Big Data Analysis: Apache Storm Perspective

Muhammad Hussain Iqbal[1], Tariq Rahim Soomro[2]

Faculty of Computing,
SZABIST Dubai

***Abstract*— the boom in the technology has resulted in emergence of new concepts and challenges. Big data is one of those spoke about terms today. Big data is becoming a synonym for competitive advantages in business rivalries. Despite enormous benefits, big data accompanies some serious challenges and when it comes to analyzing of big data, it requires some serious thought. This study explores Big Data terminology and its analysis concepts using sample from Twitter data with the help of one of the most industry trusted real time processing and fault tolerant tool called Apache Storm.**

***Keywords*— Big Data, Apache Storm, real-time processing, open Source.**

I. Introduction

Big data has become one of the most important technologies that have changed the way business use information to improve their business models and experiences. Big data is a combination of management of data technologies that have emerged over the period of time. Big data empowers business to manage, store, and use huge data at the correct motion and at the correct time to gain accuracy and efficiency. The main key for big data is that data has to be managed in a way so that business requirements are met. Many enterprises are working on different ways that will allow them to collect huge data, which help them to find the hidden patterns, which exists within that huge data that can bring the revolution to their business model. As organizations start to assess new sorts of big data results, a lot of new doors will unfold [1]. Executing a big data result obliges that the foundation be set up to backing the administration, dissemination and versatility of that data. Hence, it is critical to put both the business and specialized procedure set up to make utilization of this vital engineering pattern. In stream processing, as opposed to the idea of volume, the thought of speed is determinant. Hence requires a processing tool, which can process the new generated information at a greater pace with guaranteed processing and low latency. Subsequently, new apparatuses for preparing this enormous speed of information are needed. “Apache Storm” is the leading real time processing tool, which guarantees the processing the newly generated information with very low latency. Hadoop is the mostly used tool currently; although Hadoop works well, but it processes the data in batch only that is why it is for sure not a best tool for analyzing the latest form of data. Processing the data in real time is now a normal requirement. This phenomena is called stream processing, in other words analyze the real time data in continues motion. This study is limited to twitter as a source of big Data and will not address all the issues which normally occurred during Big Data Analysis, but it will try to process the continuous stream of tweets unlike Hadoop, which process the data in batches. This paper will focus on exploring the Big Data Analysis and its tools, including Apache Strom and will cover comparison of available tools with Apache Strom along with its justification in section 2; section 3 will discuss material and methods; section 4 will discuss the results of analyzing of big data using Strom; and finally section 5 will be the discussion and future work.

II. Review of Related Literature

The term Big Data is used by numerous organizations but there is not a standard definition of it. Big Data speaks to huge sets of information; however it is not only the prominent feature to categorize information as Big Data. Rather 3 different attributes namely volume, variety and velocity, combined together to create Big Data. These three attributes are generally referred as the 3-Vs of Big Data. These days, organizations aren't just managing their information. More real time information important to skilfully comprehend the business is created by outsiders outside the organization. Along these lines, real time information originates from numerous different sources in different sorts; it could be content from informal community, picture information, geo area, logs, and sensors information and so on. Traditional

Databases, which rely on homogenous data structures, cannot process this heterogeneous data, hence are not a great fit for Big Data [2]. So what do we mean by "real-time"? Firstly, it is important to understand that this term can have two different perspectives depends under what context this term is used. If it is used in context of the information, it means transforming the latest available information, handling the numerous data as it is generated. On the other hand, if a real-time framework is utilized to catch drifts in Twitter stream, the thought of real-time might be deferred by a couple of seconds. Nowadays, this term is really a puzzling word and is frequently misused. Anyway by and large, when talking about real-time processing, it means processing the data with very low latency [3]. Stream processing is designed to analyze and act on data which is generated in real-time, i.e. using “continuous queries” that operate repeatedly over time and buffer windows. Stream processing enables us to analyze the stream i.e. to extract mathematical or statistical information analytics on the runtime within the stream. Stream processing solutions are designed to handle Big Data in real time with a highly scalable, highly available and highly fault tolerant architecture. This empowers to analyze the data in motion [4].

