BigData – HADOOP, Hive, Spark Developer

Agenda

Day 1
Introduction to BigData
BigData Roles and Responsibilities
Introduction to Hadoop and Its EcoSystem

Day 2
Hadoop Distributed Filesystem
Ingesting Data into HDFS

Day 3
MapReduce Framework
Hive Programming

Day 4
Advanced Hive Programming
Introduction to YARN
Spark SQL

Introductions

- Your name
- Job responsibilities
- Previous Hadoop experience (if any)
- What brought you here

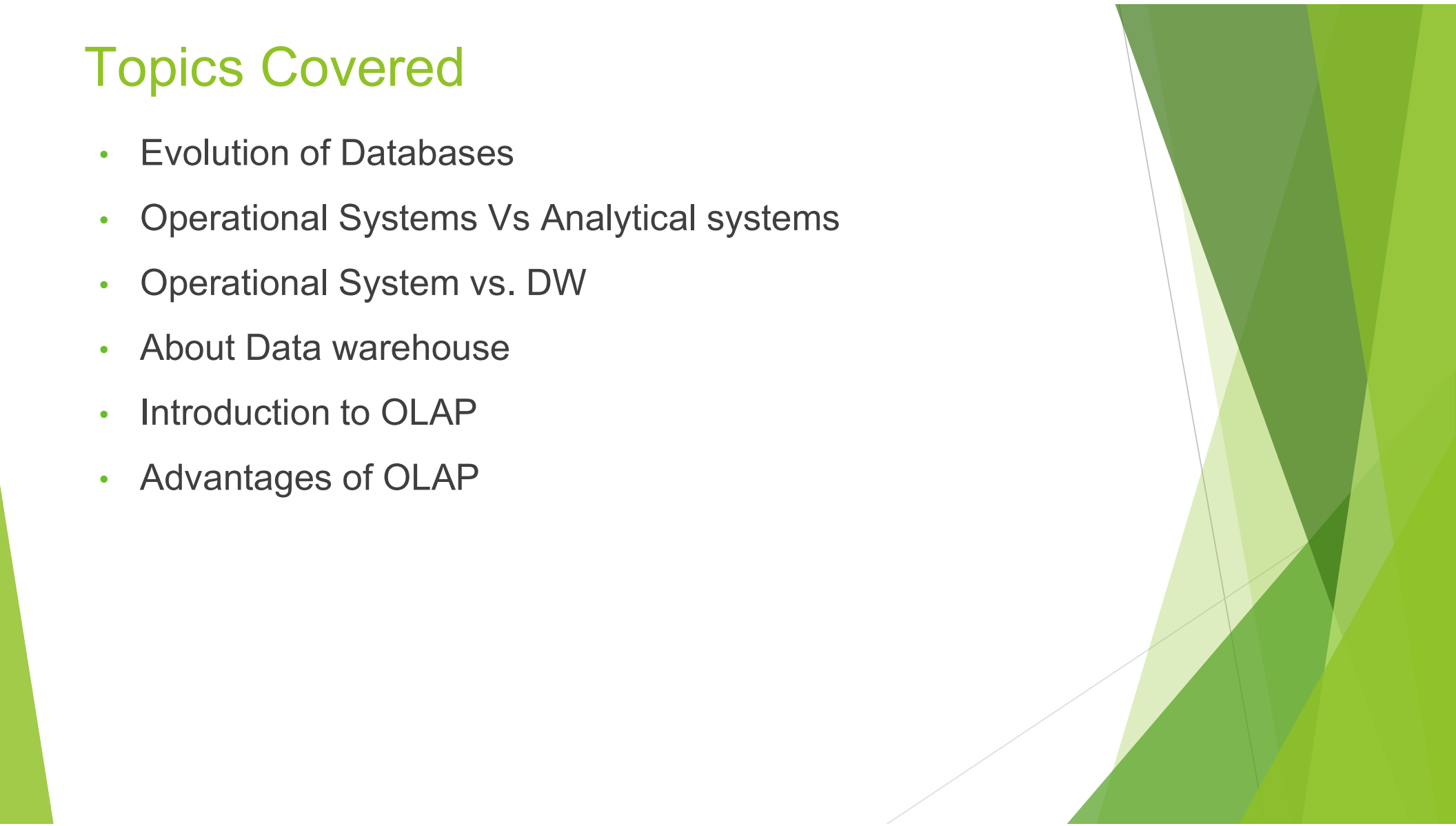
Class Logistics

- Schedule
  10:00 a.m. – 5:00 p.m.
- Restrooms
- Lunch
- Computers and Wireless Access

Introduction To BigData

Topics Covered

- Evolution of Databases
- Operational Systems Vs Analytical systems
- Operational System vs. DW
- About Data warehouse
- Introduction to OLAP
- Advantages of OLAP



Evolution of Databases

- The main objective of the database is to ensure that data can be stored and retrieved easily and effectively
- It is a compilation of data (records) in a structured way.

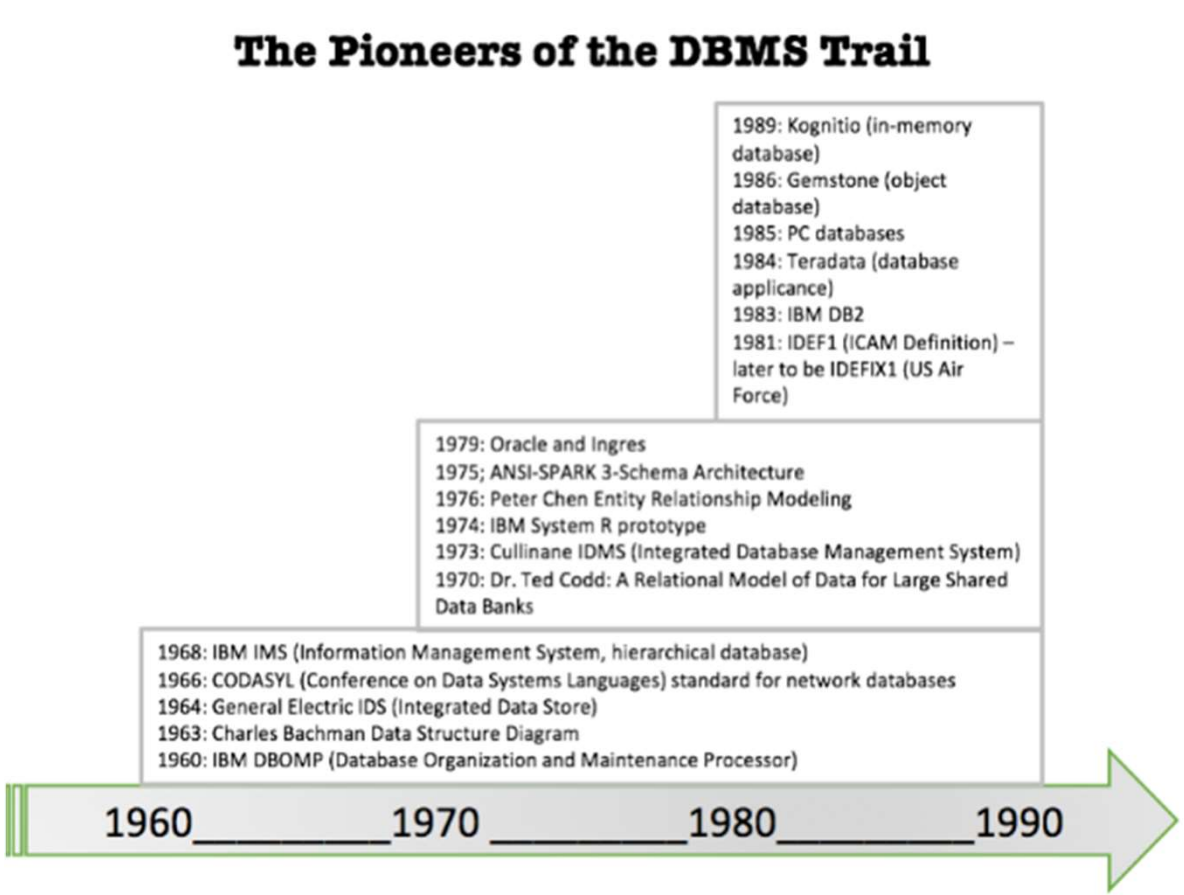
Evolution of Databases and Database Models

- The database evolution happened in four “waves”:
  - The first wave consisted of network, hierarchical, inverted list, and (in the 1990’s) object-oriented DBMSs; it took place from roughly 1960 to 1999.
  - The relational wave introduced all of the SQL products (and a few non-SQL) around 1990 and began to lose users around 2008.
  - The decision support wave introduced Online Analytical Processing (OLAP) and specialized DBMSs around 1990, and is still in full force today.
  - The NoSQL wave includes big data, graphs, and much more; it began in 2008.

Evolution of Databases and Database Models

- There was a lot of ground to cover for the pioneers of Database Management Systems.
- The first twenty to twenty-five years introduced and fine-tuned important technological fundamentals.

Evolution of Databases and Database Models

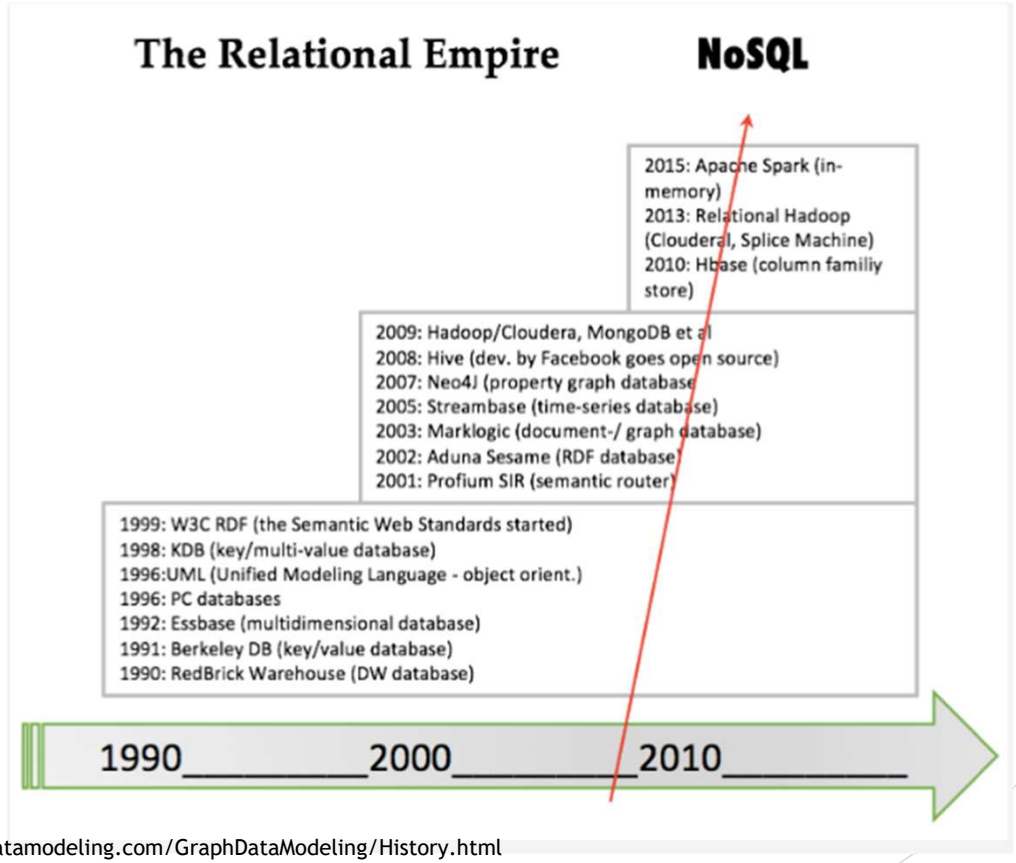


Source - http://graphdatamodeling.com/GraphDataModeling/History.html

Relational Empire to BigData/NoSql

- Today, vendors unite under the NoSQL / Big Data brand
- Around 2008, triggered by Facebook's open source versions of Hive and Cassandra, the NoSQL counter-revolution started.
- This space gets all of the attention today.
- the modern development platforms use *schema-free* or *semi-structured* approaches (also under the umbrella of NoSQL).
- "Model as you go" is a common theme

Relational Empire to BigData/NoSql

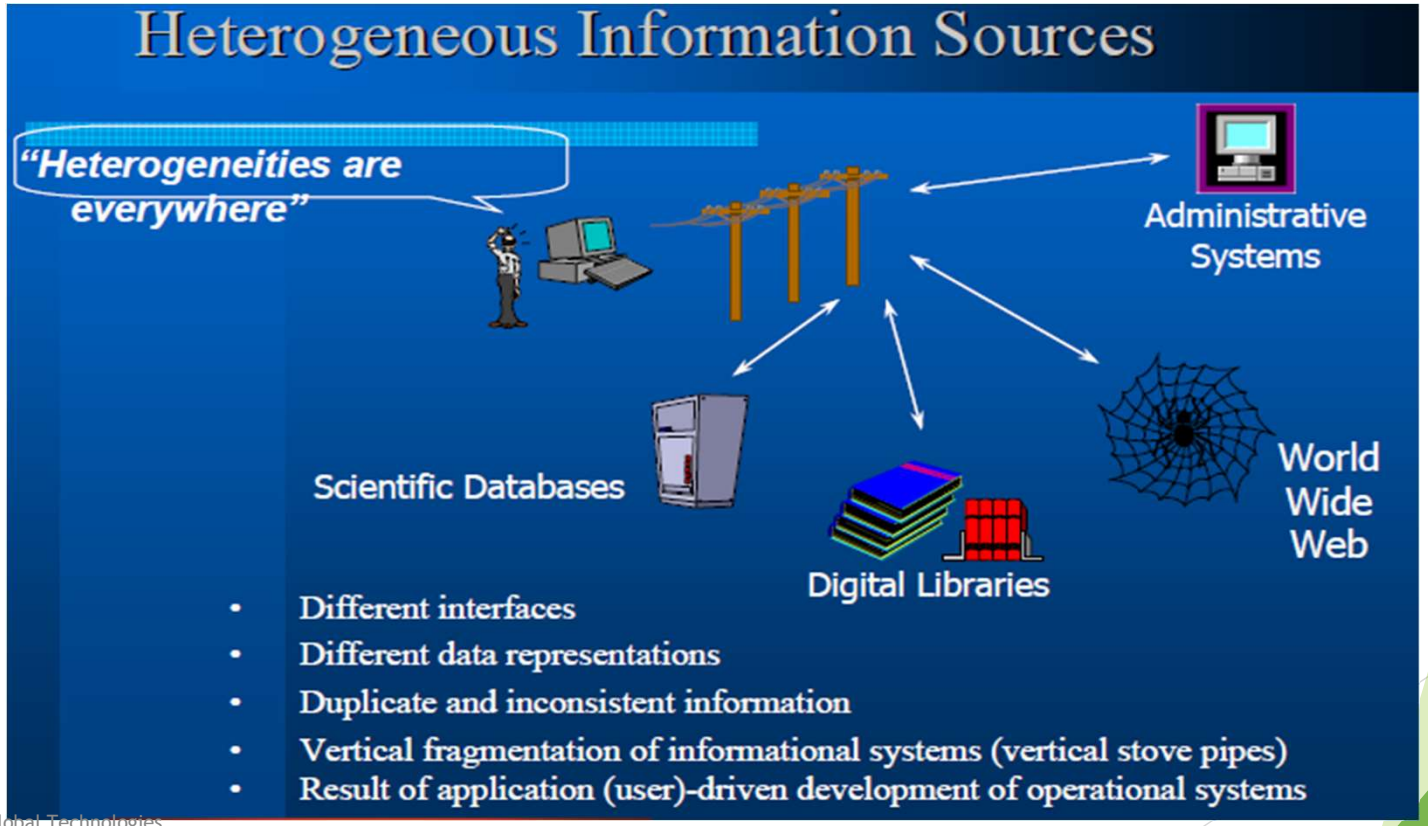


Source - http://graphdatamodeling.com/GraphDataModeling/History.html

Operational Systems Vs Data Warehousing

- Traditional database systems are designed to support typical day-to-day operations via individual user transactions (e.g. registering for a course, entering a financial transaction, etc.).
- Such systems are generally called *operational* or *transactional* systems.
- Operational system is one source of data
- There are still different sources of data
- So heterogenous information sources is the challenge
  - Does not provide integrated view
  - Does not provide uniform user interface
  - Does not support sharing
- Result of above challenges affects decision-making processes across the enterprise

