



UNIVERSITY *of* GREENWICH

Course Title: Big Data COMP1702

Student Name: Mohamed Al-kaisi

Student ID : 000931504

Contents

Introduction	3
Part 1 :	4
1.2 Produce a Big Data Architecture for ABC Investment Banking providing details of the following components in detail:	4
1.2 The Business Head has a limited understanding of IT and is aware of the data warehouse and has suggested you build this only. Explain why you will be making a Data Lake for this solution. Discuss the approach required for implementing Data Lakes.....	14
1.3 The business wants near-real-time performance when certain financial products are being discussed on social media for it to act promptly. It has been suggested the parallel distributed processing on a cluster should use Map Reduce to process this requirement. Provide a detailed assessment of whether MapReduce is optimal to meet this requirement, and If not, what would be the best approach.	16
1.4 Devise a detailed hosting strategy for this Big Data project, and how this will meet the scalability, high availability requirements for this global business.....	17
1.5 Using Cloud offering from either Amazon AWS or Microsoft Azure, how would either support this Big Data project.	19
2 Part 2	20
2.1 Using a tool from the Hadoop ecosystem, develop a solution to demonstrate <i>Schema on reading</i> . How does this differ with <i>Schema on write</i> (Include your file(s) and code in your coursework report?.....	20
2/ Compare and contrast Hadoop and Relational Database system	26

Introduction

This article explains the recommended solution for the ABC investment Bank. The bank wanted a solution that enables them to utilize social media and other data feeds to produce trading strategies and portfolio rebalancing decisions to give them a competitive advantage. The bank required an IT solution that will inform them whether they should buy or sell or hold a financial instrument. The Bank gave in the requirement that the answer should use our data from social media, market data, Online news feed, Broker notes and corporate data.

The bank suggested as well that the solution should be highly available, scalable, and accessible from worldwide offices such as London, New York, Tokyo, Hong Kong, Sydney of ABC investment bank. Lastly, the data volumes expected to be in 100's Petabytes.

The Business Head of ABC investment Bank has been asked questions at the start of this project to gain a detailed understanding of the requirement. The interview has informed us that the bank is using Microsoft products such as Microsoft Office and SQL Server as a database. The data volume should be scalable and not just limited to 100's Petabytes scale.

.

Part 1 :

1.2 Produce a Big Data Architecture for ABC Investment Banking providing details of the following components in detail:

-Data sources,

The sources of data to produce this Trading decision retrieved from the requirement given, which will be from Social media, Market data, Online news feed, Broker notes, Corporate data.

- Data storage

Data storage refers to it in Big data terms as a data lake. When data lake appeared at the start, its build on HDFS clusters on-premises. There are different ways of creating data lakes for big data projects by building the data on-premises or using Cloud services such as AWS data lake or Azure data lake.

Building data on-premises comes with challenges which are, building data pipeline, and it can be complex on-premise as it requires to manage both the hardware and software infrastructure. Hardware can be spinning up servers, orchestrating bath “ETL” jobs, and dealing with downtime. The software infrastructure requires data engineers to integrate a wide range of tools to organize, ingest, and pre-process and query the data stored in the lake. The cost for implementing the data storage on-premise can be high aside from the upfront investment needed for purchasing the server and storage equipment, the price for operating, and management involved mainly manifesting in IT engineering cost. In terms of scalability, on-premises will cost more money as when you want to increase the size of data, and then you will need to add and configure servers manually.

Moving data lake into the cloud comes with several advantages which are,

Storing big data into the cloud removes the idea of the need to build and maintain the infrastructure like the one built-in on-premises. The engineering costs are lower when using the cloud for building a data pipeline, which done more efficiently using the cloud. The data pipeline in the cloud is pre-integrated, which a solution can implement quickly without having to invest hundreds of hours in data engineering.

Some cloud providers such as amazon s3 and Athena provide data lakes that provide transparent scaling, which removes the idea of having to add machines or manage clusters. The cloud data lake does update automatically, and it usually makes the latest technology available. Cloud new services enable you to add any without changing architecture. Lastly, cloud providers work to prevent service interruptions, storing redundant copies of data on different servers to avoid data of loss and make fast access to data.

The bank requirement suggested that the application should be highly available, scalable, and accessible from the worldwide offices. Based on that requirement, we decide to use cloud data storage instead of building the data lake on-premises due to scalability, high availability using the cloud provides better security with automatic maintenance. Few cloud services offer data lakes such as AWS, AZURE data lake, and Google. Azure also provides one more storage that is different from the data lake which is called Azure blob storage. The main difference that enabled us to choose data lake over azure blob storage due to the size that is limited where the data lake store has no limits on size.

There are disadvantages of moving into the cloud such as storage costs where some cloud providers required you pay per hour for storage like Amazon offer multiple options for storing data with a variable per hour expenses so it makes it easier to optimize but the fact the store will keep increasing, and the payment keeps growing. The research shows building the data lake on the premise's it's cheaper, but that would not be the case at all time as it would require engineering and IT cost but also when data growing then new servers need to put in place. Even though moving to the cloud can be cost-effective but more tools offered within the data lake such as analytics layer that comes in with AZURE which takes cost off from other needed products.

Conclusion

Looking at both data lake amazon and Azure, clearly shows that even though AWS provide robust functionality for data lake as it mentioned on their website (Data Lakes Storage | AWS, 2019). Amazon s3 provides a cost-effectively build and scalable data lake of any size. The lake secured by 99.999999999% of durability and with that level of strength suggests that if you store 10,000,000 objects in amazon s3, then you may lose a single purpose every 10,000 years. The service automatically creates, and stores copy all the uploaded data across multiple systems to ensure it is available when it's needed but also protected against failure, threats, and errors. However, Azure in (Data Lake | Microsoft Azure, 2019) seems to provide more suitable functionality for this project and will be easier to implement the AZURE with rest of the component needed for the project due to the reason that the bank uses Microsoft products already. Another reason for choosing Azure because Microsoft Azure provides data lake architecture built on HDFS standard which makes it easier to migrate existing Hadoop data. You can access Azure data lake from Hadoop that is available through HDInsight using the WebHDFS-compatible with Rest APIs. The data lake architecture that AZURE provides consist of two layers, one for storage and one for analysis. The analytic layer made of azure data lake analytics and HDInsight which is a cloud-based analytics service. This layer will support the rest components needed to build on the big data.

- Batch processing

This component is where processing happens of blocks of data stored over a period. Batch processing, in our case, operates on large data sets where the computation takes significant time. If we wanted to process data in real-time as they arrive and quick to detect conditions within the small-time period of receiving the data, then the best component for this situation would be stream processing.

Batch processing enables the bank to have further interactive exploration that provides the modeling-ready data for machine learning or writes the data to a data store which optimized for analytics and visualization.

An example of batch processing would be transforming an extensive data set of flat, semi-structured CSV or JSON files into a schematized and structured format that is ready for further querying. The data is converted from raw formats into ingestion (such as CSV) into binary forms that are more performant for querying because they store data in a columnar format, and often provide indexes and inline statistics about the data.

Some challenges come in with the use of this component, such as data format, encoding. Data format and encoding happen when the stored files using unexpected formats or encodings such as the use of a mix of UTF-16 and UTF-8 encoding, or it may contain unexpected delimiters, or it has unexpected characters. Another example would be text files that have tabs, spaces, or commas interpreted as delimiters. To resolve those challenges by ensuring the data loading and parsing logic must be flexible enough to detect and handle these issues.

Technology provided by Azure that comes in with Azure data lake,

U-SQL: This is a query processing language used by the Azure data lake analytics. This language combines both the declarative nature of SQL with the procedural extensibility of C# and takes advantage of parallelism to enable efficient processing data at a massive scale.

Hive: This is an SQL-like language supported in all Hadoop distribution. Colony used to process data from any HDFS-compatible store such as Azure blob storage and Azure Data Lake Store.

Pig: This language used in much Hadoop distribution, including HDInsight. The language is useful for processing data that is either unstructured or semi-structured.

