UE18 CS 322- Big Data- Unit 1

Question Bank and answers

Topic 1- Introduction to Big Data:

1. Describe the data, Web data and Big Data

Definitions of Data Data has several definitions.

Usages can be singular or plural.

- "Data is information, usually in the form of facts or statistics that one can analyze or use for further calculations." [Collins English Dictionary]
- "Data is information that can be stored and used by a computer program.". [Computing] "Data is information presented in numbers, letters, or other form". [Electrical Engineering, Circuits, Computing and Control]
- "Data is information from series of observations, measurements or facts". [Science]
- "Data is information from series of behavioural observations, measurements or facts". [Social Sciences]

Definition of Web Data Web is large scale integration and presence of data on web servers. Web is a part of the Internet that stores web data in the form of documents and other web resources. URLs enable the access to web data resources. Web data is the data present on web servers (or enterprise servers) in the form of text, images, videos, audios and multimedia files for web users. A user (client software) interacts with this data. A client can access (pull) data of responses from a server. The data can also publish (push) or post (after registering subscription) from a server. Internet applications including web sites, web services, web portals, online business applications, emails, chats, tweets and social networks provide and consume the web data. Some examples of web data are Wikipedia, Google Maps, McGraw-Hill Connect, Oxford Bookstore and YouTube.

1. Wikipedia is a web-based, free-content encyclopaedia project supported by the Wikimedia Foundation.
2. Google Maps is a provider of real-time navigation, traffic, public transport and nearby places by Google Inc.
3. McGraw-Hill Connect is a targeted digital teaching and learning environment that saves students' and instructors' time by improving student performance for a variety of critical outcomes.
4. Oxford Bookstore is an online book store where people can find any book that they wish to buy from millions of titles. They can order their books online at www.oxfordbookstore.com
5. YouTube allows billions of people to discover, watch and share originally-created videos by Google Inc.

2. What is the general classification of Data? Explain with examples

Structured	UnStructured	Semi structured	Multistructured
Structured data conform and associate with data schemas and data models.	Unstructured data do not conform and associate with any data models.	Semi-structured form of data does not conform and associate with formal data model structures.	Multistructured form of data does not conform and associate with formal data model structures.
Structured data is in the form of tables.	Unstructured data is in the form of csv,txt. May be of the form key value pair.	Semistructured data are XML and JSON documents	Multi-structured data sets can have many formats.
The chess moves of White and Black in two subsequent vertical columns. Volume of data, i.e. data used for analyzing erroneous or best moves in the matches, keeps growing with more and more tables, and may eventually become 'voluminous data'.	Social media generates data after each international match. The media publishes the analysis of classical matches played between Grand Masters. The data for analyzing chess moves of these matches are thus in a variety of formats.		The Voluminous data of these matches can be in a structured format (i.e. tables) as well as in unstructured formats (i.e. text documents, news columns, blogs, Facebook etc.). Tools of multi-structured data analytics assist the players in designing better strategies for winning chess championships.

3. What do you mean by 3Vs characteristics of Big Data? What are the challenges faced from large growth in volume of data?

Volume

The phrase 'Big Data' contains the term big, which is related to size of the data and hence the characteristic. Size defines the amount or quantity of data, which is generated from an application(s). The size determines the processing considerations needed for handling that data.

Velocity

The term velocity refers to the speed of generation of data. Velocity is a measure of how fast the data generates and processes. To meet the demands and the challenges of processing Big Data, the velocity of generation of data plays a crucial role.