Operational Systems Vs Data Warehousing

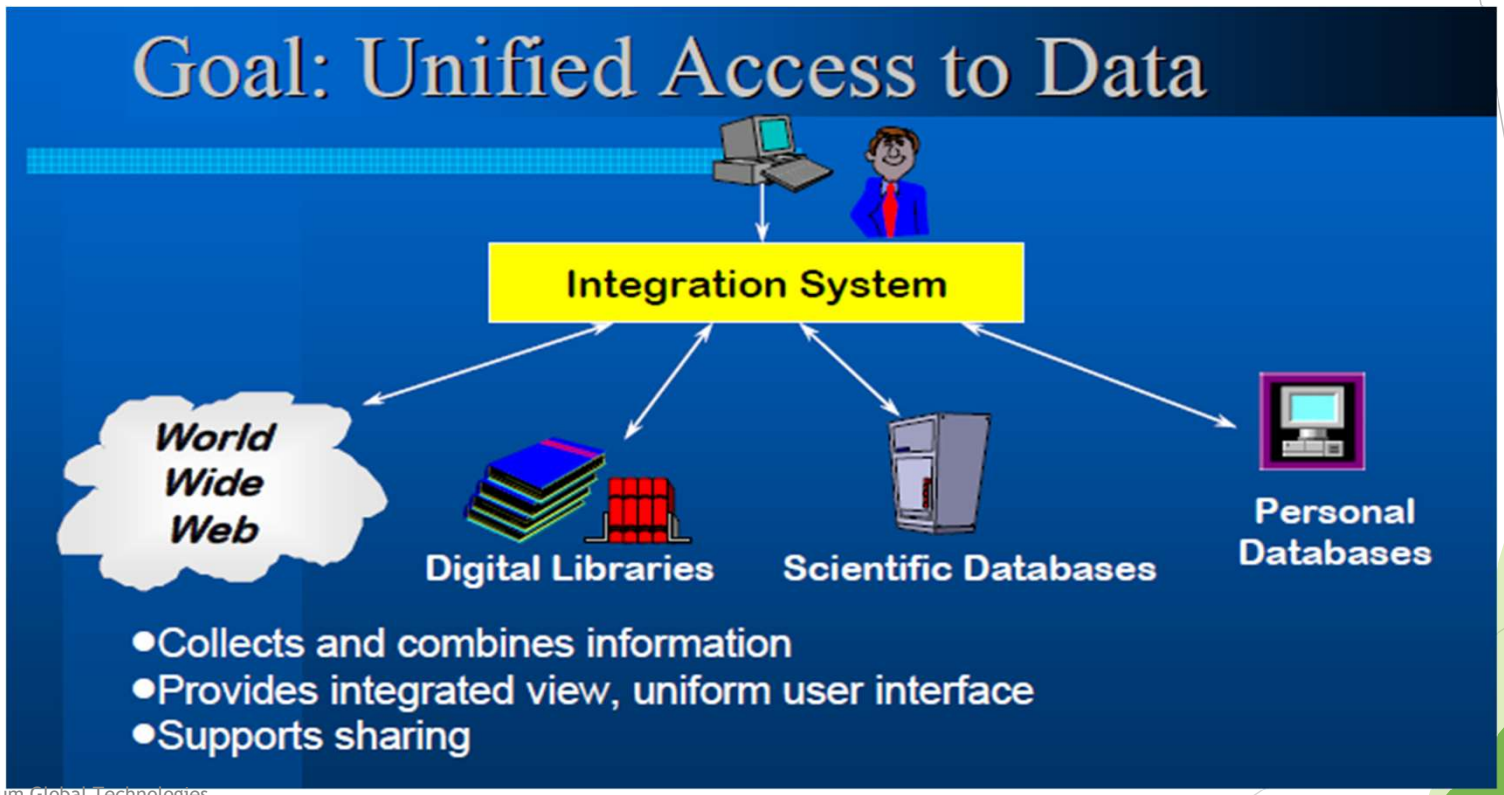


Source – UC Berkeley EDW presentation October 2006

Operational Systems Vs Data Warehousing

- A data warehouse *complements* an existing operational system by providing forecasting and decision-making processes across the enterprise
- A *data warehouse* acts as a centralized repository of an organization's data, ultimately providing a comprehensive and homogenized view of the organization.

Operational Systems Vs Data Warehousing



Source – UC Berkeley EDW presentation October 2006

What data exists in Data Warehouse

- Large volumes of detailed data already exist in transactional database systems.
- A core subset of this data will be *imported* into the data warehouse, prioritized by subject area (i.e. by business area), including finance, research, contracts and grants, enrollment analysis, alumni, etc.
- A fundamental axiom of the data warehouse is that the imported data is both *read-only* and *non-volatile*.
- As the *amount* of data within the data warehouse grows, the *value* of the data increases, allowing a user to perform longer-term analyses of the data.

What are users saying

- Data should be integrated across the enterprise
- Summary data had a real value to the organization
- Historical data held the key to understanding data over time
- What-if capabilities are required

Source – UC Berkeley EDW presentation October 2006

What does a Data warehouse do?

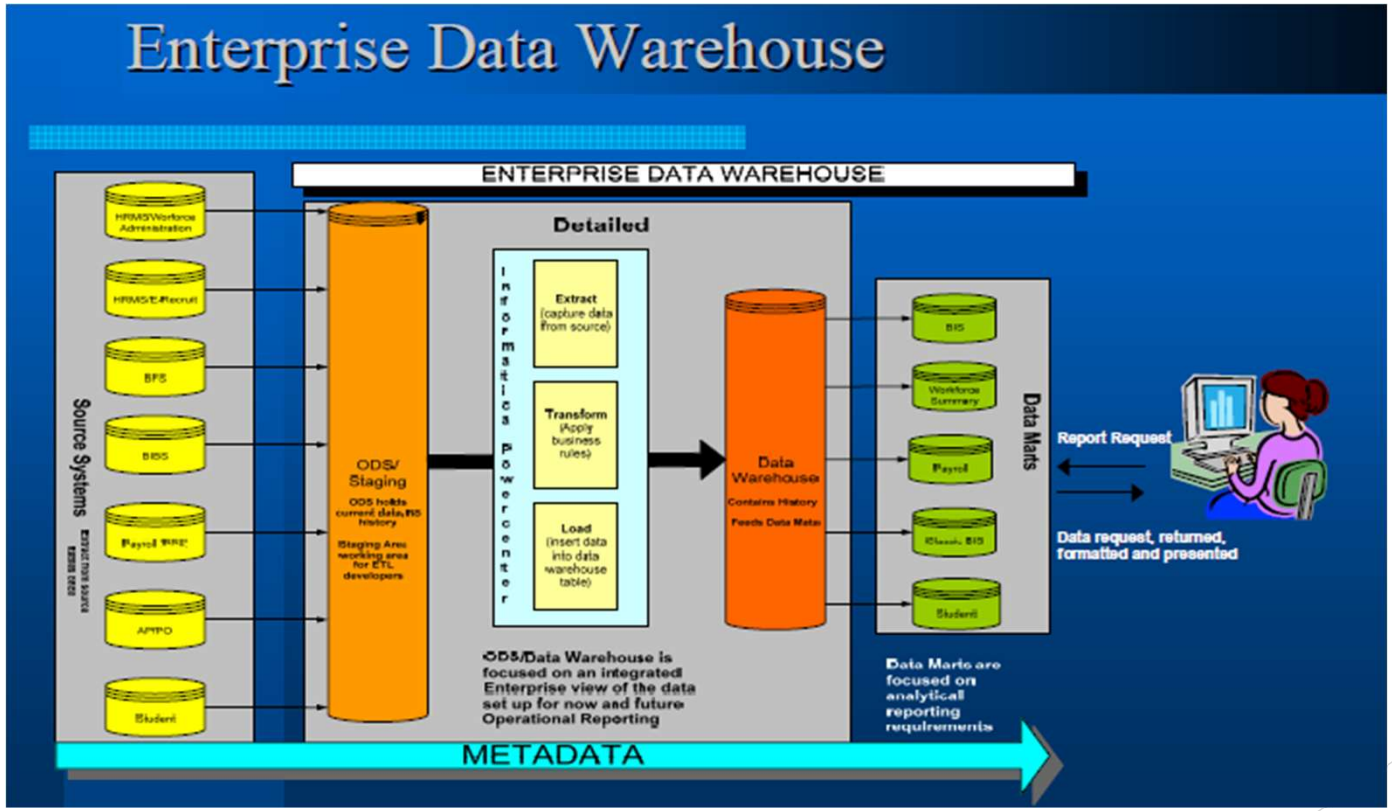
- Integrate divergent information from various systems which enable users to quickly produce powerful ad-hoc queries and perform complex analysis
- Create an infrastructure for reusing the data in numerous ways
- Create an open systems environment to make useful information easily accessible to authorized users
- Help managers make informed decisions

Source – UC Berkeley EDW presentation October 2006

Data Warehouse – Practitioners viewpoint

A data warehouse is simply a single, complete, and consistent store of data obtained from a variety of sources and made available to end users in a way they can understand and use it in a business context.

Data Warehouse – Practitioners viewpoint



Source – UC Berkeley EDW presentation October 2006

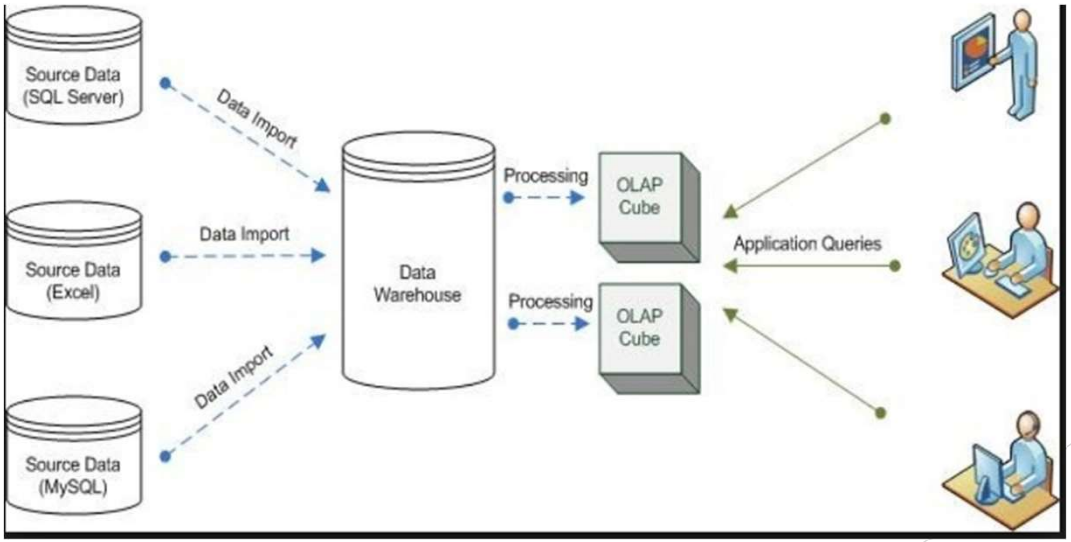
Data Warehouse Vs OLTP

Enterprise Data Warehouse	Traditional Database
Integrated Data	Application-specific Data
Current/Historical Data	Current Data
Organized by Subject	Organized for Data Entry
Non-Volatile Data	Updated Data
Denormalized Data	Normalized Data
Descriptive Data	Encoded Data
Detailed/Summarized Data	Raw Data
Knowledge user (Manager)	Clerical User

Source – UC Berkeley EDW presentation October 2006

Data Warehouse – OLAP

- **OLAP (Online Analytical Processing)** is the technology behind many Business Intelligence (BI) applications.
- OLAP is a powerful technology for data discovery, including capabilities for limitless report viewing, complex analytical calculations, and predictive “what if” scenario (budget, forecast) planning.



Advantages of OLAP

- OLAP technology has been defined as the ability to achieve “fast access to shared multidimensional information.”
- Given OLAP technology’s ability to create very fast aggregations and calculations of underlying data sets, one can understand its usefulness in helping business leaders make better, quicker “informed” decisions.

Lesson Review

1. State True or False - The decision support wave introduced Online Analytical Processing (OLAP) and specialized DBMSs ?

1. Which type of system acts as a centralized repository of an organization's data, ultimately providing a comprehensive and homogenized view of the organization ?

1. A fundamental axiom of the data warehouse is that the imported data is both _____ and _____

1. State True or False - A staging area is an intermediate storage area used for data processing during the extract, transform and load (ETL) process.

1. State any advantage of OLAP Technology

BigData Hadoop Roles and Responsibilities

Topics Covered

- Primer on some general skills expected from Hadoop Professionals
- Various Job Roles under the Hadoop domain
  - Data Enginnering
  - Data Science
  - DevOps

Primer on some general skills expected from BigData Hadoop Professionals

- Ability to work with huge volumes of data so as to derive Business Intelligence
- Analyze data, uncover information, derive insights and propose data-driven strategies
- A knowledge of OOP languages like Java, C++, Python, Scala is good to have
- Database theories, structures, categories, properties, and best practices
- A knowledge of installing, configuring, maintaining and securing Hadoop
- An analytical bent of mind and ability to learn-unlearn-relearn surely comes in handy

Various Job Roles under the BigData Hadoop domain

- Hadoop Data Engineer
- Hadoop Architect
- Hadoop Administrator
- Hadoop DevOps Engineer
- Data Scientist

BigData Hadoop Data Engg birds eye view:

- The primary job of a Hadoop Data Engineer involves coding.
- They are basically software Engineers but working in the Big Data Hadoop domain.
- They are adept at coming up with the design concepts that are used for creating extensive software applications.
- They are masters of computer programming languages.

BigData Hadoop Data Engg Responsibilities:

- Knowledge of agile methodology for delivering software solutions
- Design, develop, document and architect Hadoop applications
- Work with Hadoop Log files to manage and monitor it
- Develop MapReduce coding that works seamlessly on Hadoop clusters
- Working knowledge of SQL, NoSQL, data warehousing & DBA
- Expertise in newer concepts like Apache Spark and Scala programming
- Complete knowledge of the Hadoop ecosystem and Hadoop Common
- Seamlessly convert hard-to-grasp technical requirements into outstanding designs
- Designing web services for swift data tracking and Querying data at high speeds
- Test software prototypes, propose standards and smoothly transfer it to operations

BigData Hadoop Architect birds eye view :

- A Hadoop Architect, as the name suggests, is the one entrusted with the tremendous responsibility of dictating where the organization will go in terms of Big Data Hadoop deployment.
- He/She is involved in planning, designing and strategizing the roadmap and decides how the organization moves forward.

BigData Hadoop Architect Responsibilities:

- Hands-on experience in working with Hadoop Distribution platforms like HortonWorks, Cloudera, MapR and others
- Take end-to-end responsibility of the Hadoop Life Cycle in the organization
- Be the bridge between data scientists, engineers and the organizational needs
- Do in-depth requirement analysis and exclusively choose the work platform
- Full knowledge of Hadoop Architecture and HDFS is a must
- Working knowledge of MapReduce, HBase, Pig, Spark, Java and Hive
- Ensuring the chosen Hadoop solution is being deployed without any hindrance

BigData Hadoop Administrator birds eye view :

- The Hadoop Administrator is also a very prominent role as he/she is responsible for ensuring there is no roadblock to the smooth functioning of Hadoop framework.
- The roles and responsibilities resemble that of a System Administrator.
- A complete knowledge of the hardware ecosystem and Hadoop Architecture is critical.

BigData Hadoop Administrator Responsibilities:

- Manage and maintain the Hadoop clusters for uninterrupted jobRoutine check-up, back-up and monitoring of the entire system
- Ensuring the connectivity and network is always up and running
- Planning for capacity upgrading, downsizing as and when the need arises
- Managing the HDFS and ensuring it is working optimally at all times
- Securing the Hadoop cluster in a foolproof manner is paramount
- Regulating the administration rights depending on job profile of users
- Adding new users over time and discarding redundant users smoothly
- Proficiency in Linux scripting and also in Hive, Oozie and HCatalog

BigData Hadoop Tester birds eye view :

- The job of the Hadoop Test Engineer has become extremely critical since today Hadoop networks are getting bigger and more complex with each passing day.
- This poses some new problems when it comes to viability, security and ensuring everything works smoothly without any bugs or issues.
- The Hadoop Test Engineer is primarily responsible for troubleshooting the Hadoop Applications and rectifying any problem that he/she discovers at the earliest before it becomes seriously threatening.