Conclusion

In the previous component suggested using for this project, the data lake provided by Azure cloud service, which comes with two layers, one for storage and one for analysis. Azure Data Lake Analytics is providing different technologies for batch processing such as U-SQL, Hive, Pig.

MapReduce, hive, and pig used for batch processing only not for real-time data processing, and Spark streaming derived from Apache spark is the suitable technology. The question is related to real-time data processing which MapReduce, Hive, and pig. The best technique would not be but Apache spark which sparks streaming by Azure that choosing in point 1.1 Big data architecture is the best solution. However, I decided to make a comparison at the start for MapReduce against other processing technology used for batch processing which is Hive and pig. In question 1.1, I have selected Hive for batch processing so this to prove that there are technologies to use that consume less time and less complex than MapReduce but there some functionalities that only exist within MapReduce.

MapReduce is a framework or programming model in the Hadoop ecosystem used for processing extensive unstructured data set in a distributed manner by using many nodes. Pig and Hive are components that sit on to the top of the Hadoop framework for processing extensive data without the users having to write a Java-based MapReduce code. Both Hive and pig are converted to Java map-reduce programs before they get submitted to Hadoop for processing. Having to write SQL queries instead of Java codes ensures the ease of use of the system and will consume low time to deliver the task needed as Hadoop developers would need to learn Java. The question 1.1 for the data architecture in batch processing component Hive was selected.

In the article written by (Pol, R,2016) compared Hive against spark and MapReduce.

- MapReduce requires more lines of codes than Hadoop and pig due to the use of SQL.
- Both pig and Hive provide a higher level of abstraction, where MapReduce offers a low level of abstraction.
- Development efforts needed more with MapReduce than with Hive or pig.
- Pig and Hive are slower than a fully tuned MapReduce program.
- Coding in Hive and pig remove the possibility of having coding bugs, which an opportunity can happen with the use of MapReduce.
- Pig struggles in dealing with unstructured data such as images, videos, audio, log data, and test.
- Pig struggles with dealing with poor design of JSON, XML, and flexible schema.
- Using MapReduce requires knowledge in Java programming language.
- Pig code can extend through various defined functions written in Java, Python, JavaScript, and Ruby.
- Pig has tools for data execution, data manipulation, and data storage.
- Yahoo highly promotes pig as its data engineers for processing data on the most prominent Hadoop clusters in the world.

- Hive started by Facebook to provide Hadoop developers with a traditional data warehouse interface for MapReduce programming.
- Developers who are familiar with SQL will be more comfortable for them to use the same skills with Hive, which makes them feel at home.
- One disadvantage of Hive is that developers need to compromise on optimizing the queries as it depends on the Hive optimizer, and they need to train the hive optimizer on efficient optimization of queries.

This component will be using Hive for batch processing through the HDInsight cluster provided by Azure due to the ease of use compared to other technologies.

- Real-time message ingestion,

Real-time message ingestion is a logical component that deals with streams of data captured in real-time and processed with minimal latency defined in (Choosing a real-time message ingestion technology, 2020).

Real-time processes defined as the processing of the unbounded stream of input data and with concise latency requirements for processing that are measured either by milliseconds or second. The type of data this component will use is either unstructured or semi-structured format such as JSON and has the same processing requirement that was for batch processing but with shorter turnaround times to support real-time consumption. When the data processed, then it's often sent to the next stage, where it's written to an analytical data store optimized for analytics and visualization.

The real-time message ingestion is a logical component that deals with capturing and storing real-time messages that are to be used by a stream processing consumer. This component basically could be implemented as a simple datastore where the new data get deposited in a folder. The solution for this component with the use of Azure requires a message broker such as Azure event hub, Azure lot hub, or Apache Kafka which defined in (Choosing a real-time message ingestion technology, 2020).

Azure event hubs:

This one solution provided by Azure as a messaging solution for ingesting millions of event messages per second. Multiple consumers can process the captured data in parallel. This solution supports advanced messaging querying protocol 1.0 and as well provides a compatibility layer that enables applications using KAFKA protocol to treat events using Event Hubs without the need to change any claim.

Azure lot Hub:

Another solution by the Azure cloud, which provides bi-directional communication between millions of lot devices and cloud-based back end. Lot Hub comes with several features such as monitoring of device connectivity and device identity management events, message routing to another azure service, Quarriable store for device metadata, and synchronized

state information, securing communication and access control using per-device security keys. The last feature is having multiple options for the device to cloud and cloud to device communication and those options have one-way of messaging, request-reply methods and file transfer. Both lot Hub and event hub do similar jobs in terms of message ingestion, but for the lot, the Hub designed for managing lot device connectivity as well as message ingestion.

Apache Kafka on HDInsight:

Apache Kafka is an open distributed streaming platform that used for building a real-time data pipeline and streaming application. Kafka comes with functionalities that make it stand out, such as message broker that like a message queue that enables us to publish and subscribe to named data streams. These platforms are scalable, full-tolerant, and extremely fast. Azure provides Kafka that is managed and scalable. This component comes with several features used for different purposes which are,

- The platform supports the publish-subscribe message pattern and the fact Kafka used it as a message broker.
- The platform includes an activity tracking feature as Kafka provides in-order logging of records, which used to track and re-create activities such as user activity on a website.
- The platform enables you to aggregate information from different streams to include and centralize information into operational data.
- The platform enables you to combine and enrich data from multiple input topics into one or more output topics.

Conclusion

a lot of Hub functionality seems the best feature that demonstrates scalability and availability which chosen for this solution. It provides a feature that enables to have the cloud to device communication, enable device-initiated upload that allows the bank to upload data from their devices into the Cloud, the device state information uses device twin with use of lot Hub which are JSON documents that store the device state information such as metadata, configuration, and condition. Lot hub support protocol such as AMQP, HTTPS, which strengthens the security for this solution.

(WhatIs.com, 2019) described the Advanced protocol message queuing protocol that is an open-source published standard for asynchronous messaging by wire. This type of contract enables encrypted and interoperable messaging between organizations and applications. Lastly, this has been used in client /server messaging and in lot device management.

HTTPS mentioned in this website (HOFFMAN, 2019) as the secure version of standard hypertext protocol. The protocol used during the communication you make using the browser with another site. In our example, this means when the bank communication with the cloud system will be secure and protected.

- Stream processing

Stream processing derived from (A Gentle Introduction to Stream Processing, 2020) stated that enables users to query continuous data streams and detect conditions quickly with a small-time period from the time of receiving the data. The detection time differs from a few milliseconds to minutes.

Technology provided by Azure cloud which defined in (Choosing a stream processing technology, 2020),

Azure stream analytics:

The technology enables you to run perpetual queries against an unbounded stream of data. The questions take streams of data from storage, message brokers, filter, then aggregate the data based on temporal windows and write results to sinks such as databases, data lakes, or directly in Power BI. The stream analytics also uses a SQL based query language that can support temporal and geospatial constructs, and this can extend using JavaScript.

Storm:

The storm is an open-source framework technology that used for stream processing, which uses a topology of spouts and bolts to consume. The process then outputs the results from real-time streaming data sources. This technology you can find to be supplied by Azure HDInsight clusters and the topology implemented in Java or C#.

Spark streaming:

Another open-source platform technology designed for general data processing. Azure provides spark streaming API, which enables you to write code in any supported language such as Java, Scala and python. Azure provides Spark 2.0 within the HDInsight cluster. Spark 2.0 introduces the spark structured streaming API that provides a simple and more consistent programming model.

Conclusion:

Spark technology is compatible with lot-hub that used for real-time message ingestion, Hive, and the rest of technology choosing in this architecture. Azure documentation in (Spark Streaming in Azure HDInsight, 2019) showing in here that real-time message ingestion adopting technology was lot hub are compatible with spark streaming technology and spark streaming compatible with our choosing data lake. Another reason for choosing spark streaming due to the integration capabilities with HDFS and the scalability bounded by the cluster size.

