ISOM 671: Managing Big Data – Assessment Part II

Sean Jung

1. Provide brief answers to the following short questions:

1.1. What are the differences between relational and NoSQL databases?

Relational database (RDBMS) can only store Gigabytes (GBs) of structured data; while NoSQL can handle Petabytes (PBs) unstructured data. Scale at RDBMS usually happens vertically (by having bigger server) and thus is expensive and difficult; while NoSQL scaling is horizontal (by having more nodes) and thus is cheap and easy to implement. Additionally, RDBMS is ACID compliant, but many NoSQL solutions sacrifice ACID compliancy for performance and scalability.

1.2. What is Hadoop? What are the two core components of Hadoop?

Hadoop is an open-source software framework for storing, processing, and analyzing "big data". They are characterized as a scalable, fault-tolerant, and well-distributed computing environment. They are being widely used in a variety of industries such as Finance, Healthcare, Manufacturing, etc. The two components of Hadoop are HDFS and YARN.

1.3. What is YARN?

As one of the main components of Hadoop, YARN manages the processing resources of the Hadoop cluster, schedules jobs, and runs processing frameworks

1.4. What is a block in HDFS?

When files in HDFS are broken into block-sized chunks, these are stored as independent units called data blocks. Hadoop distributes these blocks on different slave (DataNode) machines; while the master server (NameNode) stores the metadata about location of the blocks.

1.5. What is the difference between HDFS and NAS (Network Attached Storage)?

HDFS designs to store very large files (scaled to petabytes) running on a cluster of commodity hardware that serves as flexible import/export tools; while Network-attached storage (NAS) is a file-level computer data storage server that serves individual files with caches while providing data access to a heterogeneous group of clients. While NAS is costly and network gets easily saturated by reading large portions of the network, HDFS has drawbacks of being slow query speed and having no transaction support.

1.6. What is the difference between managed and external tables in Hive?

Managed tables are Hive owned tables where the entire lifecycle of the tables' data is managed and controlled by Hive. External tables are tables where Hive has loose coupling with the data. One can use managed tables when Hive should manage the lifecycle of the table, or when generating temporary tables. One is advised to use external tables when files are already present or in remote locations, and the files should remain even if the table is dropped. (Source: Cloudera)

1.7. What is a UDF? Write a UDF to capitalize the first alphabet in a string.

User-Defined Functions (UDFs) are user-programmable language that act on one row.

```
def upperfirst(x:String): String = x[0].upper() + x[1:len(x)]

spark.udf.register("upperfirst", upperfirst)
```

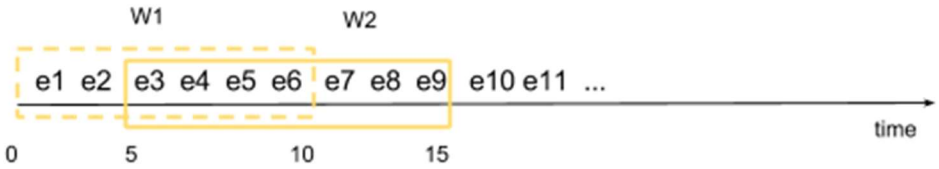
1.8. What is DStream? What is the difference between stateful and stateless stream?

A Discretized Stream (DStream), the basic abstraction in Spark Streaming, is a continuous sequence of RDDs (of the same type) representing a continuous stream of data. DStreams can either be created from live data using a StreamingContext or by transforming existing DStreams using operations.

Stateless transformations are simple RDD transformations, applying on every batch (every RDD) in a DStream. It includes common RDD transformations such as map(), filter(), reduceByKey() etc. Stateful transformation is when results of the current batch is computed when using data or intermediate results from previous batches. *Stateful transformations* are operations on DStreams that track data across time. Thus, results for a new batch is generated by making use of some data from previous batches.

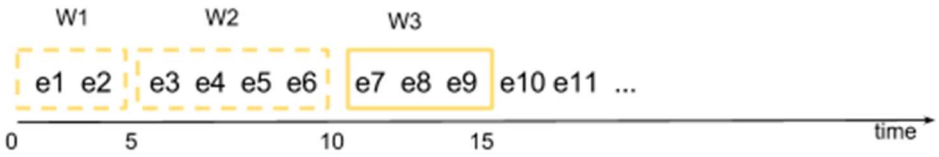
1.9. What is the difference between tumble window and sliding window?

In a *sliding window*, tuples are grouped within a window that slides across the data stream according to a specified interval. An example would be to compute the moving average of my heartbeat range using FitBit device across the sixty minutes workout session, measured every minute.



