**Big Data**

According to SiliconAngle, there was 2.5 zetabytes of stored data world over in the year 2012 and it is set to hit the 8 zetabytes mark by the end of 2015.

To put things in perspective, this data has largely been produced by websites and cross platform transactions. Add to it the fact that there would be a total of 20 billion “smart” devices connected to the internet by the end of 2020 and the numbers can be baffling!

Given that Facebook alone produces about 15 TB data on a daily basis today, imagine a situation where our refrigerators, microwaves, cars, fitness devices and a host of other connected devices start producing and storing data every second!

**The 4 V’s**

**Volume-based value**: The more comprehensive your integrated view of the customer and the more historical data you have on them, the more insight you can extract from it. In turn, you are making better decisions when it comes to acquiring, retaining, growing and managing those customer relationships.

**Velocity-based value**: The more rapidly you can process information into your data and analytics platform, the more flexibility you get to find answers to your questions via queries, reports, dashboards, etc. A rapid data ingestion and rapid analysis capability provides you with the timely and correct decision achieve your customer relationship management objectives.

**Variety-based value**: The more varied customer data you have – from the Customer relationship management (CRM) system, social media, call-center logs, etc. – the more multifaceted view you develop about your customers, thus enabling you to develop customer journey maps and personalization to engage more with customers.

**Veracity-based value**: Amassing a lot of data does not mean the data becomes clean and accurate. Data on customers must remain consolidated, cleansed, consistent, and current to make the right decisions.

**File System on a Single Disk**  
File system is a structured data representation and a set of metadata that describe the stored data.

—  Hard drives are divided into sectors of about 512 bytes each. Sectors in turn are grouped into clusters. Clusters have a defined size of 512 bytes to 64 KBs, so they usually contain multiple sectors.

—  A cluster is called as a filesystem block which represents a continuous block of space on the disk and it is the minimum amount of data that can be read or written.

—  Depending on the file system block (typically 4KBs) a single file can be stored in one or across hundreds or thousands of clusters.

—  This is generally transparent to the filesystem user who is simply reading or writing a file of whatever length.

**Why is a block in HDFS so large compared to disk block size?**

Large block size in HDFS minimizes the cost of seeks (seek time - the time taken for a disk drive to locate the area on the disk where the data to be read is stored).

If the block is large enough, the time it takes to transfer the data from the disk can be significantly longer than the time to seek to the start of the block.

Thus, transferring a large file made of multiple blocks operates at the disk transfer rate.

E.g. if the seek time is around 10 ms and the transfer rate is 100 MB/s, then to make the seek time 1% of the transfer time, we need to make the block size around 100 MB.

The default is actually 128 MB, although many HDFS installations use larger block sizes. This number will continue to be revised upward as transfer speeds grow with new generations of disk drives.

This argument shouldn’t be taken too far, however. MapReduce tasks operate on one block at a time. So if data is not distributed across many data nodes, we are not taking advantage of the parallelism possible to its fullest extent. So your jobs will run slower than they could otherwise!

**Advantages of Blocks**

A file can be larger than any single disk in the network.

In fact, it would be possible, if unusual, to store a single file on an HDFS cluster whose blocks filled all the disks in the cluster.

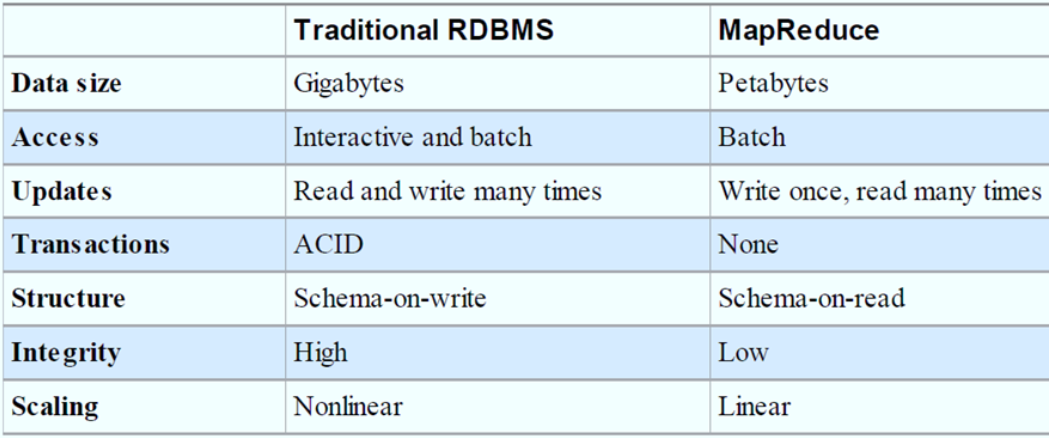
Making the unit of abstraction a block rather than a file simplifies the storage subsystem

It is easy to calculate how many blocks can be stored on a given disk and eliminates metadata concerns (file metadata such as permissions information does not need to be stored with the blocks so another system can handle metadata separately).

Blocks fit well with replication for providing fault tolerance and availability

**Comparison to RDBMS**

Hadoop isn’t the first distributed system for data storage and analysis, but it has some unique properties that sets it apart from other systems.



—  MapReduce is a good fit for problems that need to analyze the whole dataset in a batch fashion, particularly for ad hoc analysis. An RDBMS is good for point queries or updates, where the dataset has been indexed to deliver low-latency retrieval and update times of a relatively small amount of data.

—  MapReduce is suitable for applications where the data is written once and read many times, whereas RDBMS is good for datasets that are continually updated.

—  Hadoop doesn’t fully replace relational products; many architectures would benefit from both Hadoop and a Relational product(s).

—  The differences between relational databases and Hadoop systems are blurring.

Relational databases have started incorporating some of the ideas from Hadoop, and Hadoop systems such as Hive are becoming more interactive (by moving away from MapReduce) and adding features like indexes and transactions that make them look more and more like traditional RDBMSs.

**Rack in Hadoop**

A **rack** is a collection of 30 or 40 nodes that are physically stored close together and are all connected to the same network switch. Network bandwidth between any two nodes in **rack** is greater than bandwidth between two nodes on different racks. A **Hadoop** Cluster is a collection of racks.

**NameNode**

The NameNode in Hadoop is the node where Hadoop stores all the location information of the files in HDFS. In other words, it holds the metadata for HDFS. Whenever a file is placed in the cluster a corresponding entry of its location is maintained by the NameNode.  
 **Hardware Selection**

Here are the recommended specifications for DataNodes in a balanced Hadoop cluster:

12-24 1-4TB hard disks in a JBOD (Just a Bunch Of Disks) configuration

2 quad-/hex-/octo-core CPUs, running at least 2-2.5GHz

64-512GB of RAM

Bonded Gigabit Ethernet or 10Gigabit Ethernet (the more storage density, the higher the network throughput needed)

Here are the recommended specifications for NameNode:

4–6 1TB hard disks in a JBOD configuration (1 for the OS, 2 for the FS image [RAID 1], 1 for Apache ZooKeeper, and 1 for Journal node)

2 quad-/hex-/octo-core CPUs, running at least 2-2.5GHz

64-128GB of RAM

Bonded Gigabit Ethernet or 10Gigabit Ethernet