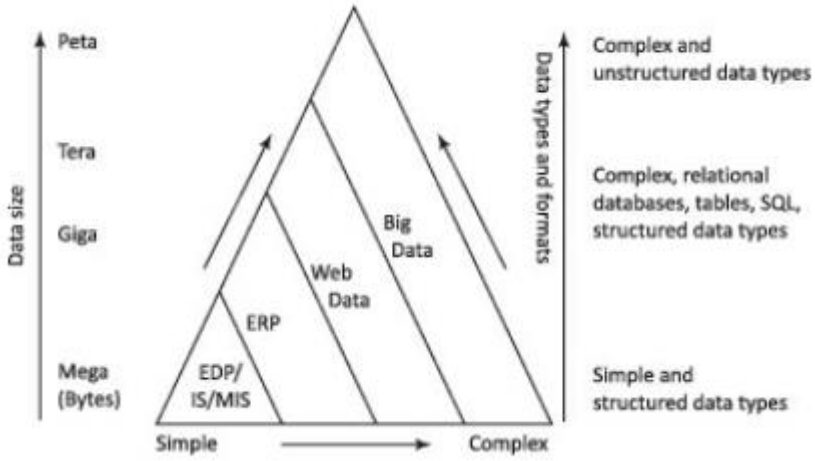
Variety

Big Data comprises of a variety of data. Data is generated from multiple sources in a system. This introduces variety in data and therefore introduces 'complexity'. Data consists of various forms and formats. The variety is due to the availability of a large number of heterogeneous platforms in the industry. This means that the type to which Big Data belongs to is also an important characteristic that needs to be known for proper processing of data. This characteristic helps in effective use of data according to their formats, thus maintaining the importance of Big Data.

Veracity is also considered an important characteristic to take into account the quality of data captured, which can vary greatly, affecting its accurate analysis.

The 4Vs (i.e. volume, velocity, variety and veracity) data need tools for mining, discovering patterns, business intelligence, artificial intelligence (AI), machine learning (ML), text analytics, descriptive and predictive analytics, and the data visualization tools.

4. Draw a diagram showing evolution of Big Data and their characteristics over the time as size, complexity increased and as unstructured data increased.



Note: The concepts explained in this figure is well known to all of you. Try to explain each of these in your own words.

5. Give three examples of the machine-generated data.

Examples of machine-generated data are:

Data from computer systems: Logs, web logs, security/surveillance systems, systems, videos/images etc.
Data from fixed sensors: Home automation, weather sensors, pollution sensors, traffic sensors etc.
Mobile sensors (tracking) and location data.

6. Think of a manufacturing and retail marketing company, such as LEGO toys. How does such a toy company optimize the services offered, products and schedules, devise ways and use Big Data processing and storing for predictions using analytics?

Assume that a retail and marketing company of toys uses several Big Data sources, such as

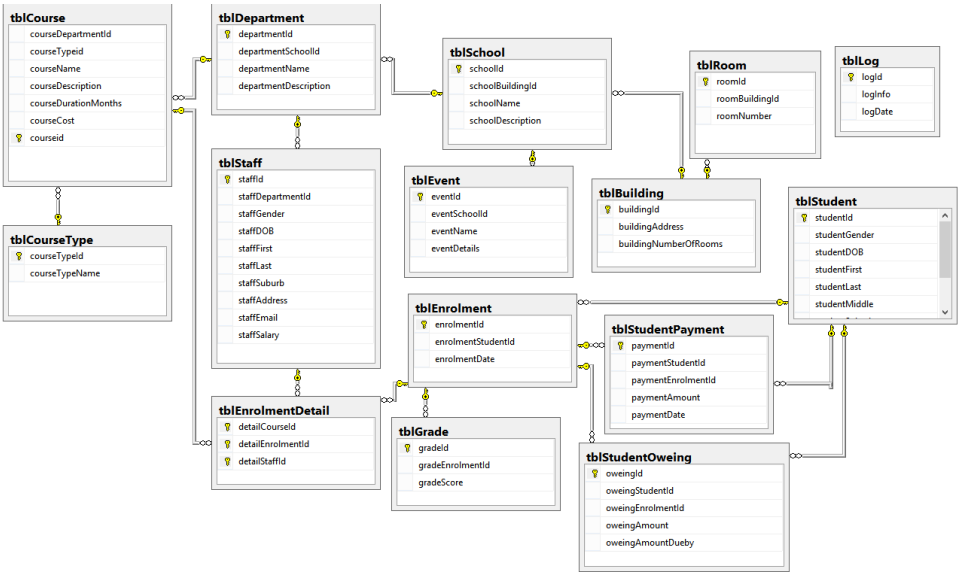
(i) machine-generated data from sensors (RFID readers) at the toy packaging,
(ii) transactions data of the sales stored as web data for automated reordering by the retail stores and
(iii) tweets, Facebook posts, e-mails, messages, and web data for messages and reports.

The company uses Big Data for understanding the toys and themes in present days that are popularly demanded by children, predicting the future types and demands. The company using such predictive analytics, optimizes the product mix and manufacturing processes of toys. The company optimizes the services to retailers by maintaining toy supply schedules. The company sends messages to retailers and children using social media on the arrival of new and popular toys.