A. *Available Tools*

Below are some open source tools which are being used for big data analysis:

1. *Apache HBase*

Apache HBase is a Java based, open-source software, which enables to store Big Data. It is highly non-relational in nature and provides Google’s Bigtable like functionality to store sparse data. HBase is widely used when random and real-time access to Big Data is required and is operates on the top of HDFS [5].

2. *Hadoop*

The Apache Hadoop project is open source software to process Big Data. The key features of Apache Hadoop are its reliability, scalability and its processing model. It allows processing the large sets of data across clusters of machines using distributed programming paradigm. It operates the information in small batches and uses MapReduce framework to process the data and is called batch processing software [6].

3. *Apache Spark*

Apache Spark project is open source based for processing fast and large-scale data, which relies on cluster computing system. Like Apache Hadoop it is also designed to operate on batches, but the batch window size is very small. It provides flexibility to develop modules in three different languages Java, Scala and Python. It also provides a rich set of tools that are to process SQL including Spark SQL, for machine learning MLlib, for process graph GraphX, and for stream analysis Spark Streaming [7].

4. *Yahoo S4*

In October 2010, Yahoo released Yahoo S4. In 2011 it joined Apache Foundation Family and it was given the status of Apache Incubator. Yahoo S4 empowers developer to design applications, which can process real-time streams of data. It is inspired by MapReduce model and process the data in distributed fashion. It supports modular programming model i.e. developers can develop plug and play modules in Java. The modules developed in Yahoo S4 can be consolidate to design more advance real-time processing applications [8].

5. *Apache Storm*

In December 2010, Nathan Marz came up with an idea to develop a stream processing system that can be presented as a single program. This idea resulted to a new project called Storm. Apache Storm empowers developers to build real-time distributed processing systems, which can process the unbounded streams of data very fast. It is also called Hadoop for real-time data. Apache Storm is highly scalable, easy to use, and offers low latency with guaranteed data processing. It provides a very simple architecture to build applications called Topologies. It enables developer to develop their logic virtually in any programming language, which supports communication over a JSON-based protocol over stdin/stdout. Apache Storm becomes the part of Apache Family on 17 September 2014.

B. Comparison of Apache Strom with other tools

Below Table 1 will compare big data open source tools with Apache Strom [9]:

TABLE 1
COMPARISON OF BIG DATA OPEN SOURCE TOOLS

Other Tools	Developer	Type	Difference
HBase	Apache	Batch	Storm provides real time data processing, while HBase (over HDFS) does not process rather offers low-latency reads of processed data for querying later.
Hadoop	Apache	Batch	The main difference is that Storm can do real-time processing of streams of Tuple's (incoming data) while Hadoop do batch processing with MapReduce jobs.
Spark	UC Berkeley AMPLab	Batch	One way to describe the difference is that Apache Spark is a batch processing framework that is capable of doing micro-batching also called Spark Streaming, while Apache Storm is real-time stream processing frameworks that also perform micro-batching also called Storm-Trident. So architecturally they are very different, but have some similarity on the functional side. With micro-batching, one can achieve higher throughput at the cost of increased latency. With Spark, this is unavoidable and with Storm, one can use the core API (spouts and bolts) to do one-at-a-time processing to avoid the inherent latency overhead imposed by micro-batching. And finally, many enterprises use Storm as a mature tool while Spark Streaming is still new.
S4	Yahoo!	Streamin g	The main difference is that, storm gives guaranteed processing with high performance and thread programming support

C. Why Apache Storm?

Five key attributes, which make Apache Storm as a first choice tool for processing real-time unbounded data, are described as follows [10]:

- Easy to use
- Fast – benchmarked for processing millions byte data per second per node
- Fault-tolerant – keep the track of all worker nodes, whenever a node dies, Apache Storm restart the process on another node
- Reliability –Guaranteed data processing with at least once semantics
- Scalability – process the data in parallel across a cluster of machines

Below listed are the main criterion, on basis of which one can decide when to use Apache Storm [11].

