What is Streaming Data?

What is streaming data?

Streaming data is data that is emitted at high volume in a continuous, incremental manner with the goal of low-latency processing. Organizations have thousands of data sources that typically simultaneously emit messages, records, or data ranging in size from a few bytes to several megabytes (MB). Streaming data includes location, event, and sensor data that companies use for real-time analytics and visibility into many aspects of their business. For example, companies can track changes in public sentiment on their brands and products by continuously analyzing clickstream and customer posts from social media streams then responding promptly as needed.

What are the characteristics of streaming data?

A data stream has the following specific characteristics that define it.

Chronologically significant

Individual elements in a data stream contain time stamps. The data stream itself may be time-sensitive with diminished significance after a specific time interval. For example, your application makes restaurant recommendations based on the current location of its user. You have to act upon user geolocation data in real time or the data loses significance.

Continuously flowing

A data stream has no beginning or end. It collects data constantly and continuously as long as required. For example, server activity logs accumulate as long as the server runs.

Unique

Repeat transmission of a data stream is challenging because of time sensitivity. Hence, accurate real-time data processing is critical. Unfortunately, provisions for retransmission are limited in most streaming data sources.

Nonhomogeneous

Some sources may stream data in multiple formats that are in structured formats such as JSON, Avro, and comma-separated values (CSV) with data types that include strings, numbers, dates, and binary types. Your stream processing systems should have the capabilities to handle such data variations.

Imperfect

Temporary errors at the source may result in damaged or missing elements in the streamed data. It can be challenging to guarantee data consistency because of the continuous nature of the stream. Stream processing and analytics systems typically include logic for data validation to mitigate or minimize these errors.

Why is streaming data important?

Traditional data processing systems capture data in a central data warehouse and process it in groups or batches. These systems were built to ingest and structure data before analytics. However, in recent years, the nature of enterprise data and the underlying data processing systems have changed significantly.

Infinite data volume

Generated data volumes from stream sources can be very large, making it a challenge for real-time analytics to regulate the streaming data's integrity (validation), structure (evolution), or velocity (throughput and latency).

Advanced data processing systems

At the same time, cloud infrastructure has introduced flexibility in the scale and usage of computing resources. You use exactly what you need and pay only for what you use. You have the options of real-time filtering or aggregation both before and after storing streaming data. Streaming data architecture uses cloud technologies to consume, enrich, analyze, and permanently store streaming data as required.

What are the use cases for streaming data?

A stream processing system is beneficial in most scenarios where new and dynamic data is generated continually. It applies to most of the industry segments and big data use cases.

Companies generally begin with simple applications, such as collecting system logs and rudimentary processing like rolling min-max computations. Then, these applications evolve to more sophisticated near real-time processing.

Here are some more examples of streaming data.

Data analysis

Applications process data streams to produce reports and perform actions in response, such as emitting alarms when key measures exceed certain thresholds. More sophisticated stream processing applications extract deeper insights by applying machine learning algorithms to business and customer activity data.

IoT applications

Internet of Things (IoT) devices are another use case for streaming data. Sensors in vehicles, industrial equipment, and farm machinery send data to a streaming application. The application monitors performance, detects potential defects in advance, and automatically places a spare part order, preventing equipment downtime.

Financial analysis

Financial institutions use stream data to track real-time changes in the stock market, compute value at risk, and automatically rebalance portfolios based on stock price movements. Another financial use case is fraud detection of credit card transactions using real-time inferencing against streaming transaction data.

Real-time recommendations

Real estate applications track geolocation data from consumers' mobile devices and make real-time recommendations of properties to visit. Similarly, advertising, food, retail, and consumer applications can integrate real-time recommendations to give more value to customers.

Service guarantees

You can implement data stream processing to track and maintain service levels in applications and equipment. For example, a solar power company has to maintain power throughput for its customers or pay penalties. It implements a streaming data application that monitors all panels in the field and schedules service in real time. Thus, it can minimize each panel's periods of low throughput and the associated penalty payouts.

Media and gaming

Media publishers stream billions of clickstream records from their online properties, aggregate and enrich the data with user demographic information, and optimize the content placement. This helps publishers deliver a better, more relevant experience to audiences. Similarly, online gaming companies use event stream processing to analyze player-game interactions and offer dynamic experiences to engage players.

Risk control

Live streaming and social platforms capture user behavior data in real time for risk control over users' financial activity, such as recharge, refund, and rewards. They view real-time dashboards to flexibly adjust risk strategies.

What is the difference between batch data and streaming data?

Batch processing is the method computers use to periodically complete high-volume, repetitive data jobs. You can use it to compute arbitrary queries over different sets of data. It usually derives computational results from all the data it encompasses and allows for deep analysis of big data sets. MapReduce-based systems, like Amazon EMR, are examples of platforms that support batch jobs.

