gathr

Data Science for Real-time Streaming Analytics

Introduction

Real-time streaming analytics (RTSA) technology allows for the collection, integration, analysis, and visualization of data in real-time. It does so without disrupting the activity of existing sources, storage, and enterprise systems.

Streaming analytics applications are designed to handle high volumes of data in real-time with a scalable, highly available and fault tolerant architecture, allowing organizations to extract business value from data in motion in much the same way that traditional batch analytics tools have allowed them to do with data at rest, but much, much faster.

Fraud Detection

To illustrate the need to analyze and act on information while the data is still in motion, consider an example of how RTSA technology is currently at work. Imagine a customer is making a payment using a credit card. At the moment of the payment, the bank analyzes the payment pattern on that particular credit card to detect the possibility of fraud. This analysis involves the history, frequency, and amounts of previous transactions for that credit card from its database records.

Depending on the scoring analysis, the bank authorizes the transaction, keeps it on hold, or declines it, all in real-time. The typical duration within which the bank has to validate a transaction, is typically less than five seconds.

Keep in mind that the window of opportunity may be only a few seconds, maybe minutes or as much as several hours, but the key is the ability to be proactive - automatically or via human interaction based on predictions that are calculated in real-time by an analytic model.

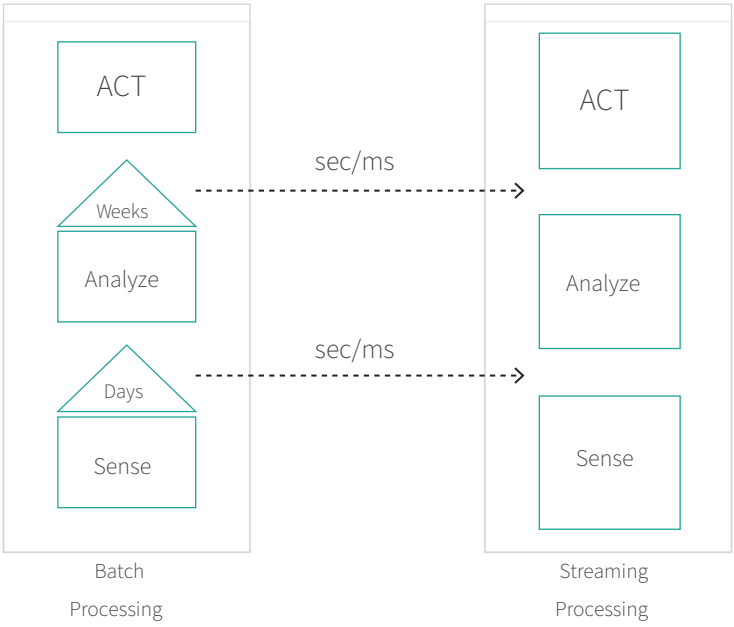

Figure 1: Difference Between Batch and Streaming Data Processing

The demand for similar use cases that deliver actionable insights as they occur are increasing in many business domains, including trading, social networks, the Internet of Things, system monitoring, and more.

Types of Data Analytics

Data analytics applications can typically be categorized as either batch or streaming data analytics. Deciding which mode is needed is generally determined by your business objectives as well as the type of data you are working with.

Batch data analytics provides an efficient way of analyzing high volumes of data at rest to get actionable business insights. Such data typically represents a series of events that are continuously captured in different application contexts, such as web logs, user transactions, system logs, sensor networks, etc. In batch data analytics, the events' data is collected based on the number of records or a certain period of time (a day, for example). The data is then stored and processed as a finite data set. The goal of such an analytics solution is to get actionable insights from the collected data; however, the time required to collect, process the data, and get insights is quite long.

By contrast, streaming data analytics enables organizations to process data immediately upon arrival and get actionable insights with sub-second latency. RTSA technology accelerates time-to-insight from massive amounts of data, whether that data originates from market data, sensors, mobile phones, the Internet of Things, web clickstreams, or transactions. For example, you can use RTSA to enable real-time alerts for customers or to leverage new business opportunities – like making promotional offers to customers based on their location. Streaming analytics capabilities are also vital in the security-monitoring context because they allow organizations a way to quickly correlate seemingly disparate events to detect threat patterns and risks. The following table summarizes the key differences between batch and streaming data analytics applications.

Batch Data Analytics	Streaming Data Analytics
High latency application	Low latency application
Static files to process	Event streams
Process after store	Process then store
Periodic jobs	Always ON
Delayed action	Instant action

Table 1: Key Characteristics of Batch and Streaming Data Analytics Applications

Recently there has been a movement in business applications from batch data analytics to streaming data analytics driven by the fovllowing factors:

1. Real-time Data-driven Decision Making

Growing opportunities to collect and leverage digital information have led many business leaders to change how they make decisions - relying less on intuition and history and more on in-the-moment data.

