/machine learning allow the platform to provide:

* Predictive analytics: to predict the success of product or crop or predict the ill effect of a natural event on crops
* Historical data analysis: process vast volumes of historical data regarding crops, geography, etc
* Real-time analytics: to provide farmers with real-time assistance by analyzing information provided by them in real-time

Business benefit

The benefit expected from such application as follows

* Increase agricultural productivity: through agriculture data
* The great success of fertilizing products across a variety of geographic conditions for agriculture companies