Real Time Decision Making in Advertising Networks using STORM

Suresh Ramanujam[1] , Sharadraj Caliamourthy[2] , Breme Arunprasath[3], Hemaprakash N[4]

1. Assistant Professor, Dept. Of Information Technology, SMVEC, Puducherry
2. Final Year, Dept. Of Information Technology, SMVEC, Puducherry
3. Final Year, Dept. Of Information Technology, SMVEC, Puducherry
4. Final Year, Dept. Of Information Technology, SMVEC, Puducherry

***Abstract*— The irresistible quantity of data that is currently occupying all areas has made the large scale applications become a hot topic in the research area. There is a very high growth in the online advertising sector, advertising networks have to deal with an enormous amount of data to process. In recent years, Hadoop has been used to collect the data and process them. Though it is very efficient in processing large amount of data, it suffers a serious setback that it could not process these large amount of data in real time. Usually in conventional systems, data analysis will be made over a period of time say over a day or a week, to decide the Ad placements. In order to overcome this limitation, we propose a real time decision making system, which will provide real time analysis of the most trending sites at that instant to the advertisers to place their advertisements. The streamlined aggregation of real time data is achieved through STORM, a distributed real time computation system**

Keywords*— *Real Time Decision making, Hadoop, BigData, Storm, Advertising Networks, MapReduce

I. INTRODUCTION

The vast amount of data that is presently occupying all public fields has made the large scale applications become a hot subject in the research area. One of the main challenges to face is how to store and organize all information in order to provide efficient and reliable access to users. The previous period has seen a revolution in data handling with MapReduce, Recent technologies such as Hadoop, can store these large volume of data and manipulate them which was not possible in early times. But a major drawback with these systems is that they does not support real time processing of these data. This is because the real-time data processing involves a different kind of methodologies when compared with the batch processing. Yet, real-time data processing is inevitable for all present systems and hence the absence of one is a serious problem.

II. PROBLEM DEFINITION

When we go to a website and view an advertisement, it is due to the background process done by an advertising network. The advertisement will be displayed according on the basis of bidding system. The advertiser claiming the maximum bid will have their ads displayed on websites. This bidding process is done by the ad servers and the advertisers use different approaches to claim their bids such as user interest, context, number of users and other parameters. All these log impressions from the internet are sent to the ad servers and it is aggregated there. This aggregation will be used to detect the total impression for a particular ad in the internet in a particular interval of time. The data pipeline usually processes humongous amount of logs daily and it is very difficult to have the resources to handle this large volume of data. The results of the aggregation process should be obtainable to advertisers and publishers as earlier as possible so that it would be beneficial to them for their advertising process. With these results the advertisers will pay their bids to a specific publisher and help them to display their ads on specific target websites. So, the sooner they get the data, the better, they can make use of them in their promotions even up to the last second.

III. ILLUSTRATION WITH A REAL TIME SCENARIO

Usually every advertisement will have a set of contexts. For example, the ad for Reebok shoe will have the context as sports/gears. Similarly every website will have a context for itself. For example, the site espncricinfo.com has the context as sports/cricket and say flipkart.com will have the context as gadgets, laptops, furniture etc.

If the Reebok ad is placed on the espn site, then it is context based ad, as both the AD context and publisher context matches. Now the parties involved here are:

- Advertisers (Reebok shoe company)
- Publishers (Website owners)
- Ad agency (say Google)

Now there has to be a decision system to help Google to place the right ads in right websites for the money paid by advertisers. The reachability of an AD is based on the number of hits it gained from the web users. Now say there are three levels of budget for advertisers offered by Google, tier A, tier B and tier C for 24 HOURS time period.

- Tier A guarantees that atleast million users watches it and atleast 100 users clicks it.
- Tier B guarantees that atleast half million users watches it and atleast 50 users clicks it.
- Tier C guarantees that atleast 1/4 million users watches it and atleast 25 users clicks it.

