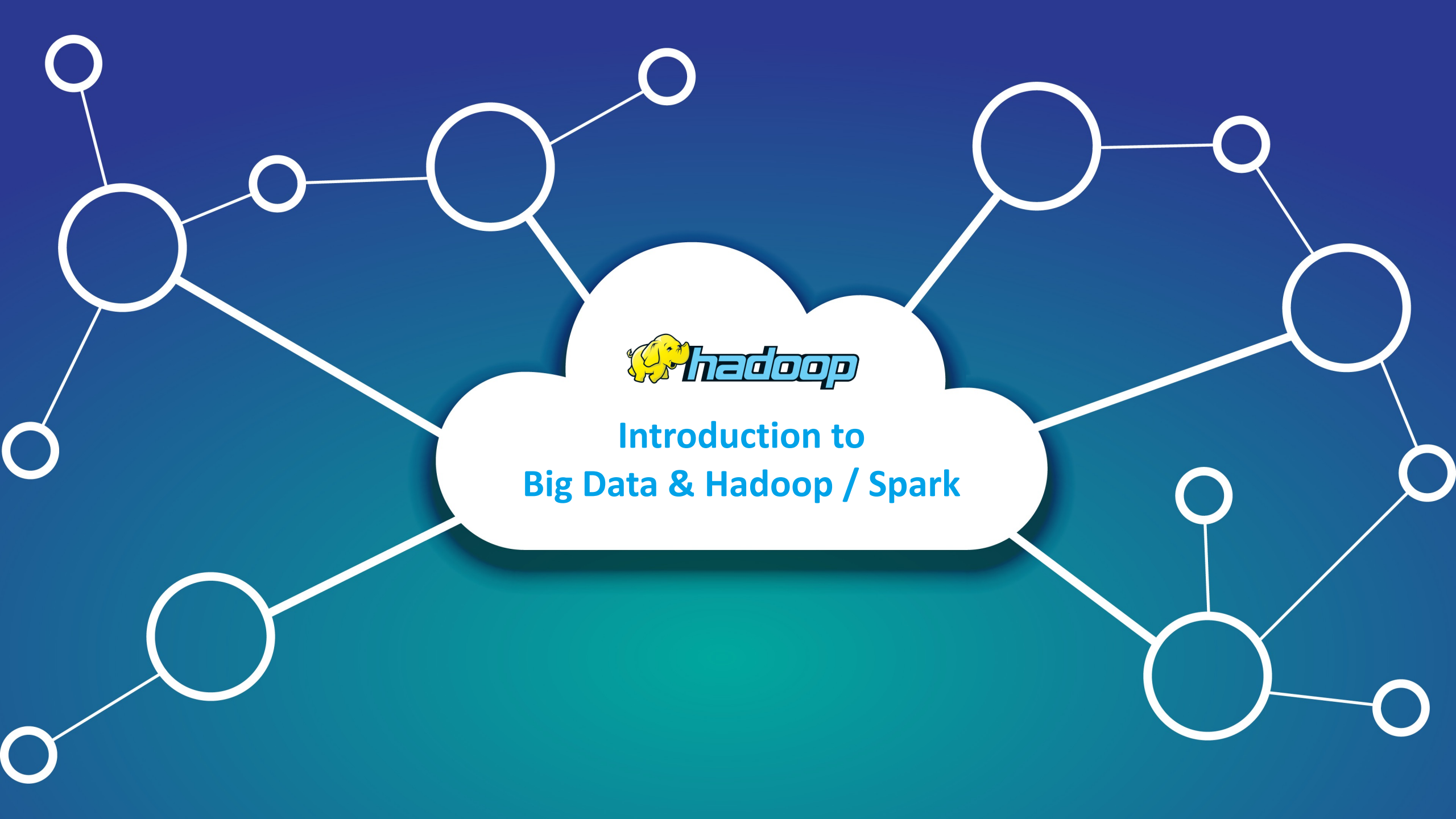

hadoop



Introduction to
Big Data & Hadoop / Spark

Contents

Relational Databases

- ✓ A Relational Database is a data structure that allows you to link information from different tables
- ✓ It normalizes data in to structures
- ✓ A schema is used to strictly define Tables, Columns, Indexes and relations between the tables
  - ✓ Same items in the table are stored in the same table locations(rows/columns)
  - ✓ Relational databases can save data in multiple-joined tables
- ✓ All Relational databases use Structured Query Language (SQL)

Relational Databases

- ✓ Are best suited for OLTP(Online Transaction Processing)
  - ✓ OLTP facilitates and manages transaction oriented applications(Typically data entry and retrieval)
  - ✓ An ATM machine transaction is an OLTP example
- ✓ Relational DBs are used in Enterprise applications/scenarios
  - ✓ Exception is MySQL which is used for web applications

Relational Databases

- ✓ Disadvantages
  - ✓ Inability to scale out (Horizontally) to the needs of Big data Applications
  - ✓ Requires expensive hardware to scale up(Vertically), since the performance is dependent on that
  - ✓ Requires more investment to span a distributed system

What is a Data Warehouse?

- ✓ Is a collection of data integrated from multiple sources , which then undergoes complex long queries for analytical decision and management reporting.
- ✓ Used for Business Intelligence
- ✓ Tools like Cognos, JasperSoft, SQL Server Reporting Service,Oracle Hyperion,SAP Netweaver
- ✓ Used to pull in very large and complex datasets
- ✓ Usually used by management to do queries on data.(Such as current performance Vs. targets, Sales Reporting, Analysing Sensor information etc)
- ✓ Examples: Informatica, Teradata, SAP HANA

Online Transaction Processing (OLTP)

OLTP Example:

- ✓ Order number: 21210021
  - ✓ Pulls up a data row such as Name, date, Address to Deliver to, Delivery status etc

OLAP Example

- ✓ Net profits for Company A for the Digital Radio product
- ✓ Pulls in a large no. of records
  - ✓ Sum of radios sold in the Company A
  - ✓ Sum of radios sold in the Pacific
  - ✓ Unit cost of a radio in each region
  - ✓ Sales price of each radio
- ✓ Data warehousing databases use different type of architecture both from a database perspective and infrastructure layer

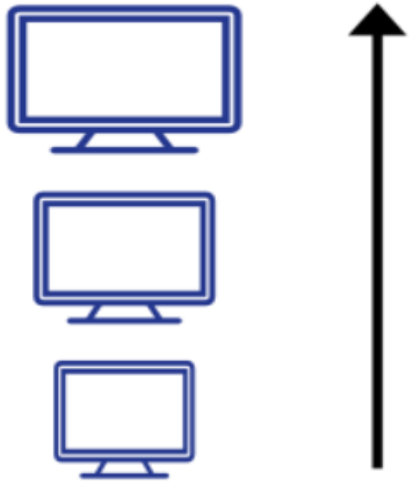
What is Scaling?