- Analytical Datastore

The role of this component is defined in (Choosing an analytical data store, 2020) to serve processed data in a structured format which queried using analytical tools. It also allows querying of both hot-path and cold-path data that collectively referred to as the serving layer or data serving storage.

The technologies below all used with the connection of Azure cloud technology,

Azure Synapse Analytics:

An Azure managed service that is based on the SQL server database and optimized to support large scale data warehousing workload, which in our case would data lake.

Spark SQL:

An API that built on SPARK that provisions the making of data frames and tables queried using SQL syntax.

HBASE:

HBase is a low latency NO SQL store offering high performance and as well flexible for querying structured and semi-structured data.

HIVE:

This technology selected above in question 1.1 used for batch processing, but Hive also offers a database architecture that theoretically likes a relational database management system and considered as a data warehouse. This technology used as a source for analytical queries in some scenarios.

Conclusion

Technologies provided by Azure such as synapse analytics, spark SQL, HIVE and HBASE which comes in the HDInsight cluster, supports all which makes it easier to integrate with the Hadoop ecosystem. It also makes the architecture integration easier by choosing compatible technology and less cost-effective. Our choice for this component would be Hive as is known for its efficient query processing by making the use of SQL-like and is used for the data stored inside HDFS. Hive mentioned in (Apache Hive vs. Apache Spark SQL - 13 Amazing Differences, 2019) that can be used with big data technology and tools such as Pig, HDFS, Sqoop and Oozie but also compatible with spark based another component such as Spark streaming. Our choice showed that Hive more compatible technology and most importantly, it works with our technology choice for stream processing which was Spark streaming.

-Analytics and reporting

The goal of having this solution build for the bank ABC to enable them to get an insight into the data through analysis and reporting. The bank will be able to analyze data when the Bigdata architecture has a data modeling layer, such tabular data model, in Azure Analysis services, or a multidimensional OLAP cube. This component defined in (Choosing data analytics and reporting technology, 2020) with technologies that support this component such as Microsoft BI software that can be used for modeling and data visualization or can use Microsoft Excel software for analysis.

- **Azure Analysis Services**

Most of Big data solutions include a centralized data model layer known as cube into their architecture. This data model will have reports, dashboards, and interactive “slice and dice” analysis. Finally, this component supports the creation of tabular models.

- **POWER BI**

This software allows data analysts to create an interactive data visualization based on data models in an OLAP or directly from the analytical data store.

- **MICROSOFT EXCEL**

This software used the bank ABC identified during the interview with a manager that they are using Microsoft office package in all their systems. Excel software known in the world for its capabilities in data analysis and visualization. The bank analyst can use this application to build data models from an analytical data store or retrieve data from OLAP data models into interactive pivot tables and charts.

Conclusion

The choosing technology is quite evident in this section is sticking with Office 365 package that comes with POWER BI and Microsoft Excel. Both suggested software has perfect functionalities for making dashboards and another analytics report. Employees at ABC bank should decide which one to use depends on their comfortability.

Orchestration:

Most big data solutions run similar operations of data processing, which encapsulated in workflows such as transforming source data, moving data between multiple sources or loading data into analytical data store or pushing the result straight into a report or dashboards. All those workflows solved using orchestration technology such as Azure data factory, apache Oozie and Sqoop.

Azure technology for orchestration defined in (Big data architecture style - Azure Application Architecture Guide, 2020).

Azure data factory:

This type of pipeline used to define a sequence of activities scheduled for recurring temporal windows. The events initiate data copy operations in technology like Hive, Pig, map-reduce, or spark if that HDInsight cluster exists, such as U-SQL jobs in Azure lake analytics and stored procedures in Azure synapse or Azure SQL database.

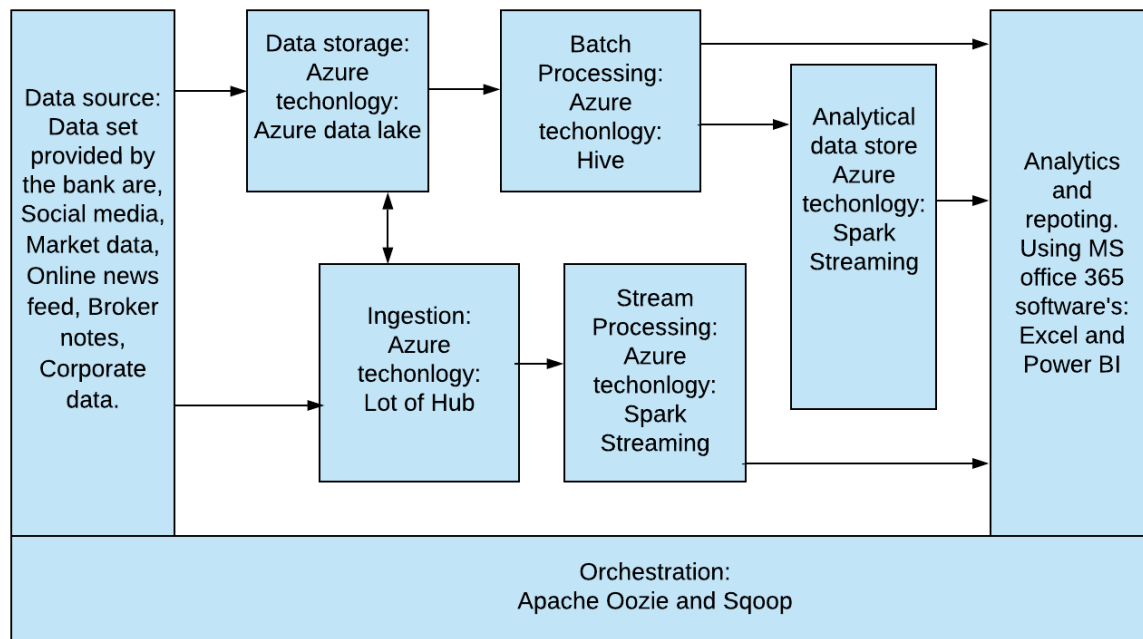
Apache Oozie and Sqoop:

Both roles designed as a job automation engine designed for the Apache Hadoop ecosystem to initiate copy operations in Hive, Pig, and MapReduce to process data. On the other hand, Sqoop used to copy data between HDFS and SQL databases.

Conclusion

An obvious choice for this component because of the requirement for data factory that to have some technologies to demand such as Azure synapse or Azure SQL database, which both are not choosing for this solution. Apache Oozie and Sqoop seem to be the compatible solution for this component due to the reason it works with Hive and allows us to copy data between HDFS and SQL databases.

The diagram below our big data architecture solution for the ABC investment Bank.



1.2 The Business Head has a limited understanding of IT and is aware of the data warehouse and has suggested you build this only. Explain why you will be making a Data Lake for this solution. Discuss the approach required for implementing Data Lakes.

Datawarehouse is a store for structured and flirtd data that has already processed for a specific purpose. On the other hand, data, the lake is a vast pool for raw data in which data purpose has not yet defined. The approach with the data lake is to keep the data on its natural form, and it's only transforming the data we need when we ready to use it. The approach data lake considers it is schema on reading and data warehouse is schema on write. The organization still needs to use both approaches using Datawarehouse for analytical purposes and data lake for storing the raw data that data warehouses cannot handle the volume, variety and complexity of today's data. Data warehouse store can be scaled up, but it will be expensive and slow as it's vertical scaling and not a horizontal scaling.

In this website (Arcadia Data, 2019) mentioned about organizations using data lake as a holding area of information which not set for immediate use, but it may come handy later. The flexibility data lake provides, and low storage cost makes the organization keep a much larger volume that can happen to us then. The blog in (Arcadia Data, 2019) mentioned as

well that analysts start using data lake to make test assumptions on massive volumes of data, then extract the useful data and load it into a warehouse for decision making. The blog in (Arcadia Data, 2019) mentioned as well that data warehouses will always be the best option for fast, interactive queries on recent data.

