uOttawa

ELG5166 Cloud Analytics
"Microsoft Azure"

Assignment 2

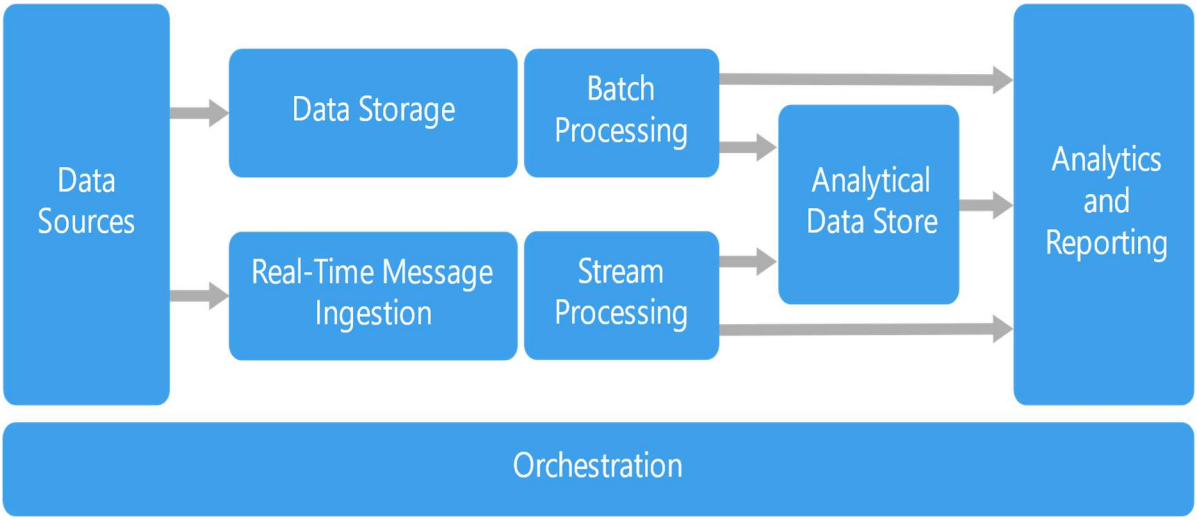
Tutor: Sara Hossam

Student Name: Ali El-Sherif

ID: 300327246

Table of Contents

1 - Describe briefly Azure Big Data Lambda Architecture



Data sources: All big data solutions start with one or more data sources. Examples include:

Application data stores, such as relational databases, Static files produced by applications, such as web server log files, Real-time data sources, such as IoT devices.

Data storage: Data for batch processing operations is typically stored in a distributed file store that can hold high volumes of large files in various formats. This kind of store is often called a *data lake*. Options for implementing this storage include Azure Data Lake Store or blob containers in Azure Storage.

Batch processing: Because the data sets are so large, often a big data solution must process data files using long-running batch jobs to filter, aggregate, and otherwise prepare the data for analysis. Usually, these jobs involve reading source files, processing them, and writing the output to new files. Options include running U-SQL jobs in Azure Data Lake Analytics, using Hive, Pig, or custom Map/Reduce jobs in an HDInsight Hadoop cluster, or using Java, Scala, or Python programs in an HDInsight Spark cluster.

Real-time message ingestion: If the solution includes real-time sources, the architecture must include a way to capture and store real-time messages for stream processing. This might be a simple data store, where incoming messages are dropped into a folder for processing. However, many solutions need a message ingestion store to act as a buffer for messages, and to support scale-out processing, reliable delivery, and other message queuing semantics. Options include Azure Event Hubs, Azure IoT Hubs, and Kafka.

Stream processing: After capturing real-time messages, the solution must process them by filtering, aggregating, and otherwise preparing the data for analysis. The processed stream data is then written to an output sink. Azure Stream Analytics provides a managed stream processing service based on perpetually running SQL queries that operate on unbounded streams. You can

also use open-source Apache streaming technologies like Storm and Spark Streaming in an HDInsight cluster.

Analytical data store: Many big data solutions prepare data for analysis and then serve the processed data in a structured format that can be queried using analytical tools. The analytical data store used to serve these queries can be a Kimball-style relational data warehouse, as seen in most traditional business intelligence (BI) solutions. Alternatively, the data could be presented through a low-latency NoSQL technology such as HBase, or an interactive Hive database that provides a metadata abstraction over data files in the distributed data store. Azure Synapse Analytics provides a managed service for large-scale, cloud-based data warehousing. HDInsight supports Interactive Hive, HBase, and Spark SQL, which can also be used to serve data for analysis.