BigData Hadoop Tester Responsibilities:

- Construct and deploy both positive and negative test cases
- Discover, document and report bugs and performance issues
- Ensure the MaReduce jobs are running at peak performance
- Constituent Hadoop scripts like HiveQL, Pig Latin are all robust
- Expert knowledge of Java to efficiently do the MapReduce testing
- Understanding of MRUnit, JUnit Testing frameworks is essential
- Full proficiency in Apache Pig and Hive is required
- Able to work with Selenium Testing Automation tool
- Able to come up with contingency plans in case of breakdown

Data Scientist birds eye view :

- A part of the attraction lies in the fact that a Data Scientist wears multiple hats over the course of a typical day at the office.
- He/She is part scientist, part artist and part magician!

Data Scientist Responsibilities:

- Data Scientists are basically Data Analysts with wider responsibilities
- Complete mastery of different techniques of analyzing data
- Expect to solve real business issues backed by solid data
- Tailor the data analytics pursuit to suit the specific business needs
- A strong grip of mathematics and statistics is expected
- Keeping the big picture in mind at all times to know what needs to be done
- Develop data mining architecture, data modeling standards and more
- An advanced knowledge of SQL, Hive and Pig is a must
- Ability to work with R, SPSS and SAS is hugely beneficial
- Ability to reason, corroborate actions with data and insights
- Creative ability to do things that can work wonders for the business
- Top-notch communication skills to take everybody onboard in the organization.

Lesson Review

1. Which Job Profile takes the responsibility of Application development coding?

1. State True or False, Hadoop Data Engineer take end-to-end responsibility of the Hadoop Life Cycle in the organization ?

1. Which Role will take the responsibility of Cluster Capacity Managemment

1. Which Role will take the responsibility of Construct and deploy both positive and negative test cases

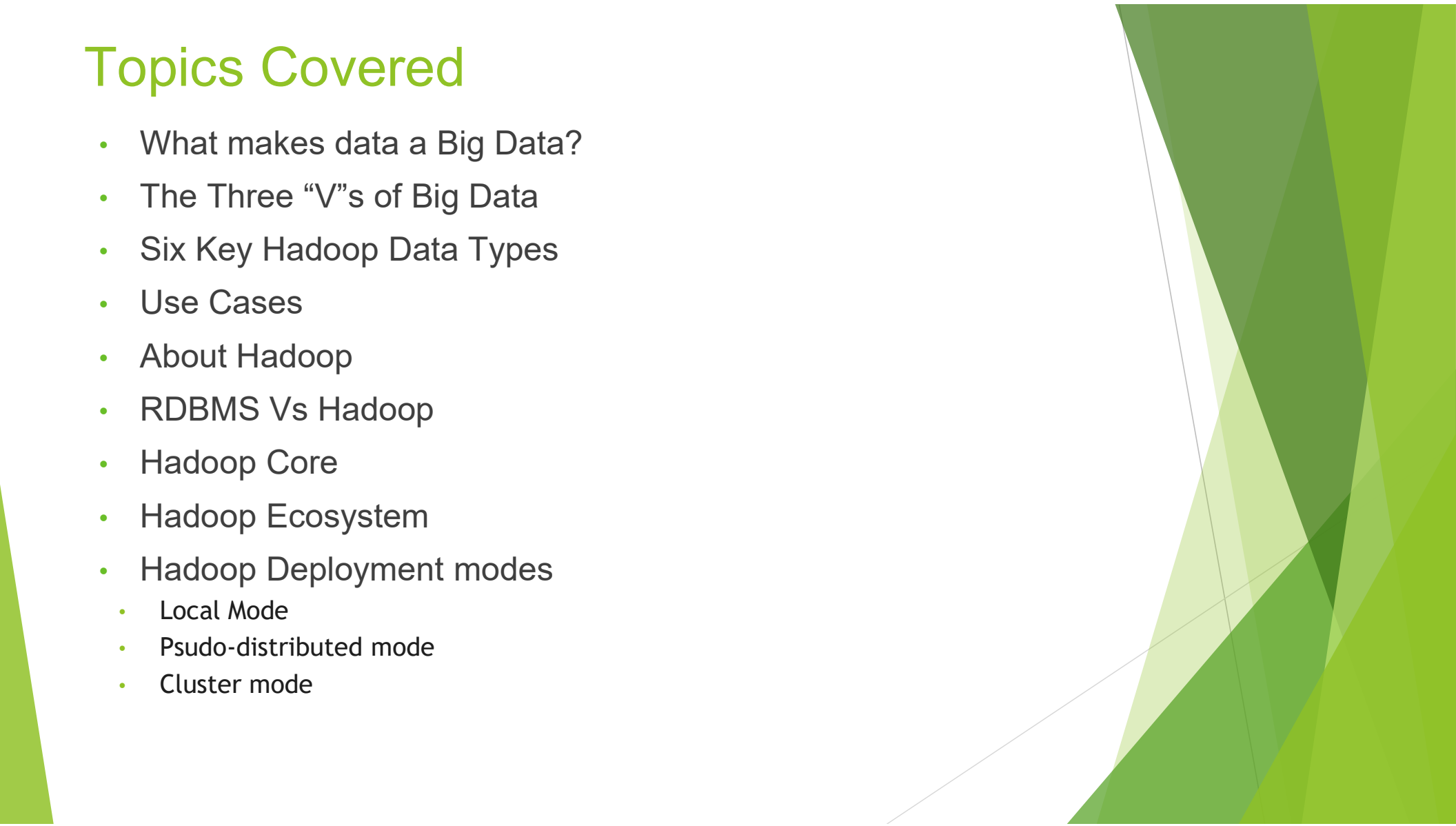
1. Which Role requires a strong grip of mathematics and statistics

Lab: Setting up HDP 2.6 Lab Environment

Introduction To Hadoop and Its EcoSystem

Topics Covered

- What makes data a Big Data?
- The Three “V”s of Big Data
- Six Key Hadoop Data Types
- Use Cases
- About Hadoop
- RDBMS Vs Hadoop
- Hadoop Core
- Hadoop Ecosystem
- Hadoop Deployment modes
  - Local Mode
  - Psudo-distributed mode
  - Cluster mode



What Makes Data BIG DATA?

- The phrase Big Data comes from the computational sciences
- Specifically, it is used to describe scenarios where the *volume* and *variety* of data types *overwhelm* the existing tools to *store* and *process* it
- In 2001, the industry analyst *Doug Laney* described Big Data using the three V's of *volume*, *velocity*, and *variety*

The Three Vs. of Big Data

Variety	Unstructured and semi-structured data is becoming as strategic as the traditional structured data.
Volume	Data coming in from new sources as well as increased regulation in multiple areas means storing more data for longer periods of time.
Velocity	Machine data, as well as data coming from new sources, is being ingested at speeds not even imagined a few years ago.

Variety

- Variety refers to the number of types of data being generated
- Varieties of data include structured, semi-structured, and unstructured data arriving from a myriad of sources
- Data can be gathered from databases, XML or JSON files, text documents, email, video, audio, stock ticker data, and financial transactions

Variety

- There are **problems** related to the variety of data. This include
  - How to gather, link, match, cleanse, and transform data across systems.
  - You also have to consider how to connect and correlate data relationships and hierarchies in order to extract business value from the data

Volume

- Volume refers to the amount of data being generated. Think in terms of gigabytes, terabytes, and petabytes
- Many systems and applications are just not able to store, let alone ingest or process, that much data

Multiples of bytes V·T·E

Decimal			Binary				
Value	Metric		Value	JEDEC		IEC	
1000	kB	kilobyte	1024	KB	kilobyte	KiB	kibibyte
1000^2	MB	megabyte	1024^2	MB	megabyte	MiB	mebibyte
1000^3	GB	gigabyte	1024^3	GB	gigabyte	GiB	gibibyte
1000^4	TB	terabyte	1024^4	–	–	TiB	tebibyte
1000^5	PB	**petabyte**	1024^5	–	–	PiB	pebibyte
1000^6	EB	exabyte	1024^6	–	–	EiB	exbibyte
1000^7	ZB	zettabyte	1024^7	–	–	ZiB	zebibyte
1000^8	YB	yottabyte	1024^8	–	–	YiB	yobibyte

Orders of magnitude of data

Volume

- Many factors contribute to the increase in data volume. This includes
  - Transaction-based data stored for years
  - Unstructured data streaming in from social media
  - Ever increasing amounts of sensor and machine data being produced and collected

Volume

- There are **problems** related to the volume of data
  - Storage cost is an obvious issue

SAN Storage	NAS Filers	Local Storage
$2-10/GB	$1-5/GB	$0.05/GB

  - Another problem is filtering and finding relevant and valuable information in large quantities of data that often contains not valuable information

Volume

- You also need a solution to analyze data quickly enough in order to maximize business value ***today*** and not just next quarter or next year

Velocity

- Velocity refers to the rate at which new data is created. Think in terms of megabytes per second and gigabytes per second
- Data is streaming in at unprecedented speed and must be dealt with in a timely manner in order to extract maximum value from the data
- Sources of this data include logs, social media, RFID tags, sensors, and many more

Velocity

- There are **problems** related to the velocity of data. These include *not reacting quickly* enough to benefit from the data
- For example, data could be used to create a *dashboard* that could warn of imminent failure or a security breach

  Failure to react in time could lead to service outages

Velocity

- Another problem related to the velocity of data is that data flows tend to be highly inconsistent with periodic peaks.
- Causes include daily or seasonal changes or event-triggered peak loads
- For example, a change in political leadership could cause a peak in social media

Hadoop Was Designed for Big Data

"Big Data is high-volume, -velocity and -variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making." – Gartner

Source:- http://www.gartner.com/it-glossary/big-data/

Hadoop Was Designed for Big Data

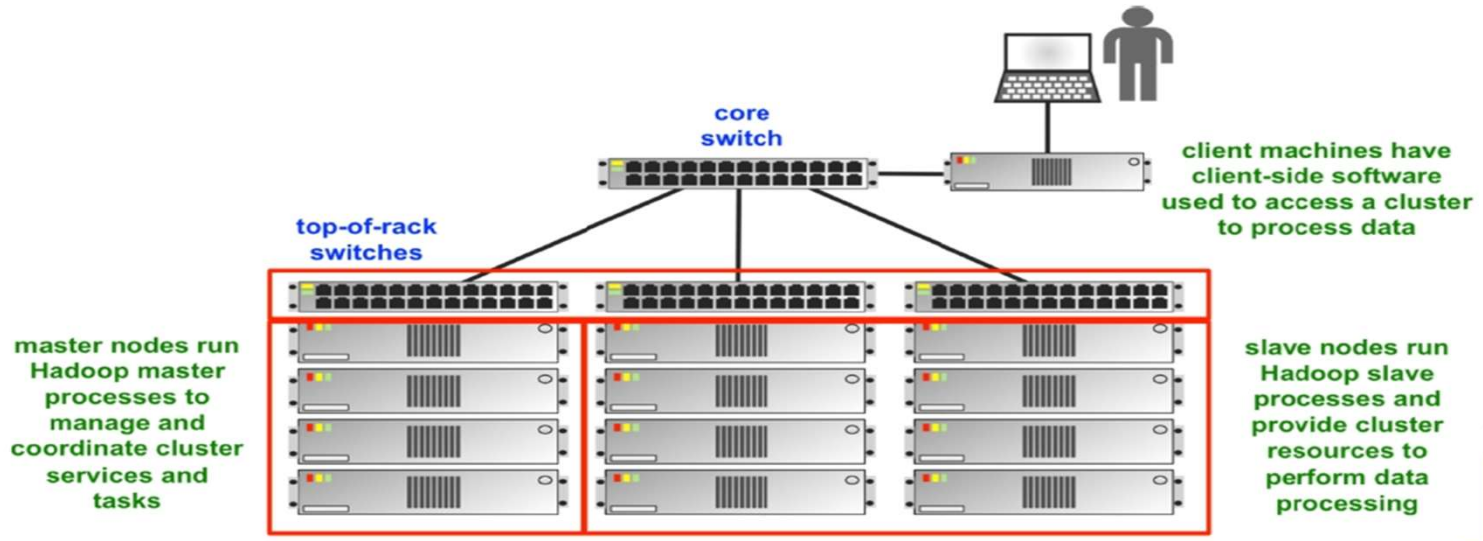
- The Gartner quote makes a good point.
  - It is not enough to understand what Big Data is and then collect it
  - You must also have a means of *processing* it in order to extract value from it
- The good news is that Hadoop was designed to *collect*, *store*, and *analyze* Big Bata
- And it does it all in a *cost-effective* way

What is Apache Hadoop?

- So what is Apache Hadoop?
  - It is a *scalable*, *fault tolerant*, *open source* framework for the *distributed storing* and *processing* of large sets of data on *commodity hardware*
- But what does all that mean?

What is Apache Hadoop scalability?

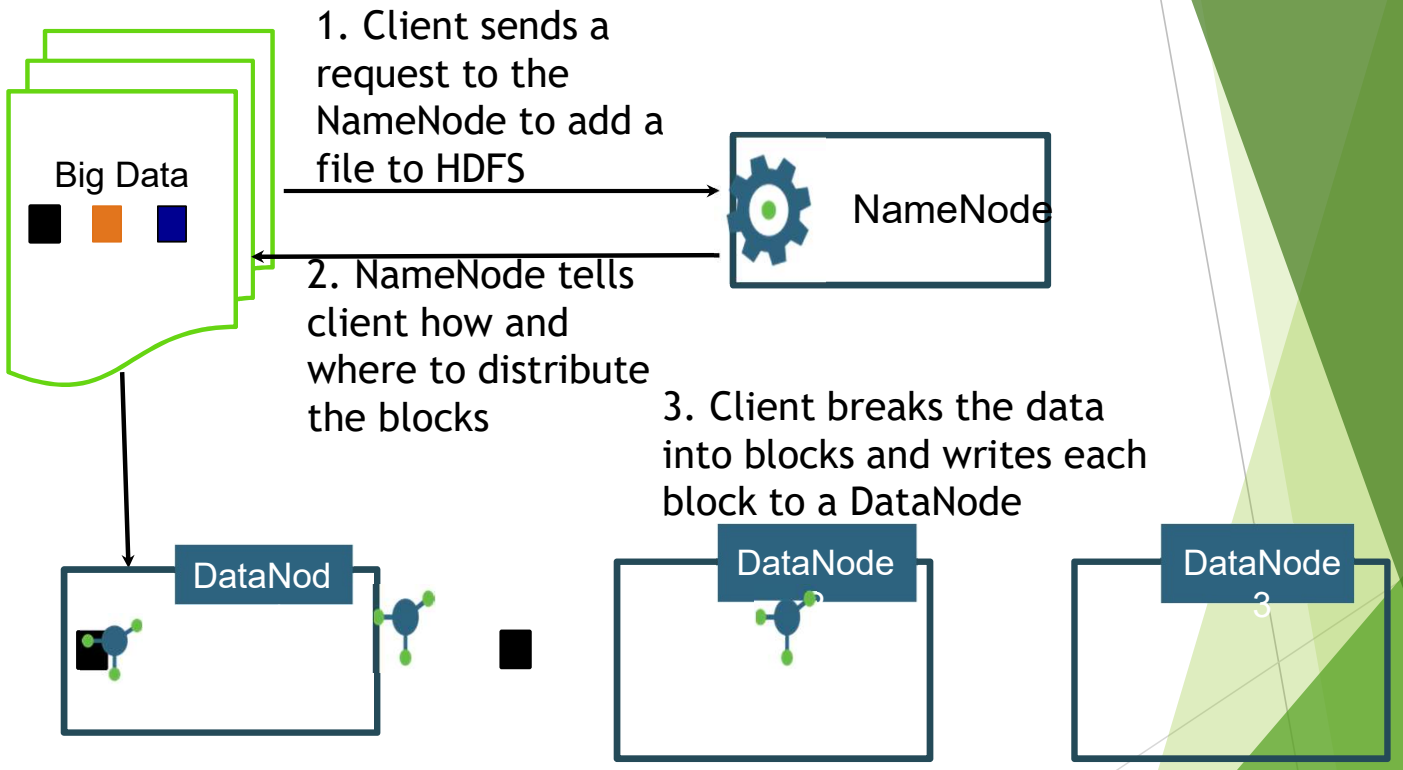
- Well first of all it is scalable.
- Hadoop clusters can range from one machine to thousands of machines. That is scalability!



What is Apache Hadoop fault tolerant?

- It is also fault tolerant
- Hadoop services become *fault tolerant* through *redundancy*
- For example, the Hadoop distributed file system, called *HDFS*, automatically replicates data blocks to three separate machines, assuming that your cluster has at least three machines in it
- Many other Hadoop services are replicated too in order to avoid any single points of failure

What is Apache Hadoop fault tolerant?



4. The DataNode replicates each block to two other DataNodes (as chosen by the NameNode)

What is Apache Hadoop open source?