The solution above in 1.1 shows that I have choosing to include both into Big data architecture such as Data lake and data warehouse, which is Hive. The data lake is used at the first stage to store all the size data with no size limits and with all formats. Enabling to store all the company data has been in the data lake without being removed or anything like the operation with data warehouse like cleaning and structuring the data before organizing the data tables then running analytical queries. The data warehouse will mainly use when the bank is trying to get a specific analysis or an answer from the data. The operation would be in here is collecting data from data lake then clean and organize it to be stored in the table inside the Data warehouse for analytical services.

Discuss the approach required for implementing Data Lakes

The information below extracted from (What is a Data Lake and How to Create One for Your Business, 2020).

1. Set up a data lake solution that our case is using Azure cloud data lake service.
2. Identity data sources that will be stored on the data lake and make a decision whether to clean or store it like how it. Ensure to identify the metadata for the individual types of data set.
3. Ensure to establish processes and automation for operations that require a higher frequency of data publishing or time-consuming. The automation could involve automating the extraction, transformation, and publishing of the data to the data lake.
4. Ensure the data lake is functioning correctly by not only focusing on putting the data lake but also allow or to facilitate the data retrieval for another system to generate data-driven informed business operations. Else, the data lake will end up as a data swamp in the long run with little to no use.
5. After you set up the data lake and it's been functioning sometime, then it's time to collect data for the data lake with the right amount of metadata. Adding the data will require to implement different processes with "extract transform and load" operations before using them to drive various business decision needed for ABC bank. If the ABC investment bank needed to get an answer from a specific data, then would be required to use the data warehouse technology, which 1.1 we suggested using Hive technology if more processing needs to create in correlation with different data set or use Microsoft Power BI for data visualization directly.

1.3 The business wants near-real-time performance when certain financial products are being discussed on social media for it to act promptly. It has been suggested the parallel distributed processing on a cluster should use Map Reduce to process this requirement. Provide a detailed assessment of whether MapReduce is optimal to meet this requirement, and If not, what would be the best approach.

Back to 1.1 questions and precisely the stream processing component, which Spark streaming technology selected, which is provided by Azure. MapReduce from research found that it is only used for batch processing and like hive and pig. However, Spark found used for batch processing and Real-time data processing which fits in with our question here. The comparison below created to show why apache spark is more suitable for this solution as it works for batch processing and Real-time data processing but also below other similarities that make it better in terms of functionalities as well.

Comparison of MapReduce and Apache spark derived from (MapReduce vs. Apache Spark- 20 Useful Comparisons To Learn, 2020).

- Both MapReduce and apache spark is scalable and limited to 1000 nodes in a single cluster
- MapReduce used only for Batch processing, where apache-spark used for batch processing and Real-Time data processing.
- MapReduce is lower than Apache Spark because of I/O disk latency. However, Apache is 100x faster in memory and 10x quicker while running on disk.
- Apache spark or spark streaming can integrate with all data sources and file formats supported by the HDInsight cluster provided by Azure. On the other hand, MapReduce is majorly compatible with all the data sources and file formats.
- MapReduce framework is more secured compared to Apache spark but is more evolving and getting matured.
- MapReduce is dependent on an external scheduler where apache-spark has its scheduler.
- MapReduce used replication for fault tolerance, where Spark uses RDD and other data storage models for fault tolerance.
- In terms of development, MapReduce is more complicated compared to Spark because the use of Java APIs and Spark is easier to use because of Rich APIs.
- Spark comes with a duplicate elimination feature that processes every record exactly once by eliminating duplication.
- MapReduce has a very high latency where the apache spark is much faster comparing to MapReduce.
- MapReduce uses persistent storage where spark uses Resilient Distributed Datasets.
- Spark can execute batch processing jobs between 10 to 100 times faster than MapReduce, but both tools used for processing big data.
- Spark more cost-effective than MapReduce because of high availability memory.

1.4 Devise a detailed hosting strategy for this Big Data project, and how this will meet the scalability, high availability requirements for this global business.

The benefit of using an on-premises solution

- Bank with use of this solution is in control of their data and entirely in control of how it's used and after. With the data protection act that forces the company to secure their data and if not secure, then the company will be faced with excellent and bad reputation. Security hesitates some businesses to leap into the cloud.

The downside of using an on-premises solution

- Building on-premises solution would require the ABC bank to hire employees with the same skills to manage both hardware and software infrastructure such as spinning servers, orchestrating batch ETL jobs and dealing with outages, downtime and in software side where requires data engineers to integrate a wide range of tool that used to ingest, organize, pre-process and query stored in the lake including software licensing. All those mentioned are considered as ongoing management and operating cost when operating data on-premises solutions.
- Data size is predictable those days, and the size is terminally increasing day by day. When the volume of the data lake is full up, they will need to manually add and configure servers which will add additional maintenance cost.
- The bank with an on-premises solution will be responsible for the ongoing costs of the server hardware, space and power consumption.

Benefits of using a Cloud solution

- Cloud service provided as self-service and on-demand, meaning you would not need to do any maintenance or worry about setting up servers or downtime.
- Most of the computing resources provided by the cloud with few mice click, meaning the ABC bank would have a downtime system or lose relevant data for example, as cloud data lake has no limit size such as Azure data lake.
- Cloud solution removes the need for buying hardware, software and setting up and running on-site data centers.
- Cloud only costs you for what you use rather than set up cost, engineering cost, and so on with the on-premises solution.
- Cloud providers not only readily available infrastructure but also enable you to scale the infrastructure with few clicks to will allow you to manage large spikes in traffic usage.
- The bank with a cloud solution will be only responsible for the cost of the resources they will need to use without the maintenance or the price to adjusts.

The downside of using a cloud solution

- Data ownership is the biggest issue that Businesses ask about when deciding to move into the cloud. Data and encryption usually reside with a third-party provider such as Azure and AWS etc. It is showing that when there is downtime, then you might not be able to access your data stored in the cloud.
- Most of the Big Data solution requires storing data in a centralized data lake. Ensure access to this data can be challenging especially when the data must be ingested and consumed by multiple applications and platforms. However, Big Data in the cloud is still extremely safe. The bank even can use some industry-standard methods with another security measure that ensures you would not lose your valuable data.

Scalability, availability, and why choosing a cloud is the right way for this solution?

- Whenever the data becomes available, then scalability becomes more critical. Having scalability as focused on this solution will allow ABC bank to increase and decrease its data-gathering capabilities. Increasing data size on-premises is harder to scale and it can cost a lot for buying, maintaining. Clouds such as AWS or Azure have no limits in terms of the data lake size and our solution for Azure data lake has no size limit in terms of the volume with a great deal of error tolerance and can always be scaled up.
- One of cloud computing characteristics is rapid elasticity by always ensuring that the provided resources from storage media, processing units, applications, and networks are available. Also, it can be increased or decreased in an almost instantaneous fashion allowing for high scalability to ensure optimal use of the resources.
- Cloud computing resources can be accessible over the network, mobile, and smart devices.
- The cloud system can measure the processes and consumption of resources as well as surveillance, control, and reporting in a completely transparent manner.

Availability

Azure documentation stated in (Azure for Amazon Web Services (AWS) Professionals - Azure Architecture Centre, 2020) that downtime or system failure would not ensure availability, which can cost the bank money for downtime. The documentation also has stated Some hardware failures such as failed disk may affect a single host machine, or a failed network can change a whole server rack. Fewer common problems can happen which can disrupt an entire datacentre such as losing power in a data centre. Failure within the cloud system found to be rarely to affect an entire region. One way of making the cloud solution is resilient through redundancy. Redundancy should be considered during design at the first stage with the level of redundancy needed for this requirement.

AWS cloud is a region that is divided into two or more availability zones. The availability zone corresponds with a physically isolated datacentre in a geographic area. Azure also provides Nemours features for providing application redundancy at every level of potential failures such as availability sets, availability zones and a paired region. Those features will minimize failure and downtime which ensures a high availability solution.