(Source: Cloudera)

In a *tumbling window*, tuples are grouped in a single window based on time or count. A tuple belongs to only one window. None of the windows overlap; each segment represents a distinct time segment. An example would be computing the average price of a stock every five distinct minutes.



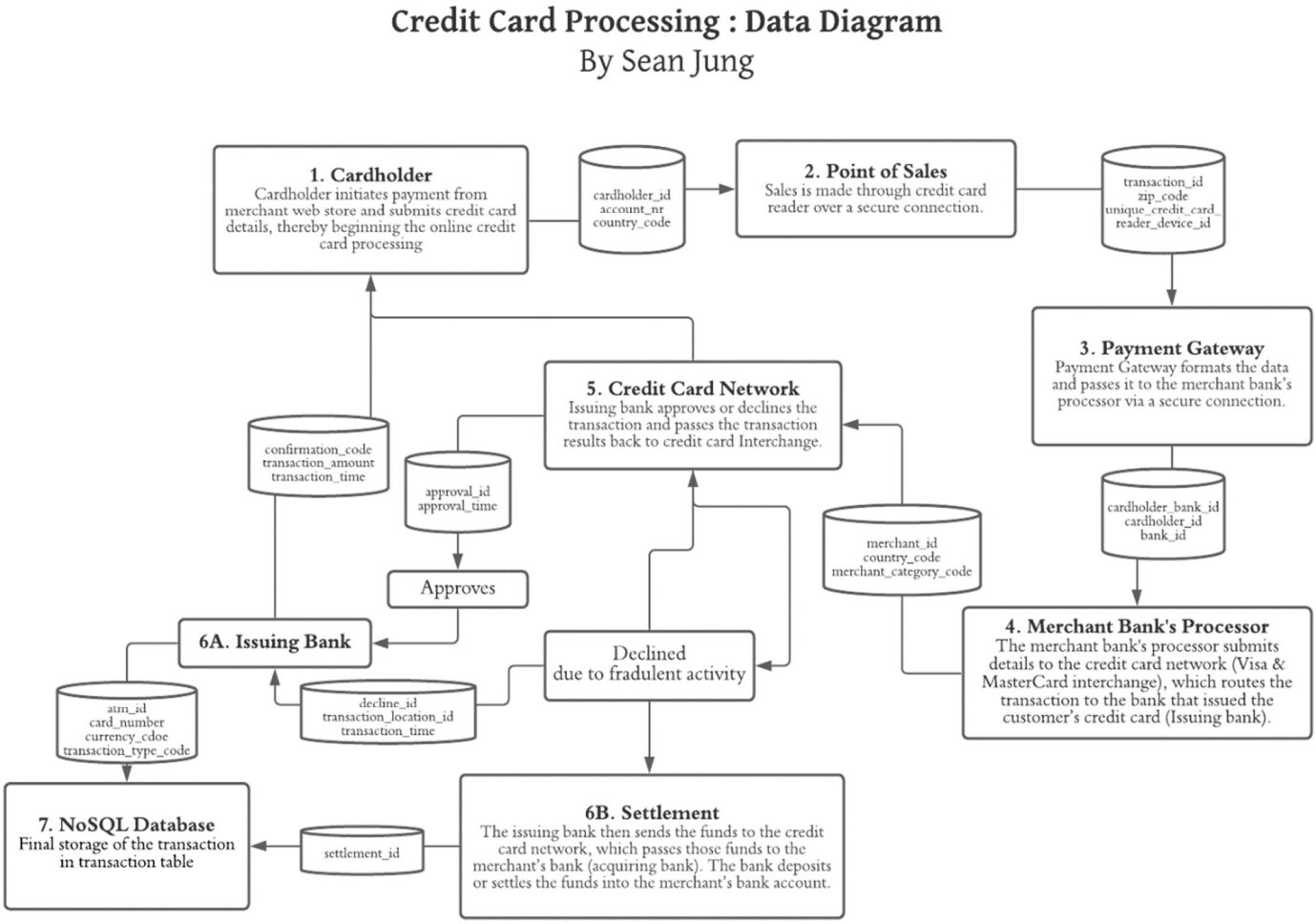
(Source: Cloudera)

1.10. How would you pick between Hadoop MapReduce or Apache Spark for a project?

It all depends on the need of the user. MapReduce (MR) only allows for batch processing; while Apache Spark (AS) allows batch processing as well as real-time data processing. MR is also mainly a data processing engine; while apache is more of a data analytics engine.MR also has very high latency and difficult to write and debug codes; while AS supports Java, Scala, Python, and R (easier to debug than MR) and is much faster than MR framework. Based on these features of each platform, one can choose to use MR if the main goal is to process data or choose AS when the main objective is to analyze the data using different ecosystem such as PySpark, MLib, Spark Streaming, and Spark SQL. (Source:Educba)

2. Let’s Pretend that you are working for JPMorgan Chase bank and are tasked to develop a solution for tracking credit card charges to identify fraud. Your application is supposed to check for repeated transactions within the same day or transactions originating from two separate locations with over 1,000 miles distance between them (recall code submission 7).