By looking at real-time data and insights, strategic, operational, and tactical decisions can be carefully reviewed to much more positively impact the bottom line. Leadership can also use streaming data to factor into predictive analytics and look at real-time KPIs to better understand employee and business performance. For instance, the practice of “dynamic” or “surge” pricing is becoming increasingly prevalent in various industry verticals. Uber the ride-sharing service, charges higher rates on Saturday nights and at other peak times of demand. Sellers on Amazon.com and other e-commerce sites are also using dynamic pricing more and more to match their inventories with demand. Such applications need to act in near real-time based on the current status of supply and demand.

2. Better Efficiency in Operations

By tracking systems, products, and equipment performance in real-time, quick decisions can be made that can greatly affect efficiency. By understanding which operational parameters have an impact on overall business performance, decision makers can be sure to track, measure, and tweak them accordingly, which can decrease costs or lead to smoother and faster processes. For instance, by analyzing a delivery truck in real-time and tracking the route and fuel use, the overall process can be evaluated to find out if there is a better route that may minimize fuel consumption and increase speed of performance.

3. Enhanced Customer Satisfaction

Customer intelligence applications can be a business’s greatest tool. With the ability to track individuals and their actions, businesses can harness this technology to create better customer experiences that are relevant and targeted. This is completely possible if data is analyzed in real-time. For instance, mobile coupon offering organizations are considering current location of their users collected via mobile GPS to instantly release geo-specific deals. The selection of the relevant deals is typically carried out in real-time by predictive matching of local merchants offers to the customers likely to accept them based on their past transaction history and other available CRM data.

4. Competitive Advantage

The ability to ingest, analyze, and take action on high-volume, multi-structured data in real-time is quickly becoming a 'must have' capability for enterprises across industry verticals to surpass or in many cases maintain parity with their competitors who are deploying these technology advancements as part of their new Enterprise Data Architecture blueprint.

How to Extract Insights from Data in Motion

The objective of streaming data analytics is to extract actionable insights from streaming data with very low latency (real-time).

These insights are generated by the underlying streaming analytical models which receive incoming streams of data and produce outcomes on that data instantly. The design, development and deployment of these models can be performed in the following ways:

A. Offline modeling on a batch of aggregated data

B. Online modeling: incremental model updates with each instance of streaming data

Offline Modeling

Making predictions on real-time data streams involves building an offline model and applying it to a stream. Models incorporate one or more machine learning algorithms which is trained using data previously collected.

Modeling or model learning is the process of applying knowledge discovery algorithms to the data to extract patterns and trends of interest, resulting in one or more models. Evaluation or model scoring involves applying the derived model to new streaming data to get predictions. Traditionally, this approach has been heavily applied to many real-world scenarios in which a large repository of previously collected data is available.

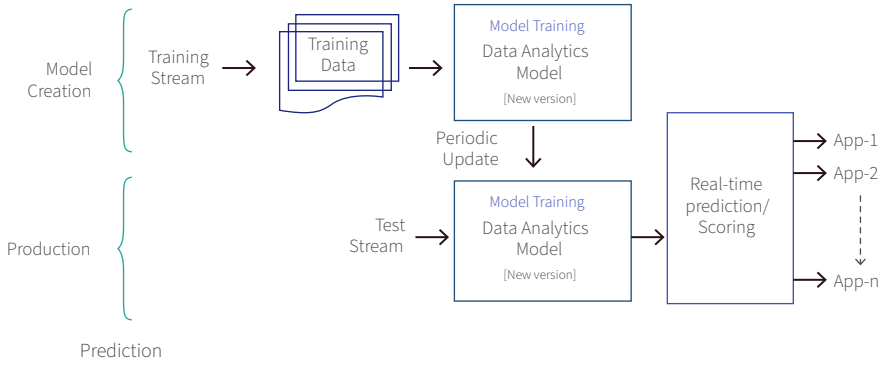

Figure 2: Offline Modeling

For example, consider an application whose goal is to spot fraudulent credit card transactions. Here's how you'd build a model for that.

1. Start with a large collection of transactions, including information about previously detected fraudulent transactions.
2. Capture the patterns that separate regular transactions from abnormal ones.

3. Base each pattern on transaction attributes such as monetary value, location of the transaction's origination, and the frequency of certain types of purchases for different types of customers.
4. Score this model against newly arriving credit card transactions to predict if individual transactions, or a group of transactions, matches the patterns that indicate fraud.

This model allows a credit card company to identify fraudulent transactions as soon as they occur and, sometimes, even before additional transactions take place.

