CDS6214

Data Science Fundamentals

Lecture 8

Big Data Architecture (Part 1)

By 2025,
an estimated 463 exabytes of data will be generated every single day
(1EB = 1 million TB)

Big Data Ecosystem

- The big data ecosystem comprises of big data tools and techniques that are used to manage big data.

- This big data ecosystem is necessary due to:
  - the nature of the big data (the 5 V's) - scalability
  - the demand from the industry
  - the change in data landscape

Big Data Architecture

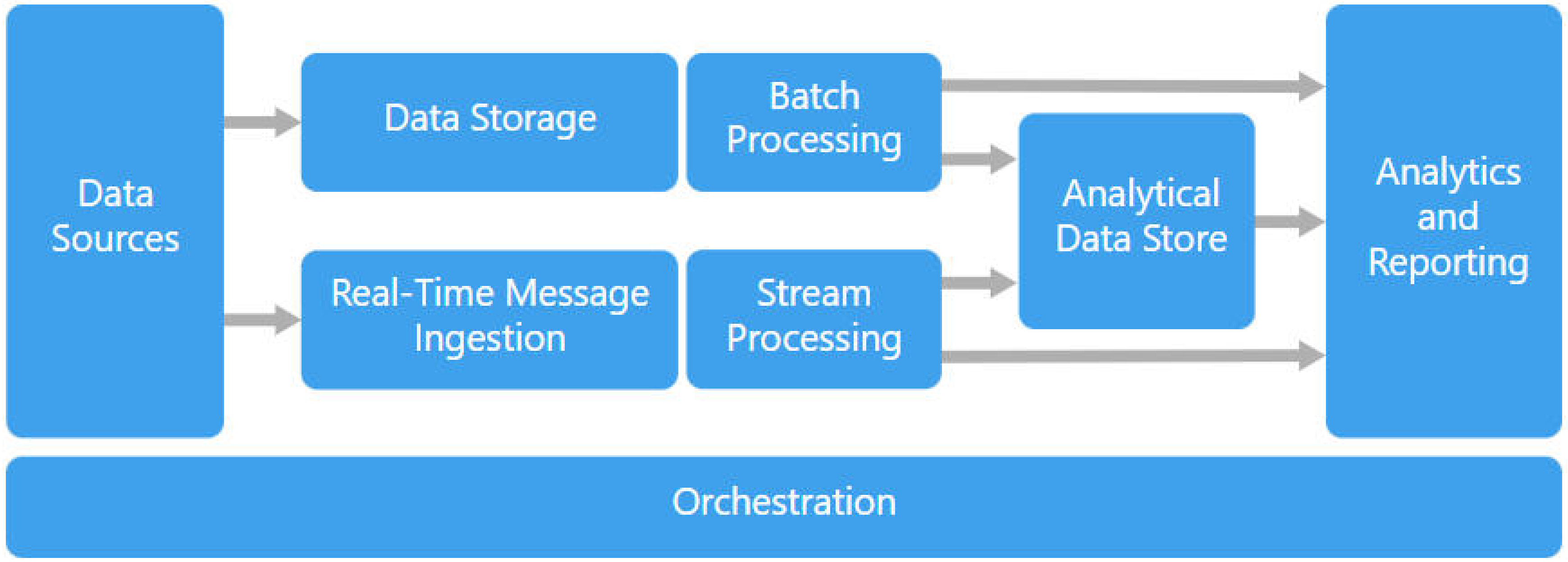
- It is a blueprint that indicates the solution and infrastructure to manage big data
- It contains the components, layers, and methods of communication.

https://www.interviewbit.com/blog/big-data-architecture/

Components of Big Data Architecture

Typical Big Data Architecture

It is up to organization to adopt all or some of the components of big data architecture



https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/big-data

Components of a typical big data architecture

- Data sources
- Data storage
- Batch processing
- Real-time message ingestion
- Stream processing
- Analytical data store
- Analytics and reporting
- Orchestration

https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures

Data Sources

- The starting point for the big data pipeline
- Refer to earlier slides of this course, data sources may consists of :
  - Open data
  - Third-party data providers
  - Relational databases
  - Data warehouses
  - Cloud-based data warehouses
  - SaaS applications
  - Real-time data from company servers and sensors such as IoT devices
  - Static files such as Windows logs

- Data can be ingested in batch mode or in real-time

https://www.interviewbit.com/blog/big-data-architecture/
https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures

Data Storage

- The data is converted into suitable format for storing and analysis.
- Data for batch processing operations is typically stored in a distributed file store that can hold high volumes of large files in variousformats : data lake.
- Can you identify the suitable data storage for :
  - Structured data?
  - Unstructured data?
- What is the difference between data warehouse and data lake ?

https://www.interviewbit.com/blog/big-data-architecture/
https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures

Batch Processing

- The size of big data poses challenge in processing them.
- Each chunk of data is split into different categories using long-running jobs, which filter and aggregate and also prepare data for analysis.
- These jobs typically require sources, process them, and deliver the processed files to new files.
- Multiple approaches to batch processing include custom Map/Reduce jobs in an HDInsight Hadoop cluster, or using Java, Scala, or Python programs in an HDInsight Spark cluster

https://www.interviewbit.com/blog/big-data-architecture/
https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures

Real-time Message Ingestion

- This component is necessary if the data source is ingested in real-time mode
- There is a need to capture and store real-time messages for stream processing
- 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. It can be data mart or data store where incoming messages are dropped into a folder for processing.
- This portion of a streaming architecture is often referred to as stream buffering.
- Examples of message-based ingestion stores include Apache Kafka, Apache Flume, Event hubs from Azure

https://www.interviewbit.com/blog/big-data-architecture/
https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures

Stream Processing

- This component processes the real-time messages in the form of windows or stream by filtering, aggregating, and otherwise preparing the data for analysis.
- The processed stream data is then written to an output sink.
- This includes Apache Spark, Flink, Storm, etc.

https://www.interviewbit.com/blog/big-data-architecture/
https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures

Analytical Data Store

- The analytical data store contains processed data which is ready for query analysis
- It can be Kimball-style relational data warehouse, as seen in most traditional business intelligence (BI) solutions.
- Alternatively, 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 can be used.

https://www.interviewbit.com/blog/big-data-architecture/
https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures

Analysis and reporting

- The ultimate goal of having big data platform is to be able to analyze the big data and derive insights from it. tions.
- The technologies to transform insights into useful visualizations such as graphs can assist organizations in interpreting the insights and make informed decisions. businesses.
- Examples include Cognos, Hyperion, Python or R with Spark.

https://www.interviewbit.com/blog/big-data-architecture/
https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures

Orchestration

- Big data solutions are built to support repeated operations such as data processing operations, data transformation, movement of 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

- Automation of pipeline / workflow is necessary . Examples of orchestration technology include Azure Data Factory or Apache Oozie and Sqoop.

https://www.interviewbit.com/blog/big-data-architecture/
https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures

Types of Big Data Architecture

(Lambda Architecture)

