Analysis of Recommendation System in Cloud Platform

Pushkar Pramod Wani

Abstract: ***A good recommendation system can recommend a group of movies to users based on their interests or the popularity of the films. The recommendation systems are important because they assist them in making good choices without requiring them to expend their time. It is difficult and expensive for programmers to build an on-premise recommendation system to automate this process as there is a massive increase in data volume which requires high computational capacity. This paper describes the deployment of a movie recommendation engine in a cloud storage environment that uses ALS algorithm and POST API. The advantage of such a deployment is the use of cloud factors in the generation of recommendations, the cloud and the cloud environment promises high availability and thus reduces downtime for recommendation services.***

Keywords: Recommendation system, cloud storage, cloud computing

1. Introduction

Recommendation Systems are pre-programmed engines that make recommendations to users. These systems have become commonplace in everyday life, with Amazon, Facebook, Netflix, and many other online platforms using e-mail or other online services for providing recommendations to its users.

Because there is a large User Database, the term "Big data" is no longer a new concept [1]. However, the challenges have existed for a long time because it was difficult to store such a large amount of data. Big Data is expanding at a break-neck speed. It is said that 90% of the world's data has been created in the last two years [2]. According to IBM, Bad data is a byproduct of Big Data. This is critical because C-level executives rely on BI Analytics to make critical business decisions, assuming that the underlying data is accurate and for accumulation of such data in the current scenario, there is technology available for efficient storage and computation, but it comes at a cost. If an organization decides to perform such tasks on-premise, it will incur significant expenses. As an alternative, many cloud platforms such as Azure, AWS, and GCP services are coming into play [4], In a cloud environment there is no need to pay for hardware or software licensing; rather, only pay for the cloud services that are used. Assume that a user requires 1TB of storage and high-end CPU usage with very little downtime. The user does not need to invest in hardware, software, or other similar items. If additional storage, hardware support, or a high-performance CPU that computes Distributed nodes and performs Multi-parallel processing for increased efficiency, are required. The user simply needs to request that configuration in the cloud, so the client will only see an increase in monthly incremental costs. In a nutshell, it's a pay-as-you-go model [3].

2. Related Literature

Big data and cloud computing can be combined. Cloud computing provides the underlying engine for analyzing and storing large datasets gathered from a variety of sources. Instead of relying on local computer power, big data makes use of dis- tributed storage technology via cloud computing. As a result, cloud computing can be viewed as a service model as well as a facilitator for big data. Google, Microsoft, and Amazon [1] are the leading providers of cloud-based big data platforms.

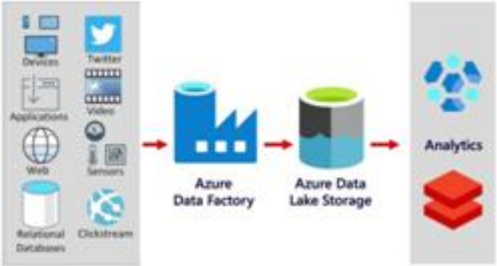


Figure 1: Block diagram of Azure services

We will concentrate on Microsoft Azure in this study. Microsoft Azure was launched in 2008 as a brand name for Microsoft's cloud computing services [3]. The research community has paid attention to the process of implementing a cloud-based big data platform.

2.1 Azure platform introduction

Azure SQL is a cloud database service from Microsoft that is based on SQL Server database technology and runs on the Azure cloud computing platform. Data is hosted and managed in Microsoft data centers, and it stores relational and NO-SQL data. Organizations can scale their database size up or down depending on their business needs. Azure SQL communicates using the same TDS protocol as SQL Server and supports the majority of SQL Server's features. Organizations can synchronize identification data with Azure Active Directory, allowing for single authentication and easier access to cloud-based applications. Authentication can be done in a variety of ways, which helps to prevent unauthorized access while also providing an authentication mechanism other than a password.

Azure Data Lake Storage service: Azure Data Lake enables you to capture data of any size, type, and ingestion speed in one single place for operational and exploratory analytics. It is optimized for storing massive amounts of unstructured data such as Text or Binary files. It is general-purpose and cost-efficient object storage.

