Survey of Realtime Stream Processing Tools and Volume Reduction Techniques

Gaurav Parashar
Research Scholar
Amity University
Noida, INDIA
Email: gaurav.parashar@student.amity.edu
gauravparashar24@gmail.com

Dr. Shailendra Narayan Singh
Associate Professor
Amity University
Noida, INDIA
Email: snsingh36@amity.edu,
sns2033@gmail.com

Dr. BK Sharma
Principal Scientific Officer
NITRA
Ghaziabad, INDIA
Email: drbksharma2002@yahoo.com

***Abstract*—In recent years Big data field have received attention by different streams like business community, researchers, academic community, financial institutions, medical practitioners and many more. Big data field has some thing for everyone. One of the best examples of the big data processing is IBM Watson. For giving an answer to the question, Watson search large databases, online text, stored text files and give answer. This realtime processing of the queries is an example of Big data processing. This paper presents a survey of different volume reduction strategies as proposed by different researchers in the field.**

INTRODUCTION

Google trends[1] for Indian Prime Minister shows the statistics on how often people on Internet used "*Narendra Modi*" as a search keyword relative to total search volume across the world. Searching the same term on Google gives around 4,04,00,000 results(as on 27 October, 2014) which includes articles, blogs, reviews, comments, videos, tweets, news, media, cross referencing documents and many more. The statistics on Trends and Google search is an example of Big Data processing.

Along with the above example, CERN's Large Hadron Collider (LHC)having about 150 million sensors, which generated data at a rate of 40 million times per second. After filtering only 100 collisions were of interest from that vast amount of generated data[2]. Data at this speed is really hard to process and manipulate. Wal-Mart, an American multinational retail chain has customer records of about 43 Tera-bytes, which is more than the total records of Internal Revenue Services[3]. Twitter, a micro blogging site have around 50 million subscribers from all around the world, generating 50M tweets per day. Tweets are small text messages that users generate to relay their thoughts, status, comments on some news and topics for discussion[4]. Social media networks like Linkedin, Facebook, Google+, are one of the biggest contributors of data. Earlier generation was termed as data generation, but now it is more than data i.e, Big Data. Big Data is a field which does not have boundaries and spans across many streams of science and technology. May it be Engineering, Telecom, Business, Bio-Medical, Healthcare, Particle Physics, social networks, each of these fields produce vast amount of data, which is distributed all across regions. When data becomes so huge that a central location cannot contain and process it, it is termed as big data. This is the first characteristic of Big Data. The next characteristic deals about the speed by which the data is generated. According to [5], everyday data of about 2.5 quintillion bytes are generated and moreover in last two years. According to[6], Facebook alone generates 500 TB of data daily which contains photos, likes, comments and shares. The last characteristic deals about variety of data which comes under Big Data. Everything in digital form comes under Big Data. Databases, videos, photos, text files, web logs, wireless sensors data, GIS data, telecom data, retail store transaction records, credit card transactions, share trading, ebooks, mp3/mpeg files, and many more. If its on your computer it is part of Big Data.

The definition of Big Data by Gartner[7] has 3 V́s Volume, Variety and Velocity. New dimensions are now being added to the definition of Big Data. From 3V́s it had now become 5 V́s. Value and Veracity is now added to the definition[8]. The improved definition of Gartner is defined by[9], It states that Big Data is a data intensive technologies target to process high-volume, high-velocity, high-variety data to extract valuable data ensuring high-veracity of original data. To harness new forms of data innovative data models with innovative technologies, infrastructure and tools applied on data life-cycle. These V́s have placed great pressure on current IT infrastructure of industries. Industries which need to harness any of these V́s have to either upgrade their current hardware and software or use cloud computing infrastructure. Both these options require investment's. For a realtime computing of data which comes with velocity and volume need algorithms to filter data in need and store the rest for future computations. Therefore, volume reduction can be one of the techniques which can be used to trim down the vast amount of data. Many technologies like Hadoop[1] which is used for scalability and storage, Flume[2] a realtime processing framework used for efficiently collecting, aggregating large amounts of log data, Storm[3] a distributed realtime computational system, Kafka[4] a distributed publish subscribe messaging system, Spark[5] a fast engine for large

[1]http://hadoop.apache.org/
[2]http://flume.apache.org/
[3]https://storm.apache.org/
[4]http://kafka.apache.org/
[5]https://spark.apache.org/

978-1-4799-8433-6/15/$/31.00©2015 IEEE

scale data processing, Drill[6] a distributed system for interactive analysis of large-scale-datasets, Cloudera Impala[7] a distributed query execution engine and many more are used to process Big Data[10][11].

According to the findings of survey by BARC[12], indicates about the trend to make data analysis in near real-time. Since at the time of the survey companies updated 4% of their business data in near real-time. It was also indicated that this increase will be increased to 7% in future.Barlow[13] in his book states that despite of several technologies in hand, increase in computing speed, handling data at massive speed, storing data in different formats realtime analytics lies beyond the realm of pure technology. This trend and the examples stated above indicates a research potential in this area.

In this paper, we make following contributions: We begin our study with an introduction to data stream processing system. In this section we will take a view of the stream processing system framework. In next section, the paper reviews different Volume reduction techniques. Section 3 concludes the paper.