1.5 Using Cloud offering from either Amazon AWS or Microsoft Azure, how would either support this Big Data project.

- Big Data relies on computing power due to the vast amounts of data needed for analysis. Azure's compute for this situation comes from its virtual machines where AWS provides EC2 instances for computing along with ancillary services such as Elastic Beanstalk and EC2 container services.
- Both Azure provides scalable storage features such as S3, Azure Blob storage or data lake to handle unstructured and is at par with each other.
- Pricing is another factor that can make strict our choice but for this solution choosing a capable analytic should be the critical factor. Both clouds offer competitive pricing such as AWS charges an upfront cost depending on the use or provide a committed instance for up to three years. However, Azure has a pay as you go model with Microsoft charging its customers by the minute. After a shorter commitment by the bank, then Microsoft may allow a mix of pre-paid and monthly charges.
- Both clouds provide the most significant security features to safeguard hacking instances and sensitive data. The survey suggests that 90% of fortune 500 companies entrusting Azure for their Big data and analytics security aspect which shows the service to be trustworthy.
- Organizations in this day of age need to access their business-critical data anytime and anywhere. Both cloud services provide Quick Sight and Power BI that helps with creating excellent reporting and business intelligence.
- The biggest challenge for processing Big data is latency and cost. Azure for this challenge provides 'Event Hubs,' and AWS provides 'kinesis' which both display enough firepower for data analysis inexpensively and with low latency.

Conclusion

AWS and Azure are showing to be the best cloud system for Big data solutions. In terms of cloud services, AWS still the market leader by 41% of market capitalization. AWS becomes like that due to the variety of features for building big data infostructure that promises scalability and price cuts for upstarts. The azure solution seems always chosen wherein the organization is invested in Microsoft using Windows operating system or Microsoft Office that can make it easier to integrate with the Azure cloud. Please refer to questions 1.1 significant data architecture where I choose Azure solutions and its components for building the big data architecture solution.

2 Part 2

2.1 Using a tool from the Hadoop ecosystem, develop a solution to demonstrate *Schema on reading*. How does this differ with *Schema on write* (Include your file(s) and code in your coursework report?)

I have downloaded the virtual machine, but I had an issue with getting the file from the local computer into the ORACLE virtual box. I tried what you informed us in your presentation about making the shared folder drag and drop “BIDIRECTIONAL” but still did not work with me. Cloudera Hadoop comes with interface Ambari that makes it easier to do all the operations, but I thought for this requirement you would like us to show you command.

Schema on write

Schema on write described in (Henson, 2020) blog designed while creating a schema for data before writing into the database. This type of schema used in development with the RDBMS database that used structured query language. Using RDBMS can be time-consuming of doing Extract transform load work. Developers or users of the database need to understand that data doesn’t start structured as given but most of the time, given unstructured and required cleansing, then defined as how you want the schema to be the write into the database. Usually, this means developers structure the data before putting in the data into the database.

Example of schema on write:

000987834, Temor AL-Kaisi, 02/03/1992, Beckenham, etc.

Year 1

Communication system 76%, Maths 80% etc.

Year 2

Programming with Java 90%, Forensic 70% etc.

Tasks need to upload the data into an RDBMS database.

- Personal details table
- Year 1 table
- Year 2 table
- Both Year 1, 2 meals will have user_id as a foreign key from Personal to help link information about a student.
- Lastly, create the tables wit the constraint primary and foreign keys.
- Run an SQL query to check for Temor year 1 grade such as,
“SELECT * FROM Year 1 Where user_id = 000987834;
- The difficult with those types of data is having to clean the data and ensure it fits with RDBMS infostructure.

Schema on Reading

Increasing the data volumes has brought another way of creating a schema that enables us to upload data as it was without any changes and transformation. The new schema approach removes the ETL process that requires an understanding of the original data schemas and the structure. ETL process used on the schema to write and consume a lot of time. Schema on reading described (Blog | iamluminousmen, 2020) as a type of schema where data would not have to follow any internal schema and its just copying and moving files. The schema-on-read used with Big Data, unstructured data or any frequent schema changes.

On the other hand, with this type of schema by not following a strict ETL and data cleansing processes, then it can have a lot of missing data or invalid data or duplicate and many other problems that can give us an incomplete query result. The point that even with schema-on-read later on after the upload that you will need to understand some level of schema design structure to be able to get insight into them.

Schema on write vs. schema on Reading

- Using the SQL query, it has faster read than with schema on reading due to map-reduce functionality.
- Loading in the schema on write is slower than schema on write due to the process of cleaning and structuring before uploading the database.
- Schema on write can only be structured where schema on reading can be structured and unstructured.
- Schema on reading is SQL query, where schema on reading is NoSQL.

Conclusion

There is no better schema than others in terms of reading and on write because it all depends on the use case. Example when you need data to support a dashboard then the result needs to be processed fast and repetitive which in this case best to use schema on write. If there are unknowns' types of data and constant new sources, then this case schema on reading will work for this use case.

Figure 1 from Lecture 9 Big Data architecture helped me to structure my example to provide schema on reading and an example giving in Hive lab as well that shows the steps needed to create a schema on reading instance.

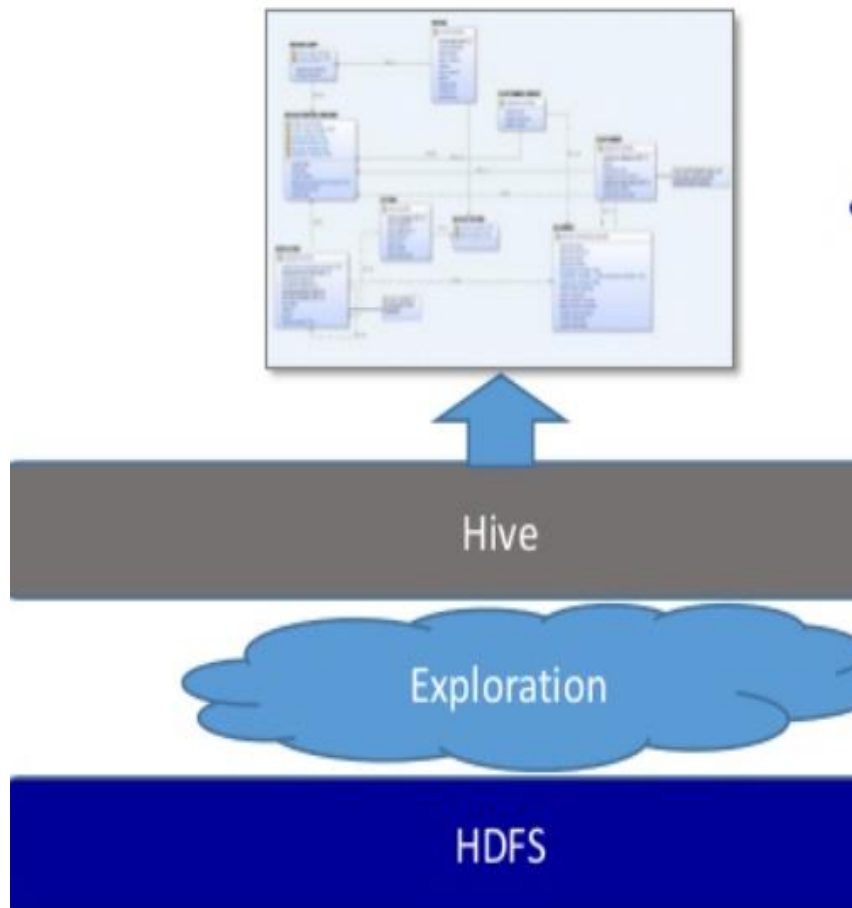


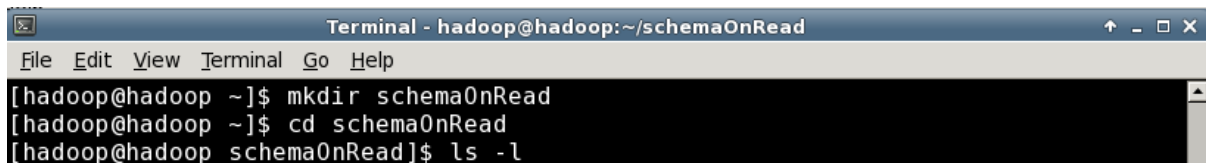
Figure 1: Lecture 9 Big Data architecture

Schema on reading demonstration example:

1. Use ORACLE virtual machine provided by the university
2. Start Hadoop requires to run both command "start-of.sh"," start-yarn.sh."
3. At the start was searching where my dataset "student-record.csv" located.
4. Data set found then its time to create a new directory folder and copy the movies folder inside it.
5. The newly created directory folder is schema On reading.
6. Copy "student-record.csv" from CW inside dataset directory stored on the Desktop into /home/Hadoop/schema on reading.