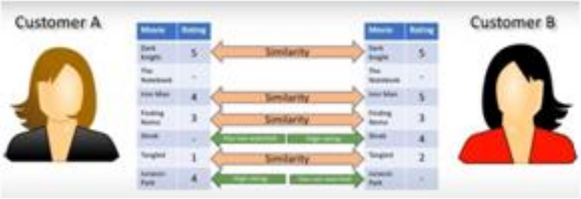
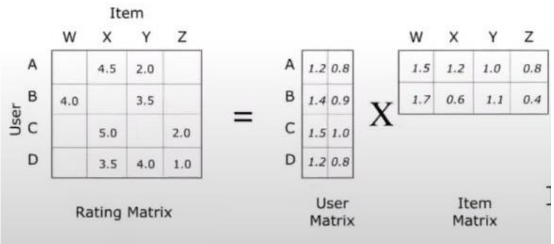


Figure 2: Collaborative filtering



$$\begin{array}{c|cccc} & W & X & Y & Z \\ \hline A & & 4.5 & 2.0 & \\ B & 4.0 & & 3.5 & \\ C & & 5.0 & & 2.0 \\ D & & 3.5 & 4.0 & 1.0 \end{array} = \begin{array}{c|cc} A & 1.2 & 0.8 \\ B & 1.4 & 0.9 \\ C & 1.5 & 1.0 \\ D & 1.2 & 0.8 \end{array} \times \begin{array}{cccc} W & X & Y & Z \\ \hline 1.5 & 1.2 & 1.0 & 0.8 \\ 1.7 & 0.6 & 1.1 & 0.4 \end{array}$$

Figure 3: Matrix factorization

Azure Data factory: Going to power the Ingestion process, place where the pipelines are used to run tasks/activities required for the Data flow. Activities can be used for Data movement, Data transformation (Ex: Copy activity, Performing JOIN, PIVOT operations for data transformation)

Azure DataBricks: This allows us to query and explore data via Python, R, Scala, SQL, etc. by creating distributed clusters i.e.., the data is processed in DataBricks.

Data Warehouse: Once the data is processed, we can analyze the data as DW has Multi parallel processing (MPP). MPP is designed to handle multiple operations simultaneously.

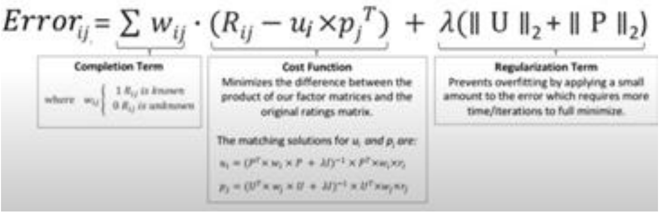
3. System Model

The recommendation engine used in this project is Collaborative Filtering (Figure 2), in collaborative filtering [7] items are published or recommended by grouping similar kinds of users. It works by collecting data from users in the form of ratings, then determining their similarity to recommend a set of movies. It is based on the users' opinions. It identifies users who share similar viewpoints, and then based on the similarity in reviews it recommends a set of movies that the user may prefer to watch [8].

Alternating least squares

The Alternating Least Squares (ALS) algorithm is a matrix factorization algorithm(Figure 3) that runs in a parallel fashion. ALS uses Apache Spark and is designed to solve large-scale collaborative filtering problems. ALS is doing a good job of resolving the Ratings data's sparsity [6].

The ALS algorithm's goal is to estimate the entire rating matrix Rij ui x pj (Figure 4). This problem can be formulated as an optimization problem in which the goal is to minimize an objective function and find the best ui and pj . We want to reduce the least-squares error of the observed ratings in particular.



$$Error_{ij} = \underbrace{\sum w_{ij}}_{} \cdot \underbrace{(R_{ij} - u_i \times p_j^{\,T})}_{} + \underbrace{\lambda(\| U \|_2 + \| P \|_2)}_{}$$

Completion Term

where $w_{ij} \begin{cases} 1 & R_{ij} \text{ is known} \\ 0 & R_{ij} \text{ is unknown} \end{cases}$

Cost Function

Minimizes the difference between the product of our factor matrices and the original ratings matrix.

The matching solutions for u_i and p_j are:

$u_i = (P^T \times w_i \times P + \lambda I)^{-1} \times P^T \times w_i \times r_i$

$p_j = (U^T \times w_j \times U + \lambda I)^{-1} \times U^T \times w_j \times r_j$

Regularization Term

Prevents overfitting by applying a small amount to the error which requires more time/iterations to full minimize.