- ✓ Scaling refers to the system in which there is a possibility of extending the system as the number of users and resources grow with time.
- ✓ The system should be capable enough to handle the increasing loads without changing the application software.
- ✓ Scalability is generally considered concerning hardware and software. In hardware, scalability refers to the ability to change workloads by altering hardware resources such as processors, memory, and hard disc space.
- ✓ Software scalability refers to the capacity to adapt to changing workloads by altering the scheduling mechanism and parallelism level.

Types of Scaling

VERTICAL SCALING

Increase size of instance
(RAM, CPU etc.)



HORIZONTAL SCALING

(Add more instances)

What is Vertical Scaling?

- ✓ Vertical Scalability refers to the addition of more power to the existing pool of resources like servers. For example, MySQL server. Here, scaling is carried out by switching from smaller to bigger machines.
- ✓ Vertical scaling, often known as “scaling up/down,” is the process of increasing/decreasing the power of an existing system, such as the CPU or RAM, to meet the changing demands.
- ✓ As there is no need to alter the logic, vertical scaling is simpler. Instead, you are only executing the same code on machines with more capacity.
- ✓ The memory, storage, or network speed can be vertically scaled.
- ✓ Vertical scaling can also refer to completely replacing a server or shifting the workload from an outdated server to an updated one. In simpler terms, Vertical scaling is about upskilling existing employees for an additional client/problem set.

Pros and cons of Vertical Scaling

Pros

- ✓ Sometimes it could be cost effective
- ✓ Less complexity involved
- ✓ Easier to maintain

Cons

- ✓ More Downtime possibilities
- ✓ Very less flexibility
- ✓ Single point of failure

What is Horizontal Scaling?

- ✓ Horizontal scaling involves adding more machines or nodes to a system instead of adding extra resoures to an existing machine
- ✓ Horizontal scaling is typically used to handle increasing amounts of internet traffic or workloads
- ✓ Horizontal scaling, often known as “scaling out/in,” is the process of increasing/decreasing the number of nodes and machines in the resource pool.
- ✓ For a sequential piece of logic to be processed in parallel across numerous devices, horizontal scaling calls for breaking it into smaller chunks and delegating the logic to the new machine.
- ✓ In simpler terms, scaling horizontally is about hiring new employees for an additional client/problem set.

Pros and cons of Horizontal Scaling

Pros

- ✓ Easier Scaling by Hardware additions
- ✓ Enhanced flexibility
- ✓ Lesser downtime
- ✓ Offers Redundancy

Cons

- ✓ Higher initial costs
- ✓ Harder to maintain
- ✓ Requires addition software to manage and monitor the whole system

Horizontal Vs Vertical Scaling

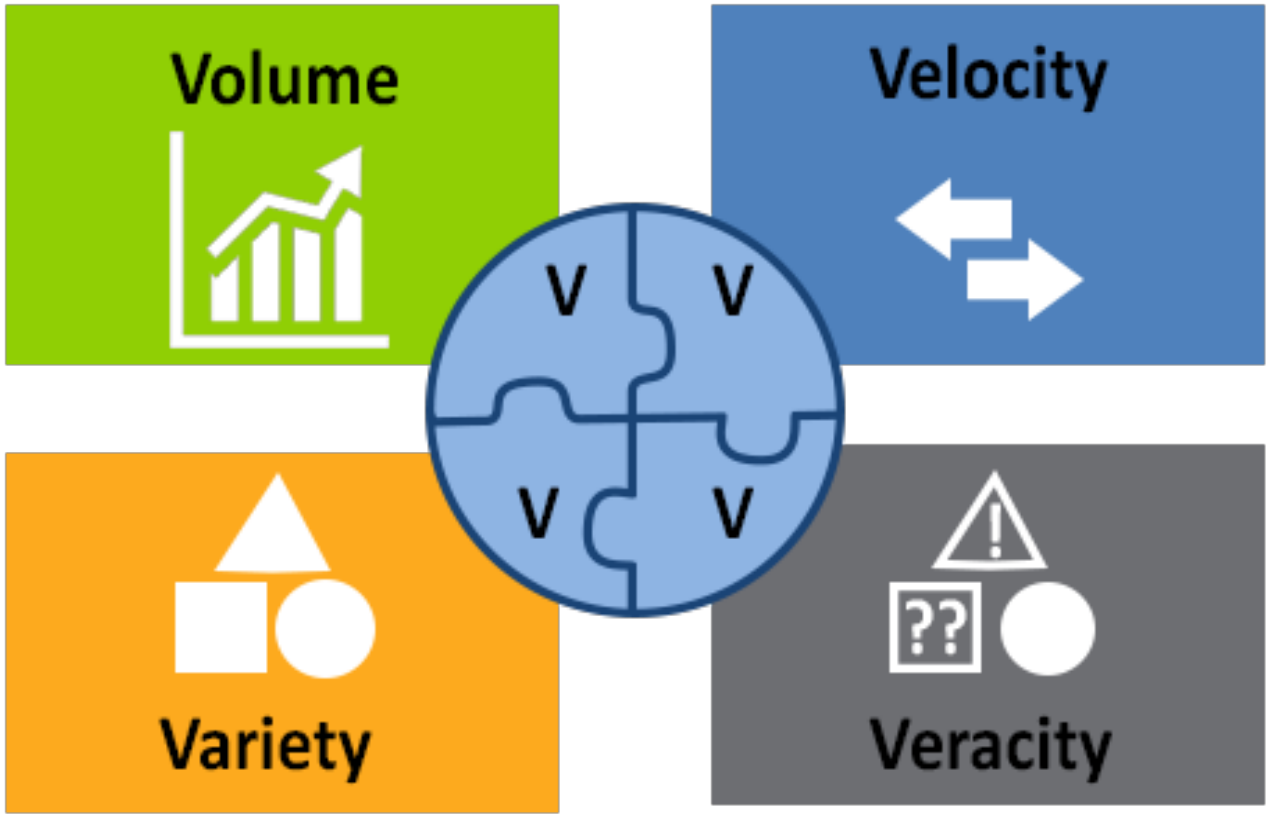
	Horizontal Scaling	Vertical Scaling
Data Management	Scaling horizontally typically relies on data partitioning as each node only contains part of the data.	In vertical scaling, the data resides on a single node, and scaling is accomplished by multi-core, primarily by distributing the load among the machine's CPU and RAM resources.
Examples	Cassandra, MongoDB, Hadoop, Spark	Relational Databases
Downtime possibility	You can scale with less downtime by adding additional computers to the existing pool because you are no longer constrained by the capacity of a single device.	There is an upper physical limit to vertical scaling, which is the scale of the current hardware. Vertical scaling is restricted to the capacity of one machine because expanding over that limit can result in downtime.

