19/03/24 **Lecture 8: Data Processing**

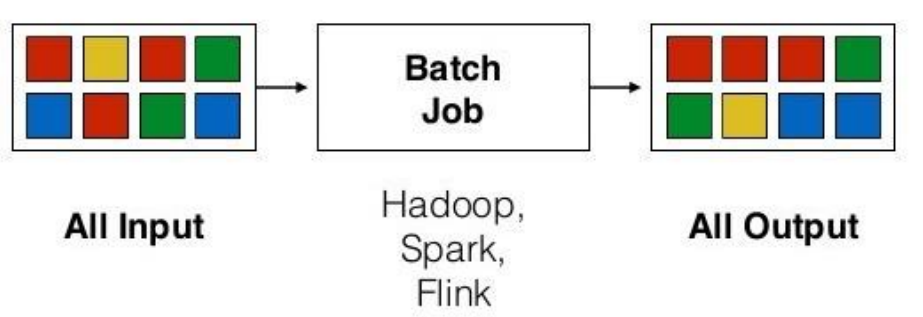
**Streaming Data Processing**

• Provides low latency analysis of high-volume data flows.

• Contrasts with batch processing.

**Batch processing**

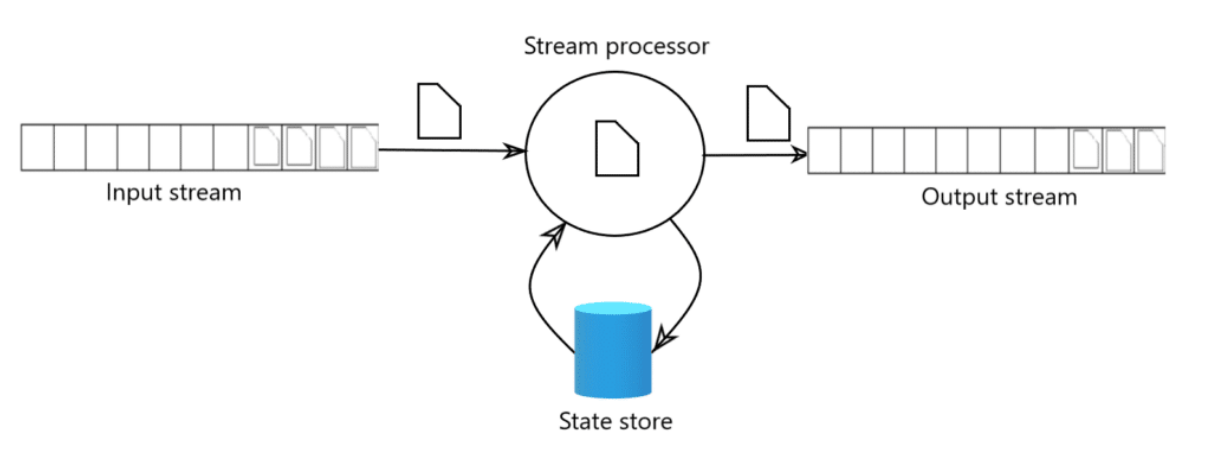
Large dataset processing methods include iterative MapReduce, collaborative filtering, and Deep Neural Network training. These methods use parallelism across Hadoop clusters or GPUs to achieve acceptable performance. However, they have a fixed input dataset and a long wall-clock time for processing the entire batch of data.



**Stream Processing**

Batch processing can handle large volumes of data but may not be suitable for high-velocity data. Organisations need quick processing for constantly changing applications like travel booking systems, air traffic control platforms, and social media. Stream processing systems are crucial for handling this demand.

Stream processing involves continuously incorporating new data to compute results, with input data unbounded and undetermined. It is expected to have low latency, with the time between raw data arrival and processed data availability being as small as possible. Batch processing can provide high throughput but always has high latency.



Stream processing systems process input data into real-time outputs, enabling fraud detection, IoT device sensor readings, personalisation, marketing, and advertising in real-time.

**Stream processing technologies**

• Apache Storm, Kafka, Samza, Flink, Beam, Spark.

• Streaming SQL for queries over data streams.

• Similar to ordinary SQL queries over RDB rows.

**Event Time in Stream Processing**

• Importance of treating event time in stream processing.

• Two timestamps associated with incoming data: Event time (time the initiating event occurred) and Processing time (time the event record arrived at the data centre).

• Technologies like Beam and Spark Structured Streaming support event processing in the order they occurred.