Figure 4: Cost Function

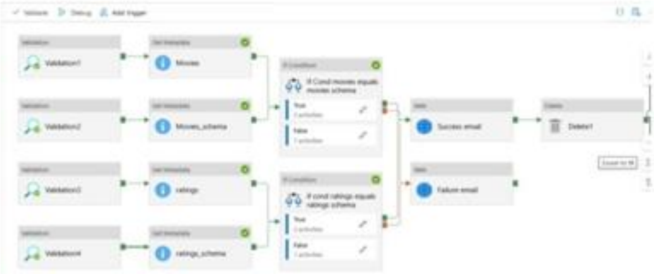


Figure 5: Data Pipeline in Azure Data Factory

Building a Data pipeline in Azure Data Factory

In this section, we'll outline the steps for constructing a pipeline that will ingest data from Azure Data lake storage (ADLs) and process it further.

The pipeline can be constructed in Azure Data Factory(ADF), [5] ADF Consists of Integration Runtime (IR), Pipeline, Activity, Linked services.

Activity: Activities within Azure Data Factory define the actions that will be performed on the data.

Linked service: Services linking the Dataset with the Activity. *Integration Runtime:* It is a fully managed, serverless compute infrastructure. It executes a particular activity.

Pipeline: A pipeline is a logical grouping of activities that together perform a task.

Step 1: Once Debug runs successfully. Then the pipeline can be Published in the ADF, followed by the pipeline can be triggered using different triggering methods such as schedule trigger, tumbling window trigger, or event-based trigger.

Step 2: Initially we check whether the ADLS has the valid data files

a) If **False** then the pipeline will wait for a specified amount of time as hard-core by the user.
b) If **True** then the output across GetMetaData (Buffer) entity is read by the if-else conditions, if the schema matches with the Raw data then the Raw data is copied to the intermediate or staging phase (VALIDATED FOLDER) in ADLS and if the schema does not match then the Raw data is copied to the intermediate or staging phase (REJECTED FOLDER) in ADLs.

Step 3: Once the condition is satisfied then the Raw- data (Movies.csv, Ratings.csv) will be fed into the Notebook, To run the Notebook the spark clusters are created and the ALS algorithm is run in a multi-parallel processing fashion and obtained result is then shared with the user via email.

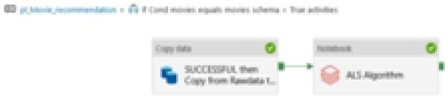

pl_Movie_recommendation > If Cond movies equals movies schema > True activities
Copy data
SUCCESSFUL then Copy from Rawdata t...
Notebook
ALS Algorithm

Figure 6: Success

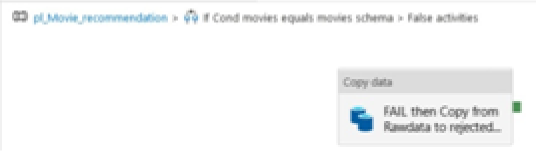

pl_Movie_recommendation > If Cond movies equals movies schema > False activities
Copy data
FAIL then Copy from Rawdata to rejected...

Figure 7: Failure

Step 4: If the email is sent successfully then the data stored in the staging phase will be deleted.

4. Conclusion

Azure offers benefits such as global access, reduced risk of data breach and loss, scalability, and almost no downtime. In addition, by receiving feedback on the recommendations, we can improve the recommendation system.

References

[1] Riahi, Youssra. (2018). Big Data and Big Data Analytics: Concepts, Types, and Technologies. 5. 524-528. 10.21276/ijre.2018.5.9.5.
[2] IBM (2014), "The Four V's of Big Data", IBM, available at: www.ibmbigdatahub.com/infographic/four- vs big-data
[3] Kumbhar, Vijaykumar Kharade, Kabir Kharade, Shraddha. (2017). A Comparative Study of Traditional Server and Azure Server. Journal of Advances in Science and Technology. 13. 329-331.
[4] CLOUD-BASED RECOMMENDATION SYSTEM Ricardo Batista Rodrigues, Carlo M. R. da Silva, Wilton O. Ferreira, Glaucia M. M. Campus1, Vinicius C. Garcia, Frederico A.
[5] Durão and Rodrigo E. Assad
[6] "Azure Data Factory", Microsoft, available at: https://docs.microsoft.com/en-us/azure/data-factory/
[7] Distributed Algorithms and Optimization, Spring 2015
[8] https://medium.com/radon-dev/als-implicit-collaborative- filtering-5ed653ba39fe
[9] Gupta, Meenu Co, Gupta Thakkar, Aditya Gupta, Vishal Pratap, Dhruv Rathore, Singh. (2021). Movie Recommender System Using Collaborative Filtering. 978-979.