Horizontal Vs Vertical Scaling

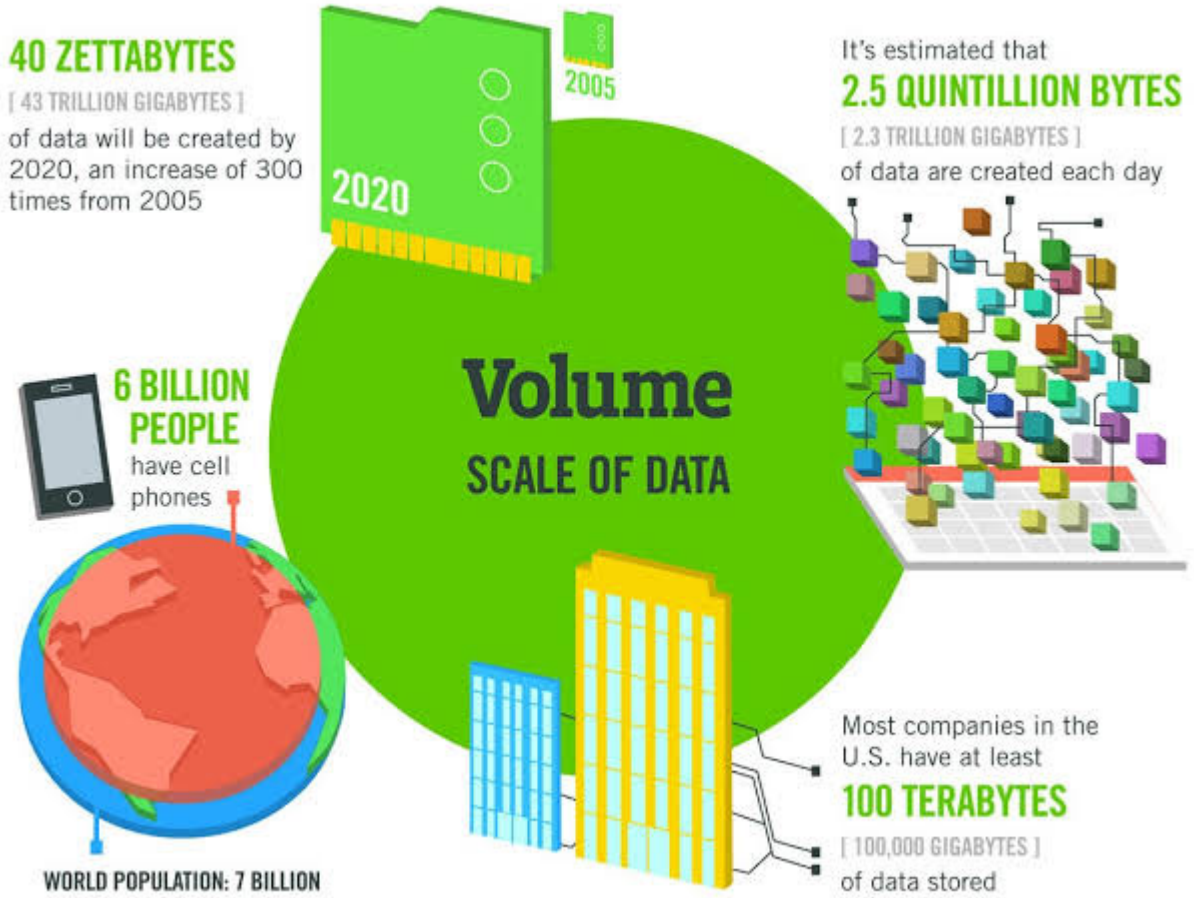
Concurrency Models	As it entails distributing jobs among devices over a network is known as distributed programming. Several patterns are connected to this model: MapReduce, Master/Worker*, Blackboard, and many spaces.	It involves the Actor model: Multi-threading and in-process message forwarding are frequently used to implement concurrent programming on multi-core platforms.
Message Passing	Data sharing is more difficult in distributed computing because there isn't a shared address space. Since you will send copies of the data, it also increases the cost of sharing, transferring, or updating data.	Data sharing and message passing Can be accomplished by passing a reference in a multi-threaded scenario since it is reasonable to presume that there is a shared address space.

What is Big Data?