Advertisers can choose any of the tier and pay for it. Now to achieve the hits promised to the advertisers, Google should find the top sites which yields maximum hits based on contexts. To find the appropriate sites and dynamically place the relevant ads, Google needs a decision system. The decision system (DS) will determine the trending sites or popular sites at the moment, which will be used for AD placement, such as

1. The site getting maximum hits at the moment.
2. The site getting maximum hits for a particular context
3. The location where users are more interested in a particular context.

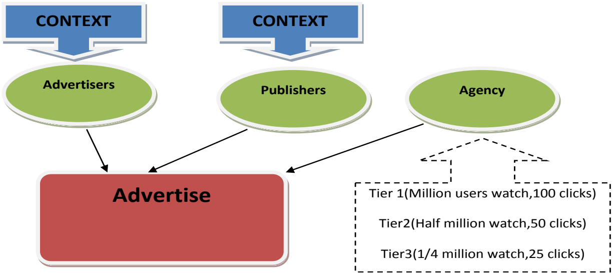

CONTEXT
CONTEXT
Advertisers
Publishers
Agency
Advertise
Tier 1(Million users watch,100 clicks)
Tier2(Half million watch,50 clicks)
Tier3(1/4 million watch,25 clicks)

Fig. 1. Advertising model

We assume that Reebook has registered for tier B in Google, which requires 5,00,000 views and 50 hits.

But unfortunately after 20 hours, only 4,00,000 views and 30 hits were there for Reebok ad. So, Google has to ensure that 20 hits has to be done in next 4 hours. To do this, Google have to place the rebook AD in sites, which are more popular at the moment and having relevant contexts. So in order to help this real time dynamic decision making, we propose the use of our STORM system in Decision Making.

IV. CHALLENGES IN PRESENT HADOOP SYSTEM

Fundamentally Hadoop is one of the batch processing system well known. In the Hadoop file system (HDFS) the data are been fed which for processing is been distributed across nodes. The resulting data is returned to HDFS for use by the originator once the processing is completed.

To aggregate data Hadoop is used in most of the ad networks. Through Hadoop is really efficient in processing a large amount of data it is not suited for real-time aggregation where data need to be available to the minute. Through a queue mechanism the ad server sends its logs to the data pipeline continuously this how usually it works. Every hour the Hadoop is scheduled to run an aggregation which is then stored in a data warehouse. This ensures that Hadoop is not suited for real-time aggregation.

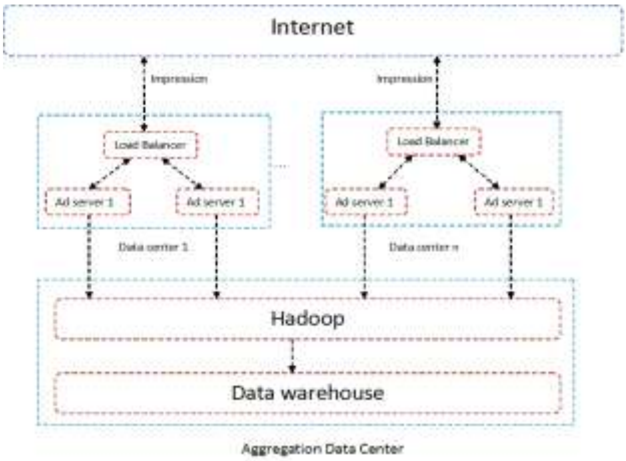
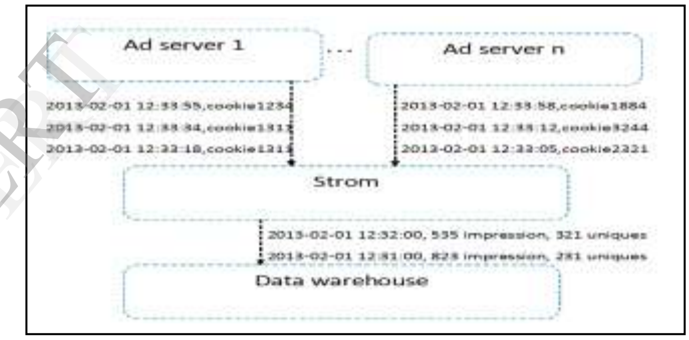