7. List the unstructured data forms used in 'Automotive Components and Predictive Automotive Maintenance Services'.

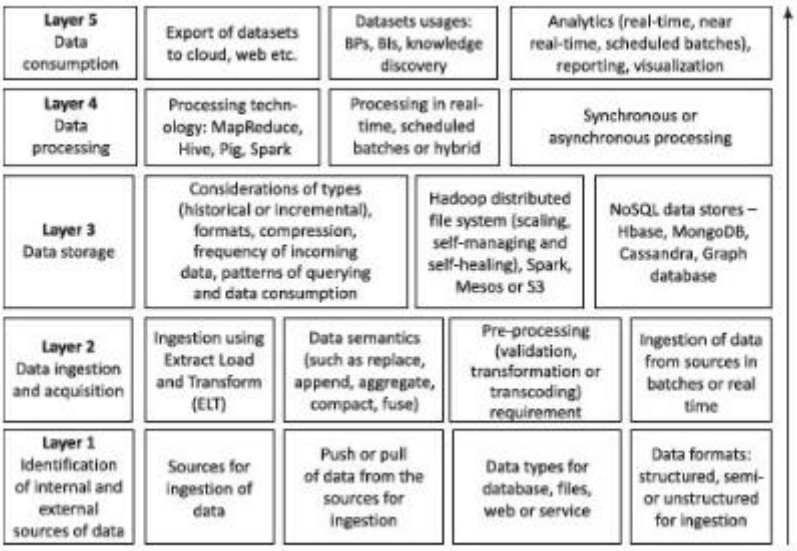
ACPAMS uses Big Data types as: machine-generated data from sensors at automotive components, such as brakes, steering and engine from each car; transactions data stored at the service website; social networks and web data in the form of messages, feedback and reports from customers. The service provides messages for scheduled and predictive maintenances. The service generates reports on social networks and updates the web data for the manufacturing plant. The service generates reports about components qualities and needed areas for improvement in products of the company.

8. Give data example of student records at a University and explain structured data, hierarchical relationships between them.



9. Explain the function of each of the five layers in Big Data architecture design.

"Big Data architecture is the logical and/or physical layout/structure of how Big Data will be stored, accessed and managed within a Big Data or IT environment. Architecture logically defines how Big Data solution will work, the core components (hardware, database, software, storage) used, flow of information, security and more." Characteristics of Big Data make designing Big Data architecture a complex process. Further, faster additions of new technological innovations increase the complexity in design. The requirements for offering competing products at lower costs in the market make the designing task more challenging for a Big Data architect. Data analytics need the number of sequential steps. Big Data architecture design task simplifies when using the logical layers approach. Figure 1.2 shows the logical layers and the functions which are considered in Big Data architecture. Five vertically aligned textboxes on the left of Figure 1.2 show the layers. Horizontal textboxes show the functions in each layer. Data processing architecture consists of five layers: (i) identification of data sources, (ii) acquisition, ingestion, extraction, pre-processing, transformation of data, (iii) data storage at files, servers, cluster or cloud, (iv) data-processing, and (v) data consumption in the number of programs and tools.



Data consumed for applications, such as business intelligence, data mining, discovering patterns/clusters, artificial intelligence (AI), machine learning (ML), text analytics, descriptive and predictive analytics, and data visualization. Data ingestion, pre-processing, storage and analytics require special tools and technologies.

- Logical layer 1 (L1) is for identifying data sources, which are external, internal or both.
- The layer 2 (L2) is for data-ingestion.
- Data ingestion means a process of absorbing information, just like the process of absorbing nutrients and medications into the body by eating or drinking them (Cambridge English Dictionary).
- Ingestion is the process of obtaining and importing data for immediate use or transfer. Ingestion may be in batches or in real time using pre-processing or semantics.
- The L3 layer is for storage of data from the L2 layer.
- The L4 is for data processing using software, such as MapReduce, Hive, Pig or Spark. The top layer L5 is for data consumption. Data is used in analytics, visualizations, reporting, export to cloud or web servers.

L1 considers the following aspects in a design:

Amount of data needed at ingestion layer 2 (L2)

Push from L1 or pull by L2 as per the mechanism for the usages Source data-types: Database, files, web or service Source formats, i.e., semi-structured,unstructured or structured.