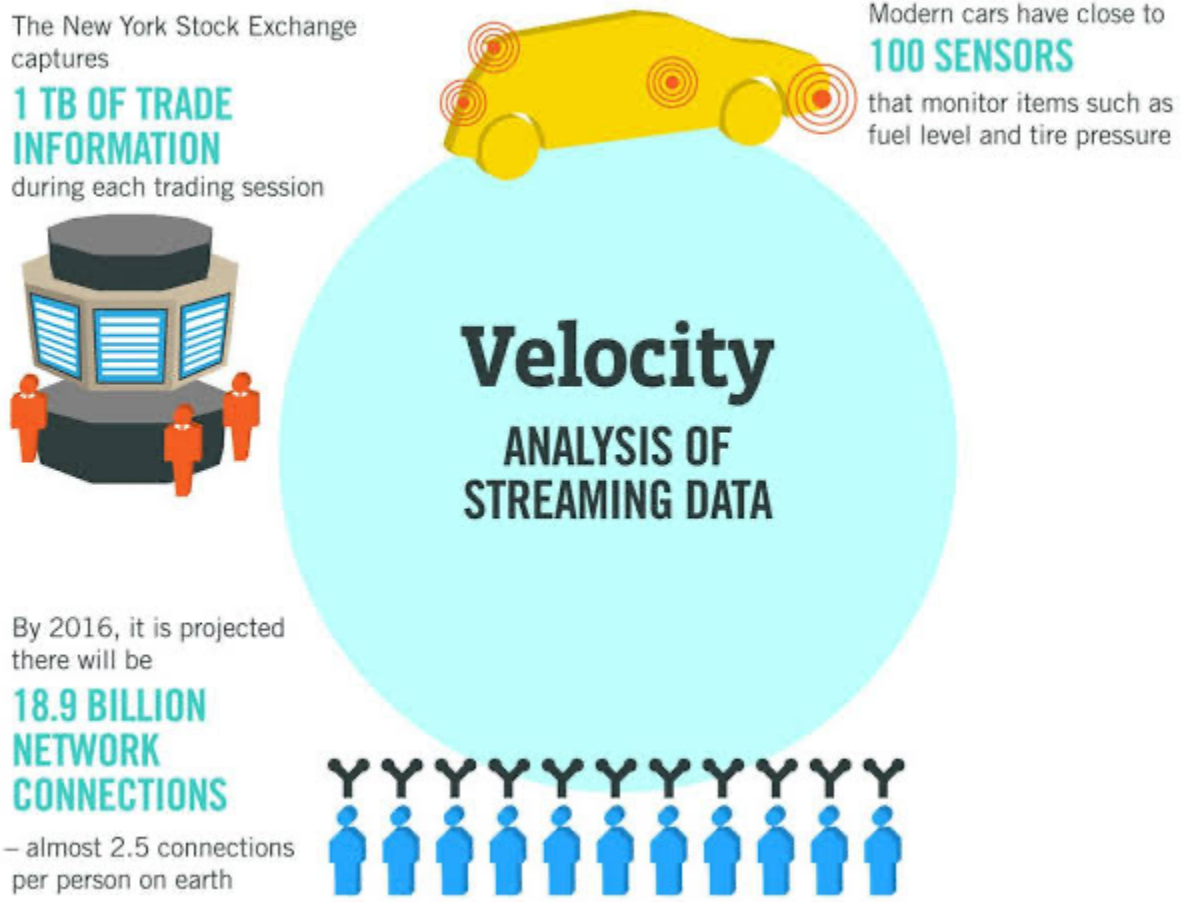
The Four V's of Big Data



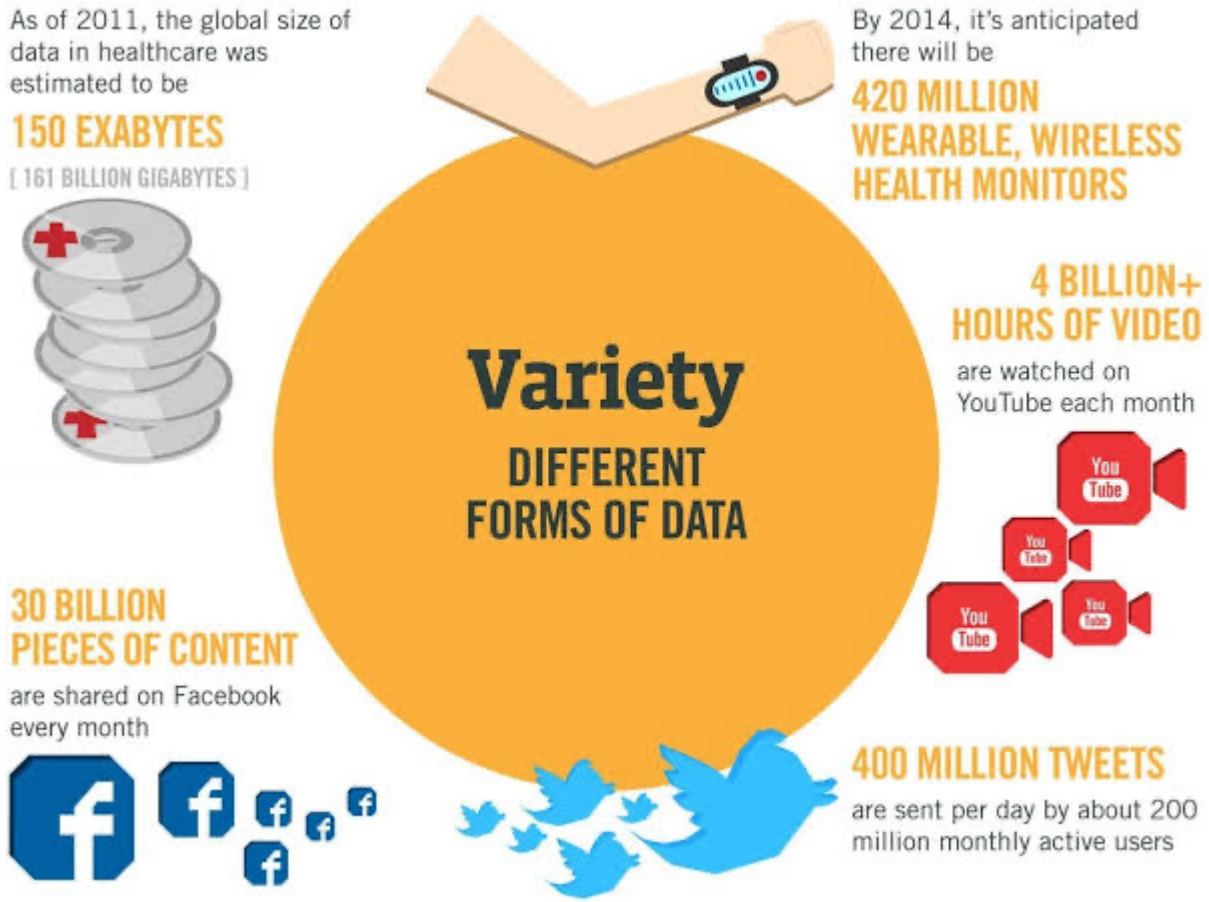
Volume.