Fig. 2. Hadoop Aggregation model

V. REAL TIME DECISION MAKING

Real Time decision making system is capable of reading the data in the form of a continuous flow from the ad servers and can process them at that instance itself. Thus they are capable of providing multiple aggregation at the same time.



```
2013-02-01 12:33:55,cookie1234
2013-02-01 12:33:34,cookie1312
2013-02-01 12:32:18,cookie1311
```

```
2013-02-01 12:33:58,cookie1884
2013-02-01 12:33:12,cookie3244
2013-02-01 12:32:05,cookie2321
```

```
2013-02-01 12:32:00, 535 impression, 321 uniques
2013-02-01 12:31:00, 823 impression, 281 uniques
```

Fig. 3. Real Time Storm Aggregation

VI. METHODOLOGY

Storm is an open source distributed real-time computation system. Storm makes it easy to reliably process limitless streams of data, doing for real-time processing what Hadoop performed by batch processing. Storm is efficient and compatible with all programming language. In a Storm topology, there are 2 main types of nodes:

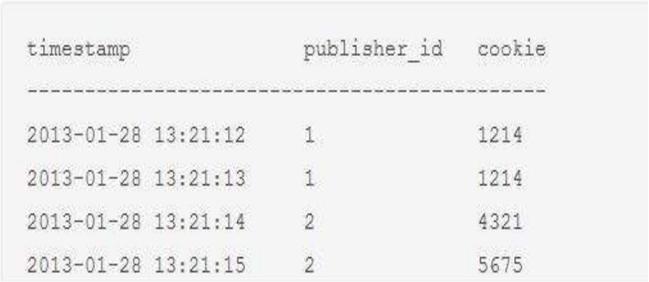
1. SPOUT: They get an input stream and pass them into the Storm cluster. The data which is been obtained from a JMS queue, a Twitter Stream, a database is released to the input stream of the cluster that will be handled by bolts.

2. BOLT: The output from a spout or another bolt which is considering as an input stream is processed. After the data being processed, it is either stored in a database or passed into another stream to other bolts.

Our topology for ad networks will contain:

- ImpressionLogSpout has a stream which is aggregated by 3 bolts at the same instance.
- The AggByMinuteBolt is used to aggregate the impression logs in real time as a stream and they are available to the publishers. This gives data on how many unique impressions are available at that instance.

Initially when an impression arrives, it is checked whether it belongs to that minute, then it is added to the cookie set. A sample impression log will be as shown in the figure. It is in the simplified form of the log.



```
timestamp                 publisher_id   cookie
--------------------------------------------------
2013-01-28 13:21:12       1              1214
2013-01-28 13:21:13       1              1214
2013-01-28 13:21:14       2              4321
2013-01-28 13:21:15       2              5675
```

Fig. 4. Sample Log file

We know that no further impression for the previous minute will be obtained when an impression comes in that belongs to the next minute so we can persist the number of impressions and the number of unique impressions to MongoDB.

- Main.java
- RandomImpressionTupleSpout.java
- AggregateByTimeAndPersistBolt.java

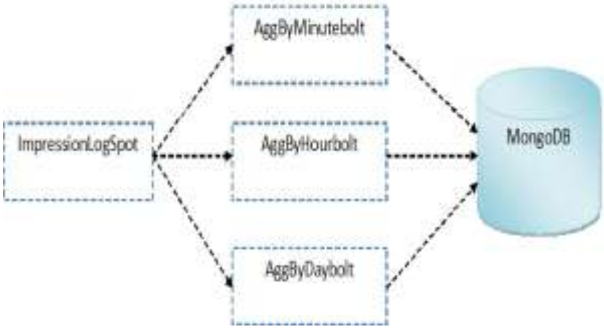
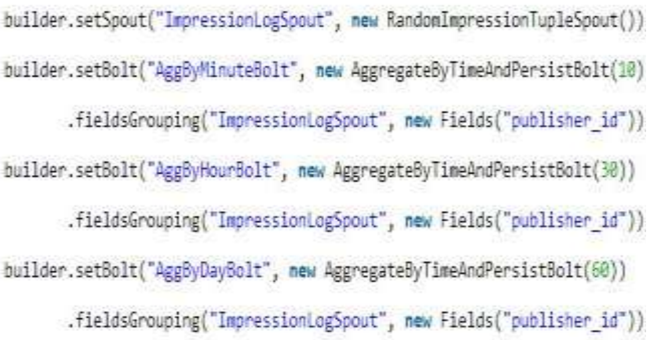


