UNIT III MINING DATA STREAMS **9**

Introduction To Streams Concepts – Stream Data Model and Architecture - Stream Computing - Sampling Data in a Stream – Filtering Streams – Counting Distinct Elements in a Stream – Estimating Moments – Counting Oneness in a Window – Decaying Window - Real time Analytics Platform(RTAP) Applications - Case Studies - Real Time Sentiment Analysis, Stock Market Predictions

---

DECAYING WINDOWS

We have assumed that a sliding window held a certain tail of the stream, either the most recent N elements for fixed N, or all the elements that arrived after some time in the past. Sometimes we do not want to make a sharp distinction between recent elements and those in the distant past, but want to weigh the recent elements more heavily. In this section, we consider "exponentially decaying windows," and an application where they are quite useful: finding the most common "recent" elements.

The Problem of Most-Common Elements

Suppose we have a stream whose elements are the movie tickets purchased all over the world, with the name of the movie as part of the element. We want to keep a summary of the stream that is the most popular movies "currently." While the notion of "currently" is imprecise, intuitively, we want to discount the popularity of a movie like Star Wars–Episode 4, which sold many tickets, but most of these were sold decades ago. On the other hand, a movie that sold n tickets in each of the last 10 weeks is probably more popular than a movie that sold 2n tickets last week but nothing in previous weeks. One solution would be to imagine a bit stream for each movie. The ith bit has value 1 if the ith ticket is for that movie, and 0 otherwise. Pick a window size N, which is the number of most recent tickets that would be considered in evaluating popularity. Then, use the method of Section 4.6 to estimate the number of tickets for each movie, and rank movies by their estimated counts. This technique might work for movies, because there are only thousands of movies, but it would fail if we were instead recording the popularity of items sold at Amazon, or the rate at which different Twitter-users tweet, because there are too many Amazon products and too many tweeters. Further, it only offers approximate answers.

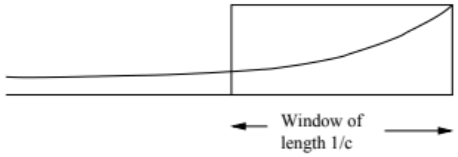
Definition of the Decaying Window

An alternative approach is to redefine the question so that we are not asking for a count of 1's in a window. Rather, let us compute a smooth aggregation of all the 1's ever seen in the stream, with decaying weights, so the further back in the stream, the less weight is given. Formally, let a stream currently consist of the elements a1, a2, . . . , at, where a1 is

the first element to arrive and at is the current element. Let c be a small constant, such as 10−6 or 10−9. Define the exponentially decaying window for this stream to be the sum

$$\sum_{i=0}^{t-1} a_{t-i}(1-c)^i$$

The effect of this definition is to spread out the weights of the stream elements as far back in time as the stream goes. In contrast, a fixed window with the same sum of the weights, 1/c, would put equal weight 1 on each of the most recent 1/c elements to arrive and weight 0 on all previous elements. The distinction is suggested by Fig. 4.4.



It is much easier to adjust the sum in an exponentially decaying window than in a sliding window of fixed length. In the sliding window, we have to worry about the element that falls out of the window each time a new element arrives. That forces us to keep the exact elements along with the sum, or to use an approximation scheme such as DGIM. However, when a new element at+1 arrives at the stream input, all we need to do is:

1. Multiply the current sum by 1 − c.
2. Add at+1.

The reason this method works is that each of the previous elements has now moved one position further from the current element, so its weight is multiplied by 1 − c. Further, the weight on the current element is (1 − c) 0 = 1, so adding at+1 is the correct way to include the new element's contribution.

Finding the Most Popular Elements

Let us return to the problem of finding the most popular movies in a stream of ticket sales.6 We shall use an exponentially decaying window with a constant c, which you might think of as 10−9. That is, we approximate a sliding window holding the last one billion ticket sales. For each movie, we imagine a separate stream with a 1 each time a ticket for that movie appears in the stream, and a 0 each time a ticket for some other movie arrives. The decaying sum of the 1's measures the current popularity of the movie.

