Case Study

Sai Hemanth Nirujogi
G01065588.

Spotify

“Spotify is a Data-driven music, podcast and radio streaming service that was launched on 7th of october 2008. It was first developed by a startup Spotify AB in stockholm, Sweden. It provides digital rights management protected content from record labels and media company. It currently has over 140 million registered active users, of which 60 million are paid subscribers. With over 20 million songs online and over 20,000 being added to its database every day. Spotify by the fiscal year 2015 has a revenue of $2.8 billion”. [1]

Most of the massive and growing data in Spotify is user driven, which helps the organization to give music recommendations, personalized radio and concert dates of the most played artist.

Spotify would not have been what it is today without the use Big data. With a developing user presence in many places and growing audience, there will be a significant surge of data in the near future.

Goal of Spotify

The main goal of Spotify is access to all the world’s music at any time and place. By the time Spotify started there are vast majority of people who consume their music via Pirates bay, Kazaa, Napster and other piracy websites. At the same time, musicians were having problems with making money with their work due to this. To promote their music they have to tour and do different kinds of things. The only way for the people to access music legally by then is to buy music from iTunes. But, the access is limited to apple devices and quality of music (160Kb) can be pretty much available for lossless access in any piracy websites like Pirates bay and Kazaa. Spotify made discovering and sharing

music easy, which allows the users to access the music simply through their personal devices and better than accessing pirated music. This also helps the music industry and artists to keep the growth of music consistent.

“The future of music is access, not ownership” [3] - Daniel Ek, CEO of Spotify.

Spotify has undergone three stages to become the world’s leading music streaming service. The stages are described throughout the years, [3]

1. Utility - 2008
2. Curation - 2013
3. Personalization - 2015

Utility

Spotify’s aim in the beginning is only one thing and that is access to all the world’s music. It has given users access to an unlimited music library right at their fingertips. At that time spotify is not really working on personalised music, for some users it is a deal breaker.

Curation

Spotify was not able to give users personalised playlists and radio even though they know the data about the user like artist, genre and mood. Spotify started to create different playlists designed by music specialists by the listening habits of the users.

Personalization

In 2015, Spotify changed the game by introducing “Discover Weekly”, this furnishes a personalised playlist every week on monday. This playlist consists of 2 hours of music generated using Spotify’s algorithms from the gathered data. This became a hit in no time, because the algorithm understands what the users are listening to and delivers the next song using the same data.

Spotify has more than 30 million songs in their database, with over 20,000 new songs added to the database everyday, 1 billion playlists and 60 million registered active users

this organization would not exist without Big Data. As a data driven company, Spotify's huge numbers in every aspect are growing everyday.

- A massive 600 GB of data is created daily and 150 GB of data from various sources by users at Spotify.
- 4 Terabytes of data is generated every single day in Hadoop and a 700 node cluster running over 1600+ jobs daily.
- Spotify has over 28 Petabytes of data distributed across 4 data centers across the globe. [3]

Data Sources

The Spotify ecosystem uses Apache Crunch to process terabytes of user generated data every day. The playlists created, artists followed and music shared by users are organized and processed by Spotify into significant data out to the other side.

The following information can be obtained from the data:

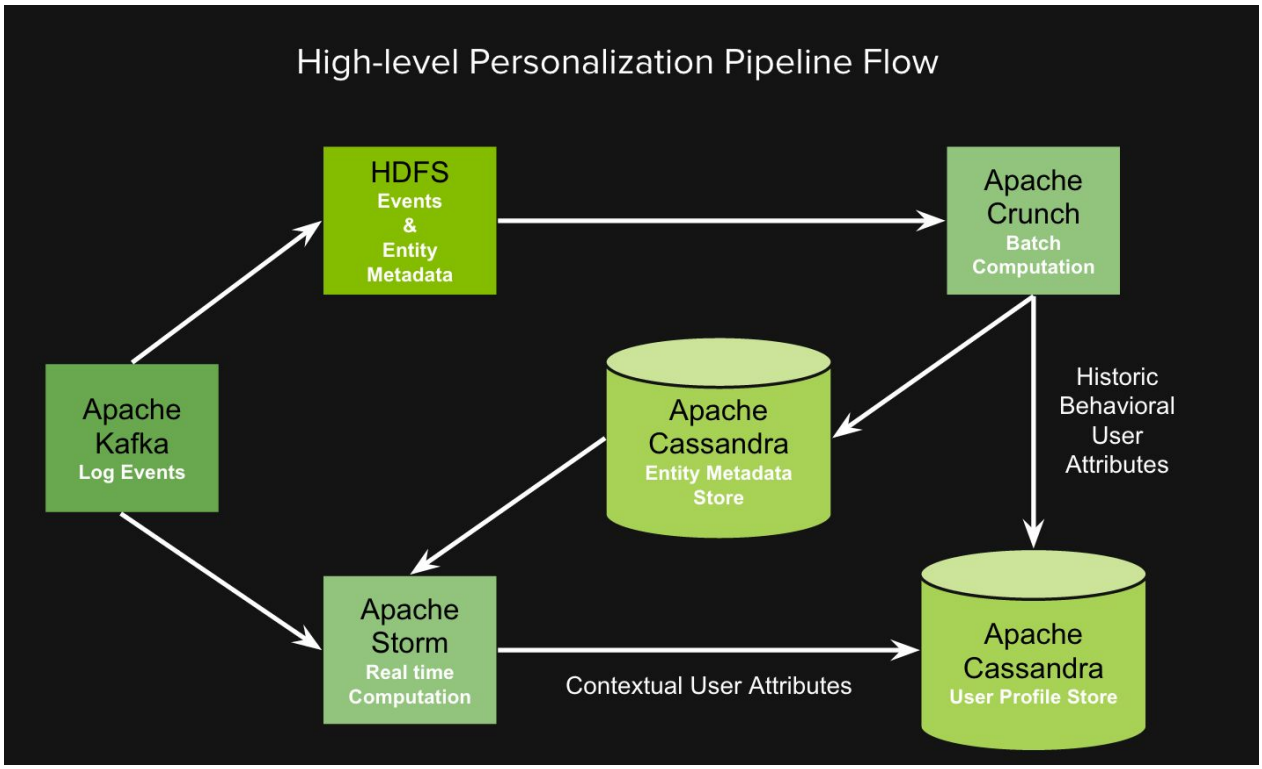
- Creating popular music charts of the music that is trending around the world.
- Reporting to record labels, artists and rights holders so they can make sure everyone gets paid.
- Discover weekly and intelligent radio can be powered up.
- Receiving feedback on how different features of the product are working so they can enhance the user experience.

Spotify uses Hadoop framework to process and store all the huge data collected. Since 2009, Hadoop is been part of the Spotify's developer branch helping developers and analysts to integrate the data by MapReduce approach in Python. Luigi is Spotify's python module that helped them to create complicated pipelines of jobs. This controls workflow management, visualization and etc. Python code over Hadoop streaming API does not seem to be working very well for Spotify, huge data sets and volumes of data keeps streaming through the database, jobs are hard to complete and they experienced

a lot of runtime errors. Using Apache Crunch on top of the existing infrastructure helped the organisation in abstraction for parallelisable data processing.

Crunch, a framework implemented on top of hadoop for writing, testing, and running MapReduce pipelines. Goal of Apache Crunch is to make pipelines that are made of many user-defined functions that are easy to write, simple to test and efficient to run.

Personalizing user experience involves learning their tastes and distastes in different contexts. [2] Spotify had an insight to build a system that personalizes and analyzes both real-time and history data like user listening habits and context. Over time they have created a personalization engine using Kafta for log collection, Storm for real time event processing, for running MapReduce jobs on Hadoop they used Crunch and Cassandra to gather profile attributes of users and playlist, artist and radio metadata. There are different Crunch pipelines which are used generate metadata for new entities. Volumes of data that is recorded is stored in HDFS and uses Crunch to Cassandra to export for real-time findings in Storm pipelines. The Cassandra cluster which stores the metadata of different entities is called Entity Metadata Store.



[2] High-level personalization Pipeline Flow

Above image shows the pipeline flow of the personalization engine, raw log events from Kafka are processed, unwanted events are filtered, entities with metadata is decorated and user level attributes from algorithms are determined by Storm pipelines. When combining these user attributes a user profile is represented and is stored in the Cassandra Cluster, which is called User Profile Store. UPS is the central to the Spotify's personalisation system. Cassandra is optimal for data storage because

- Replication of data through cross-site is supported.
- Horizontal scaling
- Ability to load bulk and streaming data from Crunch and Storm.
- Low latency even at the cost of consistency
- Ability to model different data schemas for different use-cases of entity metadata.