L2 considers the following aspects:

Ingestion and ETL processes either in real time, which means store and use the data as generated, or in batches. Batch processing is using discrete datasets at scheduled or periodic intervals of time.

L3 considers the followings aspects:

Data storage type (historical or incremental), format, compression, incoming data frequency, querying patterns and consumption requirements for L4 or L5 Data storage

using Hadoop distributed file system or NoSQL data stores—HBase, Cassandra, MongoDB.

L4 considers the followings aspects:

Data processing software such as MapReduce, Hive, Pig, Spark, Spark Mahout, Spark Streaming Processing in scheduled batches or real time or hybrid Processing as per synchronous or asynchronous processing requirements at L5.

L5 considers the consumption of data for the following:

Data integration Datasets usages for reporting and visualization Analytics (real time, near real time, scheduled batches), BPs, BIs, knowledge discovery Export of datasets to cloud, web or other systems.

10. List the functions in data management.

Data managing means enabling, controlling, protecting, delivering and enhancing the value of data and information asset. Reports, analysis and visualizations need well-defined data. Data management also enables data usage in applications. The process for managing needs to be well defined for fulfilling requirements of the applications.

Data management functions include:

1. Data assets creation, maintenance and protection
2. Data governance, which includes establishing the processes for ensuring the availability, usability, integrity, security and high-quality of data. The processes enable trustworthy data availability for analytics, followed by the decision making at the enterprise.
3. Data architecture creation, modelling and analysis
4. Database maintenance, administration and management system. For example, RDBMS (relational database management system), NoSQL
5. Managing data security, data access control, deletion, privacy and security
6. Managing the data quality
7. Data collection using the ETL process
8. Managing documents, records and contents
9. Creation of reference and master data, and data control and supervision
10. Data and application integration
11. Integrated data management, enterprise-ready data creation, fast access and analysis, automation and simplification of operations on the data,
12. Data warehouse management
13. Maintenance of business intelligence
14. Data mining and analytics algorithms.

11. How are data architecture layers used for analytics?

Note: Refer to the answer of Question No:9 for getting answer to this question

12. Show architectural design layers in 'Automotive Components and Predictive Automotive Maintenance Services.

ACPAMS uses Big Data types as: machine-generated data from sensors at automotive components, such as brakes, steering and engine from each car; transactions data stored at the service website; social networks and web data in the form of messages, feedback and reports from customers. The service provides messages for scheduled and predictive maintenances. The service generates reports on social networks and updates

the web data for the manufacturing plant. The service generates reports about components qualities and needed areas for improvement in products of the company.

Note: Using these sources, related to the generic architecture and customize the answer accordingly.

13. What are common pitfalls in analysis of data? Give example

Spurious Correlation:

Write explanantion in terms of the example " Do strokes deliver babies" and show that how data modelling resulted in false prediction.

- http://en.wikipedia.org/wiki/Spurious_relationship
- http://www.cut-the-knot.org/do_you_know/misuse.shtml

Gaps in the Data:

Explain about selection Bias done according to the conveneice results in false prediction/bad modelling. The examples to be explained here are Rutgers university study and Medicine.

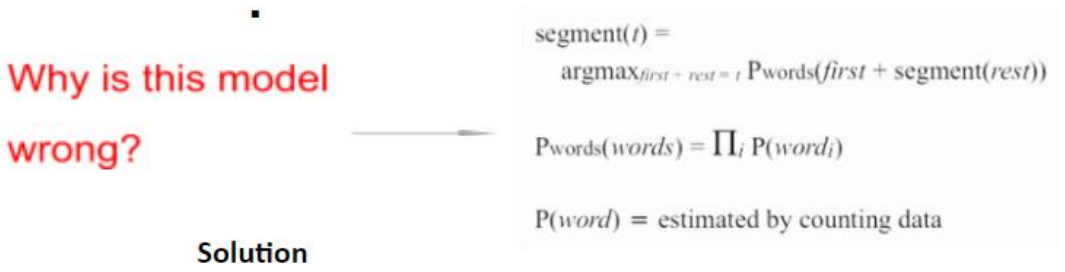
- *More Data, More Probl/ems: Is Big Data Always Right?* ARI ZOLDAN http://www.wired.cominsights/2013/05/more-data-more-problems-is-big-data-always-right/
- *The Information Architecture of Medicine is Broken* Ben Goldacre http://strataconf.com/strata2012/public/schedule/detail/22941
- https://www.youtube.com/watch?v=AK_EUKJyusg

Do refer to the links provided to understand the pitfalls clearly.