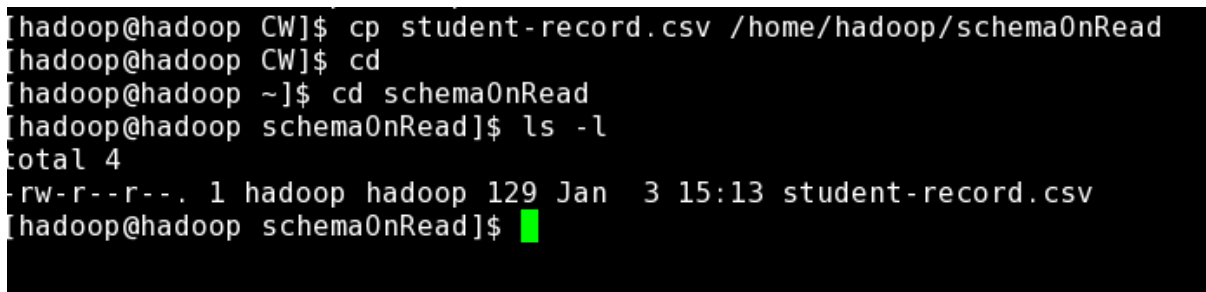
```
[hadoop@hadoop ~]$ ls -l
total 2628
drwxr-xr-x. 2 hadoop hadoop 4096 Jan 3 12:34 coursework
-rw-r--r--. 1 hadoop hadoop 2628107 Jan 3 12:29 coursework
drwxrwxr-x. 3 hadoop hadoop 4096 Oct 10 19:05 data
drwxrwxr-x. 4 hadoop hadoop 4096 Oct 10 19:17 Data
-rw-rw-r--. 1 hadoop hadoop 645 Jan 3 14:00 derby.log
drwxrwxr-x. 4 hadoop hadoop 4096 Jul 23 2018 Desktop
drwxrwxr-x. 3 hadoop hadoop 4096 Sep 5 2017 Download
drwxr-xr-x. 13 hadoop hadoop 4096 Sep 16 2015 hadoop
drwxrwxr-x. 3 hadoop hadoop 4096 Jul 16 2014 hadoopdata
drwxrwxr-x. 2 hadoop hadoop 4096 Jan 3 14:27 HIVERESULTS
drwxrwxr-x. 5 hadoop hadoop 4096 Jan 3 14:00 metastore_db
-rw-rw-r--. 1 hadoop hadoop 6063 Jul 17 2014 pig_1405589896738.log
drwxrwxr-x. 2 hadoop hadoop 4096 Oct 10 19:33 Results
drwxrwxr-x. 2 hadoop hadoop 4096 Jan 3 15:11 schemaOnRead
drwxrwxr-x. 6 hadoop hadoop 4096 Nov 7 17:08 workspace
[hadoop@hadoop ~]$ cd Desktop
[hadoop@hadoop Desktop]$ ls -l
total 16
drwxr-xr-x. 6 hadoop hadoop 4096 Jan 3 12:18 Datasets
-rw-rw-r--. 1 hadoop hadoop 709 Sep 5 2017 derby.log
-rw-r--r--. 1 hadoop hadoop 171 Oct 25 2011 eclipse.desktop
drwxrwxr-x. 5 hadoop hadoop 4096 Sep 5 2017 metastore_db
[hadoop@hadoop Desktop]$ cd Datasets
[hadoop@hadoop Datasets]$ ls -l
total 16
drwxr-xr-x. 2 hadoop hadoop 4096 Jan 3 15:11 CW
drwxr-xr-x. 2 hadoop hadoop 4096 Jul 23 2018 Lab5
drwxr-xr-x. 2 hadoop hadoop 4096 Jul 23 2018 Lab6
drwxr-xr-x. 2 hadoop hadoop 4096 Jan 3 14:33 Lab7
[hadoop@hadoop Datasets]$ cd CW
[hadoop@hadoop CW]$ ls -l
total 4
-rw-r--r--. 1 hadoop hadoop 129 Jan 3 15:09 student-record.csv
```

Figure 2: Create new directory folder to copy movies. tab

A terminal window titled "Terminal - hadoop@hadoop:~/schemaOnRead". It shows the following commands and output:

```
[hadoop@hadoop ~]$ mkdir schemaOnRead
[hadoop@hadoop ~]$ cd schemaOnRead
[hadoop@hadoop schemaOnRead]$ ls -l
```

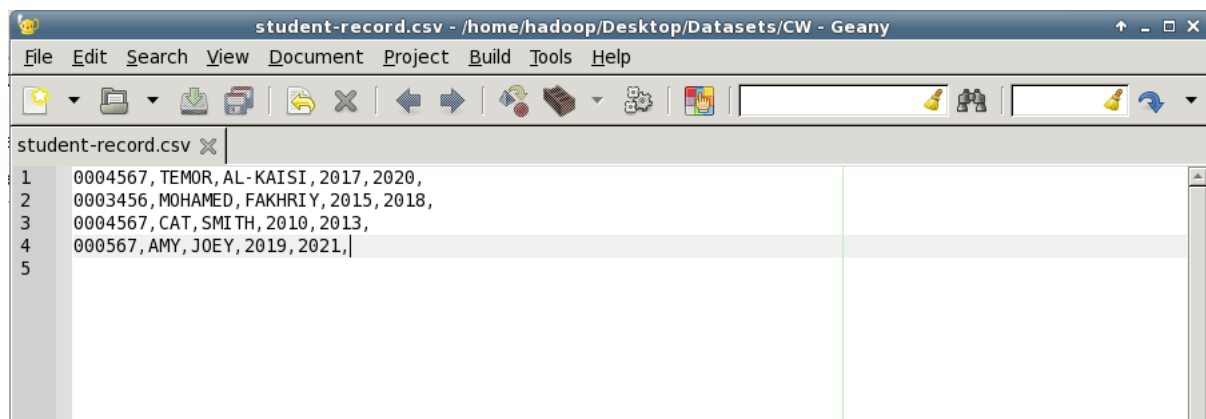
Figure 3: Creating new directory folder

A terminal window showing the process of copying a file and verifying its presence:

```
[hadoop@hadoop CW]$ cp student-record.csv /home/hadoop/schemaOnRead
[hadoop@hadoop CW]$ cd
[hadoop@hadoop ~]$ cd schemaOnRead
[hadoop@hadoop schemaOnRead]$ ls -l
total 4
-rw-r--r--. 1 hadoop hadoop 129 Jan  3 15:13 student-record.csv
[hadoop@hadoop schemaOnRead]$
```

Figure 4: copying the file into the new folder and check if it exist on the new folder

After the dataset student-record.csv stored in the directory folder, then it's time to explore the data found in the file. I will explore the data using Geaney software provided by ORACLE Linux virtual machine.

A screenshot of the Geany text editor window titled "student-record.csv - /home/hadoop/Desktop/Datasets/CW - Geany". The editor displays the contents of the CSV file:

```
1 0004567, TEMOR, AL- KAISI, 2017, 2020,
2 0003456, MOHAMED, FAKHRIY, 2015, 2018,
3 0004567, CAT, SMITH, 2010, 2013,
4 000567, AMY, JOEY, 2019, 2021,
5
```

Figure 5: Data exploration

The file was showing in figure 4 example showing it contains “user-id, first_name, last_name, start_year and end_year.