- Fault tolerance: High fault tolerance
- Latency: Sub Seconds
- Processing Model: Real-time stream processing model
- Programming language dependency: Any programming language
- Reliable: Each tuple of data should be processed at least once.
- Scalability: High scalability.

III. MATERIAL & METHODS

Big data is modern day technology term that have changed the way world have looked at data and all of methods and principles towards data. The Data gather of big data is totally different than our traditional ways of data gathering and techniques. So technically speaking, this paper will be using Twitter streaming API to get access to twitter big data as a big data sample. Storm makes it easy to reliably process unbounded streams of data, doing for real-time processing what Hadoop did for batch processing. This experiment will execute three different scenarios with live data and will collect the statistics, which will be used to analyze the processing tool and to draw some conclusion.

A. Research Instrument

In this study Twitter has been used as a Big Data source and some tests are being done using Apache Storm tool and following research instruments were used:

- Apache Storm: to process the real time data
- Big Data API(s): For example Twitter API, to fetch the data in real time

- A .NET application: to record the performance and reliability of the tool.

IV. Results

This section illustrates and analysis the data collected for the experiment purpose using twitter streaming API. The study was aimed at analyzing the twitter big data streams using state of art Apache Storm open source tools to recognize particular patterns from huge amount of data. Following scenarios were executed for experiment purpose on live streams of twits on twitter:

- Top ten words collected during a particular period of time.
- Top ten languages collected during a particular period of time.
- Number of times a particular "word" being used in twits, twitted in a particular period of time.

A. *Scenario-1: Top ten words collected in last 10 minutes*

Statistics:

- Total time duration=10 minutes (603 seconds).
- The total number of tweets analyzed during this time=67271
- The total number of words=482874
- See Table II for top ten words in tabular form.
- See Figure 4-1 for top ten words shown graphically in charts.

Table II
Top ten words in last 10 minutes

S. No.	Word	Frequency
1	Jessie	15585
2	Lady	18543
3	Gaga	18552
4	Rey	23664
5	Lana	23677
6	Del	23690
7	Swift	23881
8	Taylor	24284
9	Coldplay	25330
10	Mtvstars	62726

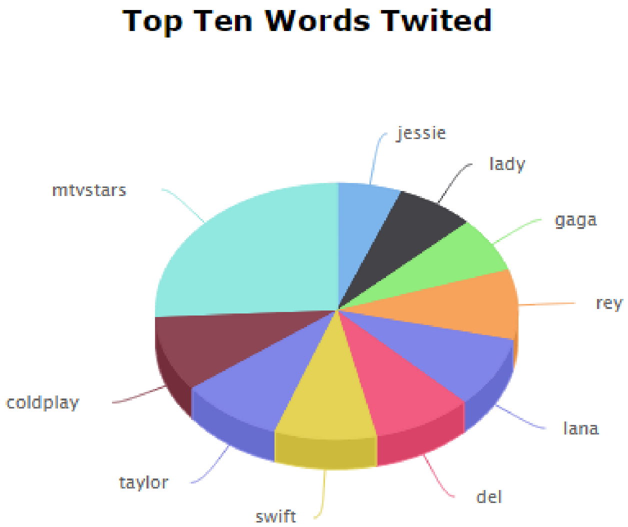

Fig. 1 Top ten words in last 10 minutes

B. *Scenario-2: Top ten languages collected in last 10 minutes.*

Statistics:

- The total number of tweets analyzed during this time=24000
- The total number of unique languages=48
- Total time duration=10 minutes (605 seconds).
- See Table III for top ten languages in tabular form
- See Figure 2 for top ten languages shown graphically in charts

