Big Data Analytics

VIT Vidyalankar Institute of Technology
Accredited A+ by NAAC

Academic Year 2022-23

Agenda

1. Batch Processing
2. Streaming processing
3. Batch processing v/s Streaming processing
4. The Need for Streaming Processing
5. Sources of Streaming data
6. Benefits of Streaming Data
7. DBMS v/s DSMS
8. Stream Processing Model
9. Stream Queries - Standing Queries & Ad-Hoc Queries
10. Key Issues in Stream Processing

Prof. Prakash Parmar
Data Engineer | Data Analyst
Assistant Professor
Vidyalankar Institute of Technology, Mumbai
GATE Instructor: Vidyalankar Classes

Batch Processing

VIT Vidyalankar Institute of Technology
Accredited A+ by NAAC

- **Batch processing refers to the processing of blocks of data that have already been stored over a period of time.**
- Example, processing transactions that have been performed by a financial firm in a week. This data contains millions of records for a day that can be stored as a file or record. The particular file will undergo processing at the end of the day for various analyses that the firm requires, and it will be a time taking process.
- **Batch processing is often used for tasks that do not require immediate or real-time processing and can tolerate some delay.**

Characteristics of Batch processing

VIT Vidyalankar Institute of Technology
Accredited A+ by NAAC

Data Collection:

Data is collected and stored until a sufficient amount of the predetermined batch size is reached.

Higher Latency:

Due to the waiting period for batch accumulation and processing, batch processing inherently has higher latency compared to stream processing.

Offline Processing:

Batch processing is typically performed offline or at scheduled intervals. It is not designed for real-time or immediate processing.

Use Cases:

Batch processing is well-suited for tasks such as data warehousing, ETL (Extract, Transform, Load) operations, generating reports, and historical analytics.

Streaming Processing

VIT Vidyalankar Institute of Technology
Accredited A+ by NAAC

- **Streaming processing, also known as real-time data processing or event stream processing, is a method of handling and analyzing data as it is generated or received in real-time.**
- It is designed to handle continuous and often rapidly changing data streams.

Characteristics of Streaming processing

VIT Vidyalankar Institute of Technology
Accredited A+ by NAAC

Continuous Data Flow:

Streaming processing deals with a continuous and unending flow of data in the form of small, discrete units called events or data records

Real-Time or Near-Real-Time:

Streaming processing aims to process data as soon as it becomes available or with minimal delay.

Low Latency:

Streaming processing is optimized for low-latency data analysis, minimizing the time between data generation and processing.

Event-Driven:

Streaming processing is event-driven, meaning that actions or responses can be triggered based on incoming events or patterns detected within the data stream.

Characteristics of Streaming processing

Scalability:

Streaming systems are often designed to be highly scalable, allowing them to handle increasing data volumes and processing loads. Scalability is achieved by adding more resources, such as processing nodes or partitions, to accommodate dynamic data sources and fluctuations in data rates.

Complex Event Processing (CEP):

Streaming processing platforms can perform complex event processing (CEP) to detect patterns, correlations, or anomalies in real-time data. This enables automated responses or alerts based on predefined rules or conditions.

Fault Tolerance:

Reliable streaming processing systems are designed to handle hardware or network failures gracefully, ensuring that data is not lost, and processing continues even in the presence of issues.