If the company wants analytical answers to some of their questions, they can use Hive that suggested part of the Architecture as a Data warehouse. Firstly, you will need to type “hive” as the command to log in to Hive. Afterward, in the data exploration stage, we identified the columns then its time to create a table in Hive.


```
hive> CREATE table students(Id INT, first_name STRING ,last_name STRING,start_year INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;
OK
Time taken: 0.982 seconds
```

Figure 6: Creating table in Hive

The table created in Hive data warehouse then its time to load our inside HDFS into our Hive using the command below provided in the screenshot figure 6.

```
hive> LOAD DATA INPATH 'student-record.csv' OVERWRITE INTO TABLE students;
Loading data to table default.students
rmr: DEPRECATED: Please use 'rm -r' instead.
Deleted hdfs://localhost:9000/user/hive/warehouse/students
Table default.students stats: [numFiles=1, numRows=0, totalSize=129, rawDataSize=0]
OK
Time taken: 1.186 seconds
```

Figure 7: Loading student-record.csv into students table in Hive

```
Time taken: 17.988 seconds
hive> select * from students where start_year=2017;
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1578053603805_0014, Tracking URL = http://localhost:8088/proxy/application_1578053603805_0014/
Kill Command = /home/hadoop/hadoop/bin/hadoop job -kill job_1578053603805_0014
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2020-01-03 15:28:37,840 Stage-1 map = 0%, reduce = 0%
2020-01-03 15:28:45,345 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.31 sec
MapReduce Total cumulative CPU time: 1 seconds 310 msec
Ended Job = job_1578053603805_0014
MapReduce Jobs Launched:
Job 0: Map: 1 Cumulative CPU: 1.31 sec HDFS Read: 352 HDFS Write: 30 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 310 msec
OK
4567    TEMOR    AL-KAISI    2017    2020
Time taken: 16.112 seconds, Fetched: 1 row(s)
hive>
```

Figure 8: Running SQL like query to extract information about students started in year 2017

2/ Compare and contrast Hadoop and Relational Database system

The comparison below mostly derived from (Data Flair, 2019).

- Hadoop and relational database systems have similar functions, such as collecting, storing, processing, retrieving, extracting, and manipulating data. However, both differ in terms of processing data as RDBMS focuses on structured data, whereas Hadoop specializes in semi-structured and unstructured data.
- RDBMS is used mostly for OLTP processing and Hadoop for analytical and big data processing.
- The relational database systems used widely by most companies and proven to be consistent, matured, but this system only works better when the data definitions are data types, relationships among the data, and constraints. Approving again, RDBMS is suitable for real-time OLTP processing. In terms of the Hadoop, the system developed recently and it's in demand for the big data and unstructured data that is in a different format.
- The database cluster with the RDBMS system uses the data files stored in the shared storage. However, In Hadoop, the storage data can be stored independently in each processing node.
- RDBMS tuning performance can go down even in a proven environment, whereas in Hadoop system enables hot tuning by creating extra nodes that will be self-managed.
- The RDBMS systems need downtime to have storage maintenance. Standalone databases needed to add processing powers such as CPU or physical memory in a non-virtualized environment, and all of those need a downtime of RDBMS such as ORACLE, DB2 and SQL server whereas Hadoop system is individual independent nodes added to as need basis.

References

A Gentle Introduction to Stream Processing, (2020) *Medium*, [online] Available at <https://medium.com/stream-processing/what-is-stream-processing-1eadfca11b97> (Accessed 2 January 2020).

Apache Hive vs Apache Spark SQL - 13 Amazing Differences, (2019) *EDUCBA*, [online] Available at: <https://www.educba.com/apache-hive-vs-apache-spark-sql/> (Accessed 30 December 2019).

Arcadia Data. (2019). *Why Organizations Need Data Warehouses and Data Lakes*. [online] Available at: <https://www.arcadiadata.com/blog/data-lakes-and-data-warehouses-why-you-need-both/> [Accessed 31 Dec. 2019].

Azure for Amazon Web Services (AWS) Professionals - Azure Architecture Center, (2020) *Docs.microsoft.com*, [online] Available at: <https://docs.microsoft.com/en-us/azure/architecture/aws-professional/> (Accessed 3 January 2020).

Azure vs AWS: Which platform to choose for Big Data & Analytics solutions?, (2020) *Saviantconsulting.com*, [online] Available at: <https://www.saviantconsulting.com/blog/azure-vs-aws-platform-big-data-analytics.aspx> (Accessed 3 January 2020).

Big data architecture style - Azure Application Architecture Guide, (2020) *Docs.microsoft.com*, [online] Available at: <https://docs.microsoft.com/en-us/azure/architecture/guide/architecture-styles/big-data> (Accessed 3 January 2020).

Blog | iamluminousmen. (2020). *Schema-on-Read vs Schema-on-Write*. [online] Available at: <https://luminousmen.com/post/schema-on-read-vs-schema-on-write> [Accessed 3 Jan. 2020].

Choosing a data analytics and reporting technology, (2020) *Docs.microsoft.com*, [online] Available at: <https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/analysis-visualizations-reporting> (Accessed 3 January 2020).

Choosing a real-time message ingestion technology, (2020) *Docs.microsoft.com*, [online] Available at: <https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/real-time-ingestion> (Accessed 3 January 2020).

Choosing a stream processing technology, (2020) *Docs.microsoft.com*, [online] Available at: <https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/stream-processing> (Accessed 3 January 2020).

Choosing an analytical data store, (2019) *Docs.microsoft.com*, [online] Available at: <https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/analytical-data-stores> (Accessed 29 December 2019).

Choosing an analytical data store, (2020) *Docs.microsoft.com*, [online] Available at: <https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/analytical-data-stores> (Accessed 2 January 2020).

Data Lake | Microsoft Azure, (2019) *Azure.microsoft.com*, [online] Available at: <https://azure.microsoft.com/en-gb/solutions/data-lake/> (Accessed 26 December 2019).

Data Lakes Storage | AWS, (2019) *Amazon Web Services, Inc.*, [online] Available at: <https://aws.amazon.com/products/storage/data-lake-storage/> (Accessed 26 December 2019).

DataFlair. (2019). *Difference between RDBMS with Hadoop MapReduce - Data Flair*. [online] Available at: <https://data-flair.training/forums/topic/difference-between-rdbms-with-hadoop-mapreduce/> [Accessed 18 Dec. 2019].

Guru99.com. (2019). *Data Lake vs Data Warehouse: What's the Difference?*. [online] Available at: <https://www.guru99.com/data-lake-vs-data-warehouse.html> [Accessed 30 Dec. 2019].

Henson, T. (2020). *Schema On Read vs. Schema On Write Explained - Thomas Henson*. [online] Thomas Henson. Available at: <https://www.thomashenson.com/schema-read-vs-schema-write-explained/> [Accessed 3 Jan. 2020].

HOFFMAN, C. (2019) What Is HTTPS, and Why Should I Care?, *How-To Geek*, [online] Available at: <https://www.howtogeek.com/181767/htg-explains-what-is-https-and-why-should-i-care/> (Accessed 28 December 2019).

Medium. (2019). *Big Data Battle : Batch Processing vs Stream Processing*. [online] Available at: <https://medium.com/@gowthamy/big-data-battle-batch-processing-vs-stream-processing-5d94600d8103> [Accessed 27 Dec. 2019].

Pol, U.R., 2016. Big data analysis: comparison of hadoop mapreduce, pig and hive. *Int. J. Innov. Res. Sci. Eng. Technol*, 5(6).

Spark Streaming in Azure HDInsight, (2019) *Docs.microsoft.com*, [online] Available at: <https://docs.microsoft.com/en-us/azure/hdinsight/spark/apache-spark-streaming-overview> (Accessed 30 December 2019).