40 ZETTABYTES
[43 TRILLION GIGABYTES]
of data will be created by 2020, an increase of 300 times from 2005
2020
2005
It's estimated that
2.5 QUINTILLION BYTES
[2.3 TRILLION GIGABYTES]
of data are created each day
Volume
SCALE OF DATA
6 BILLION PEOPLE
have cell phones
WORLD POPULATION: 7 BILLION
Most companies in the U.S. have at least
100 TERABYTES
[100,000 GIGABYTES]
of data stored

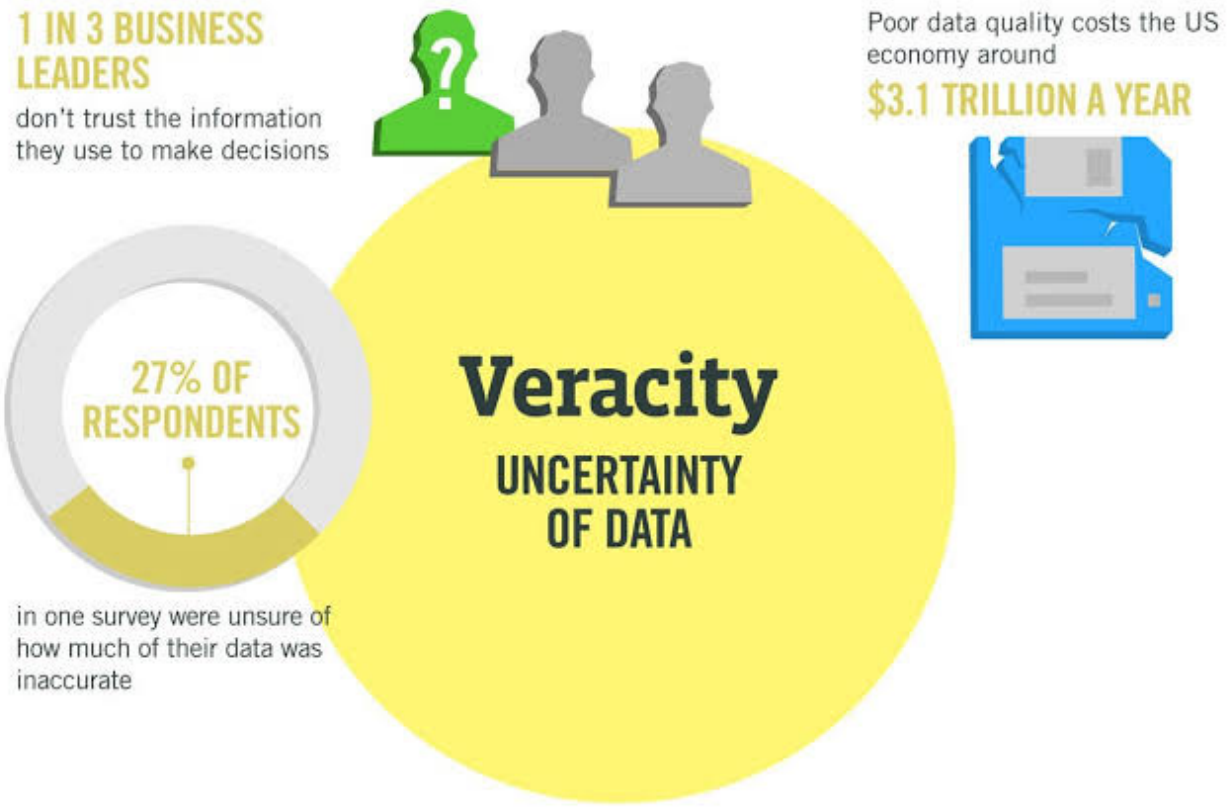
Velocity.


The New York Stock Exchange captures
1 TB OF TRADE INFORMATION
during each trading session
Modern cars have close to
100 SENSORS
that monitor items such as fuel level and tire pressure
Velocity
ANALYSIS OF STREAMING DATA
By 2016, it is projected there will be
18.9 BILLION NETWORK CONNECTIONS
– almost 2.5 connections per person on earth

Variety.


As of 2011, the global size of data in healthcare was estimated to be
150 EXABYTES
[161 BILLION GIGABYTES]
By 2014, it's anticipated there will be
420 MILLION WEARABLE, WIRELESS HEALTH MONITORS
Variety
DIFFERENT FORMS OF DATA
4 BILLION+ HOURS OF VIDEO
are watched on YouTube each month
30 BILLION PIECES OF CONTENT
are shared on Facebook every month
400 MILLION TWEETS
are sent per day by about 200 million monthly active users

Veracity.


1 IN 3 BUSINESS LEADERS
don't trust the information they use to make decisions
Poor data quality costs the US economy around
$3.1 TRILLION A YEAR
27% OF RESPONDENTS
in one survey were unsure of how much of their data was inaccurate
Veracity
UNCERTAINTY OF DATA

What is Big Data?