2.1. Develop a data flow diagram that includes all data elements from card swipe at a point-of-sale (POS) system to final storage of the transaction in transaction table in a NoSQL database. In case of potential fraud send notification to customer and the bank.



2.2. Briefly discuss what database systems will you implement to store all relevant data that is generated in the process?

Column-based NoSQL database may be most appropriate for storing such credit card transactional data. This type of database is suitable for reading and writing across large volumes of data as it runs on clusters of multiple servers. Since they are schema-free, they are considered to be highly flexible in terms of type of data that it can store. Column-based database is also one of the best platforms to store key-value pairs in a massively parallel system, which can aptly store the customers' information regarding the transaction as noted in the data flow diagram above. Due to the structural nature of what is being stored in this type of database, it can also allow for better compression of the data as values in columns can be similar (different instances of the same attributes) while values in rows can be dissimilar (different attributes of the same instance), allowing the version controlling of the data values (also having the ability to overwrite the values in the database, if necessary).

Most importantly, column-based databases can be geographically distributed over multiple data centers, making it very suitable in this context where multiple transaction can happen at the same time under the different location. Apache HBase is the most well-known column-based database that integrates either HDFS or Amazon's S3 for data storage. As it utilizes the Hadoop database that can read and write access to the data in real time, not only can it provide automatic failover support between Region Servers, simple alert mechanism can be deployed to enable the fraud-detection system in the times when two different transaction happen within the proximal time bound but under massively distant geographic location, making it suitable database in this context.

3. In 2016 Marr listed 17 predictions around the future of big data [1]. Whereas Reis, Braatz, and Chang 2016 [2] mention that big data does pose challenges that needs to be considered for the future. More specifically, they took the context of chemical industry and mentioned that volume may not always be good representation of the target population, and that people and processes evolve over time so the existing data may not be relevant. Taking both articles in the context of financial industry (say JPMorgan Chase Bank), discuss:

3.1. How the company should handle the 4Vs of big data during the next five years. i.e. for each V, pick one area where the company could use (or is using) data for decision making.

Volume: In the context of financial industry, some of the phone maker such as Apple and Samsung have used their own payment system (Apple pay and Samsung pay, respectively) to collect not only the type of transformation but also the attributes of the customer who processes a list of transaction using their payment system. Of course, this result in humongous amount of new data that are being generated every minutes

Velocity: In the context of financial industry, they must process these transactions to detect potential fraud or process billions of balance-transfer records via mobile banking application to detect malicious activity. Velocity is helpful in detecting trends among people that are interacting with financial industry in any way shape or form. Processing of streaming data for analysis also involves the velocity dimension.

Variety: Different ways of data collection in financial industry have led to various types of data that are being collected. It can include mobile data (e.g., mobile banking and what people say in the bank's social media account), research data (e.g., surveys and industry reports), location data (e.g., mobile device data and geospatial data of the bank's user), images (e.g., scanned image of check, credit card), e-mails, signal data (e.g., sensors and RFID/NFC devices).

Veracity: In the context of financial industry, they always go through some degree of sanity-check to make sure the transaction is valid before they accept the data for analytical or other applications.

3.2. For each of those four areas, discuss what data and technology should company plan to incorporate in their decision making in future and the how they address any associated challenges with data. **(Please do not count question stem as a part of the page limit)**

Volume Challenges: Currently, most financial big data projects rely on happenstance data (data passively collected from processes operating under normal circumstances), meaning that even if the size of the data is large, the span of the data is not wide enough to capture all the possible variability within all regular operating scenario. For the process control and optimization activities under financial activities, process description must capture the actual influence of each manipulated input variable on the process outputs to increases the data's value for predictive, control and optimization techniques. Currently, the relevant or interesting information related to financial transactions may happen on only a few dispersed occasions despite the sheer volume of industrial data, and data mining and knowledge discovery tools (such as LASSO regression and ensemble method) can handle very large volumes of data that are rich in information.