Batch processing v/s Streaming processing

Vidyalankar Institute of Technology
Accredited A+ by NAAC

Traditional Processing

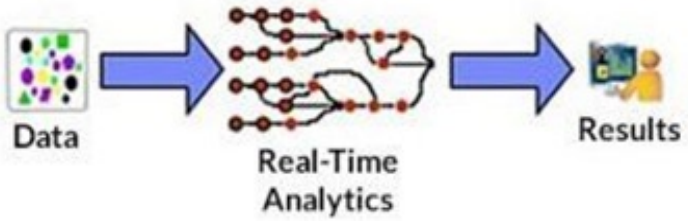
Historical fact finding

Find and analyze information stored on disk

Batch paradigm, pull model

Query-driven: submits queries to static data

Stream Processing



Current fact finding

Analyze data in motion – before it is stored

Low latency paradigm, push model

Data driven: bring data to the analytics

Batch processing v/s Streaming processing

VIT Vidyalankar Institute of Technology
Accredited A+ by NAAC

Aspect	Batch Processing	Stream Processing
Data Flow	Data collected and processed in fixed-size batches	Continuous and real-time data flow in small units (events)
Real-Time Processing	Typically performs processing offline or on a scheduled basis, resulting in higher latency	Processes data as it arrives, aiming for real-time or near-real-time analysis
Latency	Higher latency due to data waiting for batch accumulation before processing	Low-latency processing, minimizing the time between data generation and processing
Event-Driven	Typically, not event-driven; processing occurs at scheduled intervals	Event-driven; actions triggered based on incoming events or patterns
Data Size	Processes fixed-size batches, often with a substantial volume of data	Handles data streams of varying volumes and rates
Scalability	Scalability is limited to batch size and may require additional resources for larger batches	Highly scalable to accommodate dynamic data sources and fluctuations in data rates
Immediate Insights	Results are available after batch processing is complete	Provides immediate insights and enables real-time decision-making
Use Cases	Data warehousing, ETL (Extract, Transform, Load), historical analytics	Real-time monitoring, fraud detection, recommendation engines, IoT applications
Examples	Monthly financial reports, daily website analytics, quarterly sales summaries	Social media updates, sensor data, financial transactions, log streams

Need for Streaming Processing

Vidyalankar Institute of Technology
Accredited A+ by NAAC

- The need for streaming processing arises from the scenarios where real-time or near-real-time data analysis is essential for decision-making, security, automation, and user experience.
- **Timely Decision-Making**: In certain domains like finance and cybersecurity, timely decisions are critical to mitigate risks, prevent fraud, and respond to events as they happen.
- **IoT Data**: The Internet of Things (IoT) generates vast amounts of data continuously. Processing this data in real-time is necessary for monitoring and controlling IoT devices effectively.
- **Customer Experience**: Industries like e-commerce, content streaming, and social media benefit from real-time recommendations and personalization to enhance user experiences.
- **Efficiency**: Streamlining business processes and workflows often requires real-time data analysis to identify bottlenecks and optimize operations.
- **Data Quality**: Real-time data processing can help identify and address data quality issues as they occur, ensuring that only accurate and valid data is used for decision-making.

Sources of Streaming data

It is usually generated simultaneously and at high speed by many data sources, which can include applications, IoT sensors, log files, and servers.

Social Media Platforms:

Description: Social media platforms, such as Twitter, Facebook, and Instagram, produce continuous streams of user-generated content. This content includes text posts, images, videos, likes, shares, and comments.

Use Cases: Social media streams are used for sentiment analysis, trend detection, real-time news updates, and social network analysis.

Web Server Logs:

Description: Web servers generate log files that record every user interaction with a website. These logs include information such as IP addresses, user agents, URLs, and timestamps.

Use Cases: Analyzing web server logs can help track website traffic, identify security threats, and optimize web application performance.

Financial Markets:

Description: Financial markets generate real-time data on stock prices, currency exchange rates, commodities, and trading volumes. These data streams are crucial for traders and financial analysts.

Use Cases: Financial market data is used for algorithmic trading, risk management, and market trend analysis.

Sources of Streaming data

Vidyalankar Institute of Technology
Accredited A+ by NAAC

Sensor Networks:

Description: Sensor networks consist of a collection of sensors that continuously monitor physical or environmental conditions. These sensors can include temperature sensors, humidity sensors, motion detectors, GPS devices, and more.

Use Cases: Sensor networks are used in applications like environmental monitoring (e.g., weather forecasting), industrial automation (e.g., manufacturing process control), and IoT devices (e.g., smart home sensors).