Big data is a term applied to data sets whose size is beyond the ability of commonly used software tools to capture, manage, and process the data within a tolerable elapsed time.

Gartner Predicts 800% data growth over next 5 years.

80-90% of data produced today is unstructured.

Challenges of Big Data.

- How we can capture and deliver data to right people in real-time?
- How we can understand and use big data when it is in Variety of forms?
- How we can store/analyse the data given its size and computational capacity?
- While the storage capacities of hard drives have increased massively over the years, access speeds—the rate at which data can be read from drives, have not kept up.
- Example: Need to process 100TB datasets

  On 1 node:

  scanning @ 50MB/s = 23 days

  On 1000 node cluster:

  scanning @ 50MB/s = 33 min
- Hardware Problems / Process and combine data from Multiple disks
- Traditional Systems: They can’t scale, not reliable and expensive.

Technologies to support Big Data.

Big Data

Scale-out everything:

- Storage
- Compute
- Analytics

What is Hadoop?

- Apache Hadoop is a software framework that supports data-intensive distributed applications under a free license.
- It enables applications to work with thousands of nodes and petabytes of data.
- Hadoop was inspired by Google File System (GFS) and Google's MapReduce papers

Why Hadoop?

Characteristics

- **Accessible** - Hadoop runs on large clusters of commodity machines or on cloud (EC2).
- **Robust** - Hadoop is architected with the assumption of frequent hardware malfunctions. It can gracefully handle most such failures
- **Scalable** - Hadoop scales linearly to handle larger data by adding more nodes to the cluster.
- **Simple** - Hadoop allows users to quickly write efficient parallel code.
- **Data Locality** - Move Computation to the Data.
- **Replication** - Use replication across servers to deal with unreliable storage/servers

Adoption Drivers

- **Business Drivers -** Bigger the data, Higher the value
- **Financial Drivers -** Cost advantage of Open Source + Commodity H/W Low cost per TB
- **Technical Drivers -** Existing systems failing under growing requirements-3 Vs

Data Processing – Old way Vs. Hadoop way

THE OLD WAY

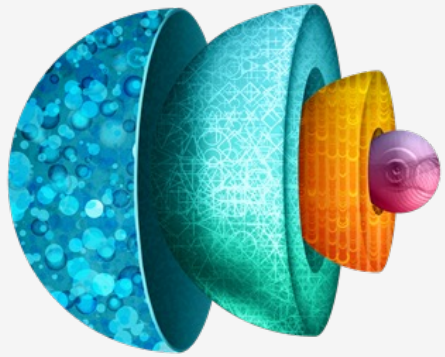
Multiple platforms
for multiple workloads

COMPLEX, FRAGMENTED, COSTLY

- Data siloed by department or line of business
- Mostly stored in expensive specialized systems
- Analysts only retrieve data into EDW when needed
- Nobody in the organization has a complete view

THE HADOOP WAY

Single data platform for
storage, BI, analytics, and
app serving



SIMPLE, UNIFIED, EFFICIENT

- Data stored on scalable, low-cost platform
- Hadoop handles a variety of workloads
- Perform end-to-end workflows on single system
- Provides broad data access across departments

Big Data Vs. DWH

Big data systems are **complement** to DWH systems, not a **replacement**.

