Dang Khoa Le

613760

ASSIGNMENT 1

1. [2] **Can you think of a use case of Big Data?  Explain it briefly.**

One of the use cases of Big Data I’ve ever heard of is about a platform to track parcels over 600+ carriers covered worldwide. By tracking parcels’ delivery information over time, the platform aims to provide streamlined logistics data at one place, helps businesses gain transparency on carrier performance, drives efficiency in customer service, and enable a better post-purchase experience.

This use case should be a Big Data scenario, because:

* Big volume of dataset. With a capacity of tracking 40 million parcels monthly, the amount of data could be stored up to Petabytes.
* Velocity is high with average 1 million of active users visiting the website of platform every hour for enquiring logistics data.
* It captures unstructured data from variety of carriers all around the world.
* The data being tracked from carriers could be wrong at some points in time. It might be duplicated, get changed in the way of presenting information, etc.

1. [2] **What are the advantages of using Hadoop and HDFS?**

The advantages of Hadoop:

* Provides a reliable shared storage and analysis system
* Has the ability to read and write data in parallel to or from multiple disks
* Enables applications to work with thousands of nodes and petabytes of data
* A free license

The advantages of HDFS:

* Be able to store large file (Terabytes, Petabytes, etc. …)
* Provides streaming data (WORM pattern, and optimized for streaming reads over random reads)
* “Cheap” commodity hardware (No needs for super computers)
* Provides Java API for applications to use
* HTTP browser can be used to browse the files of HDFS instance

1. [2] **Explain the term block abstraction in Hadoop.**

* HDFS is broken into block-sized chunks which are stored as independent units. This unit is called block abstraction or simply block.
* Block themselves are stored on top of standard single-machine filesystem.
* A file in HDFS that is smaller than a single block does not occupy a full block’s worth of underlying storage. Ex: A 1MB file stored with a block size of 128MB uses 1MB of disk space, not 128MB.
* Making the unit of abstraction a block rather than a file simplifies the storage subsystem
  + It’s easy to calculate how many blocks can be stored in 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).
* From Hadoop 2.x, by default, a block size is 128MB
* Each file in HDFS is sequence of blocks
* All blocks in the file except the last one are of the same size

1. [2] **What is the meaning of fault tolerance in HDFS and how is it achieved?**

Fault tolerance in HDFS means that it’s designed to carry on working without a noticeable interruption to the user in the face of failures such as nodes become unavailable, hard disk of node may fall.

The way to achieve it is to use data replication:

* Blocks are replicated in multiple nodes. By this way, when a given node has failed, our system still works normally, because this data on this node has been replicated to another node.
* NameNode tries to place replicas of blocks on multiple racks for improved fault tolerance.
* Number of replicas is configurable
* By default, one block is replicated to 3 nodes. In case of multiple racks, there are no more 2 nodes in the same rack. This makes sure our system is always on.

1. [2] **Consider a 560 TB of text file which needs to be stored in HDFS. The block size has been set to be 128 MB with a replication factor of 3. The cluster has 100 DataNodes each with a capacity of 15 TB.  
   Will it be possible to store this text file in this HDFS cluster? Why or why not**?

No, it won’t be possible to store this text file. Because the expected storage capacity is greater than the actual capacity. To be more specific, we need 1,680TB (560TB x 3) of storage to store the text file along with its replicas (replication factor is 3), but our system only has 1,500TB (100 nodes x 15TB) of storage.