Challenges

With the consecutive three year growth in music industry, Spotify has seen a massive revenues in these years. Even with huge numbers like that spotify still has to convince it's investors for future profitability. In the first half of 2017, Spotify has reportedly made $2.2 billion which is 70 percent of what it has made in the whole year of 2016. With that kind of revenue the organization will be making $4.4 billion by the end of the year. This kind of projections can increase the subscriber number by 40 percent than the last year. Still Spotify has significant operating losses, which has an estimate of 100 million to 200 million dollars in the first half of 2017. Investors are expecting spotify to be earning profits because of the new businesses in market like Apple, Beats, Sony and Google, which may not as popular as Spotify (except for Apple) may grab the user attention due to this.

Partnerships with other leading network operators like Sprint has made Spotify to appeal to a more mainstream audience with a better monthly price of subscription. But, rivals like Apple and Google Play are aggressive at this partnerships, but offers free subscriptions for more than what Spotify's offering. Spotify with it's playlists recommendations, personal radio and discover alone can help the company to raise the

bar comparing to its rivals. Raising fueled up competition in music industry has been giving the company hard time in appealing to a larger set of audience. Spotify has reached its goal by becoming the most accessible and music tech streaming service in the world.

Spotify has faced many technical challenges since it has started. A crucial one being keeping up with the data that is being generated by the users every day. It is using Hadoop framework for data management since 2009, with an open source workflow manager called Luigi. It is a Python framework on top of hadoop used for executing, defining and crunching loads of data. In the year 2013, the organization found out that running a python framework on top of Hadoop cluster is not keeping up with the velocity of data that has been streaming into Spotify's data base. The execution of jobs in the Hadoop cluster facing issues with runtime. With change in the time, the environment has a lack in higher-level data abstractions which may have been helped in expressing developer intent. Code duplication and runtime errors which are costly to debug are reported instead.

To create a better platform on for writing code for data processing, they started using Apache Crunch: [2]

- **Type-Safety** helps to avoid making mistakes in the code, which are very expensive running across a massive Hadoop clusters.
- **High performance** comparative to performance of Python over Hadoop streaming.
- **Higher-level abstractions** like filters, joins, nodes and aggregations rather than having to use everything in terms of MapReduce.
- **First-class Avro support** lets the organization work with strongly-typed data files with the ability to evolve schemas over time.
- **Pluggable execution engines** like MapReduce and Spark that helps Spotify system to keep up to date with the advancements in Big data technology without rewriting all the pipelines with each increment.
- **Simple powerful testing** using the supplied MemPipline for fast in-memory unit tests.

```
public static void main(String[] args) {
  Pipeline pipeline = new MRPipeline(Example1StreamCountJob.class);
  PCollection<TrackPlayedMessage> trackPlayedMessages =
    pipeline.read(From.avroFile("/logs/track_played/2014-01-01",
                                TrackPlayedMessage.class));
  trackPlayedMessages
    .parallelDo(new MapFn<TrackPlayedMessage, String>() {
      public String map(TrackPlayedMessage input) {
        return input.getCountry().toString();
      }
    }, strings())
    .count()
    .parallelDo(new MapFn<Pair<String, Long>, String>() {
      public String map(Pair<String, Long> input) {
        return input.first() + "\t" + input.second();
      }
    }, strings())
    .write(To.textFile("/my/output/path"));
  pipeline.done();
}
```

[2] Code to know how many tracks were played in every nation every day.

Executing the code above with Hadoop jar runs a MapReduce by reading data from HDFS and Writing it back.