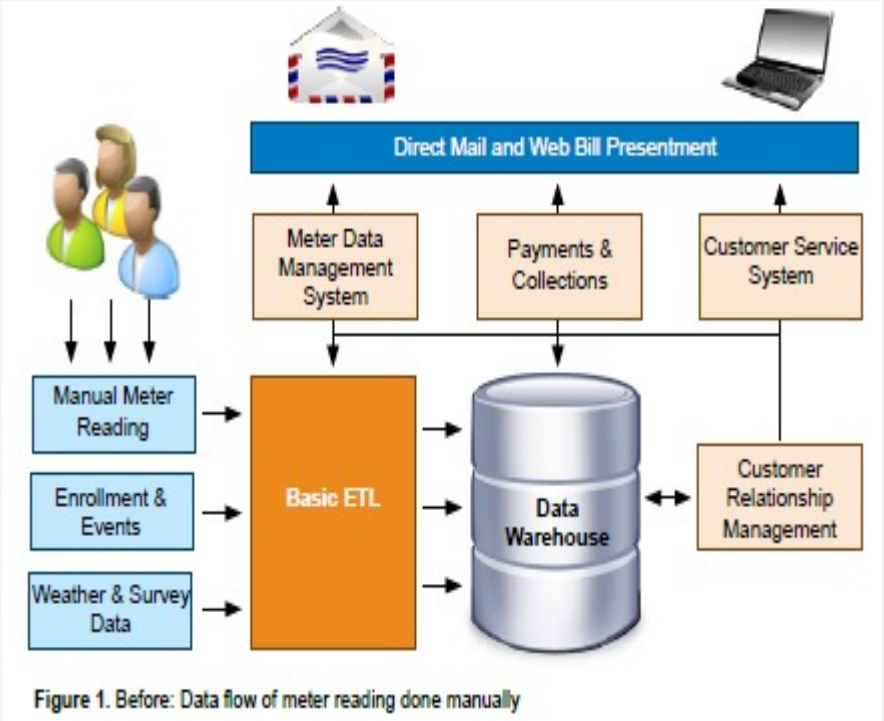


Figure 1. Before: Data flow of meter reading done manually

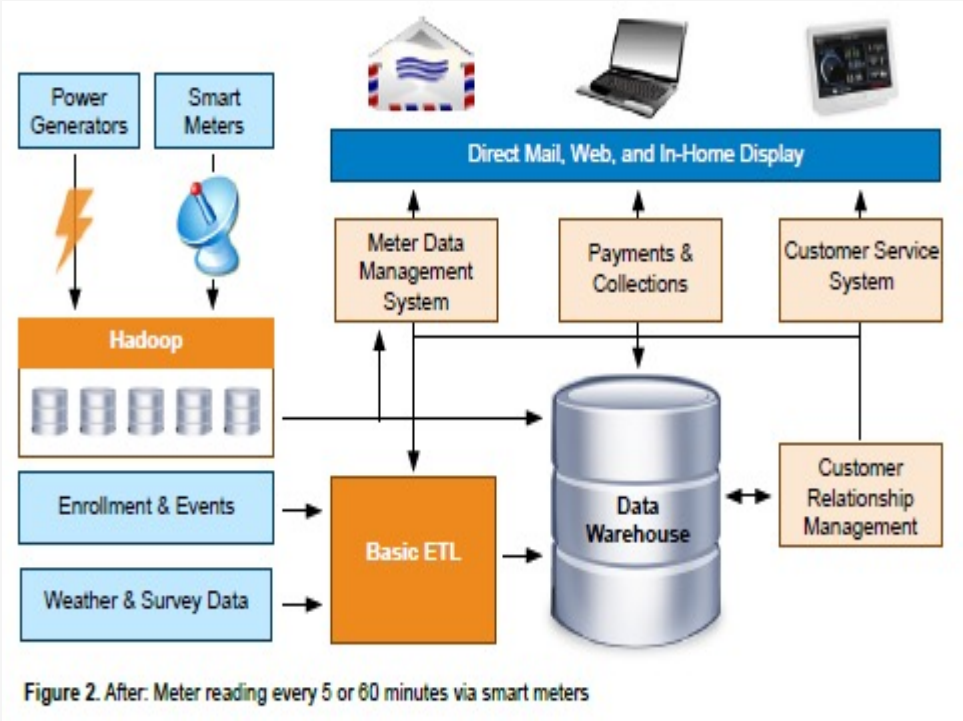
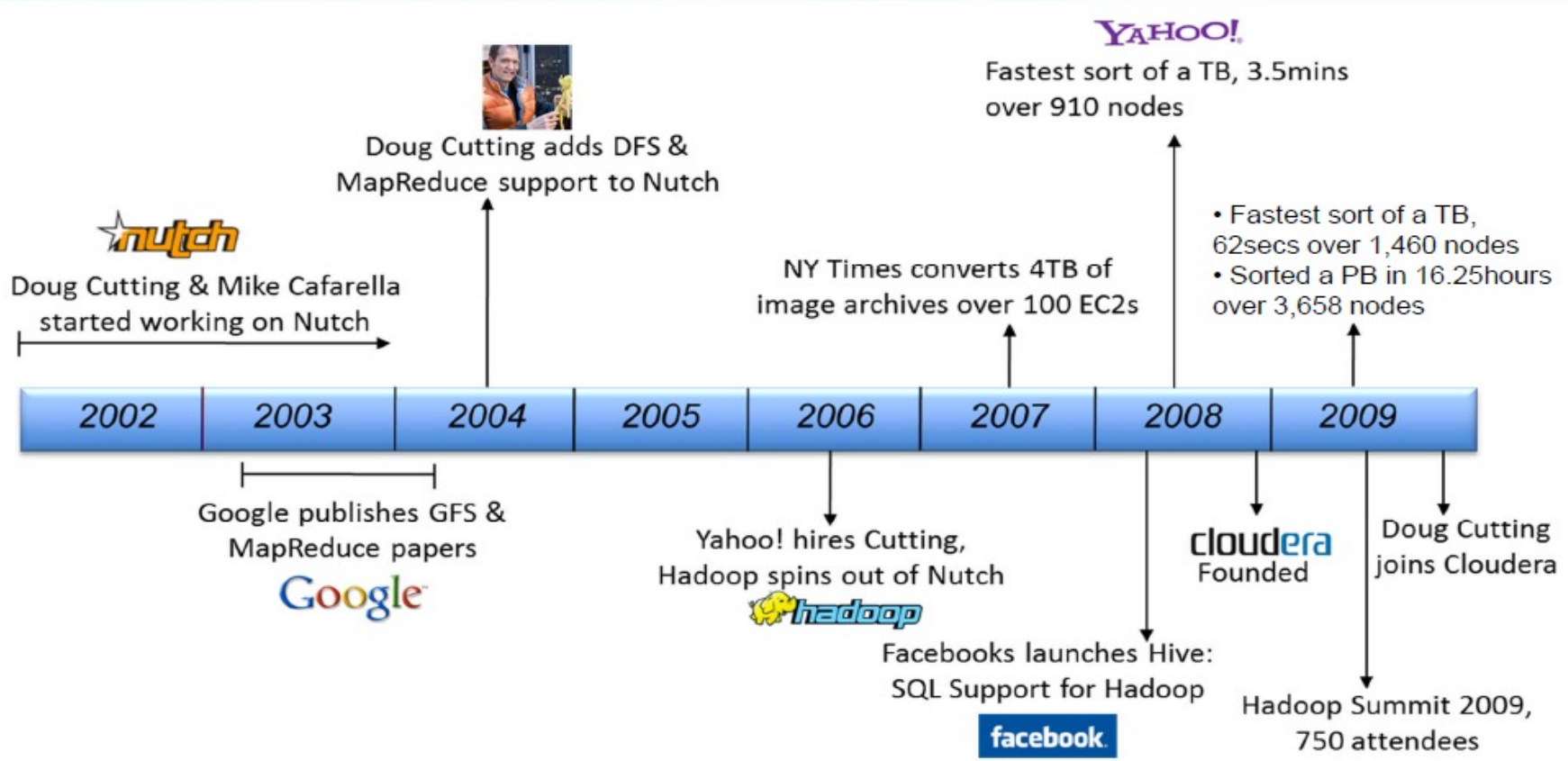