Team, D. (2019). *Hadoop MapReduce Tutorial - A Complete Guide to MapReduce - Data Flair*. [online] DataFlair. Available at: <https://data-flair.training/blogs/hadoop-mapreduce-tutorial/> [Accessed 18 Dec. 2019].

Top Reasons Big Data in the Cloud Is Raining on On-Premise - InformationWeek, (2020) *InformationWeek*, [online] Available at: <https://www.informationweek.com/big-data/top-reasons-big-data-in-the-cloud-is-raining-on-on-premise/a/d-id/1334722> (Accessed 3 January 2020).

What is a Data Lake and How to Create One for Your Business, (2020) *Hackernoon.com*, [online] Available at: <https://hackernoon.com/what-is-a-data-lake-im-confused-8e12d554f7a0> (Accessed 1 January 2020).

WhatIs.com. (2019). *What is Advanced Message Queuing Protocol (AMQP) ? - Definition from WhatIs.com*. [online] Available at: <https://whatIs.techtarget.com/definition/Advanced-Message-Queuing-Protocol-AMQP> [Accessed 28 Dec. 2019].

Bibliography

Apache Spark Streaming vs. Azure Stream Analytics Comparison | IT Central Station, (2020) *Itcentralstation.com*, [online] Available at: https://www.itcentralstation.com/products/comparisons/apache-spark-streaming_vs_azure-stream-analytics (Accessed 2 January 2020).

Awadallah (2020). *Schema-on-Read vs Schema-on-Write*. [online] Slideshare.net. Available at: <https://www.slideshare.net/awadallah/schemaonread-vs-schemaonwrite> [Accessed 3 Jan. 2020].

Azure Data Lake Analytics and U-SQL, (2020) InfoQ, [online] Available at: <https://www.infoq.com/articles/azure-data-lake-analytics-usql/> (Accessed 2 January 2020).

Azure Data Lake vs. Amazon Redshift: Data Warehousing for Professionals - DZone Big Data, (2020) *dzone.com*, [online] Available at: <https://dzone.com/articles/azure-data-lake-vs-amazon-redshift-data-warehousin> (Accessed 2 January 2020).

Balkenende, M., Cohen, L. and Pandey, V. (2020) The Big Data Debate: Batch Versus Stream Processing - The New Stack, *The New Stack*, [online] Available at: <https://thenewstack.io/the-big-data-debate-batch-processing-vs-streaming-processing/> (Accessed 2 January 2020).

Best available Cloud-based Big Data solution, (2020) *Newgenapps.com*, [online] Available at: <https://www.newgenapps.com/blog/best-available-cloud-based-big-data-solution> (Accessed 2 January 2020).

Big data architecture style - Azure Application Architecture Guide, (2020) *Docs.microsoft.com*, [online] Available at: <https://docs.microsoft.com/en-us/azure/architecture/guide/architecture-styles/big-data> (Accessed 3 January 2020).

Campbell, C. (2020) Top Five Differences between Data Lakes and Data Warehouses, *Blue-granite.com*, [online] Available at: <https://www.blue-granite.com/blog/bid/402596/top-five-differences-between-data-lakes-and-data-warehouses> (Accessed 2 January 2020).

Carey, S. (2020) AWS vs Azure vs Google: What's the best cloud platform for enterprise?, *Computerworld*, [online] Available at: <https://www.computerworld.com/article/3429365/aws-vs-azure-vs-google-whats-the-best-cloud-platform-for-enterprise.html> (Accessed 2 January 2020).

Charting the data lake: Using the data models with schema-on-read and, (2020) *IBM Big Data & Analytics Hub*, [online] Available at: <https://www.ibmbigdatahub.com/blog/charting-data-lake-using-data-models-schema-read-and-schema-write> (Accessed 2 January 2020).

Data Lake vs Data Warehouse: Key Differences - KDnuggets, (2020) *KDnuggets*, [online] Available at: <https://www.kdnuggets.com/2015/09/data-lake-vs-data-warehouse-key-differences.html> (Accessed 2 January 2020).<https://www.cloudmoyo.com/blog/data-architecture/difference-between-a-data-warehouse-and-a-data-lake/>

Data Lake vs Data Warehouse: Key Differences - Talend, (2020) *Talend Real-Time Open Source Data Integration Software*, [online] Available at: <https://www.talend.com/resources/data-lake-vs-data-warehouse/> (Accessed 2 January 2020).

Filanovskiy, A. (2020). *Big Data SQL Quick Start. Schema on Read and Schema on Write - Part11..* [online] Blogs.oracle.com. Available at: <https://blogs.oracle.com/datawarehousing/big-data-sql-quick-start-schema-on-read-and-schema-on-write-part11> [Accessed 3 Jan. 2020].

Filanovskiy, A. (2020). *Big Data SQL Quick Start. Schema on Read and Schema on Write - Part11..* [online] Blogs.oracle.com. Available at: <https://blogs.oracle.com/datawarehousing/big-data-sql-quick-start-schema-on-read-and-schema-on-write-part11> [Accessed 3 Jan. 2020].

HBase vs. Hive vs. Spark SQL Comparison, (2020) *Db-engines.com*, [online] Available at: <https://db-engines.com/en/system/HBase%3BHive%3BSpark+SQL> (Accessed 2 January 2020).

IBM Big Data & Analytics Hub. (2020). *Why is Schema on Read So Useful?*. [online] Available at: <https://www.ibmbigdatahub.com/blog/why-schema-read-so-useful> [Accessed 3 Jan. 2020].

ime?, H., guy, L. and Miner, D. (2020). *Hive enforces schema during read time?*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/11764237/hive-enforces-schema-during-read-time> [Accessed 3 Jan. 2020].

LaRock, T. (2019). *Azure vs. AWS Analytics and Big Data Services Comparison - Thomas LaRock*. [online] Thomas LaRock. Available at: <https://thomaslarock.com/2018/03/azure-vs-aws-analytics-and-big-data-services-comparison/> [Accessed 27 Dec. 2019].

Lin, H.K., Harding, J.A. and Chen, C.I., 2016. A hyperconnected manufacturing collaboration system using the semantic web and Hadoop Ecosystem System. *Procedia Cirp*, 52, pp.18-23.

Linkedin.com. (2020). *Schema on Read and ELT (Extract Load and Transform)*. [online] Available at: <https://www.linkedin.com/pulse/schema-read-elt-extract-load-transform-swapnil-kothari/> [Accessed 3 Jan. 2020].

Map-Reduce - an overview | ScienceDirect Topics, (2020) *Sciencedirect.com*, [online] Available at: <https://www.sciencedirect.com/topics/computer-science/map-reduce> (Accessed 2 January 2020).

O'Reilly | Safari. (2020). *Programming Hive*. [online] Available at: <https://www.oreilly.com/library/view/programming-hive/9781449326944/ch04.html> [Accessed 3 Jan. 2020].

Panni, J. (2020) AWS vs Azure vs Google Cloud Platform – Analytics & Big Data, *endjin blog*, [online] Available at: <https://blogs.endjin.com/2016/08/aws-vs-azure-vs-google-cloud-platform-analytics-big-data/> (Accessed 2 January 2020).

Strohbach, M., Daubert, J., Ravkin, H. and Lischka, M., 2016. Big data storage. In *New horizons for a data-driven economy* (pp. 119-141). Springer, Cham.

Systems, D., Banin, F., Banin, F., Banin, F. and Banin, F. (2020) Distributed Computing Principles and SQL-on-Hadoop Systems – SQLServerCentral, *SQLServerCentral*, [online] Available at: <https://www.sqlservercentral.com/articles/distributed-computing-principles-and-sql-on-hadoop-systems> (Accessed 2 January 2020).

What Is Big Data Architecture? - DZone Big Data, (2020) *dzone.com*, [online] Available at: <https://dzone.com/articles/what-is-big-data-architecture> (Accessed 2 January 2020).