Variety Challenges: Currently, financial industry seeks to utilize a priori knowledge (some domain knowledge about the main sources of variety in financial information affecting a massive dataset). However, making use of it in conventional financial analytics is not always straightforward. Incorporating information about the structure of the processes in data-driven analysis is an important research path for the future, especially in the fields of faults diagnosis and predictive modeling. For example, Bayesian approaches and data transformation based on network inference can be adopted to convert a priori knowledge into data-driven financial modeling to

better understand the customers. Currently, multiple data structures are made analyzable by focusing on developing financial analytic platforms that can effectively incoroporate and fuse all of these heterogenous sources of financial information found in different processes. However, future research can be conducted to develop methods that detect and handle such time-varying nature of the processes via dynamic and adaptive schemas.

Veracity Challenges: Big data cannot replace the need to understand how these financial data are acquired and underlying mechanisms that generate variability. A major concern in the analysis of large dataset has to with the quality of the data. Quantity doesn't necessarily equate to quality. Currently, financial sectors collect information about measurement uncertainty (a parameter associated with the result of a measurement that characterizes the dispersion of the values that could reasonably be attributed to the quantity to be measured) They combine such uncertainty data with the raw measurements to improve the data analysis, empirical modeling, and subsequent decision making. However, jumping into the analysis of massive data sets is contrary to a reliable statistical engineering approach to address process improvement activity.

Velocity Challenges: large quantities of financial data being collected at high speed can pose a problem such as implementation of appropriate online collection technique and defining the appropriate granularity to adopt for data analysis. It is crucial to select the most effective resolution for the particular analysis, and a default resolution by a third party without domain knowledge of the specific data will be risky business to conduct. To mitigate this, they can focus more on developing sound ways for selecting the proper resolution that consider the variables' dynamic and noise features by adopting adaptive fault detection and diagnosis.

4. Read: https://assets.kpmg/content/dam/kpmg/pdf/2015/11/cloud- economics.pdf and answer the following short questions:

4.1. What is the difference between private, hybrid, and public clouds?

Private Cloud retains everything in-house and thus maximizes the legacy investments but leaves a capital intensive and high-cost structure in place. This type of cloud is suitable for the type of application that requires strict security or regulatory compliance where the migration costs are excessive. *Hybrid Cloud* have the similar advantages of public cloud, mainly in terms of having cost-effective elastic surge capacity. They are considered to be somewhat in the middle between private and public cloud. Public Cloud adopts rapid increase and capacity and maximizes the benefits of a low-cost structure, but as a trade-off its downside requires significant rundown of the legacy environment.

4.2. Why do you think the "cost of ownership gap of 30 to 40% between traditional IT and public cloud services is predicted to continue"?

I suppose that this is attributed to the snowball effect; When the early adopters initially went into this field, the cost of implementation was relative cheap while gaining market share was relatively easy. However, as competition gets fiercer more firms have now entered this market, and as a result the level of innovation has increased. Consequently, this results in a positive feedback loop in a way that increased competition leads to increased level of sophistication and innovation in this technology, making it even more difficult for newcomers to enter into this field. Pareto principle also may have applied in this context that only 20 percent of the cloud technology companies dominate 80 percent of the market share, meaning 80 percent of the emerging cloud firms are basically fighting for 20 percent of the entire market share. Such increase in demand without having the equivalent level of supply will definitely even more increase the cost of adoption with ownership gap from current baseline of 30-40 percent.

4.3. Do you agree with KPMG's phased-approach to developing a business case for cloud-based big data solution? Why (not)?

When we want to jump into a water for swimming, we don't want to dive straight into a water without acclimating our body to the temperature of the water because sudden change in body temperature could lead to hypothermia, and heat attack in the worst-case scenario. Same analogy applies to the cloud migration strategy as KPMG's phased-immersive-approach anecdotally led to 50–60 percent steady-state savings while 30–70 percent increase in utilization and paybacks beginning in less than 18 months. It suggests that firms can maximize the likelihood of success by focusing most of its resource into mobilizing and accelerating action that capture the benefits of cloud computing, developing the baseline cost model for existing infrastructure, etc. Phased approach would maximize the odds that phased approach is not only about lowering the cost of implementation, but it is also about increasing agility to rapidly respond to the market while minimizing the potential threats such as security or compliance with regulatory, legal, and contractual obligations.

4.4. If cloud solutions are economical, should an organization invest in data center? When and why?

With software running on cloud is becoming the de facto standard, major share of all new servers is now going to cloud providers where the percentage of the cloud service provider increases each year. If the organization is left behind from such innovation, they will eventually get outcompeted and eventually excluded from the market. While other firms are building apps in the cloud to ensure they can maximize their efficiency to analyze their data to extract insights that will be of strategic value of the firm, those firms whose infrastructure is not compatible to IoT integrations would not be able to monetize on billions of data points helping firms to capture, analyze, and respond to fast-changing market trends. Regardless of up-front investment and short-term lead time, firms MUST consider investing in such cases if the still want to survive in the coming years of digital transformation era.