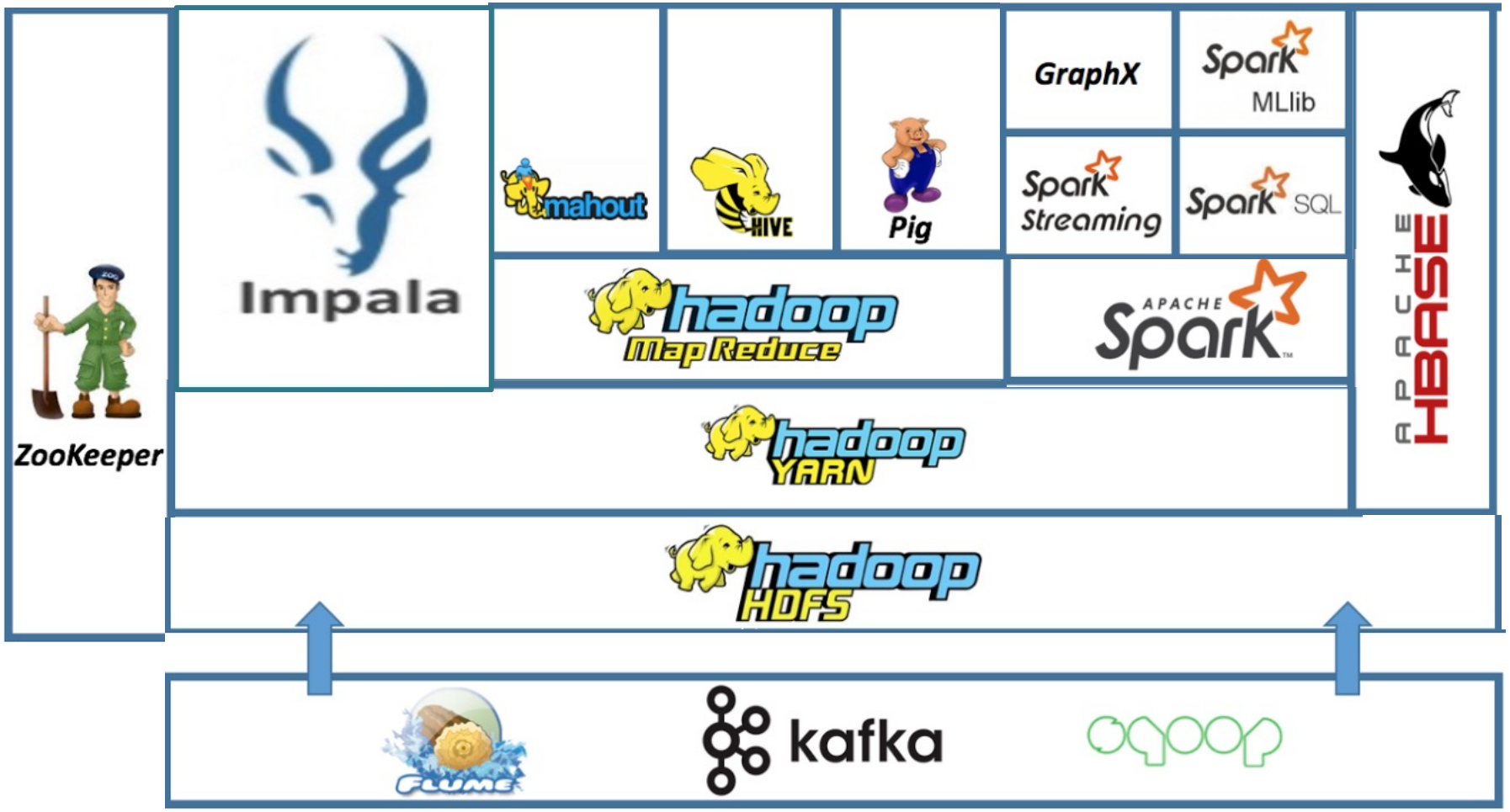
Figure 2. After: Meter reading every 5 or 60 minutes via smart meters

History Of Hadoop



The name Hadoop is not an acronym; it's a made-up name. The project's creator, Doug Cutting, explains that it was named after his kid's stuffed yellow elephant.

Hadoop Eco system.



Hadoop Eco system.

HDFS: A distributed file system that provides high throughput access to application data.
MapReduce: A software framework for distributed processing of large data sets on compute clusters.
Other Hadoop/Spark-related projects at Apache include:
Sqoop: Import/export the data from RDBMS databases to/from HDFS
Hive: A data warehouse infrastructure that provides data summarization and ad hoc querying.
Impala: SQL Realtime queries - Should work both for analytic and transactional workloads. Response Time: microseconds to hours
Pig: A high-level data-flow language and execution framework for parallel computation.
HBase: A scalable, distributed database that supports structured data storage for large tables.
ZooKeeper: A high-performance coordination service for distributed applications
Flume/Kafka: Reliable Real-time distributed streaming solutions
Spark: In-memory lightning fast processing engine

Who uses Hadoop?

Yahoo - 100,000 CPUs in >36,000 computers
Facebook - 1100-machine cluster with 8800 cores and about 12 PB storage and A 300-machine cluster with 2400 cores and about 3 PB raw storage.
Linkedin
Twitter
Ebay
IBM
IIIT, Hyderabad - 10 to 30 nodes ,Quad 6600s, 4GB RAM and 1TB disk PSG Tech, Coimbatore - 5 to 10 nodes. Cluster nodes vary from 2950 Quad Core Rack Server, with 2x6MB Cache and 4 x 500 GB SATA Hard Drive to E7200 / E7400 processors with 4 GB RAM and 160 GB HDD.
Rackspace
Google – University initiative
Adobe
NewYork Times

Common use cases Hadoop?

Financial Services

- Detect/prevent fraud
- Model and manage risk
- Improve debt recovery rates
- Personalize banking/insurance products

Healthcare

- Remote patient monitoring
- Predictive modeling for new drugs
- Personalized medicine
- Optimal treatment pathways

Retail

- In-store behavior analysis
- Cross selling, recommendation engines
- Optimize pricing, placement, design
- Optimize inventory and distribution

Web / Social / Mobile

- Sentiment analysis
- Web log, image, and video analysis
- Location-based marketing
- Price comparison services

Manufacturing

- Design to value
- Improve service via product sensor data
- Crowd-sourcing
- “Digital factory” for lean manufacturing

Government

- Detect/prevent fraud
- Segment populations, customize action
- Support open data initiatives
- Cyber-security

Questions?