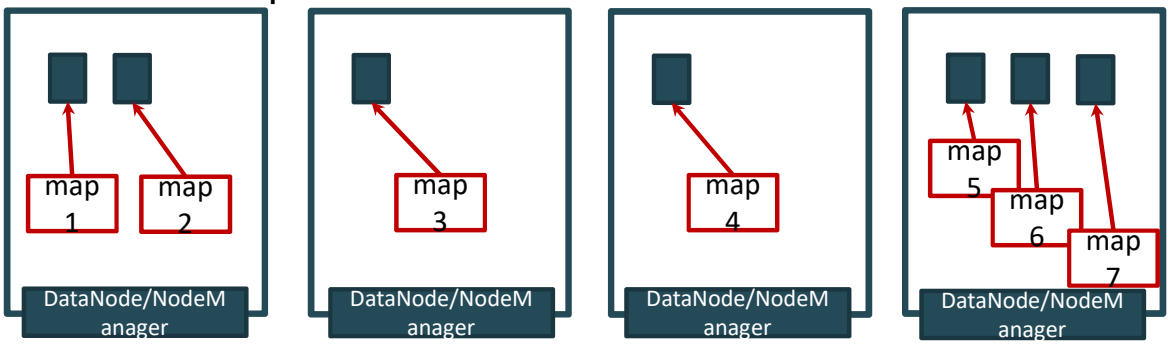
- Hadoop is also open source
- Hadoop development is a community effort governed under the licensing of the Apache Software Foundation
- Anyone can help to improve Hadoop by adding features, fixing software bugs, or improving performance and scalability

What is Apache Hadoop distributed storage and processing?

- Hadoop also uses distributed storage and processing
- Large datasets are automatically split into smaller chunks, called *blocks*, and distributed across the cluster machines
- Not only that, but each machine processes its local block of data. This means that processing is distributed too, potentially across hundreds of CPUs and hundreds of gigabytes of memory

Distributed Processing - MapReduce

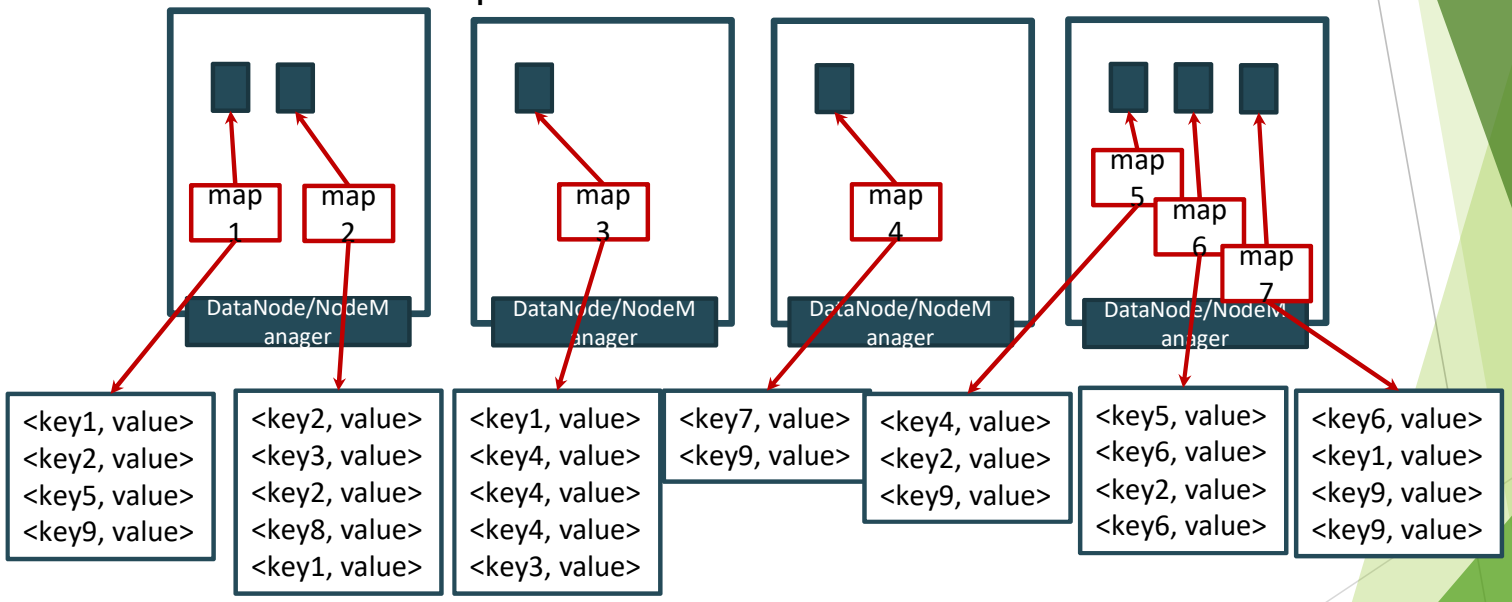
1. Suppose a file is the input to a MapReduce job. That file is broken down into blocks stored on DataNodes across the Hadoop cluster.



2. During the Map phase, map tasks process the input of the MapReduce job, with a map task assigned to each Input Split. The map tasks are Java processes that ideally run on the DataNodes where the blocks are stored.

Distributed Processing - MapReduce cont.

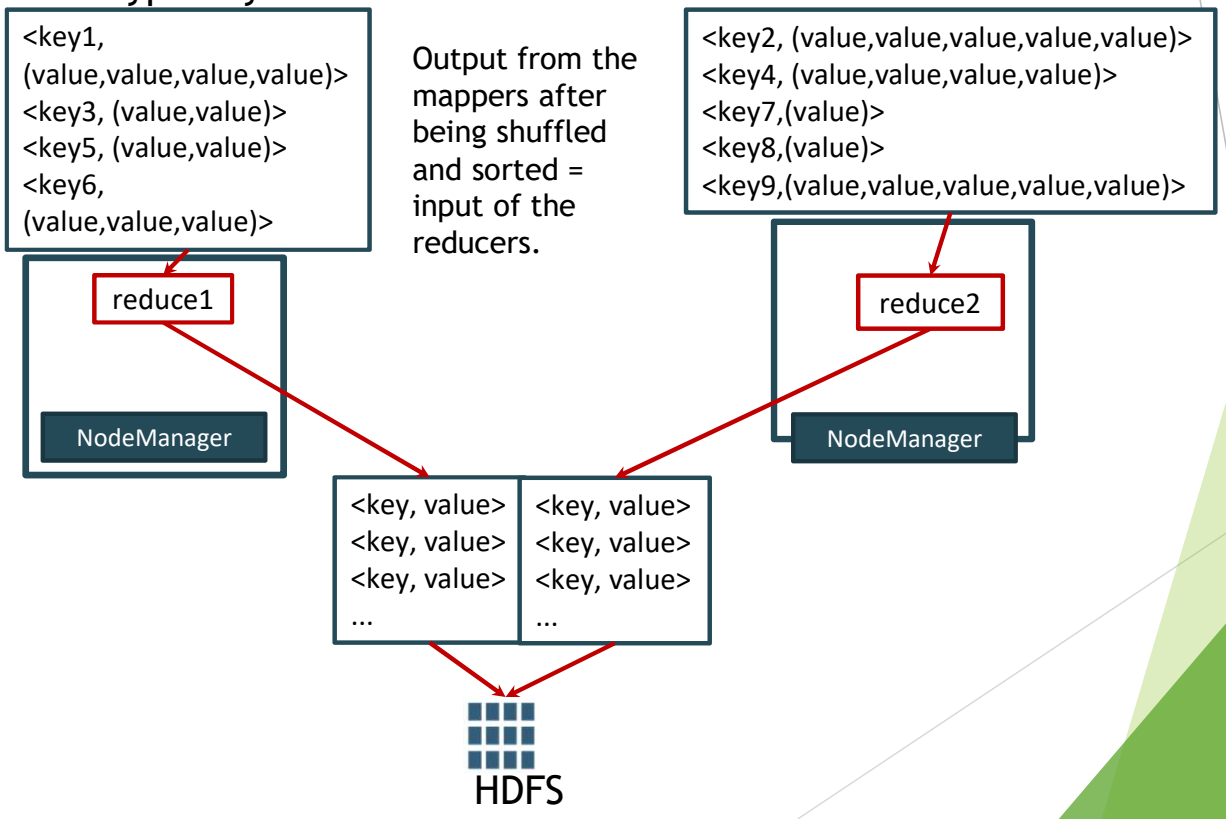
3. Each map task processes its Input Split and outputs records of <key, value> pairs.



4. The <key,value> pairs go through a shuffle/sort phase, where records with the same key end up at the same reducer. The specific pairs sent to a reducer are sorted by key, and the values are aggregated into a collection.

Distributed Processing - MapReduce cont.

5. Reduce tasks run on a NodeManager as a Java process. Each Reducer processes its input and outputs <key,value> pairs that are typically written to a file in HDFS.



What is Apache Hadoop uses comodity hardware?

- All of this occurs on commodity hardware which reduces not only the original purchase price, but also potentially reduces support costs too

Six Key Hadoop DATA TYPES

1. **Sentiment**
   How your customers feel
2. **Clickstream**
   Website visitors' data
3. **Sensor/Machine**
   Data from remote sensors and machines
4. **Geographic**
   Location-based data
5. **Server Logs**
6. **Text**
   Millions of web pages, emails, and documents

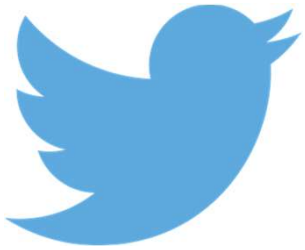
Value

Sentiment Use Case

™ Marvel Comics

- Analyze customer sentiment on the days leading up to and following the release of the movie *Iron Man 3*.
- Questions to answer:
  - How did the public feel about the debut?
  - How might the sentiment data have been used to better promote the launch of the movie?

Getting Twitter Feeds into Hadoop



Flume
Agent

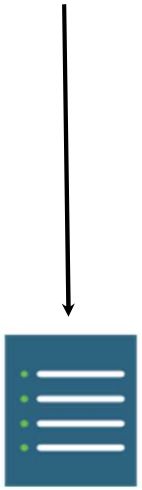
- Iron Man 3 was awesome. I want to go see it again!
- Iron Man 3 = 7.7 stars
- Tony Stark has 42 different Iron Man suits in Iron Man 3
- Wow as good as or better than the first two
- Thor was way better than Iron Man 3

Flume is a tool for streaming data into Hadoop.

Hadoop
cluster

Use HCatalog to Define a Schema

```
CREATE EXTERNAL TABLE tweets_raw (
   id BIGINT,
   created_at STRING,
   source STRING,
   favorited BOOLEAN,
   retweet_count INT,
   text STRING
)
```

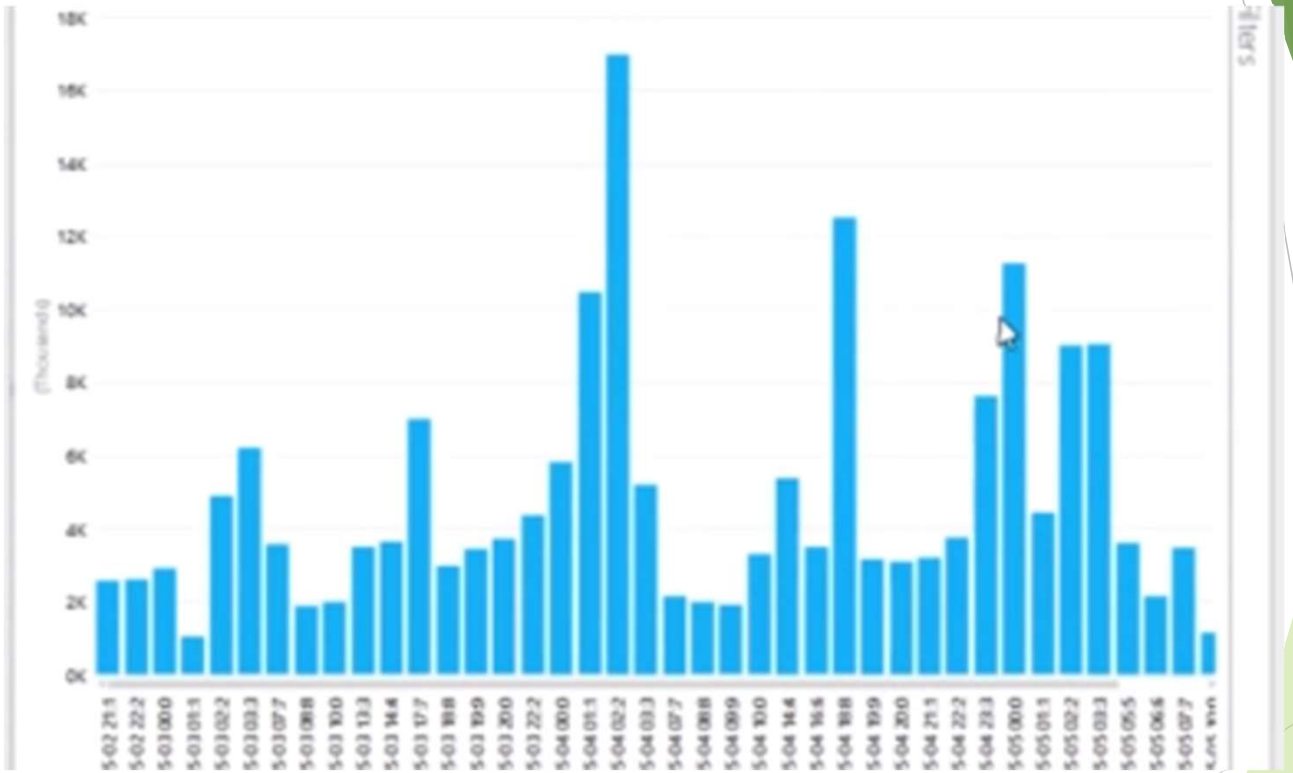


HCatalog metastore

Use Hive to Determine Sentiment

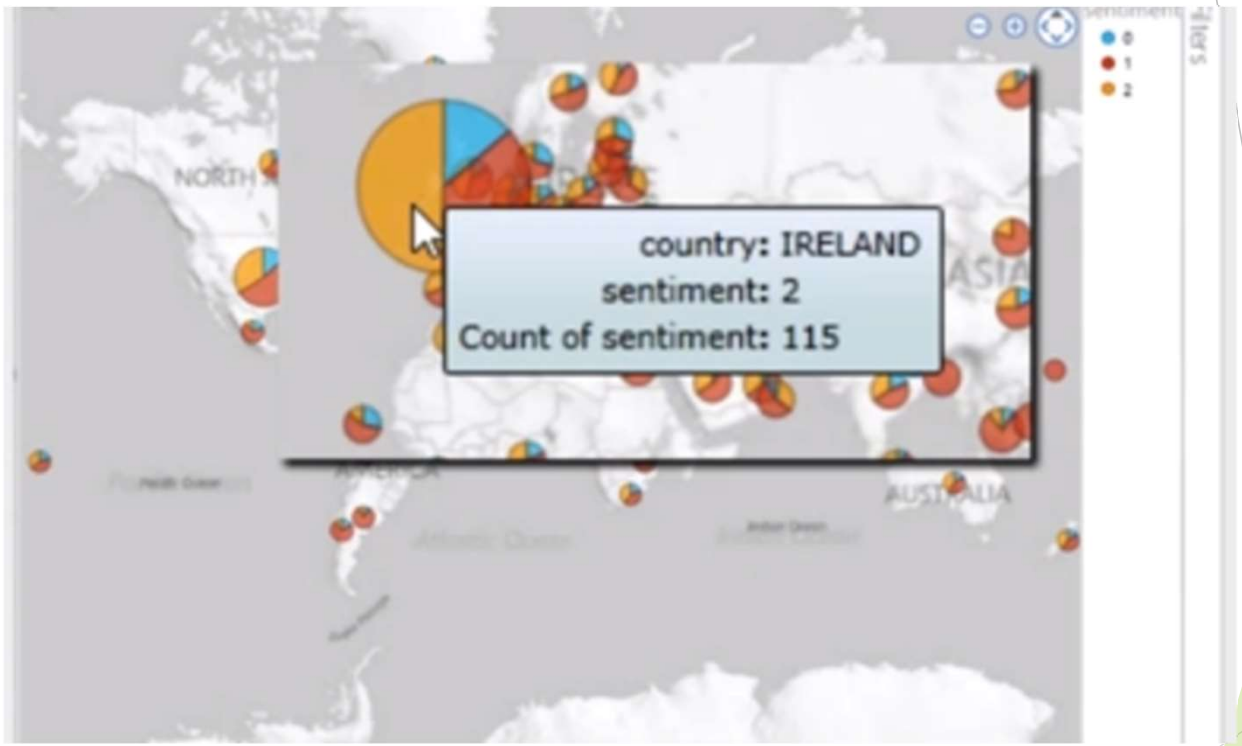
```
CREATE TABLE tweetsbi
STORED AS RCFile
AS
SELECT
  t.*,
  case s.sentiment
    when 'positive' then 2
    when 'neutral' then 1
    when 'negative' then 0
  end as sentiment
FROM tweets_clean t LEFT OUTER JOIN
tweets_sentiment s on t.id = s.id;
```

View Spikes in Tweet Volume



Notice a large spike in tweets around the Thursday midnight opening and spikes around the Friday evening, Saturday afternoon, and Saturday evening showings.