In contrast, stream processing requires ingesting a data sequence and incrementally updating metrics, reports, and summary statistics in response to each arriving data record. It is better suited for real-time analytics and response functions.

	Batch processing	Stream processing
Data scope	Queries or processing over all or most of the data in the dataset.	Queries or processing over data within a rolling time window, or on just the most recent data record.
Data size	Large batches of data.	Individual records or micro batches consisting of a few records.
Performance	Latencies in minutes to hours.	Requires latency in the order of seconds or milliseconds.
Analysis	Complex analytics.	Simple response functions, aggregates, and rolling metrics.

Many organizations are building a hybrid model by combining the two approaches to maintain a real-time layer and a batch layer. For example, you can first process data in a streaming data platform such as Amazon Kinesis to extract real-time insights. Then, you can persist it into a store like Amazon Simple Storage Service (Amazon S3). There, it can be transformed and loaded for various batch processing use cases.

Amazon Redshift Streaming Ingestion allows users to ingest data directly from Amazon Kinesis Data Streams without having to stage it in Amazon S3. The service can also ingest data from Amazon Managed Streaming for Apache Kafka (Amazon MSK) into Amazon Redshift.

How can you process streaming data?

Streaming data architecture contains two main types of components.

Stream producers

Stream producers are software components in applications and IoT systems that collect data. They transmit records to the stream processor that contain a stream name, data value, and sequence number. The processor either buffers

or temporarily groups the data records by stream name. It uses the sequence number to track the unique position of each record and process data chronologically.

Stream consumers

Stream consumers are software components that process and analyze the data streams buffered in the processor. Each consumer has analytics capabilities like correlations, aggregations, filtering, sampling, or machine learning. Each stream can have multiple consumers, and each consumer can process numerous streams. Consumers can also send the changed data back to the processor to create new streams for other consumers.

Architecture implementation

To implement streaming data architecture, you require storage and processing layers. The storage layer must support record ordering and strong consistency to enable fast, inexpensive, and replayable reads and writes of large data streams. The processing layer is responsible for consuming data from the storage layer, running computations on that data, and then notifying the storage layer to delete data that is no longer needed.

What are the challenges in working with streaming data?

Streaming data architecture requires special considerations due to the nature and volume of data.

Availability

Streaming data applications require consistency, low latency, and high availability. Consumers are constantly taking new data from the stream to process it. Delays from the producer could back up the system and cause errors.

Scalability

Raw data streams can surge rapidly and unexpectedly. For example, social media posts spike during a big sporting event. Therefore, the system should prioritize proper data sequencing, availability, and consistency—even during peak loads.

Durability

Because of the time sensitivity of data, the stream processing system has to be fault tolerant. Otherwise, the data will be lost forever in an outage or failure.

How can AWS support your streaming data requirements?

AWS provides several options to work with streaming data.

Amazon Kinesis

Kinesis is a platform for streaming data on AWS. It offers robust services that make it easy to load and analyze streaming data, while also allowing you to build custom streaming data applications for specialized needs.

Kinesis provides three services: Amazon Data Firehose, Amazon Kinesis Data Streams, and Amazon Managed Streaming for Apache Kafka (Amazon MSK).

Amazon Data Firehose

Amazon Data Firehose can capture and automatically load streaming data into Amazon Simple Storage Service (Amazon S3) and Amazon Redshift. This allows you to perform real-time analytics with existing business intelligence tools and dashboards you're already using today.

Kinesis Data Streams

Kinesis Data Streams can continuously capture and store terabytes (TB) of data per hour from hundreds of thousands of sources. It supports your choice of stream processing framework, including Amazon Kinesis Client Library (KCL), Apache Storm, and Apache Spark Streaming.

Amazon MSK

Amazon MSK is a fully managed service that makes it easy for you to build and run applications using Apache Kafka to process streaming data. Apache Kafka is an open-source platform for building real-time streaming data pipelines and applications.

Amazon Redshift

Amazon Redshift Streaming Ingestion allows users to ingest streaming data into their data warehouse for real-time analytics from multiple Kinesis data streams. You can perform rich analytics using familiar SQL and easily create and manage ELT pipelines. It also lets you process large volumes of streaming data with low latency and high throughput to derive insights in seconds.

Other streaming solutions on Amazon EC2

You can install streaming data platforms of your choice on Amazon Elastic Compute Cloud (Amazon EC2) and Amazon EMR and build your custom stream storage and processing layers. As a result, you can avoid the friction of infrastructure provisioning, plus gain access to various stream storage and processing frameworks. Options for the data storage layer include Amazon MSK and Apache Flume. Options for the stream processing layer include Apache Spark Streaming and Apache Storm.

Get started with streaming data on AWS by creating a free AWS account today!