IoT Devices:

Description: The Internet of Things (IoT) encompasses a wide range of devices, from smart thermostats to wearable fitness trackers, which continuously generate data related to their status, usage, and environment.

Use Cases: IoT data is used for remote monitoring, predictive maintenance, healthcare, and smart city applications.

Network Traffic:

Description: Network routers and switches continuously generate logs of network traffic, including source and destination IP addresses, protocols, packet sizes, and timestamps.

Use Cases: Network traffic data is used for network security, anomaly detection, and bandwidth optimization.

Security Cameras:

Description: Surveillance cameras continuously capture video feeds. These feeds can be processed in real-time for security and surveillance applications.

Use Cases: Security cameras are used for video analytics, object detection, and facial recognition in security systems.

Benefits of Streaming Data

VIT Vidyalankar Institute of Technology
Accredited A+ by NAAC

- Real-Time Insights
- Timely Actions
- Immediate Feedback
- Continuous Monitoring
- Event-Driven Architectures
- Improved Security
- Enhanced Customer Experience
- Competitive Advantage

Data Stream Management System

Vidyalankar Institute of Technology
Accredited A+ by NAAC

- A Data Stream Management System (DSMS) is a specialized software system designed to manage and process continuous streams of data in real-time or near real-time.
- Unlike traditional databases that focus on storing and querying static data, DSMSs are built to handle dynamic, high-velocity data streams that can be generated from various sources, such as sensors, social media, financial transactions, IoT devices, and more.
- Names of DSMS

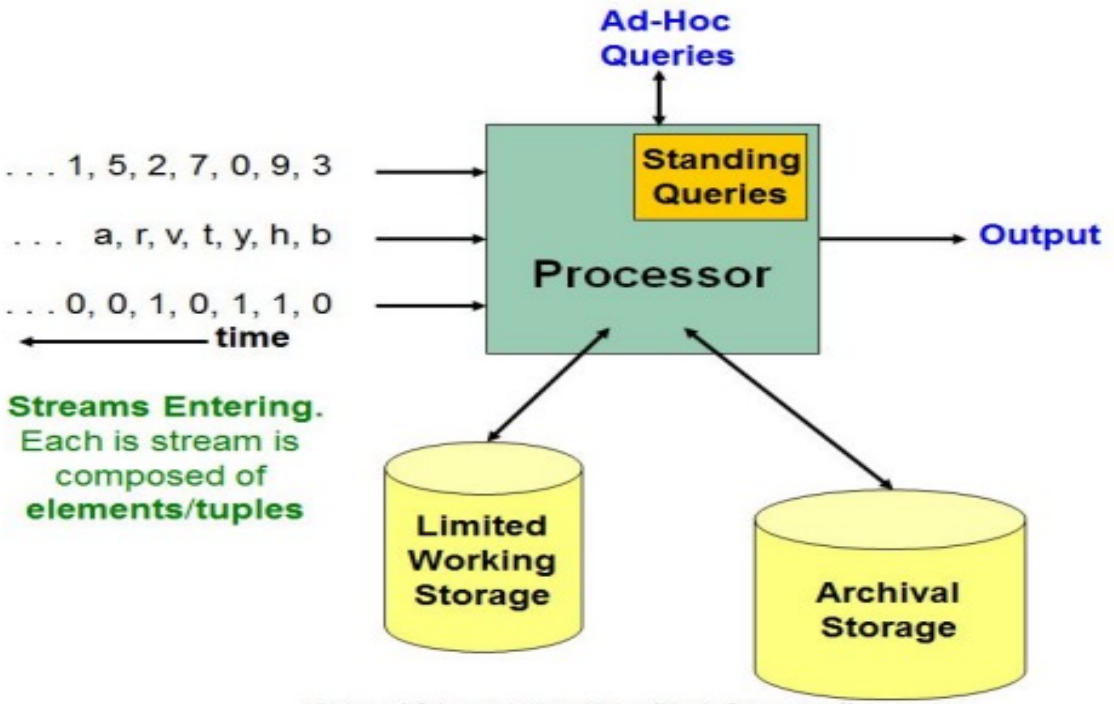
1. Apache Kafka Streams
2. Apache Flink
3. Apache Storm
4. Google Cloud Dataflow
5. Microsoft Azure Stream Analytics
6. MapR Streams etc.