View Sentiment by Country



Viewing the tweets on a map shows the sentiment of the movie by country. For example, Ireland had 50% positive tweets, while 67% of tweets from Mexico were neutral.

Geolocation Use Case

- A trucking company has over 100 trucks.
- The geolocation data collected from the trucks contains events generated while the truck drivers are driving.
- The company's goal with Hadoop is to:
  - reduce fuel costs
  - improve driver safety

The Geolocation Data

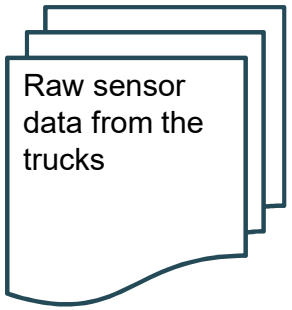
Here is what the collected data from the trucks' sensors looks like:

- truckid
- driverid
- event
- latitude
- longitude
- city
- state
- velocity
- event_indicator (0 or 1)
- idling_indicator (0 or 1)

For example:

- A5 A5 unsafe following distance 41.526509 -124.038407 Klamath California 33 1 0
- A54 A54 normal 35.373292 -119.018712 Bakersfield California 19 0 0
- A48 A48 overspeed 38.752124 -121.288006 Roseville California 77 1 0

Getting the Raw Data into Hadoop



A5 A5 unsafe following distance 41.526509 -124.038407 Klamath California
33 1 0
A54 A54 normal 35.373292 -119.018712 Bakersfield California 19 0 0
A48 A48 overspeed 38.752124 -121.288006 Roseville California 77 1 0
...

Flume
Agent

Flume is a tool for streaming data into Hadoop.

Hadoop
cluster

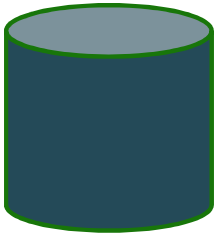
The Truck Data

The truck data is stored in a database and looks like:

- driverid
- truckid
- model
- monthyear_miles
- monthyear_gas
- total_miles
- total_gas
- mileage

The miles and gas figures go back to 2009.

Getting the Truck Data into Hadoop



A table in RDBMS containing the info on the fleet of trucks.

A Sqoop job

Hadoop cluster

Sqoop is a tool for transferring data between an RDBMS and Hadoop.

HCatalog Stores a Shared Schema

```
create table trucks (
driverid string,
truckid string,
model string,
monthyear_miles int,
monthyear_gas int,
total_miles int,
total_gas double,
mileage double
);
```

```
create table events (
truckid string,
driverid string,
event string,
latitude double,
longitude double,
city string,
state string,
velocity double
event_indicator boolean,
idling_indicator boolean
);
```

```
create table
riskfactor (
    driverid string,
    riskfactor float
);
```

HCatalog metastore

Data Analysis

We want to answer two questions:

- Which trucks are wasting fuel through unnecessary idling?
- Which drivers are most frequently involved in unsafe events on the road?

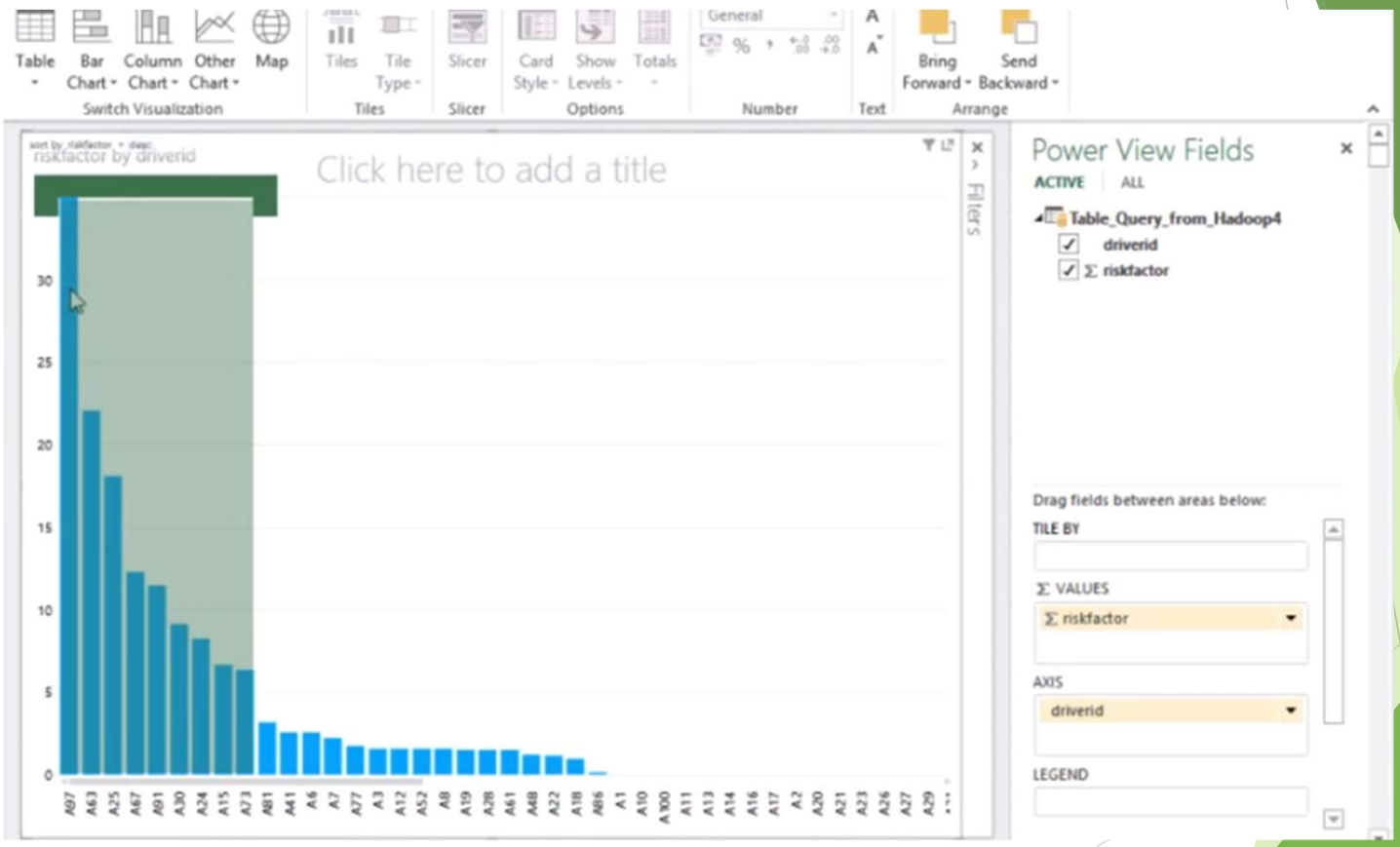
Use Hive to Compute Truck Mileage

```
CREATE TABLE truck_mileage AS
  SELECT truckid, rdate, miles,
gas,
         miles/gas mpg
  FROM trucks
  LATERAL VIEW stack(54,
'jun13',jun13_miles,jun13_gas,'may1
3',may13_miles,may13_gas,'apr13',ap
r13_miles,apr13_gas,...
) dummyalias AS rdate, miles, gas;
```

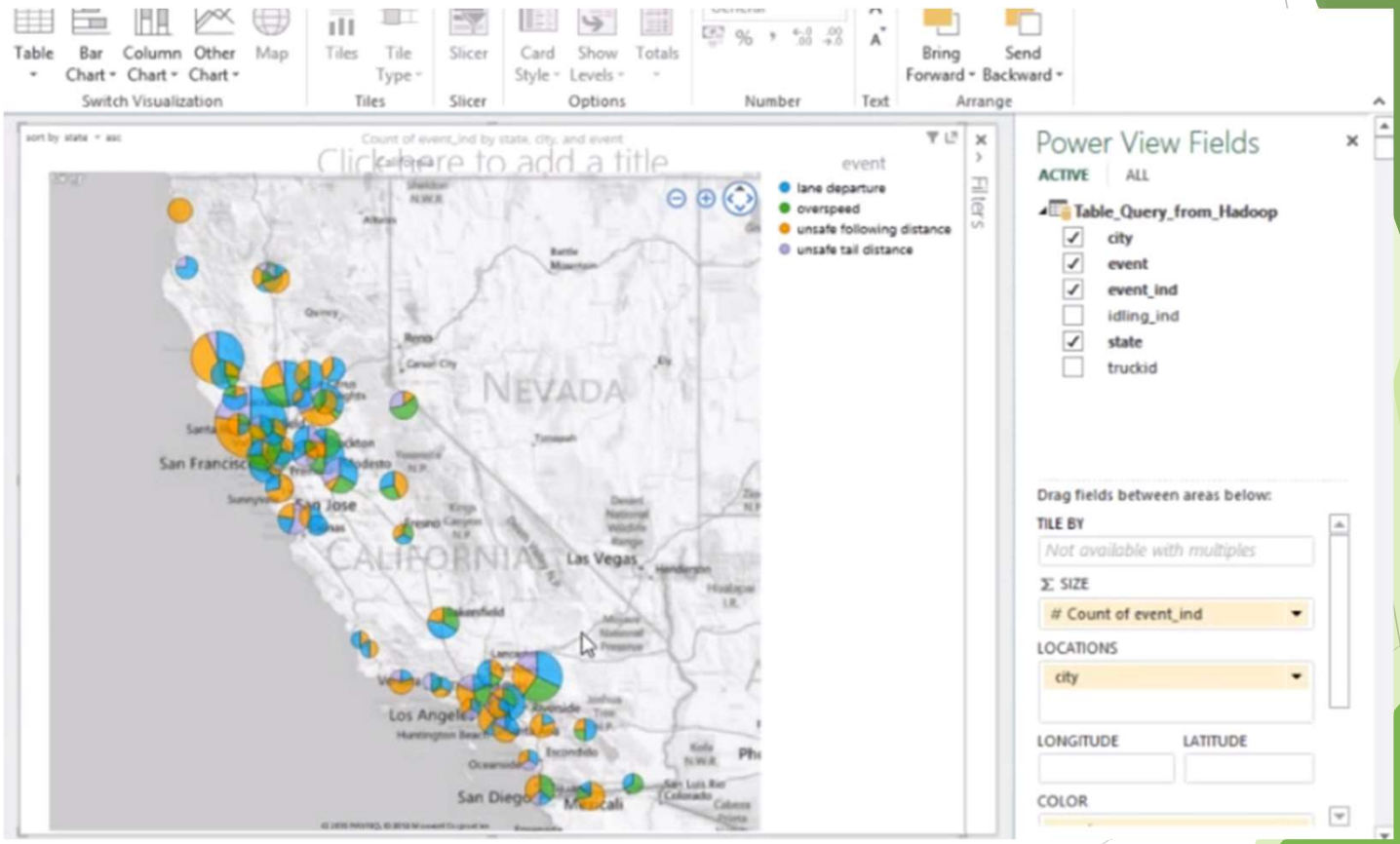
Use Pig to Compute a Risk Factor

```
a = LOAD 'events'
      using org.apache.hive.hcatalog.pig.HCatLoader();
b = filter a by event != 'Normal';
c = foreach b
        generate driverid, event, (int) '1' as occurance;
d = group c by driverid;
e = foreach d generate group as driverid,
                       SUM(c.occurance) as t_occ;
f = LOAD 'trucks'
      using org.apache.hive.hcatalog.pig.HCatLoader();
g = foreach f generate driverid,
    ((int) apr09_miles + (int) apr10_miles) as t_miles;
join_d = join e by (driverid), g by (driverid);
final_data = foreach join_d generate
           $0 as driverid, (float) $1/$3*1000 as riskfactor;
store final_data into 'riskfactor'
      using org.apache.hive.hcatalog.pig.HCatStorer();
```

Risk Factors Viewed in a Graph



Risk Factors Viewed on a Map



About Hadoop

- Framework for solving data-intensive processes
- Designed to scale massively
- Very fast for very large jobs
- Variety of processing engines
- Designed for hardware and software failures

Relational Databases vs. Hadoop

Relational		Hadoop
Required on write	schema	Required on read
Reads are fast	speed	Writes are fast
Standards and structured	governance	Loosely structured
Limited, no data processing	processing	Processing coupled with data
Structured	data types	Multi- and unstructured
Interactive OLAP Analytics Complex ACID Transactions Operational Data Store	best fit use	Data Discovery Processing unstructured data Massive Storage/Processing

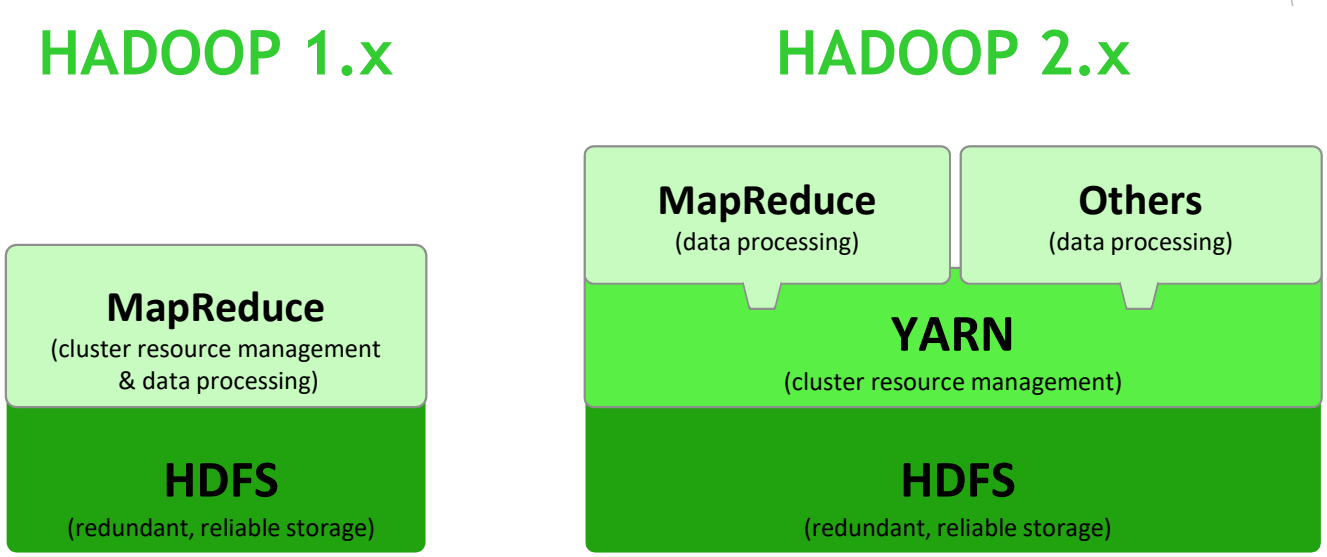
About Hadoop 2.x

The Apache Hadoop 2.x project consists of the following modules:

- **Hadoop Common**: the utilities that provide support for the other Hadoop modules
- **HDFS**: the Hadoop Distributed File System
- **YARN**: a framework for job scheduling and cluster resource management
- **MapReduce**: for processing large data sets in a scalable and parallel fashion

New in Hadoop 2.x

YARN is a re-architecture of Hadoop that allows multiple applications to run on the same platform