We imagine that the number of possible movies in the stream is huge, so we do not want to record values for the unpopular movies. Therefore, we establish a threshold, say 1/2, so that if the popularity score for a movie goes below this number, its score is dropped from the counting. For reasons that will become obvious, the threshold must be less than 1, although it can be any number less than 1. When a new ticket arrives on the stream, do the following:

1. For each movie whose score we are currently maintaining, multiply its score by (1 − c).
2. Suppose the new ticket is for movie M. If there is currently a score for M, add 1 to that score. If there is no score for M, create one and initialize it to 1.
3. If any score is below the threshold 1/2, drop that score.

It may not be obvious that the number of movies whose scores are maintained at any time is limited. However, note that the sum of all scores is 1/c. There cannot be more than 2/c movies with a score of 1/2 or more, or else the sum of the scores would exceed 1/c. Thus, 2/c is a limit on the number of movies being counted at any time. Of course in practice, the ticket sales would be concentrated on only a small number of movies at any time, so the number of actively counted movies would be much less than 2/c.

REAL TIME ANALYTICAL PLATFORM

Real-time analytics in Big Data provides the ability to extract useful insights quickly from massive datasets. Real-time analytics stands at the forefront of this transformation, enabling organizations to analyze data streams as they are generated, rather than relying on historical analysis alone. This capability not only enhances decision-making processes but also empowers businesses to respond dynamically to changing market conditions, customer behaviors, and operational challenges.

Real time analytics makes use of all available data and resources when they are needed. It consists of dynamic analysis and reporting based on the entered on to a system less than one minute before the actual time of use. Real time denotes the ability to process as it arrives, rather than storing the data and retrieving it at some point in the future. Real time analytics is thus delivering meaningful patterns in the data for something urgent.

Real-time analytics involves continuously processing and analyzing data immediately as it is collected, enabling instant decision-making and responses. businesses to get awareness and take action on data immediately or soon after the data enters their system. Real-time app analytics respond to queries within seconds. They grasp a large amount of data with high velocity and low reaction time. For example, real-time big data analytics uses data in financial databases to notify trading decisions. Analytics can be on-demand or uninterrupted. On-demand notifies results when the user requests it. Continuous renovation users as events happen and can be programmed to answer automatically to certain events.

Examples of real-time customer analytics include the following.

1. Viewing orders as they happen for better tracing and to identify fashion.

2. Continually modernize customer activity like page views and shopping cart use to understand user etiquette.
3. Choose customers with advancement as they shop for items in a store, affecting real-time decisions.

Types of real time analytics

On Demand Real Time Analytics – It is reactive because it waits for users to request a query and then delivers the analytics. This is used when someone within a company needs to take a pulse on what is happening right this minute.

Continuous Real Time Analytics – It is more proactive and alerts users with continuous updates in real time. Example: Monitoring stock market trends provides analytics to help users make a decision to buy or sell all in real time.

Real-Time Analytics – working

Real-time analytics involves a comprehensive and intricate process that encompasses several critical components and steps. Here's a more detailed breakdown of how it operates:

1. Data Ingestion

Continuous Data Collection: Real-time analytics systems continuously collect data from various sources, such as sensors, IoT devices, social media feeds, transaction logs, and application databases. This data can come in various formats, including structured, semi-structured, and unstructured data.

Stream Processing: Data is ingested as streams, meaning it is captured and processed in real-time as it arrives. Technologies like Apache Kafka, RabbitMQ, and Amazon Kinesis are commonly used for data ingestion due to their ability to handle high-throughput data streams reliably.

2. Data Processing Engines

Stream Processing Platforms: Once ingested, the data is processed by stream processing engines such as Apache Flink, Apache Storm, or Spark Streaming. These platforms are designed to handle continuous data flows and perform complex event processing, transformations, aggregations, and filtering in real-time.

- **In-Memory Processing:** To ensure low-latency processing, many real-time analytics solutions use in-memory computing frameworks. This allows data to be processed directly in memory rather than being written to disk, significantly speeding up the processing time.
- **Parallel Processing:** Real-time analytics systems often employ parallel processing techniques, distributing the workload across multiple nodes or processors to handle large volumes of data efficiently.