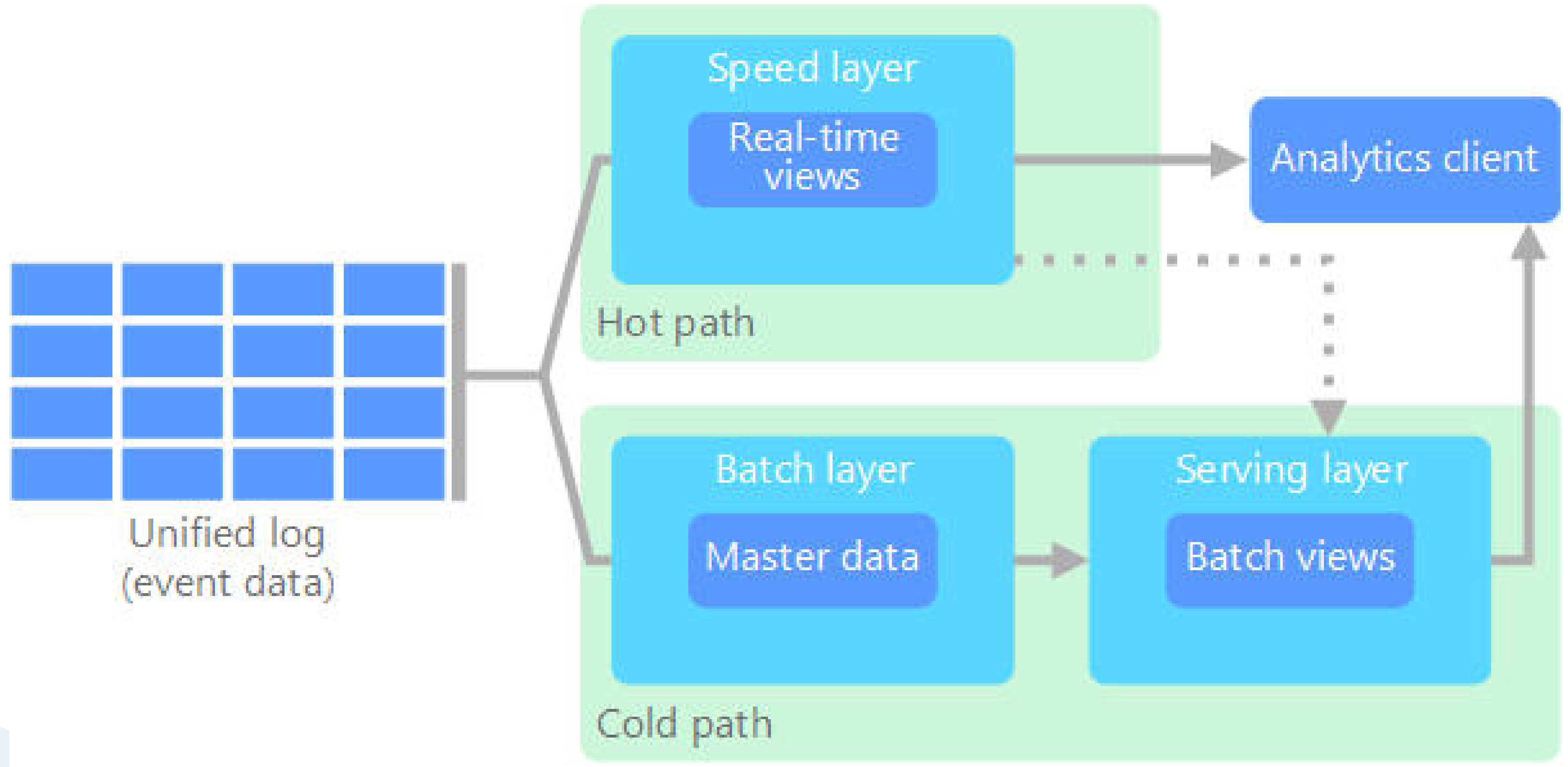
- Consider the following scenario:

Querying very large data sets cannot be done in real-time and it requires algorithm to apply the query on chunks of data. Often the results from the query are stored separately from the raw data.

Latency exists but you would like to get some results in real time (perhaps with some loss of accuracy), and combine these results with the results from the batch analytics

https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures

Lambda Architecture



https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures

Lambda Architecture

- Proposed by Nathan Marz, to solve the scenario by creating two paths for data flow which are batch layer (cold path) and speed layer (hot path).

  - A batch layer (cold path) stores incoming data in its raw form and performs batch processing on the data. The result of this processing is stored as a batch view in serving layer.
  - A speed layer (hot path) analyzes data in real time. This layer is designed for low latency, at the expense of accuracy. The speed layer updates the serving layer with incremental updates based on the most recent data.

https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures

Lambda Architecture

- The raw data stored at the batch layer is immutable. Incoming data is always appended to the existing data, and the previous data is never overwritten.
- Any changes to the value of a particular datum are stored as a new timestamped event record. This allows for recomputation at any point in time across the history of the data collected.
- The ability to recompute the batch view from the original raw data is important, because it allows for new views to be created as the system evolves.

https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures

Drawback of Lambda Architecture

- Complexity
  - Processing logic appears in two different paths using different frameworks.
    - Duplicate computation logic
    - Complexity of managing the architecture for both paths.
- Scalability. The batch layer is more difficult to scale than the real-time/stream processing layer.