14. How is error estimation of model done in Big Data problems? How is this different from the traditional approach?

- Traditional Approach estimates error mainly based on Gaussian models, however this error estimation is completely different when it comes to Big data problems.
- Purely empirical: cannot be analysed by theory
- Divide data into *training set* and *testing set*
- Develop algorithm using training set; estimate error from testing set
    - Can be used to compare analytics algorithms
- Examples
    - Nate Silver: weather prediction: human adjustment
    - Amazon recommendations
        - Derive model using historical data; make recommendations
        - Get statistics on how many people look at or buy recommendations

15. In the Peter Norvig video he defines the following function to compute the best segmentation for a sentence?



- Model assumes that each word in a sentence is independent of other words that follow which is not necessarily true
- For example

Consider "smallandinsignificant" It can be broken up as "small and in significant" or "small and insignificant"

- In and significant are common so the model picks that.
- Insignificant is rare word, but is used commonly with small so, independence assumption is incorrect**.**

16. Explain the steps involved in Google search.

https://www.electronicsforu.com/resources/how-google-search-engine-works

Kindly refer to the above link for understanding the steps involved.

17. Write short notes on the various Big Data Platforms.

https://medium.com/@satyanageshan8/5-prominent-big-data-analytics-tools-to-learn-in-2019-ee1d4f5b98ce

Topic 2- HDFS:

1. Assume that you are been given a 1TB of data. How much time it takes for read data?

(i) Single machine (4 I/O channels each channel 100 Mb/s)

(ii) 10 machines (4 I/O channels each channel 100 Mb/s)

Solution:

General Formulae to compute the time taken for reading Data:

$$\text{Time Taken} = \frac{\text{Total Size of the data} \times \text{Transfer rate}}{\text{Number of I/O Channels}}$$

On substitution in the formulae for a single machine =45 mts and 10 machines it is 4.5 mts.

2. What are the two ways that can be adopted to build a computational system?

The monolithic way is the equivalent of building a cricket team with a single star player, an all-rounder who is expected to excel in every department of the game, who can be relied on completely. A monolithic system typically is one powerful server, and if there is an increase in demand for computing capacity, spend more to increase the resources of one server. However, note that the computing power does not increase in the same proportion as it increases in resources. This is called **vertical scaling** and it does not vary linearly with the cost of resources.

On the other hand, a distributed system is the equivalent of a team of good players, none of them being a star all-rounder, but they work together efficiently. Here, the ownership of improving performance does not lie with a single machine, but with all the machines. The individual machines in the distributed system are called nodes, and the entire system is called a cluster. None of the individual nodes in this cluster is a supercomputer. As a matter of fact, they are typically cheap machines or commodity hardware. However, these machines can serve together all the data processing needs in a large scale.

What makes such an arrangement ideal for Big Data processing is the fact that, a cluster of machines can scale linearly with the amount of data that has to be processed. The capacity of the cluster increases with the number of machines that are added to the cluster. For example, if the number of nodes in the cluster is doubled, then obviously the storage capacity will be twice. Each machine comes with its own hard disk and doubling the number of machines naturally doubles the storage capacity. It is imperative to know how such a distributed configuration is attractive, in addition to linearly increasing storage, when the number of machines in a cluster is doubled, there is nearly twice the speed of execution, resulting from parallelism. Also, a distributed system satisfies the third critical requirement in big data processing, i.e., scale and it known as **horizontal scaling.**

3. What is Hadoop cluster and in how many different modes can you configure a Hadoop cluster?

One Hadoop cluster consists of master and slave machine (Linux box). The main configuration files of Hadoop cluster are 'hadoop-env.sh', 'core-site.xml', 'hdfs-site.xml', 'mapred-site.xml' and 'yarn-site.xml'. Hadoop package has a defined file structure and these files are in the path '$HADOOP_HOME/etc/hadoop'. Here, $HADOOP_HOME is the Hadoop software package path like '/usr/local/hadoop'. Hadoop cluster can be configured in three modes as explained below.

• **Standalone Mode:** This is the default mode to configure a Hadoop cluster. This mode is mainly used for debugging and testing purposes, and it does not support the use of HDFS operations.