3. Real-Time Querying

- **Low-Latency Query Engines:** Real-time query engines like Apache Druid, ClickHouse, and Amazon Redshift Spectrum allow users to run queries on streaming data with minimal delay. These engines are optimized for low-latency query execution, providing near-instantaneous results.
- **Complex Queries:** Users can perform complex queries and analytical operations on streaming data, such as joins, aggregations, window functions, and pattern matching, enabling sophisticated real-time analysis.

4. Data Storage

- **Time-Series Databases:** Real-time analytics often involves storing data in time-series databases like InfluxDB, TimescaleDB, or OpenTSDB. These databases are optimized for handling time-stamped data and can efficiently store and retrieve real-time data points.
- **NoSQL Databases:** For unstructured or semi-structured data, NoSQL databases like MongoDB, Cassandra, and HBase provide flexible storage solutions that can scale horizontally to accommodate large data volumes.

5. Visualization Tools

- **Dashboards and BI Tools:** Real-time data is visualized using dashboards and business intelligence (BI) tools like Tableau, Power BI, Grafana, and Kibana. These tools provide interactive and customizable visualizations that allow users to monitor and analyze data in real-time.
- **Alerts and Notifications:** Real-time analytics systems can be configured to trigger alerts and notifications based on predefined conditions or thresholds. This enables proactive responses to critical events, such as system failures, security breaches, or significant business metrics.

Real Time Analytics Applications

- **Predictive Maintenance:** Manufacturing, utilities, and transportation sectors utilize real-time analytics to monitor equipment health and predict failures before they occur.
- **Fraud Detection:** Financial services, e-commerce platforms, and insurance companies continuously monitor transactions and user behavior, to identify anomalies and take immediate action to mitigate fraud risks.
- **Customer Experience Management:** Retailers, hospitality providers, and online services analyze customer interactions and feedback in real-time, businesses can personalize services, optimize marketing campaigns, and promptly address customer issues, leading to higher satisfaction and loyalty.
- **Smart Cities:** Urban planners and city administrations employ real-time analytics in traffic management, public transportation optimization, and real-time monitoring of public safety and environmental conditions.

- **Healthcare:** Healthcare providers use real-time analytics to monitor patient vitals, manage hospital resources, and provide timely interventions. For instance, real-time analysis of patient data can alert medical staff to potential emergencies, improving patient outcomes and operational efficiency.
- **Financial Trading:** Financial institutions and traders rely on real-time analytics to make quick, informed trading decisions. By analyzing market data as it happens, traders can identify trends, detect anomalies, and execute trades at the optimal moment to maximize profits.
- **Supply Chain Management:** Logistics and supply chain companies use real-time analytics to track shipments, manage inventory, and optimize delivery routes. This ensures timely deliveries, reduces costs, and improves overall supply chain efficiency.
- **Telecommunications:** Telecom operators use real-time analytics to monitor network performance, detect outages, and manage bandwidth. This helps in maintaining service quality, reducing downtime, and enhancing customer satisfaction.
- **Energy Management:** Utility companies and large enterprises employ real-time analytics for energy consumption monitoring and optimization. By analyzing real-time data from smart meters and sensors, businesses can optimize energy usage, reduce costs, and support sustainability initiatives.
- **Marketing and Advertising:** Marketers and advertisers use real-time analytics to measure the effectiveness of campaigns and adjust strategies on the fly. Real-time insights into customer behavior and engagement help in creating targeted and impactful marketing efforts.
- **Retail and E-commerce:** Retailers and e-commerce platforms leverage real-time analytics to manage inventory, optimize pricing strategies, and enhance the shopping experience. Analyzing real-time sales data and customer interactions helps in making informed decisions that drive sales and improve customer satisfaction.

Generic Design of an RTAP

Companies like Facebook and twitter generate petabytes of real time data. This data must be harnessed to provide real time analytics to make better business decisions. Today Billions of devices are already connected to the internet with more connecting everyday. Real time analytics will leverage information from all these devices to apply analytics algorithms and generate automated actions within milliseconds of a trigger. Real time analytics needed the following aspects of data flow,

- Input - An event happens (New sale, new customer, someone enters a high security zone etc.)
- Process and Store Input – Capture the data of the event, and analyze the data without leveraging resources that are dedicated to operations.