The Hadoop Ecosystem

Apache Ambari
http://incubator.apache.org/ambari

APACHE FALCON

oozie

Spark

HIVE

Hadoop

Apache Solr

APACHE HBASE

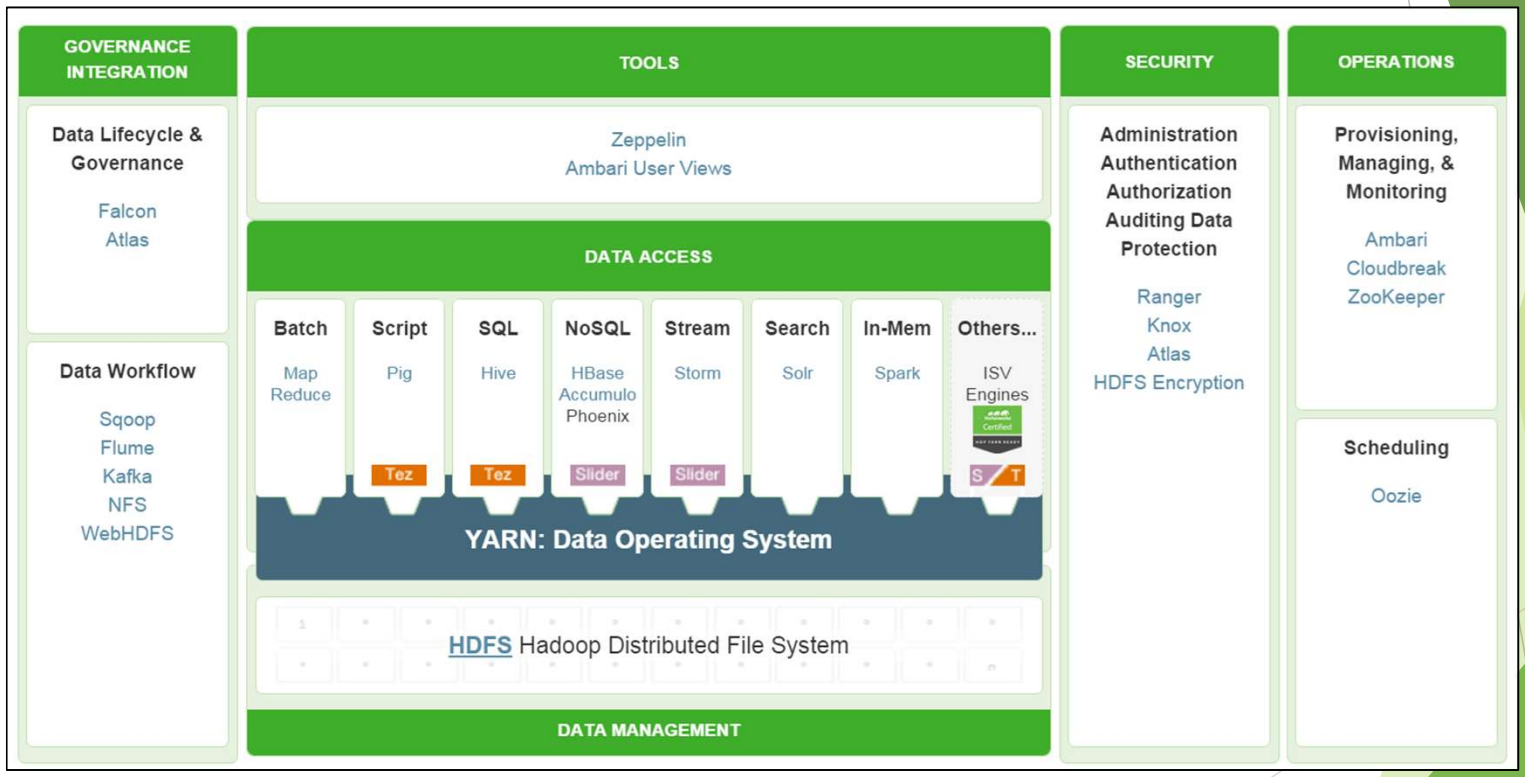
accumulo

mahout

FLUME

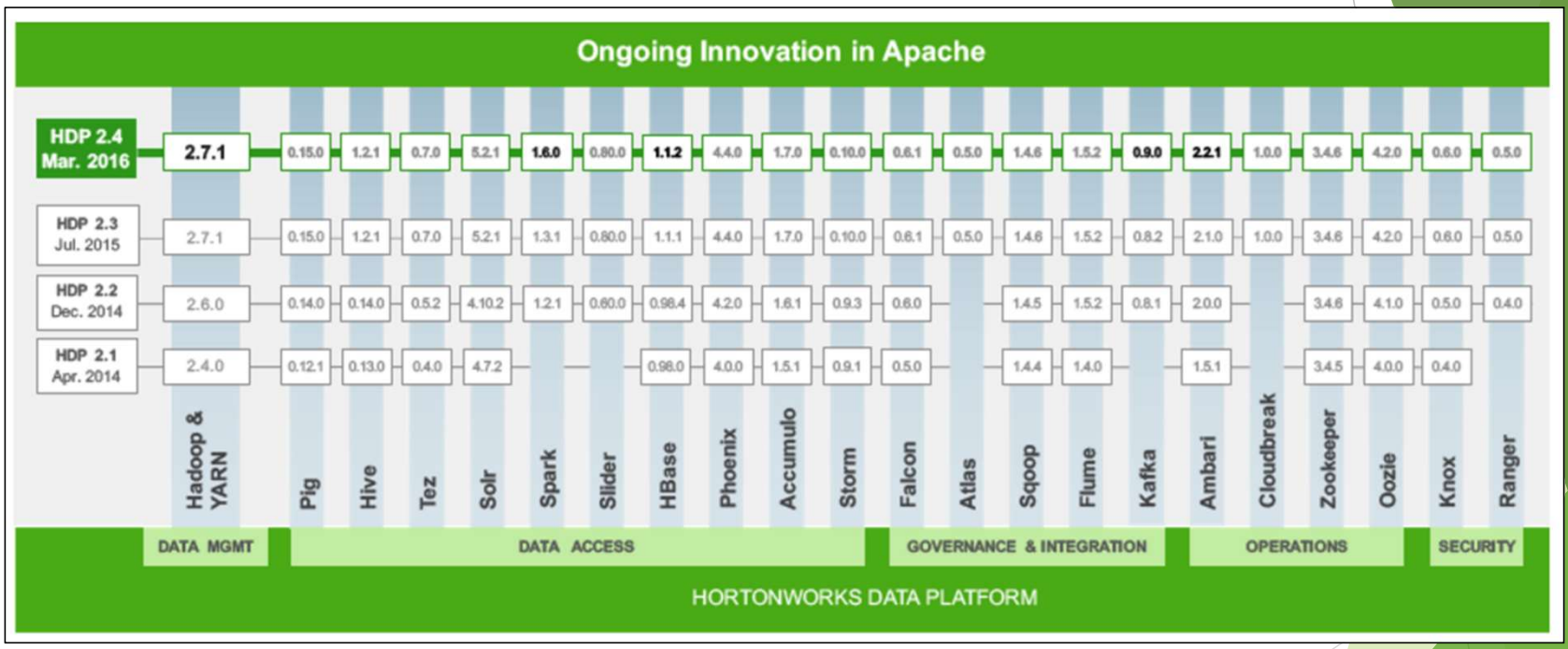
STORM

Enterprise ready Hadoop Platforms



Source – https://hortonworks.com/products/data-center/hdp/

Hadoop distros - HDP



Source – https://hortonworks.com/products/data-center/hdp/

Data Management and Operations Frameworks

Framework	Description
Hadoop Distributed File System (HDFS)	A Java-based, distributed file system that provides scalable, reliable, high-throughput access to application data stored across commodity servers
Yet Another Resource Negotiator (YARN)	A framework for cluster resource management and job scheduling

Framework	Description
Ambari	A Web-based framework for provisioning, managing, and monitoring Hadoop clusters
ZooKeeper	A high-performance coordination service for distributed applications
Cloudbreak	A tool for provisioning and managing Hadoop clusters in the cloud
Oozie	A server-based workflow engine used to execute Hadoop jobs

These brief descriptions are provided for quick convenience. More detailed descriptions are available online

Data Access Frameworks

Framework	Description
Pig	A high-level platform for extracting, transforming, or analyzing large datasets
Hive	A data warehouse infrastructure that supports ad hoc SQL queries
HCatalog	A table information, schema, and metadata management layer supporting Hive, Pig, MapReduce, and Tez processing
Cascading	An application development framework for building data applications, abstracting the details of complex MapReduce programming
HBase	A scalable, distributed NoSQL database that supports structured data storage for large tables
Phoenix	A client-side SQL layer over HBase that provides low-latency access to HBase data
Accumulo	A low-latency, large table data storage and retrieval system with cell-level security
Storm	A distributed computation system for processing continuous streams of real-time data
Solr	A distributed search platform capable of indexing petabytes of data
Spark	A fast, general purpose processing engine use to build and run sophisticated SQL, streaming, machine learning, or graphics applications.

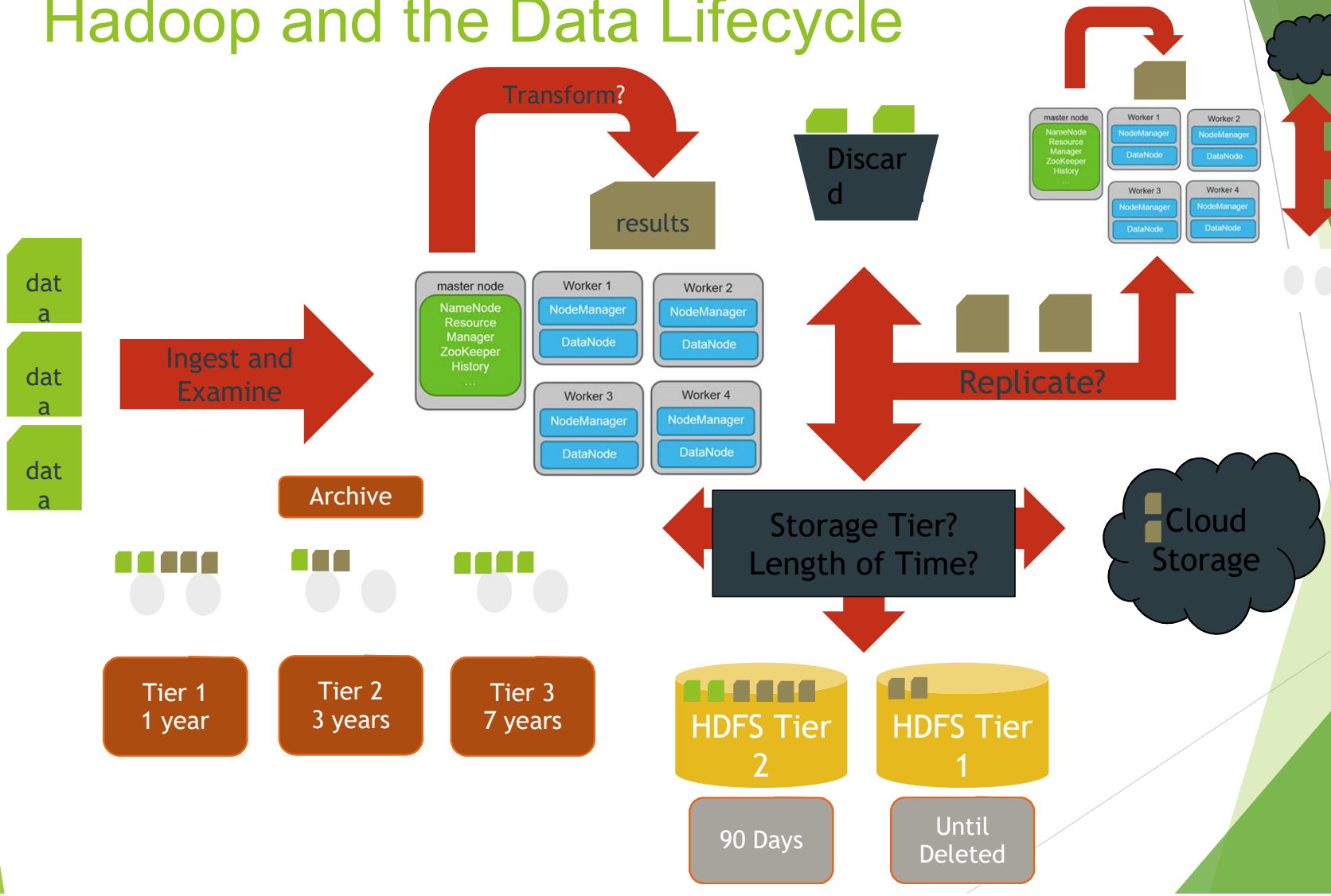
Governance and Integration Frameworks

Framework	Description
Falcon	A data governance tool providing workflow orchestration, data lifecycle management, and data replication services.
WebHDFS	A REST API that uses the standard HTTP verbs to access, operate, and manage HDFS
HDFS NFS Gateway	A gateway that enables access to HDFS as an NFS mounted file system
Flume	A distributed, reliable, and highly-available service that efficiently collects, aggregates, and moves streaming data
Sqoop	A set of tools for importing and exporting data between Hadoop and RDBM systems
Kafka	A fast, scalable, durable, and fault-tolerant publish-subscribe messaging system
Atlas	A scalable and extensible set of core governance services enabling enterprises to meet compliance and data integration requirements

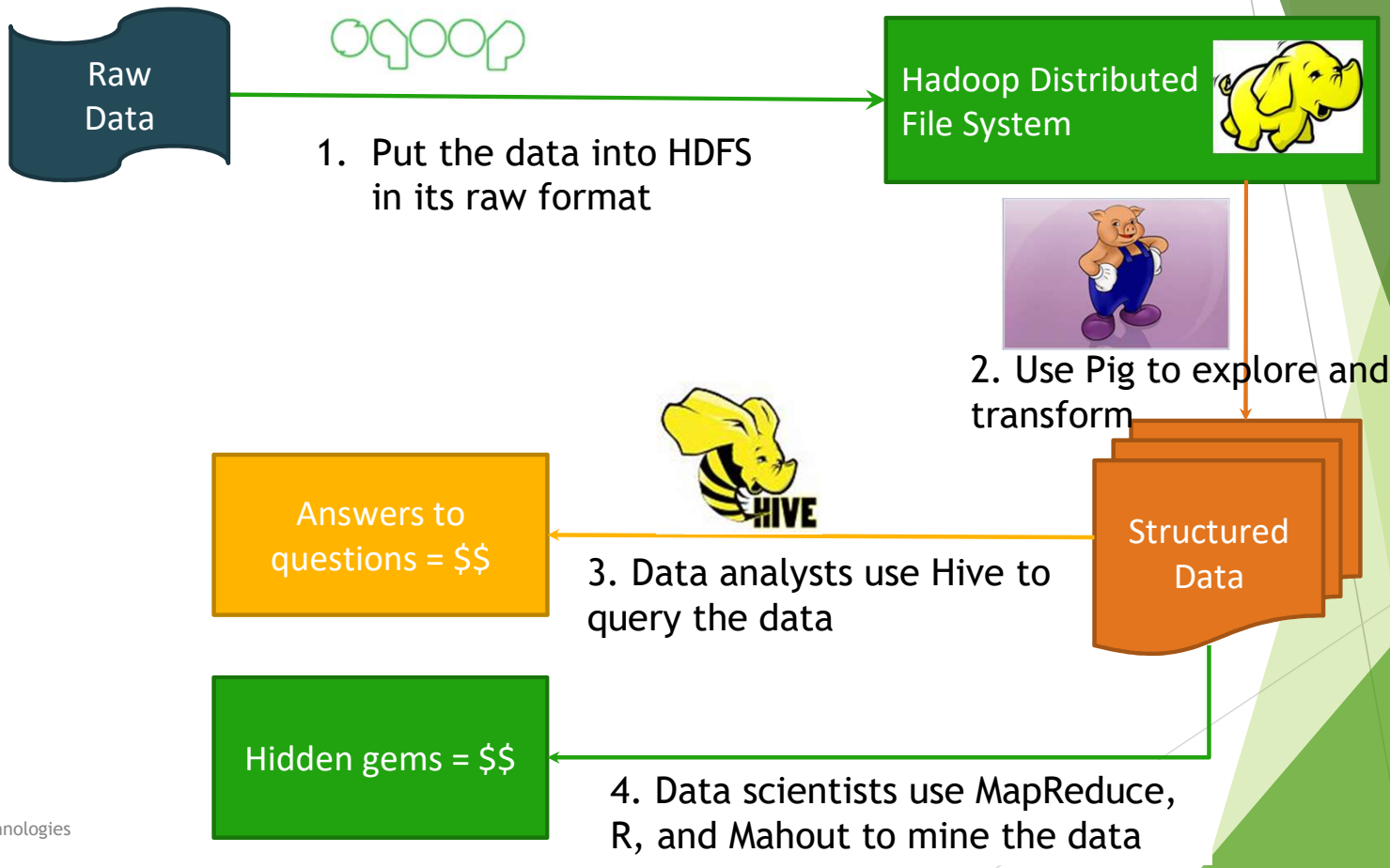
Security Frameworks

Framework	Description
HDFS	A storage management service providing file and directory permissions, even more granular file and directory access control lists, and transparent data encryption
YARN	A resource management service with access control lists controlling access to compute resources and YARN administrative functions
Hive	A data warehouse infrastructure service providing granular access controls to table columns and rows
Falcon	A data governance tool providing access control lists that limit who may submit Hadoop jobs
Knox	A gateway providing perimeter security to a Hadoop cluster
Ranger	A centralized security framework offering fine-grained policy controls for HDFS, Hive, HBase, Knox, Storm, Kafka, and Solr