Table III
Top ten LANGUAGES in last 10 minutes

S. No.	Language	Frequency
1	Korean	588
2	Thai	600
3	Turkish	612
4	French	860
5	Spanish	1011
6	Indonesian	1257
7	Russian	1818
8	Arabic	2559
9	English	5826
10	Japanese	7053

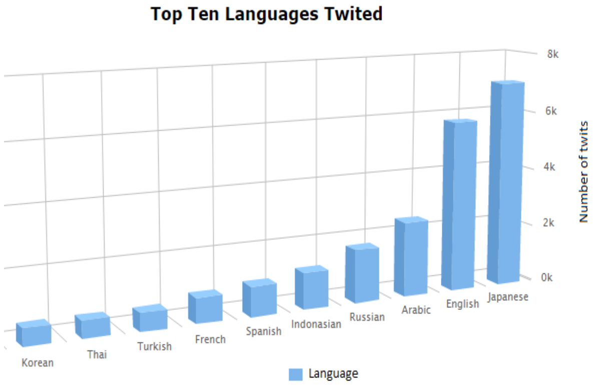

Fig. 2 Top ten languages in last ten minutes

C. *Scenario-3: Number of times "mtvstars" being used in twits twitted in last 10 minutes.*

Statistics:

- Search String =mtvstars
- Time duration = 10 minutes (600 seconds)
- Number of twits =42125
- See Table IV for number of twits posted using word "mtvstars" in tabular form
- See Figure 3 for number of twits posted using word "mtvstars" shown graphically in charts

TABLE IV
NUMBER OF TWITS "MTVSTARS" USED TO POST A TWIT IN LAST 10 MINUTES

Twits frequency	Time duration in seconds	Twits frequency	Time duration in seconds
401	16	22535	312
977	22	22758	318
1544	29	23062	324
2331	35	23443	330
2876	41	24073	336
3579	47	24510	342
4136	53	25004	348
4826	59	25456	355
5282	65	25916	361
5525	71	26379	367
6186	77	26656	373
6602	83	27024	379
6933	89	27586	385
7162	95	27948	391
7769	101	28379	397
8023	107	28651	403
8291	113	28930	409
8506	119	29361	415
8809	125	29630	421
9365	131	30063	427
9615	137	30463	433
10044	143	30950	442
10405	149	31606	448
10884	155	31995	454
11207	161	32590	460
11879	167	32921	466
12309	173	33530	472
12888	179	34134	479
13391	185	34464	485
14024	191	34812	491
14362	197	35048	497
14602	203	35441	503
14921	209	35801	509
15154	215	36317	515
15521	221	36704	521
15947	227	37210	527
16482	233	37668	533
16788	240	37988	539
17304	246	38448	545
17664	252	39018	551
18211	258	39537	557
18720	264	40207	563
19387	270	40640	569
19859	276	40963	575
20341	282	41396	581
20599	288	41547	587
21114	294	41688	593
21605	300	41837	600
22232	306	42125	606