- Output – Consume this data without distributing operations

The following key capabilities must be provided by any analytical platform

- Delivering in Memory TransactionSpeed
- Quickly Moving Unneeded data to disk for long term storage
- Distributing data and processing for speed
- Supporting continuous queries for real time events
- Embedding data into apps or apps into database
- Additional Requirements

Many technologies support real time analytics, they are,

- Processing in memory
- In database analytics
- Data warehouse applications
- In memory analytics
- Massive parallel programming

REAL TIME SENTIMENT ANALYSIS

Sentiment analysis, also known as opinion mining, refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials.

Sentiment analysis is widely applied to reviews and social media for a variety of applications ranging from marketing to customer service. A basic task in sentiment analysis is classifying the polarity of a given text at the document, sentence, or feature/aspect level where the expressed opinion in a document, a sentence or an entity feature aspect is positive, negative or neural.

Applications

News media websites are interested in getting an edge over its competitors by featuring site content that is immediately relevant to its readers. They use social media analysis topics relevant to their readers by doing real time sentiment analysis on twitter data. Specifically to identify what topics are trending in real time on twitter.

Twitter has become a central site where people express their opinions and views on political parties and candidates. Emerging events or news or often followed almost instantly by a burst in twitter volume which if analyzed in real time can help explore how these events affect public opinion While traditional content analytics takes days or weeks to complete, RSTA can look into entire content about elections and deliver results instantly and continuously.

Ad agencies can track the crowd sentiment during commercial viewing on TV and decide which commercials are resulting in positive sentiment and which are not.

Analyzing sentiments of messages posted to social media or online forums can generate countless business values for the organizations, which aims to extract timely business intelligence about how their products or services are perceived by their customers. As a result proactive marketing or product design strategies can be developed to efficiently increase the customer base.

Tools

- **Apache Storm** is a distributed real time computation system for processing large volumes of data. It is part of Hadoop. Strom is extremely fast with the ability to process over a million records per second per node on a cluster of modest size.
- **Apache Solr** is another tool from Hadoop which provides a highly reliable scalable search engine facility at real time.
- **RADAR** is a software solution for retailers built using a Natural language processing based sentiment analysis engine and utilizing Hadoop technologies including HDFS, YARN, Apache Storm, Apache Solr, Oozie and Zookeeper to help them maximize sales through database continuous repricing.

Online retailers can track the following for the number of products in their portfolio,

a. Social sentiment for each product
b. Competitive pricing / promotions being offered in social media and on the web.

REAL TIME STOCK PREDICTION

Traditional stock market prediction algorithms check historical stock prices and try to predict the future using different models. But in real time scenario stock market trends continually change economic forces, new products, competition, world events, regulations and even tweets are all factors to affect stock prices. Thus real time analytics to predict stock prices is the need of the hour. A general real time stock prediction and machine learning architecture comprises three basic components,

a. Incoming real time trading data must be captured and stored becoming historical data.
b. The system must be able to learn from historical trends in the data and recognize patterns and probabilities to inform decisions.
c. The system needs to do a real time comparison of new incoming trading data with the learned patterns and probabilities based on historical data. Then it predicts an outcome and determines an action to take.

For Example consider the following, Live data from Yahoo Finance or any other finance news RSS feed is real and processed. The data is then stored in memory with a fast consistent resilient and linearly scalable system. Using the live hot data from Apache Geode, a Spark MLib application creates and trains a model computing new data to

historical patterns. The models could also be supported by other toolsets such as Apache MADlib or R.

Results of the machine-learning model are pushed to other interested applications and also updated within Apache Geode for real time prediction and decisioning. As data ages and starts to become cool it is moved from Apache Geode to Apache HAWQ and eventually lands in Apache Hadoop. Apache HAWQ allows for SQL based analysis on petabyte scale data sets and allows data scientists to iterate on and improve models. Another process is triggered to periodically retain and update the machine learning model based on the whole historical data set. This closes the loop and creates ongoing updates and improvements when historical patterns change or as new models emerge.