Hadoop and the Data Lifecycle

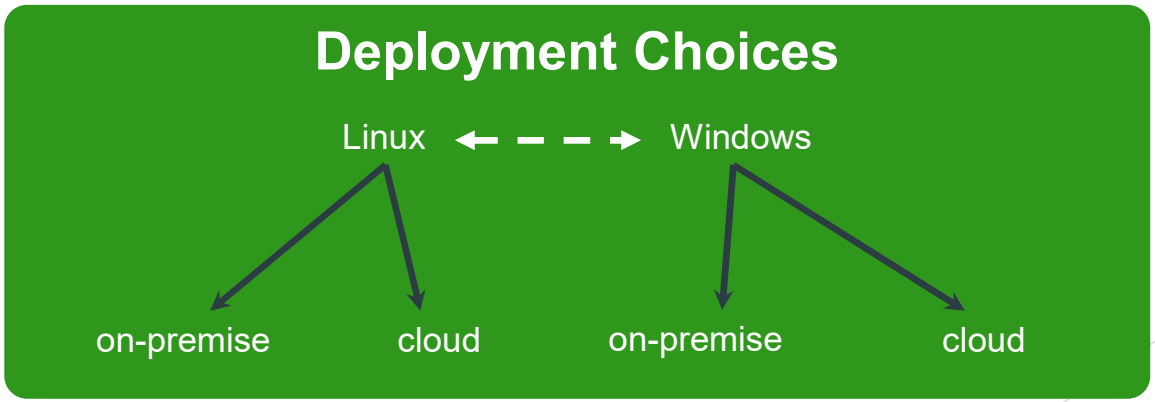


The Path to ROI



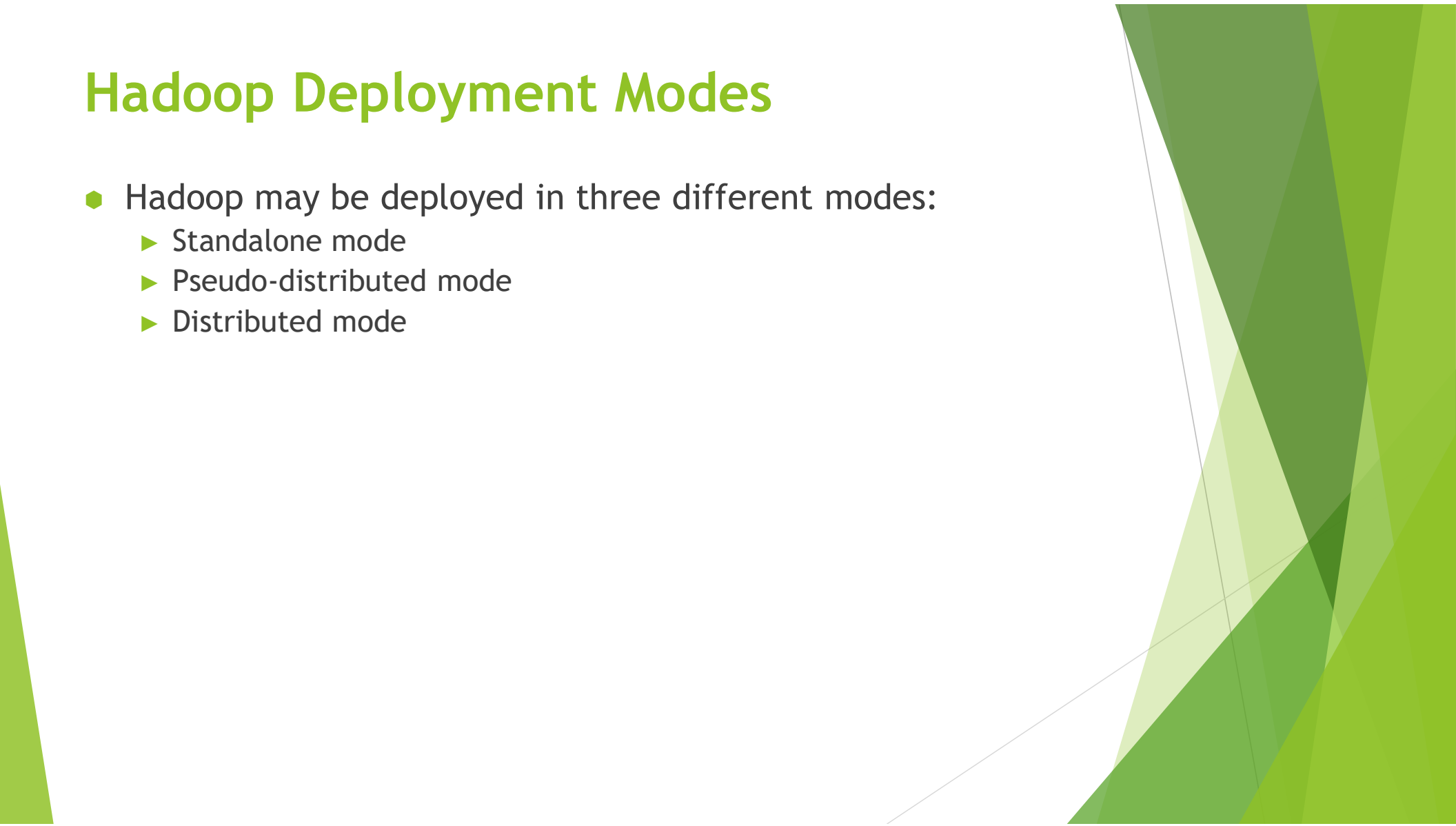
Hadoop Deployment Options

- There are choices when deploying Hadoop:
  - Deploy on-premise in your own data center
  - Deploy in the cloud
  - Deploy on Microsoft Windows
  - Deploy on Linux

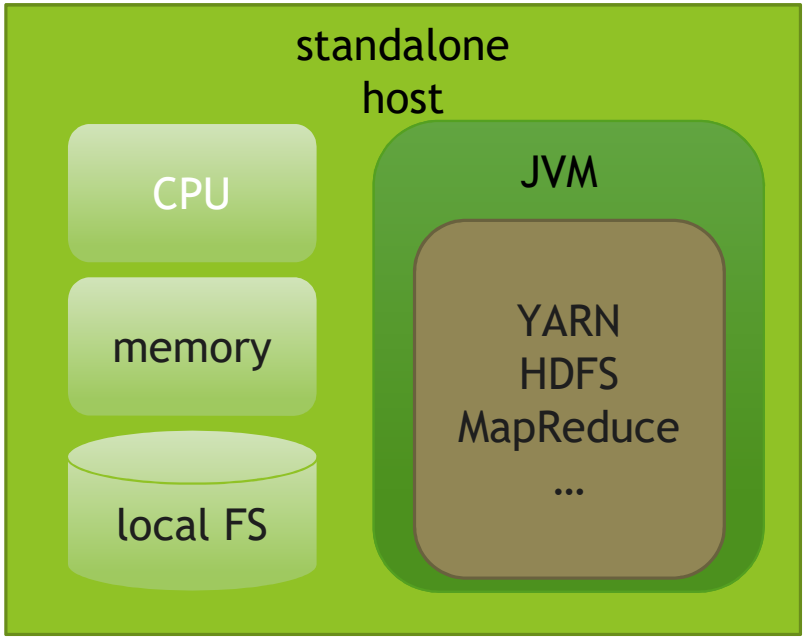


Hadoop Deployment Modes

- Hadoop may be deployed in three different modes:
  - Standalone mode
  - Pseudo-distributed mode
  - Distributed mode

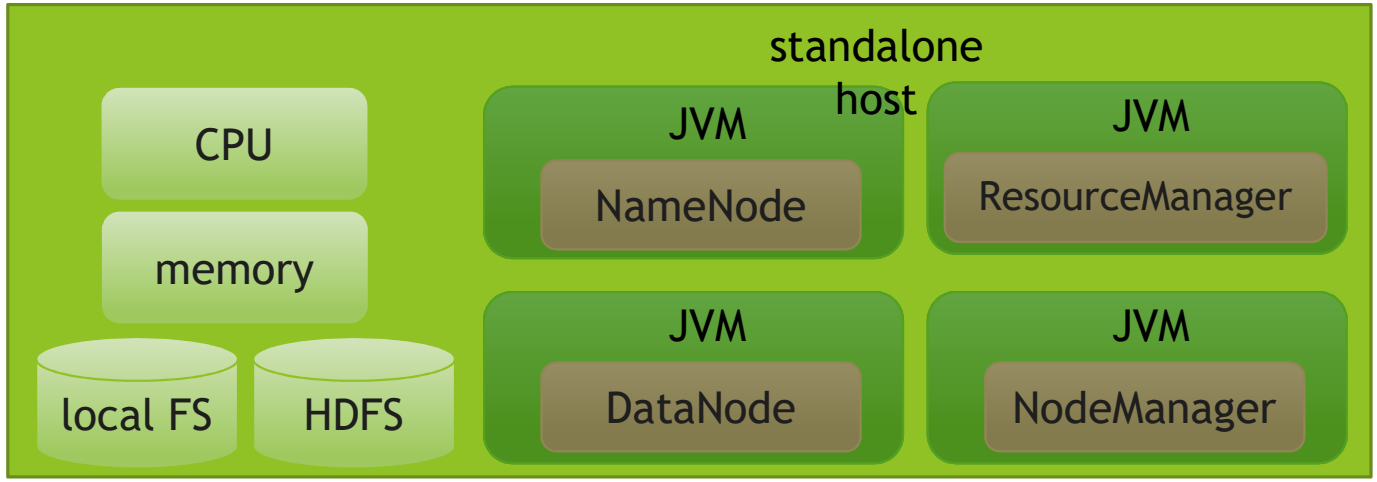


Standalone Mode



- Single system installation
- All Hadoop service daemons run in a single Java virtual machine (JVM)
- Uses the file system on local disk
- Suitable for test and development, or introductory training

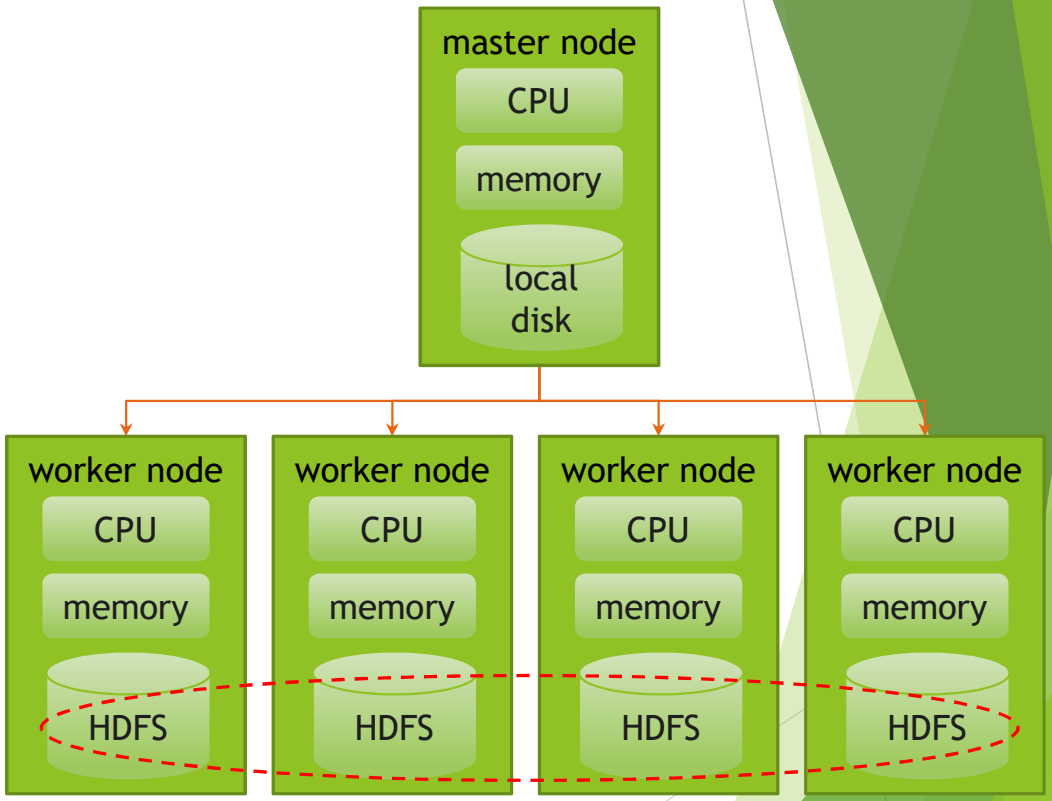
Pseudo-Distributed Mode



- Single system installation
- Each Hadoop service daemon runs in its own JVM
- Uses HDFS on local disk(s)
- Appropriate for quality assurance, test and development
- Format used for the Hortonworks Sandbox

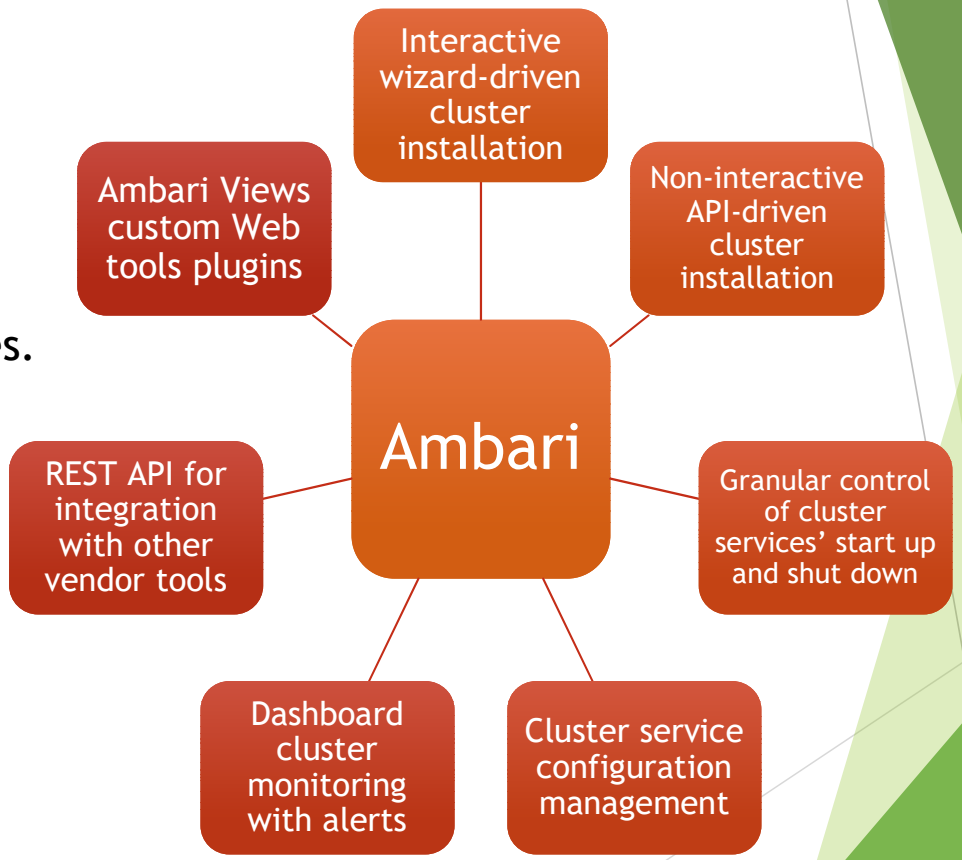
Distributed Mode

- Multi-system installation
- Each Hadoop service daemon runs in its own JVM.
  - Multiple JVMs per system is common
- Uses HDFS on local disk(s)
  - HDFS is distributed across systems
- Best and typical for production environments

