BIG
DATA

Using the Lambda Architecture on a Big Data Platform to Improve Mobile Campaign Management

Author: **Sandesh Deshmane**

talentica
SPREAD YOUR WINGS

Executive Summary

Growing data volumes and real time decision making requirements are making traditionally used processing systems redundant. Increasingly, businesses are leveraging Big Data technologies for enabling high impact decision making in real time.

In the case of mobile campaign management, high data volumes and velocities pose a data processing challenge. The lack of real time views on campaign performance often leads to over or under delivering campaigns. Often, campaigns are run for shorter durations and over delivering entails heavy losses in terms of revenues and missed opportunities.

This paper talks about the Lambda architecture- a combination of batch and real time processing capabilities. And how this was used as a solution on a big data platform, to enable real time decision making for mobile ad campaign management.

Background

Traditionally, batch jobs were used to update dashboards meant for monitoring or decision making purposes. In a batch processing system, data is processed in batches set at specified intervals, and then made available to users.

To handle the continuously growing volumes of data, batch processing technologies have evolved from Java based ones to Hadoop based big data ones. However, since processing is done in batches, there are large time lags between the data arriving onto the platform and being displayed to users. Also, if the velocity of data generation is high, each batch takes longer to process, leading to further delays in generating reports. While this issue is easily overcome by adding more servers to the system, it is a costly thing to do.

“In a highly dynamic business world, the inability to take strategic decisions almost instantly leads to heavy losses in terms of revenues and business opportunities. In turn, making real time processing capabilities crucial to have.”

Large time ranges have considerable impact on business decisions, often leading to heavy revenue losses and missed opportunities. Real time data processing tools help in overcoming this issue. Real time processing systems continuously process data as it arrives. Whenever data has to be viewed, reports are generated based on the most recent data. This makes monitoring and decision making more precise than in the case of a batch processing system.

The issue with real time processing, is that factors such as delayed events, code changes, or system failures, call for re-processing the time ranges and then re-computing the metrics. Making it a time and resource consuming affair.

Citing the example we have used for this paper. Advertising campaigns have become commonplace on the mobile platform. These campaigns are usually run for short durations. For campaigns to be successful, their perfomance needs to be constantly monitored and priorities adjusted accordingly.

If campaigns under deliver, advertisers lose visibility. However, if they over deliver, platforms lose revenue. Reprioritizing campaign objectives based on historic data often leads to under or over delivering campaigns. To ensure this does not happen real time reporting is essential.

In our case batch jobs were being used for processing campaign data. With over 500 GB data being generated per hour at almost 200k messages/second, campaign priorities took a while to calculate. In most cases, the campaigns would over deliver in terms of the number of impressions, leading to considerable losses in terms of revenue.

To minimize the revenue loss, campaign performance reports needed to be generated in real time.

Dealing with Volume and Velocity

In our case, the volume of data generated and the speed at which data was generated were considerably high. To ensure success of the campaigns, data processing needed to factor in both the volume and velocity aspects of the data.

Using a Big Data based batch processing system, large data sets could be easily handled. Depending on the batch intervals, large volumes of data could be easily processed, along with scope for managing incremental increases. However, at the high velocity this data was being generated, the time taken for processing each batch would increase, adding to the delay in generating reports. A workaround would entail an expensive addition of servers.

"The lambda architecture is an malgamation of batch and real time processing. It provides a solution where real time decision making is enabled for critical areas and cost efficiency is maintained by not making the entire processing happen in real time."

Handling high velocity data required a real-time processing tool that compensated for the high-latency of batch systems. However, using such a system alone would mean risking time consuming and costly reprocessing whenever delayed events, code changes, or system failures occurred.

Since, neither batch nor real time systems alone, could handle the volume and velocity conundrum. We looked at a solution where both the systems worked in tandem. The batch component would store the master data set and batch process all the metrics. The real time component would continuously process recent data and integrate the old data into the batch.

Lambda Architecture

Using a combination of batch and real time processing systems in parallel is an approach which businesses are increasingly exploring now. While the batch layer ensures scalability and fault resilience for the system, the real time layer takes care of processing metrics required on the go.

Nathan Marz coined the term Lambda Architecture (LA) to describe a generic, scalable and fault-tolerant data processing architecture.

Data that enters the system is dispatched to the batch and the speed layers for simultaneous processing.

The batch layer serves two functions:

- Managing the master dataset (an immutable, append-only set of raw data)
- Pre-computing the batch views.

The serving layer indexes the batch views so that they can be queried in a low-latency and ad-hoc manner.

The speed layer compensates for the high latency of updates to the serving layer and deals only with recent data.

Any incoming query can be answered by merging the results from both batch and real-time views.