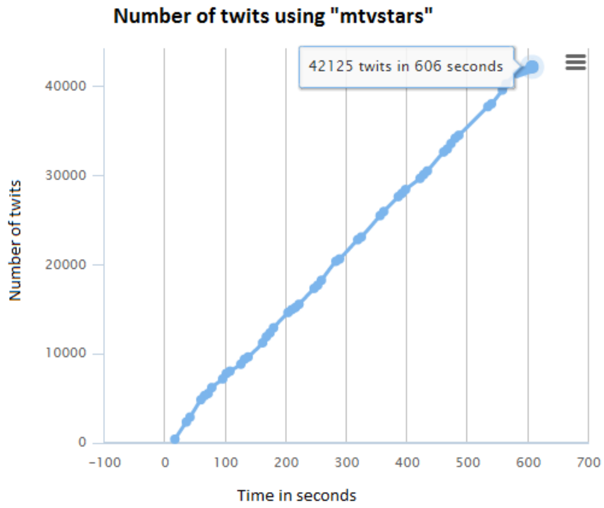

Fig. 3: number of twits using "mtvstarts" in last 10 minutes

V. DISCUSSION & FUTURE WORK

Companies are continuously looking to run across huge bits of information into their data. Most of big data exercises start from the need to answer specific business request, for instance, what customers positively consider our brand? And in what limit would we have the capacity to assemble our arrangements experiences and close more plans. Big data investigation is extremely imperative for all stack-holders of any business. Extending from top chiefs to significant other experts like IT specialists. This study explored for companies to understand the Big Data and its notions. It reviewed for the companies to choose between traditional databases and the big data tools. It is empowering the IT managers to think about the Big Data before it is too late. It is also empowering the developer to understand the use of Storm to analyze and process big data. This study was conducted under some experimental limitations in terms of infrastructure and data. In terms of data approximately 1% of the total tweets were available with Twitter free API. In terms of hardware configuration the experiment was not performed on dedicated Server, rather this study was conducted using laptop HP630 having corei3 processer, 2.53 GHz frequency, 4 GB RAM and windows 7 professional editions 32 bit and keeping in view the above mentioned consideration following three scenarios was performed:

- Top ten words twitted during last 10 minutes.
- Top ten languages used to twit during last 10 minutes.
- A list of twitted items matching a given search keyword during last 10 minutes.

All the above three mentioned scenarios were performed successfully, proving Apache Storm can process real-time streams with very low latency. All the tweets were queued¬ as they were received without any delay and calculations were performed on the tweets using bolts. The programming model was easy to build on own topologies. The execution of the topology can be drawn as a directed graph. Even though Apache is built on Clojure, the topologies were created in Java, so programming can be done in multiple languages. During this experiment, some areas of future development were identified.

- To install and configure Apache Storm is not easy task. No direct setup is available to install pre-requisites and configure the tool. All the steps have to be performed manually and there is no comprehensive guide available. So for future releases a user friendly installer and configuration module will be of great use for developers.
- Although Apache Spark provides some key performance indicator's (KPI's) to measure the performance and reliability but it is not enough to call it user friendly. There is no reporting module either. For future releases addition of a reporting module will make the tool the leading open source tool for real-time processing.

References

[1] J. M. a. M. C. Tim McGuire, August 2012. [Online]. Available: http://iveybusinessjournal.com/topics/strategy/why-big-data-is-the-new-competitive-advantage#.VKv0wSuUe9E.

[2] "Big data: The next frontier for innovation, competition, and productivity," McKinsey Global Institute, 2011.

[3] B. Perroud, "A hybrid approach to enabling real-time queries to end-users," Software Developer's Journal, 2013.

[4] K. Wähner, 10 Sep 2014. [Online]. Available: http://www.infoq.com/articles/stream-processing-hadoop.

[5] "Apache HBase," Apache, 22 December 2014. [Online]. Available: http://hbase.apache.org/. [Accessed 06 January 2015].

[6] 1 Dec 2014. [Online]. Available: http://hadoop.apache.org/ .

[7] 4 Dec 2014. [Online]. Available: https://spark.apache.org/.

[8] 4 Dec 2014. [Online]. Available: http://incubator.apache.org/s4/.

[9] J. S. Damji, "Discover HDP 2.1: Apache Storm for Stream Data Processing in Hadoop," 23 June 2014. [Online]. Available: http://hortonworks.com/blog/discover-hdp-2-1-apache-storm-stream-data-processing-hadoop/. [Accessed 06 January 2015].

[10] "Apache Storm," Hortonworks, [Online]. Available: http://hortonworks.com/hadoop/storm/. [Accessed 06 January 2015].

[11] "Apache Storm," Apache Software Foundation, 2014. [Online]. Available: https://storm.apache.org/about/integrates.html. [Accessed 01 January 2015].