Online Modeling

A preliminary model is created and trained using a small set of streaming data. It is deployed directly for online evaluation using new streaming data. The duration of a training data stream is short but the underlying model has the capability to evolve and improve itself based on performance feedback from new streaming test data. This is what distinguishes it from offline batch modeling. In other words, the underlying model learns each time it gets performance feedback on a new data stream.

Because offline modeling depends on previously stored data, it may not perform well with new and changing data streams. Therefore, the online modeling approach is more suitable to business situations in which the data is changing over time.

For example, traditional machine-learned ranking algorithms for web search are trained in batch mode. This assumes that the relevance of documents for a given query is static. In scenarios where the relevance of documents to a query changes over time, such as ranking recent documents for a breaking news story, (called recency ranking), the batch-learned ranking functions have limitations. In these cases, users' real-time click feedback is a better and more accurate judge for the varying relevance of documents. Knowing how to reflect the drift quickly and therefore generate better rankings is a key challenge.

Online learning is an excellent alternative to improve the quality of results based on users' click feedback.

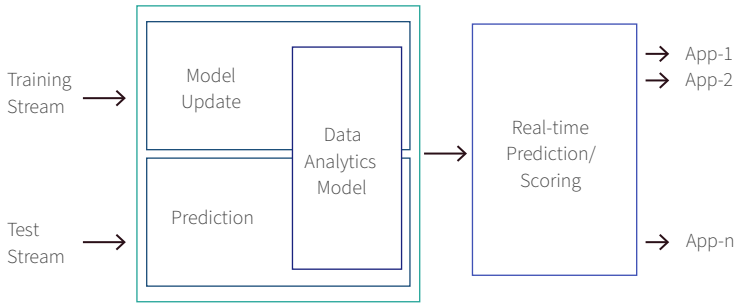

Figure 3: Online Modeling- Data Analytics model gets updated with each new test instance

Offline vs. Online Modeling in Streaming Analytics

The choice between the two modeling approaches depends on two main factors:

Data Horizon

- How quickly do you need the most recent data point to become part of your model?
- Does the next point need to modify the model immediately?

Online modeling is more favorable than offline modeling in environments where there is a shorter data horizon and in which decisions on the new streaming data has a high dependency on recent streaming data. In other words, if your answer is "I need it to learn from what happened just now," then online modeling would better suit your needs.

Data Obsolescence

This factor indicates whether data is evolving over time; if it is evolving over time, then online learning is a good choice. Offline modeling does not handle the changing nature of the data immediately and can therefore result in undesired results if the data is fluctuating.

These factors play a critical role in achieving accurate modeling results. However, there are a few secondary factors which are also important to consider from an architectural and implementation perspective.

Computational and Memory Space Requirement

Offline modeling builds models on sizable training data sets and requires a greater computational footprint with higher memory than online modeling where a single data example can be used for model initialization and subsequent updates.

Model Design and Accuracy

Since the online model makes one pass over the data in a sequential manner, it usually simplifies the model design complexity. On the other hand, offline modeling gets a larger view of the data which generally results in higher accuracy.

Fault Tolerance

Deploying online models in production typically requires constant model updates. Managing the version control of models under various faulty circumstances such as network latency, server/network failure, etc., poses implementation challenges. In contrast, offline models are deployed one at a time with periodic updates after a finite time interval; therefore, fault tolerance is much higher.

Performance Evaluation

By design, online modeling does not have "test" data sets, unlike offline learning. Here, the model performance can only be known for a data example at a given point of time.

There have been some attempts of realizing a hybrid approach in which incremental update of the model is performed based on a smaller set of relevant data streams (mini-batch) unlike one data example in online modeling. The creation of mini-batch data is done by buffering relevant data streams filtered by some relevant criteria or rules. Moreover, some business policies such as quantitative deviation in the model performance, etc., can be applied to trigger a re-training event with buffered mini-batch data streams.

New Challenges Posed by Streaming Analytics

Streaming analytics has great potential to help businesses achieve short and long term business goals. However, designing and delivering such applications can require a significant effort. In order to build an effective streaming analytics solution, data scientists and big data professionals need to address several challenges:

Real-time Transfer of Stream Data

Stream data, such as events, change, activity logs, etc., needs to be transferred in real-time.

- Data can be transferred from distributed sources as raw events or as filtered or aggregated events.
- All generated data can be transferred to a centralized processing point or to distributed intermediate processing points for pre-processing before being transferred to the streaming analytical models.

In a real-time analytical approach, event filtering and aggregation should be done to reduce processing time without negatively affecting the accuracy and optimality of the decision making.

Real-time Analytics

Real-time analytics may involve single or multiple integrated analytical models. Inter-model interactions and data abstraction for effective modeling in the streaming environment pose a new set of modeling challenges to data scientists.