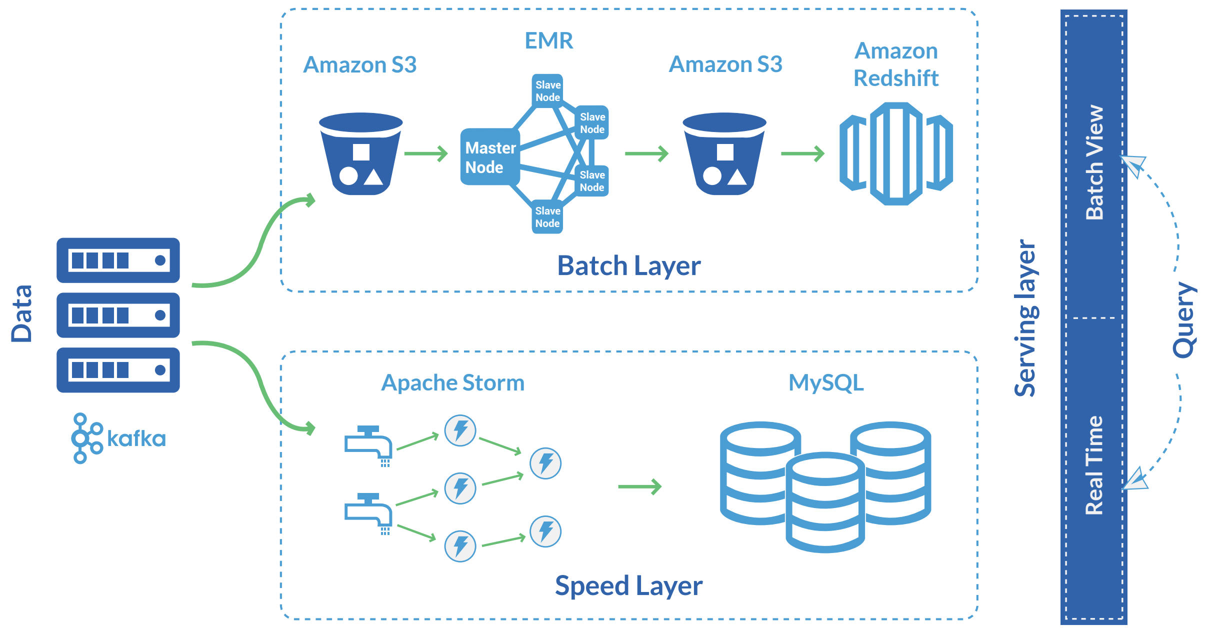
Solution

Overview

To process data at almost 200k messages / second, high velocity receivers and storage systems were required for processing and storing incoming records. Kafka proved to be the best choice as a message queue. Apache Storm Stream and Redshift seemed best suited to manage real time data and data storage respectively. For processing historic data and generating adhoc reports (service layer), batch-processing services like Amazon Elastic MapReduce, PIG and Apache Spark were used.

The Architecture

The figure below provides an overview of how the lambda architecture looks like.



Components

Batch layer

The Kafka messaging queue passes the log messages to Amazon S3 where they are stored as the master dataset. These logs are then batch processed on Amazon Elastic MapReduce using PIG scripts, MapReduce and Apache Spark. The metrics derived from this processing are loaded into the Amazon Redshift Data warehouse and are made available as key value stores for to visualization / reporting tools such as tableau.

Speed Layer

In order to generate real time reports, the Kafka messaging queue passes the logs to the Apache Storm Cluster via a Storm Topology. The storm topology reads the events from the Kafka Queue with the help of Kafka Spouts. Stream joins, data aggregations, and transformations are carried out using Storm bolts. The metrics derived are calculated on the cluster and stored in MySQL. A majority of the issues related to concurrent updates are managed on the speed layer where regular purging makes the data sizes smaller. In turn, helping to reduce the processing complexity.

Message Queue
Kafka

Batch Layer
Amazon S3, EMR, MapReduce, Apache Spark, Amazon Redshift

Speed Layer
Apache Storm, MySQL

Serving Layer

Views get created separately on batch as well as real time data. The serving layer is responsible for merging the views created on batch and real time layers. The merged views are then used for generating reports. Real time views are transient in nature show only the most recent data, older data is discarded once it passes through the serving layer and is stored progressively in the batch layer.

Error Recovery

The lambda architecture enables error rectification by allowing the views to be re-computed. If this seems time consuming in a particular case, we can simply revert to the previous, non-corrupted version of the data. Doing this was possible since the data in the master dataset was immutable - could not be altered after being created. The data in the master dataset does not get updated and is only appended to (time-based ordering). This makes for a Human Fault Tolerant System where bad data can be completely removed and re-computation can be easily done.

Conclusion

Advantages

- Input data retained without any changes
- Data can be reprocessed to re-derive an output
- Obeys the CAP theorem

Disadvantages

- Maintaining a single code to produces the same result in two complex distributed systems
- Heavy operational load of running and debugging two systems
- Latencies caused during merging data at theserving layer

In the case of mobile campaign management, the Lambda architecture works as a cost effective and scalable solution. The speed layer uses most recent data for real time processing. The batch layer maintains the master dataset for non real time processing. This framework helps churn out real time output where needed while ensuring that too much data does not remain in the real time processing system at any given point of time.

About Talentica

Talentica Software is an innovative outsourced product development company that helps startups build their own products. We help technology companies transform their ideas into successful products by partnering in their roadmap from pre-funded startups to a profitable acquisition.

We have successfully built core intellectual property for more than 60 customers so far. We have the deep technological expertise, proven track record and unique methodology to build products successfully. Our customers include some of the most innovative product companies across USA, Europe and India.

Office No. 501, Amar Megaplex
Baner, Pune 411045

Tel: +91 20 4660 4000 | Fax: +91 20 4075 6699

www.talentica.com