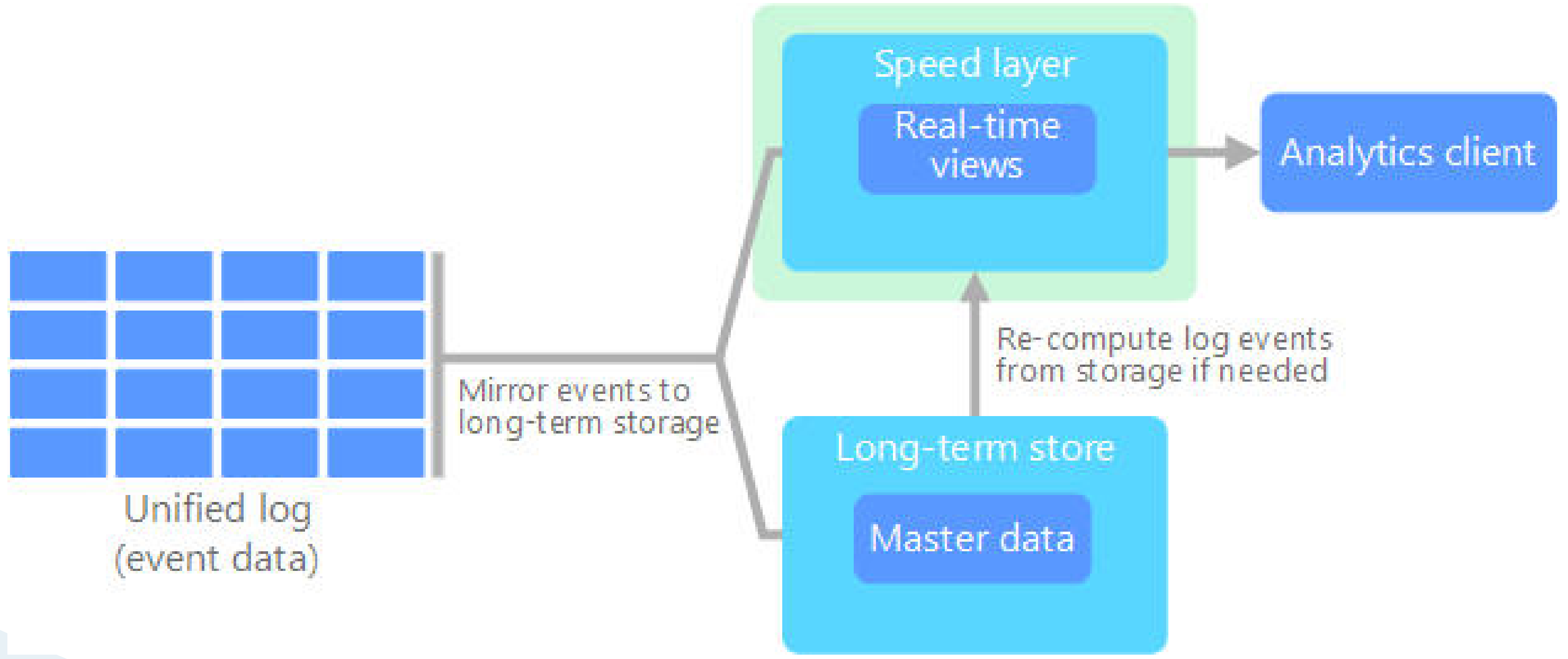
https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures

Types of Big Data Architecture

(Kappa Architecture)

Kappa Architecture

- Proposed by Jay Kreps, all data flows through a single path, using a stream processing system.



https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures

Kappa Architecture

- Intended to handle real-time processing
- No batch layer.
- All processing on single stream of data, and the results are stored in a database that can be queried in real-time.
- Speed efficiency
- The event data is immutable.The data is ingested as a stream of events into a distributed and fault tolerant unified log. These events are ordered, and the current state of an event is changed only by a new event being appended.
- All event processing is performed on the input stream and persisted in real-time view

https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures
https://pradeepl.com/blog/kappa-architecture/

Kappa Architecture

Benefits:

- Simplicity
- Efficiency
- Cost-effectiveness

Challenge
- Complexity
- Cost
- Data-loss risk

https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures
https://pradeepl.com/blog/kappa-architecture/

Benefits, Challenges and Best Practices

Benefits of Setting Up Big Data Architecture

- Mix and match of technologies are possible depending on factors such as existing skills or technology investments.
- Supports parallelism
- Elastic scale and pay only for the resources that you use.
- Interoperability with existing solutions.

https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures

Challenges of Setting Up Big Data Architecture

- Complexity imposes challenge to build, configure, test, and troubleshoot big data processes.
- Skillset. Many big data technologies are highly specialized, and use frameworks and languages that are not typical of more general application architectures.
- Some technologies have matured but many are evolving.
- Securing access to this data can be challenging, especially when the data must be ingested and consumed by multiple applications and platforms.

https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures

Best Practices to Big Data Architecture

- Leverage parallelism.
- Partition data.
- Apply schema-on-read semantics.
- Process data in-place.
- Balance utilization and time costs.
- Separate cluster resources.
- Orchestrate data ingestion.
- Scrub sensitive data early.

https://learn.microsoft.com/en-us/azure/architecture/databases/guide/big-data-architectures

Thank you