Analysis and reporting: The goal of most big data solutions is to provide insights into the data through analysis and reporting. To empower users to analyze the data, the architecture may include a data modeling layer, such as a multidimensional OLAP cube or tabular data model in Azure Analysis Services. It might also support self-service BI, using the modeling and visualization technologies in Microsoft Power BI or Microsoft Excel. Analysis and reporting can also take the form of interactive data exploration by data scientists or data analysts. For these scenarios, many Azure services support analytical notebooks, such as Jupyter, enabling these users to leverage their existing skills with Python or R. For large-scale data exploration, you can use Microsoft R Server, either standalone or with Spark.

Orchestration: Most big data solutions consist of repeated data processing operations, encapsulated in workflows, that transform source data, move data between multiple sources and sinks, load the processed data into an analytical data store, or push the results straight to a report or dashboard. To automate these workflows, you can use an orchestration technology such Azure Data Factory or Apache Oozie and Sqoop.

2 - Compare between three of the primary Azure services

Azure Stream Analytics

Discover Azure Stream Analytics, the easy-to-use, real-time analytics service that is designed for mission-critical workloads. Build an end-to-end serverless streaming pipeline with just a few clicks. Go from zero to production in minutes using SQL—easily extensible with custom code and built-in machine learning capabilities for more advanced scenarios. Run your most demanding workloads with the confidence of a financially backed SLA.

Azure Blob Storage

Blob Storage is designed for:

Serving images or documents directly to a browser, storing files for distributed access, streaming video and audio, writing to log files, storing data for backup and restore, disaster recovery, and archiving and storing data for analysis by an on-premises or Azure-hosted service.

Azure Data Lake Storage Gen2

Azure Data Lake Storage Gen2 is a set of capabilities dedicated to big data analytics, built on Azure Blob Storage, Data Lake Storage Gen2 converges the capabilities of Azure Data Lake Storage Gen1 with Azure Blob Storage. For example, Data Lake Storage Gen2 provides file system semantics, file-level security, and scale. Because these capabilities are built on Blob storage, you'll also get low-cost, tiered storage, with high availability/disaster recovery capabilities.

3 - What are ADLS features? (In details and mention examples if possible)

Hadoop compatible access: - Can manage and access data using Data Lake Storage Gen2 in a manner similar to that of the Hadoop Distributed File System (HDFS). Available in all Apache Hadoop setups is the new ABFS driver, which is used to access data. Azure Synapse Analytics, Azure Databricks, and Azure HDInsight are some of these.

Cost-effective: Transactions and storage space are inexpensive with Data Lake Storage Gen2. As data moves through its lifespan, features like the lifecycle of Azure Blob Storage minimize costs.

Optimized driver: The ABFS driver has been tailored particularly for big data analytics. The endpoint dfs.core.windows.net surfaces the associated REST APIs.

A superset of POSIX permissions: ACL and POSIX permissions are supported by the security architecture for Data Lake Gen2, in addition to some additional granularity unique to Data Lake Storage Gen2. Frameworks like Hive and Spark as well as Storage Explorer allow for the configuration

4 - Compare between windowing functions.

Tumbling Window	Hopping Window	Sliding Window	Session Window	Snapshot Window
Tumbling window functions are used to segment a data stream into distinct time segments and perform a function against them.	Hopping window functions advance in time by a fixed period. It's common to conceive of them as Tumbling windows that can overlap and emit more rapidly than the window size. Events can be assigned to multiple Hopping window result sets. To make a Hopping window look like a Tumbling window, set the hop size to the same value as the window size.	Sliding windows, unlike Tumbling or Hopping windows, only emit events when the content of the window truly changes. To put it another way, when an event enters or exits the window. As a result, every window has at least one event. Events, like hopping windows, can belong to more than one sliding window.	The session window functions aggregate events that arrive at comparable times, cutting out periods of inactivity. It has three main parameters: timeout, maximum duration, and partitioning key.	Snapshot windows gather events with the same timestamp together. Unlike other windowing kinds that need a specialized window method (such as SessionWindow()), a snapshot window can be applied simply including System.Timestamp() in the GROUP BY clause.

5 - How does ADF work?

ADF workflows use and generate time-sliced data. The steps of processing it is divided into:

Transform & enrich: – use computing resources such HDInsight Hadoop, Spark Lake Analytics, or machine learning. Results come in the form of converted data that is produced on a regulated and maintained schedule for production environments.

Connect & collect: - A centralized data store in the cloud is used to store and handle data from all sources, including file sharing, FTP, Web services, and SaaS services.

Publish: - Data was modified in the cloud before being moved to on-premises sources like SQLServer.

References

[1] Jain, N. (2019, November 27). *List of Top 10 Azure Services*. Whizlabs Blog. https://www.whizlabs.com/blog/top-azure-services/

[2] normesta. (n.d.). *Azure Data Lake Storage Gen2 Introduction*. Learn.microsoft.com. https://learn.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction

[3] rolyon. (n.d.). *Azure documentation*. Learn.microsoft.com. https://learn.microsoft.com/en-us/azure/