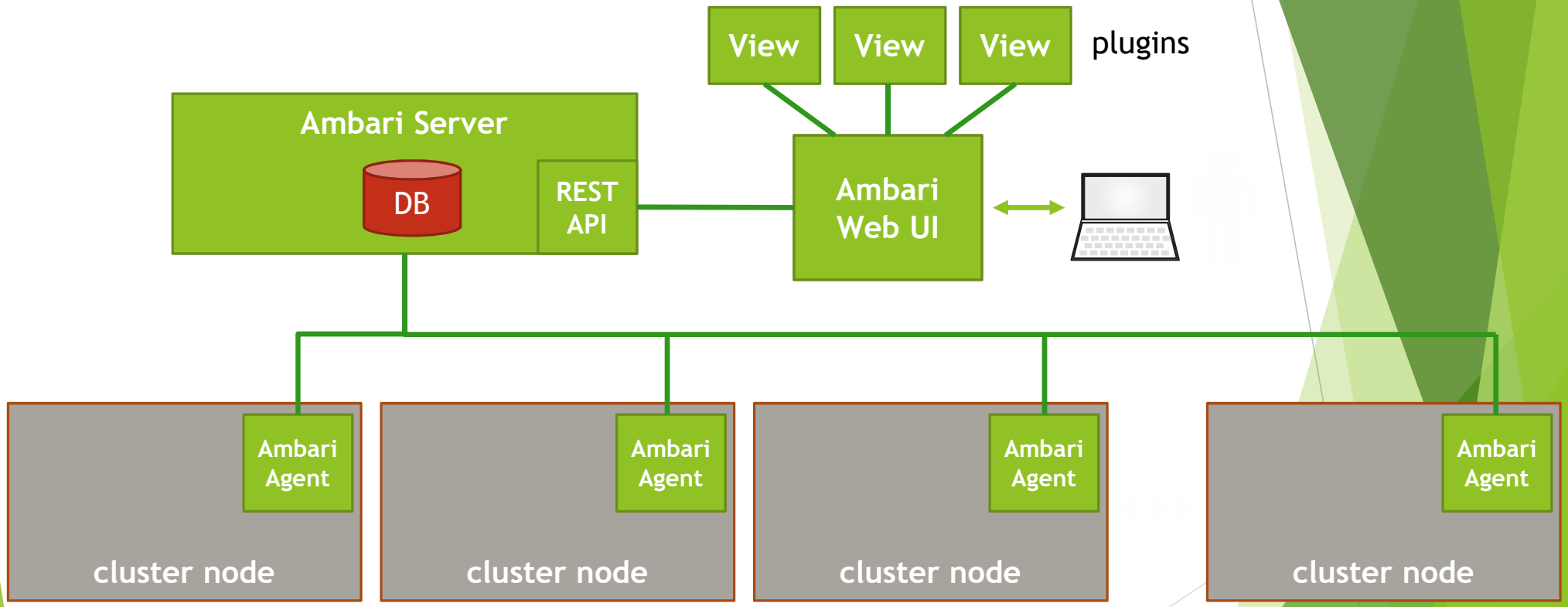


Ambari Cluster Management Features

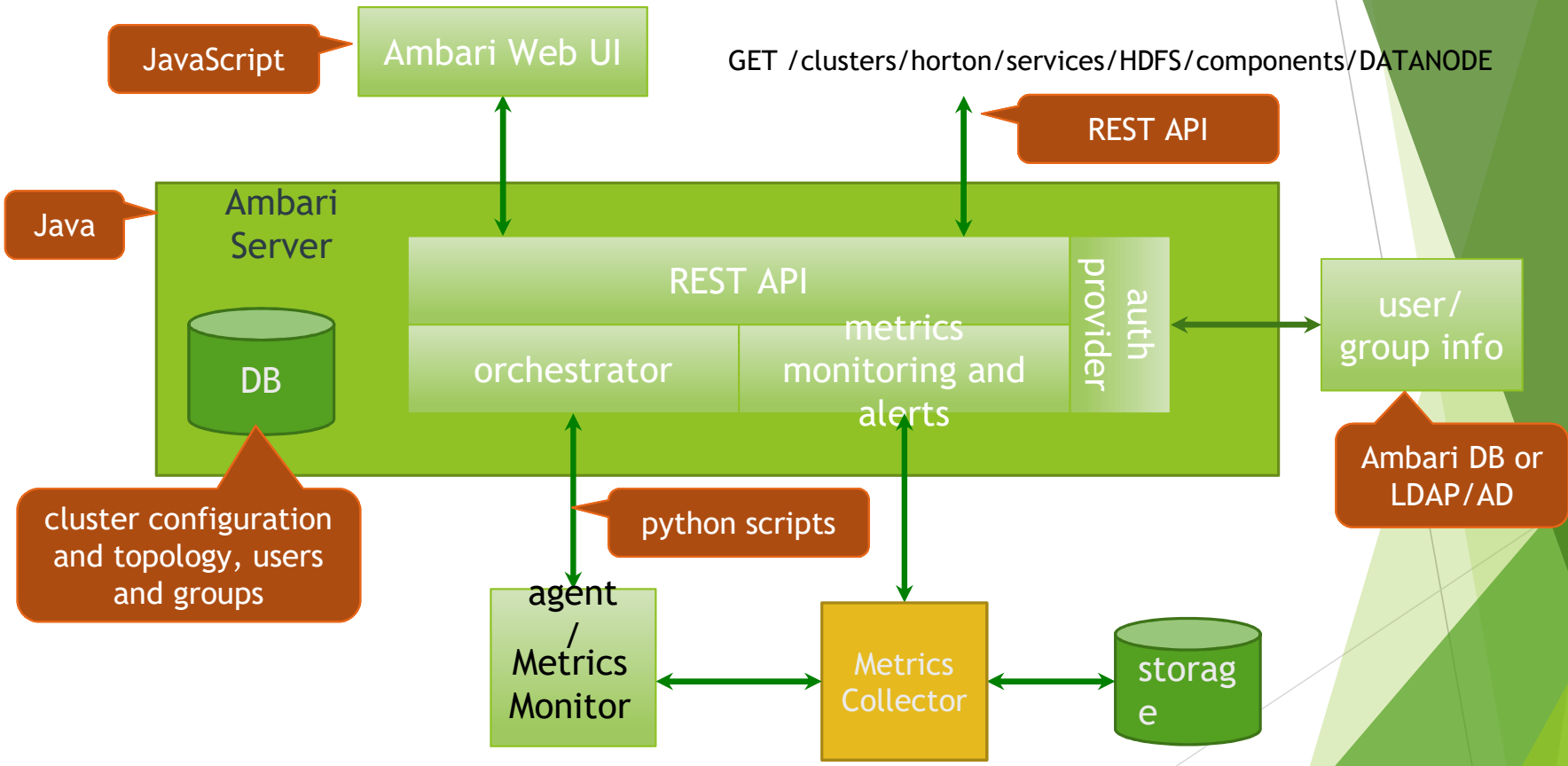
- Ambari is the primary management interface in today's Hadoop cluster.
- Ambari provides a single control point with many important features.



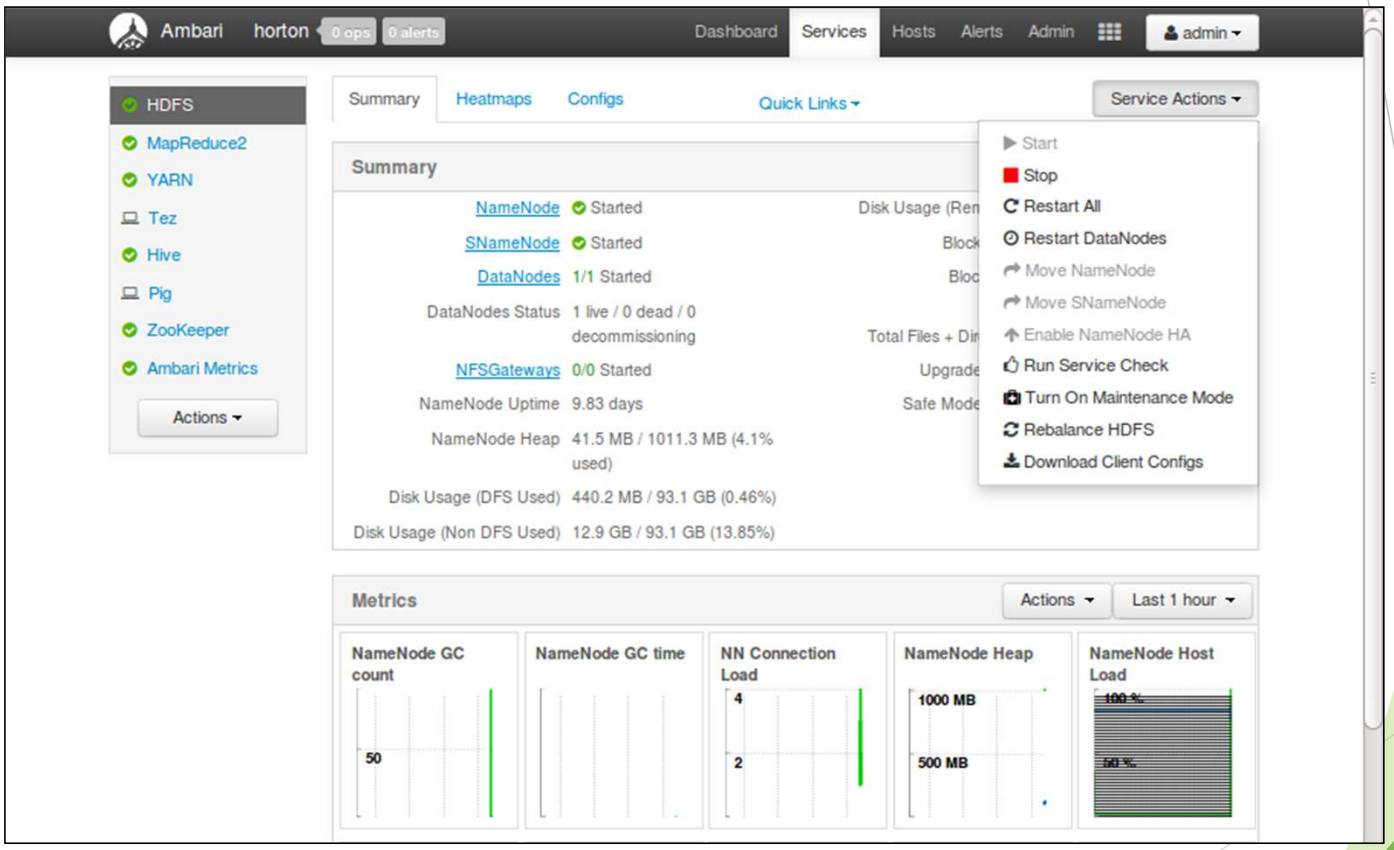
Ambari Architecture



Ambari Server Architecture

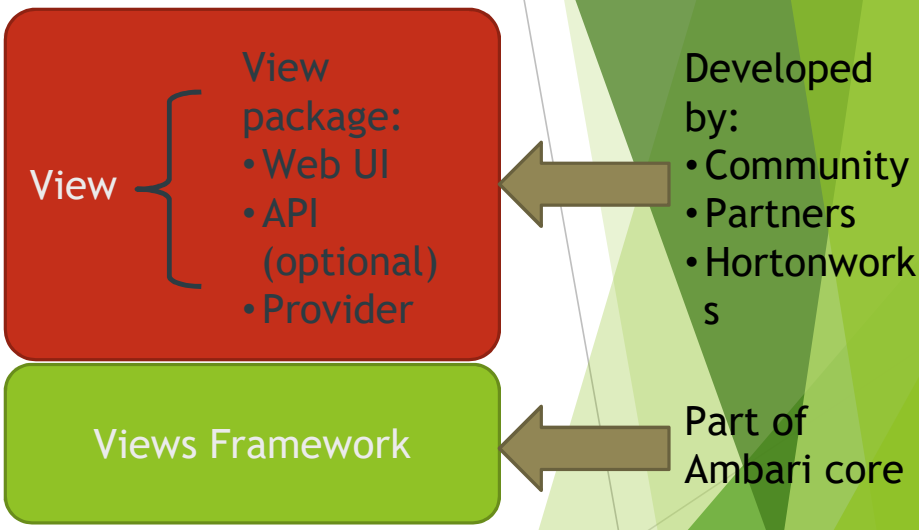


The Ambari Web UI

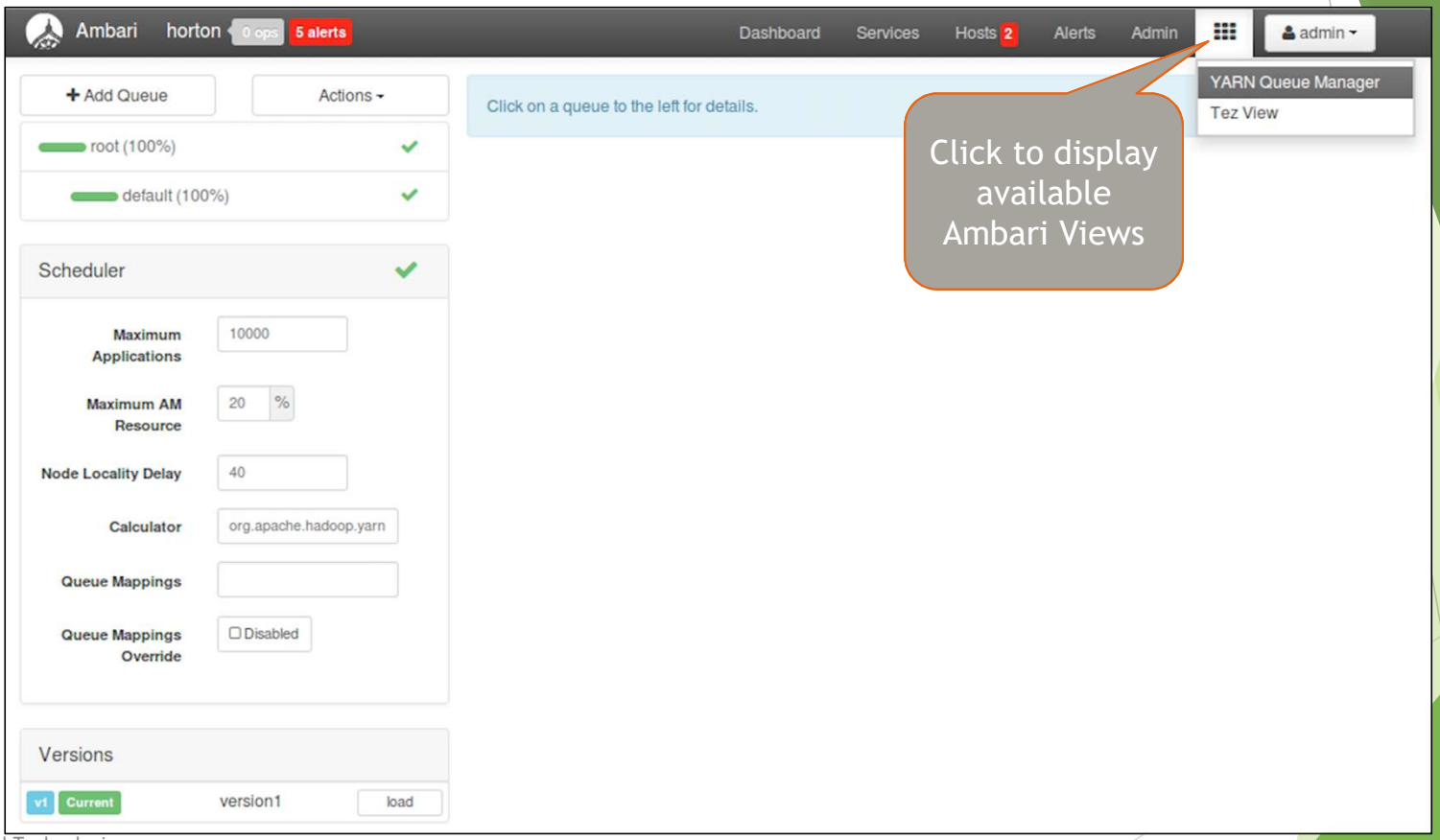


Ambari Views

- Views are Web applications that are plugged into Ambari.
- Views enable organizations to extend and customize Ambari Web.
- Developers write View packages
- Administrators deploy View packages to the Ambari server.
  - Includes server and client-side software, and possibly new APIs
- Ambari administrators create View instances.
- Administrators entitle users to access specific Views.



Sample Ambari View



Lesson Review

1. What are 1,024 petabytes known as?

1. What are 1,024 exabytes known as?

1. List the three Vs. of big data

1. Sentiment is one of the six key types of big data. List the other five.

1. What technology might you use to stream Twitter feeds into Hadoop?

1. What technology might you use to define, store, and share the schemas of your big data stored in Hadoop?

7. What are the two main new components in Hadoop 2.x?

Lab: Start an HDP 2.6 Cluster