• **Pseudo-Distributed Mode (single/double node cluster):** In this cluster, you need to configure all the four main xml files as mentioned above. All Hadoop daemons (Java processes) run on the same node. A single Linux machine acts both as master and as slave.

• **Fully Distributed Mode (multiple node cluster):** This type of cluster is used in an industrial application mainly for different layers of development, testing and production. Separate Linux boxes are allotted as master and slave. In addition, the need to configure failover for NameNode and Resource Manager here for high availability.

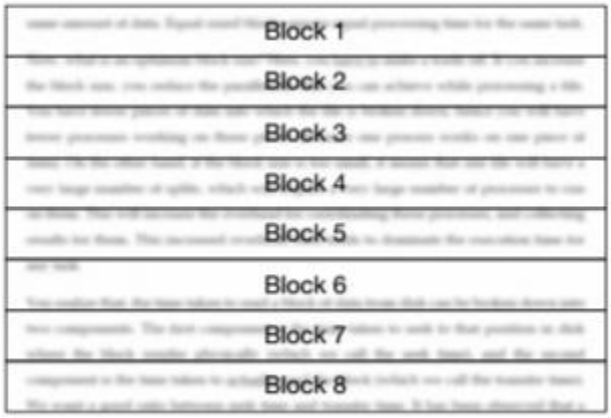
4. What is the usage of HDFS and how does it work?

- HDFS is Hadoop Distributed File System that is spread across multiple machines with commodity hardware.
- It can store huge volumes of data and execute data processing tasks at high speed and scale.

- The data stored in HDFS tends to be very large, for example, in petabytes. Also, majority of this data is semi-structured or unstructured.
- HDFS is suited for long time running job and not suited for low latency job.
- Availability of data in a cluster is always possible in HDFS with the master slave configuration.
- Within the cluster, one of the machines is designated as the master node. The master node is responsible for coordinating the storage across all other nodes on the cluster, which are slave nodes.
- On this master node, HDFS runs a process, which receives all requests that are made to the cluster and forwards it to the slave nodes which contain the data.
- All other machines in the cluster are designated as DataNodes. Thus, there is one NameNode per cluster and any number of DataNodes depends on the number of machines in that cluster.
- The NameNode serves two primary functions. First, any request from a client is passed to the NameNode because it tells us where to find the required data.
- The NameNode has the directory structure and it knows which piece of a file is located on which DataNode. Secondly, it also has the metadata for the file other than the actual content. Examples of metadata include file permissions, how the file is split up and where the replica of the file is stored. The function of the DataNode is simply that it physically stores the actual file contents.

5. What is the usage of HDFS for the read operation?

- Any file is splitted as smaller pieces of information called blocks.

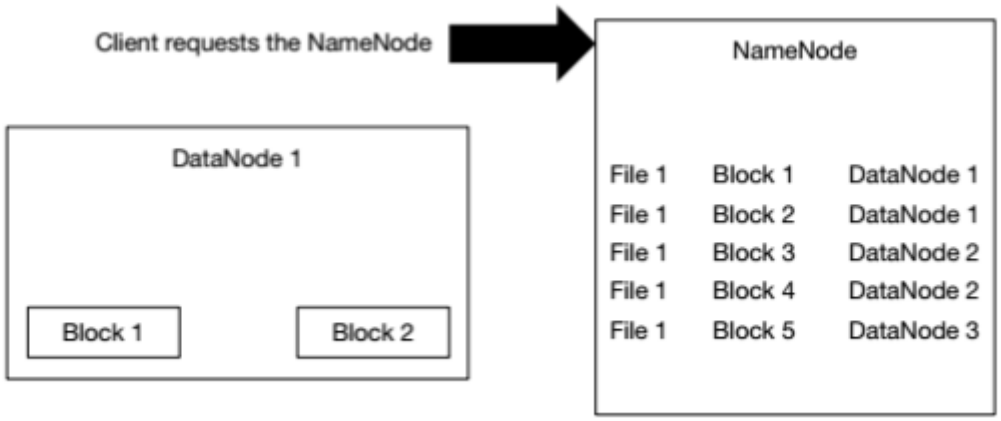


- Note that the blocks are of the same size. This allows HDFS to deal with files of different lengths in the same manner.
- The reason is due to the fact that HDFS does not deal with a file as a whole. Instead, it deals only with the blocks of a file, where each block, except the last, is of the same size.
- This makes the entire storage management mechanism within HDFS rather simple. At any point in time, only a block of data is dealt with. This block is also the unit for replication and fault tolerance.
- Therefore, there is no need to maintain multiple copies of the entire file. Instead, multiple copies of each blocks of same size into which any file is split up are kept. Hence, it brings about uniformity and standardization.