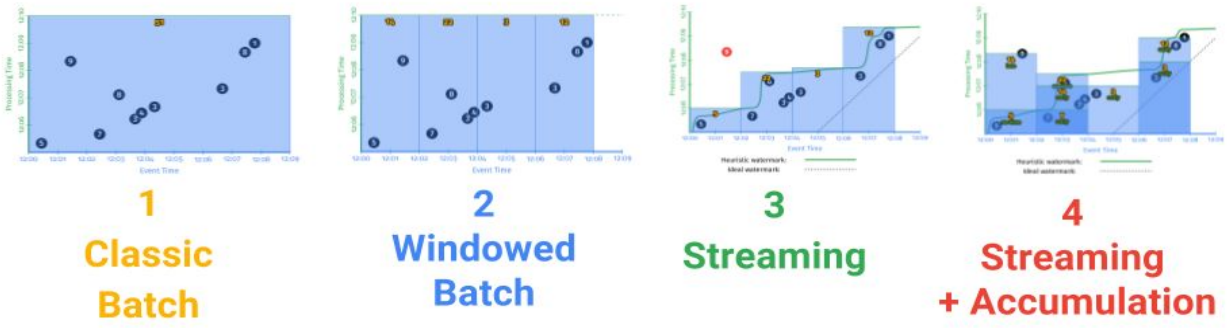
Crunch has been a great help for both product development and high-level performance execution on top of Hadoop in Spotify's system. Crunch can be used by anyone that are not satisfied with the stability of Spark and Python. It gives a relatively decent API that executes as a MapReduce and even a sub-project called Scrunch that uncovers a Scala API almost similar to that of Spark.

Requirements and Resources

Apache beam is the high-level Apache project for streaming and combined batch of data processing, known before as Google Cloud Dataflow before they donated it to the Apache Foundation. The processing of world scale data was divided into two paths:

- **Batch Systems** like Hadoop MapReduce, Hive which treats the data as a rigid and diverse volumes and process them as a single whole unit.
- **Streaming Systems** like Storm, Samza are quick at processing streams of events.

In the Dataflow paper beam has introduced a new unified programming model by Implementing both batch systems and streaming systems in the same model. Every element in the model has a window assignment and a undeclared timestamp. The system consumes data from infinite and continuous sources while in streaming mode and consumes elements from finite and discrete sources when in batch mode assigned to the same unified window.



With Beam, one can effectively write unified streaming and batch pipelines in a single API. In batch mode, the systems allows us to develop any sampled log files, analyze the timestamps and assigning windows to logs and run the streaming in same pipeline.

Observations made from different perspectives when SCIO compared to Spark and Scala:

- Both batch and streaming is supported in Spark in separate APIs. It also supports caching and execution dynamically by master node.
- Batch only with no in-memory caching or undefined algorithms supported in Scalding.
- Both batch and streaming with in the same API is supported SCIO. No in-memory or undefined algorithms support like in Spark.

Solutions

At Spotify, music personalization used be a very small team, developing models, writing pipelines and reading papers. Personalization team now has multiple teams around the globe working on producing datasets and serving users with products. Spotify has ensured consistent personalized user experience across the whole platform. Liking or Disliking a song on Spotify helps the algorithm to create a new song radio and Discover weekly playlist next time. The learning algorithms can easily over complicate and can easily satisfy none of the users. So, discover and start pages are the user's new personalization. Spotify's APIs are pretty simple which are user centric and any listening habits or music based stories can help the algorithm to be viewed on the page.

Critique:

Spotify might be struggling to be an entirely data-driven company, but they have been overcoming and leading the music streaming game since 2009. Spotify continues to be a personalized music access machine at the fingertips at right when user needs it. Google cloud integration has not been implemented in Spotify's system, the integration using Scala and Spark is not as close as and effective as the Dataflow. Being a Google software, the big data project can include cloud storage, BigQuery, Bigtable and spanner. Even though Scio can simplify the library and system maintenance, managing and replacing most of the data ecosystem with Pub/Sub, BigQuery and Bigtable reduces the cost of development and support cost.

Terms used

SCIO - Scala API for Google cloud Dataflow and Apache Beam.

Citations

1. Predicting the Next Big Hit-Big Data and Music Industry by Avantika Monnappa on Sep 8th, 2017 https://www.simplilearn.com/big-data-science-in-music-industry-article

2. Davida whiting. "Data Processing with Apache Crunch at Spotify." Labs. December 19, 2014. Accessed October 16, 2017. https://labs.spotify.com/2014/11/27/crunch.

3. Griggs, Brandon. "Spotify founder: Future of music is access, not ownership." CNN. July 21, 2011. Accessed October 16, 2017. http://www.cnn.com/2011/TECH/web/07/21/spotify.fortune.brainstorm/index.html.