DBMS v/s DSMS

VIT Vidyalankar Institute of Technology
Accredited A+ by NAAC

DBMS	DSMS
Deals with persistent data	Deals with stream data
Data access can be random	Data access is sequential
Query provides the exact answer	Query provides the approximately answer
Uses unbounded disk store	Uses bounded main memory
Queries are one-time queries	Queries are continuous
No real time requirements	Real time requirements
Suitable for OLAP, OLTP, reporting, and batch processing.	Ideal for real-time analytics, IoT data, event monitoring, and complex event processing.
Oracle, MySQL, SQL Server, PostgreSQL, MongoDB.	Apache Kafka Streams, Apache Flink, Apache Storm, Apache Beam.

General Stream Processing Model



Stream Queries

Vidyalankar Institute of Technology
Accredited A+ by NAAC

Stream queries are a fundamental component of stream processing systems, allowing users to extract meaningful information and insights from continuous data streams.

There are two main categories of stream queries: standing queries and ad-hoc queries, each serving different purposes:

Standing Queries

- Standing queries, also known as continuous queries or persistent queries, are predefined queries that run continuously on a data stream.
- These queries are set up in advance and are continuously executed against the incoming data as long as the stream is active.
- **Examples**: Examples of standing queries include monitoring for temperature anomalies in IoT sensor data, detecting fraud patterns in financial transactions, or counting the number of mentions of specific keywords in a social media stream

Ad-Hoc Queries

- Ad-hoc queries, also known as dynamic queries or on-demand queries, are queries that users create and execute as needed.
- Unlike standing queries, they are not predefined and do not run continuously; instead, they are initiated by users when they want to analyze specific data.
- **Examples:** Examples of ad-hoc queries include performing statistical analysis on recent stock market data, searching for specific keywords in a social media stream for research, or aggregating sensor data for a specific time frame.

Key Issues in Stream Processing

Vidyalankar Institute of Technology
Accredited A+ by NAAC

Data Velocity: Streams can generate data at high velocities, creating challenges in terms of data ingestion, processing, and storage. Handling fast data ingestion rates requires scalable and efficient stream processing infrastructure.

Data Volume: Large volumes of data can overwhelm processing systems. Managing and storing historical stream data for analysis can be resource-intensive, leading to increased infrastructure costs.

Data Variability: Data streams may exhibit variable data rates, leading to bursty or sporadic data patterns. Stream processing systems must adapt to these fluctuations in data velocity.

Data Complexity: Streams can contain diverse data types, including structured, semi-structured, and unstructured data. Processing and extracting insights from this complex data require flexible data handling capabilities.

Latency Management: Reducing data processing latency is crucial in stream processing. Delayed processing can result in missed opportunities or inaccurate insights. Achieving low-latency processing requires optimized algorithms and infrastructure.

Ordering and Time Synchronization: Ensuring the correct order of events in a stream is essential for accurate analysis. Handling out-of-order events and managing event time stamps are common challenges.

Fault Tolerance: Stream processing systems should be fault-tolerant to handle hardware failures, network issues, or component failures without data loss. This often involves mechanisms like replication and checkpointing.

Learn Fundamentals & Enjoy Engineering

Thank You

Prof. Prakash Parmar
Data Engineer | Data Analyst
Assistant Professor
Vidyalankar Institute of Technology, Mumbai
GATE Instructor: Vidyalankar Classes