- These blocks are of equal size, it is made sure that when a block of data is processed, the same amount of data is dealt with.
- However, a critical question arises about the optimum size of a block. Here, a trade-off has to be made. If the block size is increased, then the parallelism that can be achieved while processing a file is reduced. However, there are fewer chunks of data into which the file is broken down, so there will be fewer processes working on those chunks (because one process works on one piece of data). On the other hand, if the block size is too small, then it means that one file will have a large number of splits, which will require large number of processes to run on them.
- This will increase the overhead for coordinating those processes and aggregating the results from them. Here, the increased overhead tends to dominate the execution time for any task.
- The time taken to read a block of data from the disk can be broken down into two components. The first component is the time taken to seek to that position in disk where the block resides physically (which is called the seek time). The second component is the time taken to actually read the block (which is called the transfer time). A good ratio between seek time and transfer time needs to be achieved. It has been observed that a block size of 128 MB provides an optimum ratio between seek time and transfer time.

6. Once the blocks are distributed across the nodes in a cluster, how do you know where the blocks of a particular file are located? How do you track down the data in a single file, when you have spread it out in blocks?

Now that you understand how the blocks are distributed across DataNodes and how the NameNode contains the desired mapping, let us examine how a file is read from HDFS. Reading a file from HDFS is a two-step process. Initially, you need to know where the blocks of data in that file are located. For this, the metadata in the NameNode is used to look up block locations in the DataNodes. Once the information on block location is obtained, reach out to the respective DataNode and read the block. Let us look at an example. In a cluster where a file is distributed across DataNodes as described above, let's read the beginning of a file, say 'File 1'. A client comes in and makes a request to the NameNode, saying it wants to read 'File 1' from the beginning (Figure 3.5).



will then look up in its internal table of contents to see where the first block ('Block 1') of 'File1' is stored. In this case, it happens to be DataNode 1. The NameNode then forwards this request to the DataNode 1 to read the actual contents from Block 1, and

it is this content that is returned back to the client (Figure 3.6). For example, three files are in HDFS with the size of a.txt (256 MB), b.txt (289 MB) and c.txt (370 MB). Thus, HDFS will allocate a total of 8 blocks (default size of a block 128 MB) for these three files. Here, a.txt will consume 2 blocks, b.txt and c.txt will absorb 3 blocks, respectively.

7. Remember that each of these DataNodes in the cluster is commodity hardware, which means it is prone to failure. There are two challenges to distributed storage. First, how to manage the failure of DataNodes? Second, how to manage failure of the NameNode?

   The fault tolerance strategy, which HDFS uses, is based on a replication factor. It is already seen that a file is broken up into huge number of blocks distributed across DataNodes in a cluster. No single machine holds the entire data for a single file. Further, every block of a file that is stored in the cluster is replicated across multiple machines. Replication factor is a configuration property that can be set. Based on this replication factor, the blocks which belong to a certain file are replicated or copied to other nodes. The number of copies will depend on the replication factor has been set.

   Note: Kindly refer to the ppts for understanding various scenarios of failure management.

8. What is Block Replacement Policy?

- First replica is placed on the local node
- Second replica is placed in a different rack
- Third replica is placed in the same rack as the second replica

9. 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.
   Hadoop v1 default – 64MBHadoop v2 default – 128MB
   Review: why has this increased?
   - It has been observed that a block size of 128 MB provides an optimum ratio between seek time and transfer time.

10. Edit logs are stored on shared storage. Why?

Topic 3- Map Reduce :

Here you may be asked to write pseudocode.

i. Counting the number of times, the word appears in the text file.

ii.

2. Rearrange the main configuration parameters that the user need to specify to run Mapreduce Job.

   a. Job's input locations in the distributed file system.

   b. Input format

   c. Class containing the map function

   d. Output format

e. Job's output location in the distributed file system.

f. Class containing the reduce function.

g. Application JAR file containing the mapper, reducer and driver classes for execution and deployment.