Fig. 5. Architecture overview

- The Main java class is used to run in a single system. The spout collects the log impressions from the ad servers
- RandomImpressionTuple Spout collects the continuous data stream from ImpressionLogSpout
- AggregateByTimeAndPersistBolt.java is responsible for the aggregation of the impressions by publisher and given to the agency



```
builder.setSpout("ImpressionLogSpout", new RandomImpressionTupleSpout())

builder.setBolt("AggByMinuteBolt", new AggregateByTimeAndPersistBolt(10))

        .fieldsGrouping("ImpressionLogSpout", new Fields("publisher_id"))

builder.setBolt("AggByHourBolt", new AggregateByTimeAndPersistBolt(30))

        .fieldsGrouping("ImpressionLogSpout", new Fields("publisher_id"))

builder.setBolt("AggByDayBolt", new AggregateByTimeAndPersistBolt(60))

        .fieldsGrouping("ImpressionLogSpout", new Fields("publisher_id"))
```

Fig. 6. The Main class

VII. THE DECISION MAKING SYSTEM

Thus using the results that are obtained from the storm topology, we generate a real time graph. The graph will be constantly updating the current status of the websites. Using this graph the placements of the ads in the publisher's websites can be made effectively.

This graph is generated using Flot. Flot is a utility of JavaScript library using which we can create attractive real time graphs. A sample Flot graphs is shown in the following figure.

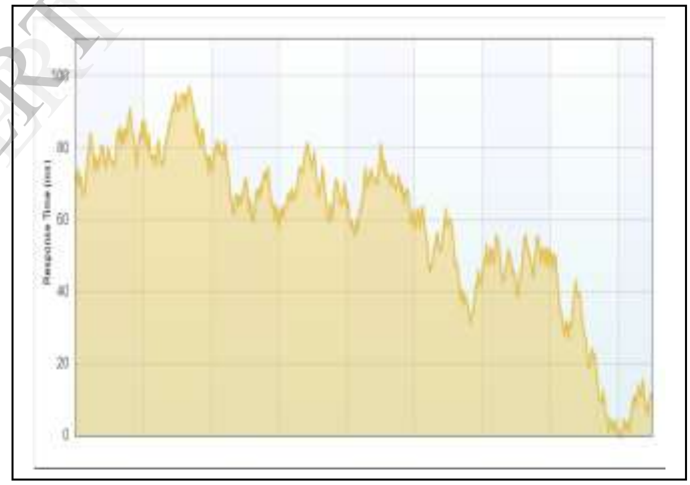


Fig. 7. Flot Graph for Decision System

The Graph will provide three important information such as

- Popular Site at the instance
- Popular Site Context at the instance
- Popular Site Location

VIII. CONCLUSION

Using this real time decision making system it has become easy to compute multiple aggregations at the same time and also faster to spit aggregations out as soon as they are computed and allowing reducing the amount of data to be sent between data centers which works well on time based data in particular.

References

[1] Nathan Marz, James Warren “Big Data: Principles and Best Practices of Scalable Realtime Data Systems” Manning Publications Company, 28-Sep-2013

[2] Jonathan Leibiusky, Gabriel Eisbruch, Dario Simonassi “Getting Started with Storm”, "O'Reilly Media, Inc.", 2012.

[3] Boris Lublinsky, Kevin T. Smith, Alexey Yakubovich. “Professional Hadoop Solutions”, John Wiley & Sons, 12-Sep-2013.

[4] Google eBook, “Storm Real-Time Processing Cookbook” , Packt Publishing Ltd

[5] Tom White, “Hadoop: The Definitive Guide,” "O'Reilly Media, Inc.", 10-May-2012.

[6] M. Tim Jones, “Process real time big data with twitter storm,” IBM.