Automated vs. Human Involved Decision Making

Despite consuming, transferring and analyzing data quickly, it is crucial to factor in human feedback received on the analytical outcome. The role of human feedback is essential to enhance the capabilities of the underlying model. However, incorporating human feedback into the analytical model design adds a new dimension of complexity to streaming analytics applications.

Gathr Can Help

Gathr is an enterprise class real-time streaming analytics platform based on a best-of-breed Open Source stack that can help organizations across industry verticals to quickly and reliably take into production a wide range of streaming data applications.

It enables use cases in areas such as the Internet of Things (IoT), sensor data analytics, e-commerce and Internet advertising, security, fraud, insurance claim validation, credit-line-management, call center analytics and log analytics. It also enables enterprise IT and business transformation with horizontal capabilities like Streaming ETL to speed up slow batch processes to near-real-time.

An analytical model can be developed on the Gathr platform or transferred from other model-building tools and environments. The deployment of any such model over live data streams is seamless with Gathr.

Gathr enables data scientists to demonstrate business insights via visualization of the model results in real-time, especially on continuous streaming data. Business analysts can better understand the range of possibilities with streaming analytical models in real-time.

With Gathr, you can develop and deploy analytical models seamlessly with advanced analytics libraries and tools such as R, Spark MLib and Spark ML to build production-level models in both online and offline mode.

Additionally, Gathr lifts restrictions on model building tools because you can use external tools such as R, SAS, SPSS and others and export them as a PMML file. In other words, the model from any of these tools can be imported and deployed in Gathr for run-time scoring, regardless of how and where you built the analytical model.

The Gathr Platform

Gathr is a platform to build and deploy streaming analytics applications for any industry vertical, any data format, and any use case.

Gathr provides a level of abstraction that allows for the deployment of multiple streaming engines depending on the use-case requirements. This affords customers a new level of "best-of-breed" flexibility in their real-time architecture.

With Gathr, you can use the visual IDE and an enhanced set of powerful stream processing operators to easily construct data pipelines in a matter of minutes. You can then deploy them to a stream processing engine of choice.

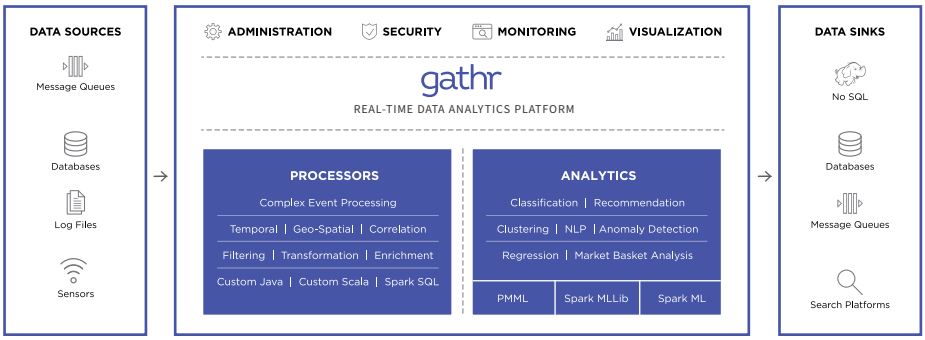

Figure 4: Gathr at a Glance

It also enables enterprise IT and business transformation with horizontal capabilities like Streaming ETL to speed up slow batch processes to near-real-time.

Additionally, Gathr provides the following:

Support for Spark Streaming

A rich array of drag-and-drop Spark data transformations including Machine Learning operations to analyze data using SQL queries and save the query output in a data store of choice. Built-in operators for predictive models with inline model-test features and graphs to visually analyze data for models like Neural Networks and Tree.

Proven Open Source Stack

Ingest, store, and analyze millions of events per second with a pre-integrated package of industry-preferred open source components: Hadoop, NoSQL, Kafka,RabbitMQ, Apache Storm, Elastic Search, and Apache Solr.

Visual Performance Monitoring

Monitor performance of running applications and their underlying compute components visually through graphs. Set alerts to get real-time notification on threshold breaches.

Rapid App Development

Integrate custom applications into the real-time data pipeline by visual drag and drop. Rapidly port predictive analytics and machine learning models built in SAS or R via PMML onto real-time data.

Open, Flexible, and Extensible

Use any fast-ingest data store of your choice. Bring in any number of proprietary or standard data sources. Integrate the real-time data pipeline with other existing applications, based on configurable conditions.

Data Lineage

Track the progress of data, capture and record data changes at every stage in the pipeline. Ability to search and view the entire path of any incoming message through the pipeline, including the time taken for processing at each stage, changes to the data attributes.

Online Debug

Visually examine the health of an application and all the pipeline components through multiple graphs, analyze logs to debug issues, and define alerts for real-time notifications.

Easily build fast and reliable data pipelines using Gathr