3. Assume that Hadoop spawned 100 tasks for a job and one of the tasks failed. What will Hadoop MapReduce framework do?

When one of the tasks out of 100 tasks fails, the map reduce framework will attempt to restart the task probably on a different node

4. What is the difference between an Input Split and HDFS Block? Please explain.

Input Split – the input file in HDFS that is provided as input to the MR job is split into multiple chunks and a mapper is started for each chunk. Each of these chunks is called an input split. It is best if the input split is the same as the HDFS block size to maximize performance.

5. Consider Example 1.6(i) of ACVMs selling KitKat, Milk, Fruit and Nuts, Nougat and Oreo chocolates. Assume 24 files are created every hour for each day. The files are at file_1, file_2, …., file_24. Each file stores as key-value pairs as hourly sales log at the large number of machines.

(i) How will the large number of machines, say 5000 ACVMs hourly data for each flavor sales log store using HDFS? What will be the strategy to restrict the data size in HDFS?

(ii) How will the sample of data collected in a file for 0-1,1-2, … 12-13,13-14, 15-16, up to 23-24 specific hour-sales log for sales at a large number of machines, say 5000?

(iii) What will be the output streams of map tasks for feeding the input streams to the Reducer?

(iv) (iv) What will be the Reducer outputs?

Solution

5000 machines send sales data every hour for KitKat, Milk, Fruit and Nuts, Nougat and Oreo chocolates, i.e., a total of 5 flavors. Assume each sales data size = 64 B, then data bytes 64 × 5 × 5000 B = 1600000 B

will accumulate (append) each hour in a file. Sales data are date-time stamped key-value pairs. Each of 24 hour hourly log files will use initially 24 data blocks at a DataNode and replicated at three DataNodes. A data file in one year will accumulate 1600000 × 24 × 365 B = 14016000000 B = nearly 16 GB. Each data block can store 64 MB. Therefore, 16 GB/64 MB = 250 data blocks in each file each year. However, hourly and daily sales analytics is required only for managing supply chain for chocolate fill service and finding insight into sales during holidays and festival days compared to other days. Therefore, a strategy can be designed to replace the hourly sales data each month and create new files for monthly sales data. A file sample-data of key-value pairs for hour-sales log in file_16 for sales during 15:00-16:00 will be as follows: ACVM_id10KitKat, 23 ACVM_id2206Milk, 31 ACVM_id2Oreo, 36 ACVM_id10FruitNuts, 18 ACVM_id16Nougat, 8 . . . ACVM_id1224KitKat, 48

ACVM_id4837Nougat, 28 . Map tasks will map the input streams of key values at files, file_1, file_2, ... file_23, file_24 every hour. The resulting 5000 key value pairs maps each hour with keys for ACVM_idNKitKats (N = 1 to 5000). The output stream from Mapper will be as follows: (ACVM_id10KitKat, 0), (ACVM_id1224KitKat, 3),.., .., ...,... ,.., .., ..., ...,....,... ,.., .., ... Hourly 5 output streams of mapped tasks for all chocolates of all 5000 machines will be input to the reduce task. The Reducer processes each hour using 5 input streams, sums all machines sales and generates one output (ACVMs_KitKat, 109624), (ACVMs_Milk, 128324), (ACVMs_FruitNuts, 9835), (ACVMs_Nougat, 2074903), and (ACVMs_Oreo, 305163). The reduced output serializes and is input to the analytics applications each hour.

6. Consider HDFS DataNodes in a cluster. Draw a diagram depicting 10 data nodes storing the data of 4 groups of students. Using the diagram, show the execution of MapReduce sub-tasks for each group in parallel on the DataNodes in a cluster.

Topic 4- YARN :

What is Resource Manager in YARN?

Resource manger in YARN is the component that allocates nodes for processing of jobs and tasks.

What are speculative tasks in YARN? Why are they sometimes killed?

Speculative tasks are tasks started by YARN when it observes that an already started task is running slowly.

How does YARN improve upon the deficiencies of Map Reducev1

What is the difference between the Capacity scheduler and the Fair scheduler?

Capacity scheduler and fair scheduler both have pools, but capacity scheduler will not allow a job to use more than its share if the usage of cluster is low. However, fair scheduler will allow a job to consume more than its share of resources if no one else is using the cluster.

Note: While you are reading the architecture, try to ask questions as why this architecture like this is…what happens if there is no input split. What happens if they is no schedulers etc.