DATA STREAM PROCESSING SYSTEM

A data stream processing system processes realtime data (called tuples) and performs statistical computations on it. A data stream processing system (P) can be defined as a system which processes Y(input stream) and outputs Z(output stream):

$$P = (Y \cup OP \cup Z)$$

Input stream(Y), Operators(OP) and output stream(Z) can be defined as

$$Y = (t_1, t_2, t_3, ..., t_n, ...)$$
$$Z = (o_1, o_2, o_3, ..., o_m, ...)$$
$$OP = \text{Operators like count, join, aggregate, max, min, ...}$$

Where tuple t_i is represented by <data set or fact> and t_{start} — t_{end} denotes start time and end time of a tuple validity, operators are the operations those are performed on streams and o_j is represented as output tuples processed by operators. While processing these input streams system(P) consume resources like memory, cpu cycles and bandwidth. Data coming at faster rate than the system processing capabilities induce stress and results in loss of efficiency. Therefore technique like load shedding and eviction are used to release stress of the system. These techniques are described in section -2. Next section describes some of the realtime platforms used in industries for realtime stream processing.

REALTIME PLATFORMS

Many realtime platforms have evolved in recent decade to process the realtime data in near realtime. We will discuss some of them here.

[6]http://incubator.apache.org/drill/

[7]http://www.cloudera.com/content/cloudera/en/products-and-services/cdh/impala.html

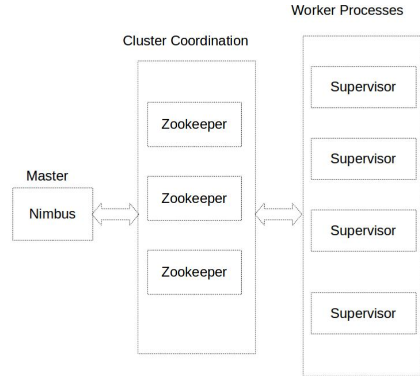

Fig. 1. Storm Cluster Setup

1) Storm: Storm is a big data realtime event processing system. It can be easily complied, deployed, has high fault tolerance, can be scaled horizontally with reliable message handling, has a service framework for hot deployment and simple programming model[11]. Storm is deployed using multi-node cluster. The organization is shown in fig1. ZooKeeper node is used to keep all the configuration, naming, coordination and provide distributed synchronization. Master node daemon "Nimbus" is used to assign tasks to all the slave nodes or worker nodes, distributing code around the clusters. Slave nodes runs an instance of supervisor daemon, which listens to Nimbus and executes the tasks assigned.

Data in storm comes from "Spout" and is transferred to "Bolts", which are processing logic. Data travels through them and computations are performed. The arrangement of Spouts and Bolts is termed as topology. The abstraction of realtime data which are fed into the system are called as streams. These streams are processed in-memory, because streams comes in varying size and speed and if stream processing systems use file I/O to manage data then the system will become too slow in processing the data, for faster data processing in near realtime. Companies like twitter, Groupon, fullcontact , Yahoo use storm for their realtime data processing[14].

2) Spark Streaming: Spark streaming is a large scale event data processing engine on top of Spark. It extends spark for big data computing. It uses in-memory computing and 100 times faster than hadoop. It batches up events, according to time window, and then processes it. Batch processing of events put latency in processing of few seconds. Spark also provides fault tolerant computation, with a guarantee that each event will be processed only once. It combines streaming with batch and interactive queries. Processing in spark streaming is done by dividing stream computation in a series of very small and deterministic batch jobs[15]. The processing is explained as under:

1) It chops the incoming data stream into batches of X seconds.
2) It treats each batch as RDD(Resilient Distributed Datasets) and process them using RDD operations.
3) Processed results of RDD operations are returned to batches.

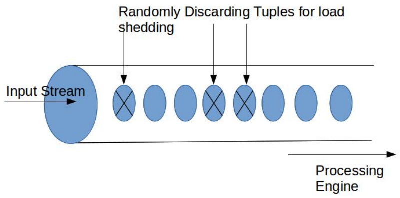

Fig. 2. Load Shedding

Comparison

Both Storm and Spark Streaming are used in production by many organizations. We have summarized the key features of both the systems in Table I.

VOLUME REDUCTION TECHNIQUES

A realtime system works in a tight constraint of time. In these systems data flow in a continuous stream of tuples. These tuples are monitored and analysis is done in near real time. Data modelled in these systems are not persistent but transient[16]. Examples of such systems are sensor networks, financial applications, network monitoring applications, web applications, security applications. Some of the characteristics of data streams[17][18][19] are:

- Data streams arrive continuously
- Data streams are unpredictable and system has no control over the order in which tuples arrive.
- Tuples have unbounded size
- Once a tuple is processed it is discarded
- Tuples may arrive in duplicates and data stream becomes possibly noisy data
- Streams have unbounded memory requirements
- Streams comes at a high speed

These characteristics incur communication, computational and memory overhead on the processing systems. Therefore many researchers have devoted their study around optimizing these characteristics. Load shedding is a method to solve the limited resources issue and is described in section-A.The details of the work contributed by different researchers in section-B.

A. Load Shedding

Load shedding is defined as an optimization problem where we have to optimize the results and time used for processing the queries. It has been studied in areas like networking and multimedia but in a different context. Many authors have given different methodologies to solve this optimization problem. Figure 2 describes that *in load shedding the stream processing engine takes streams and then deletes the tuple randomly, before they are sent for processing*. The problem with deleting tuples randomly introduce errors or induce imprecision in result computed by the processing engine. To reduce errors and increase efficiency shedding mechanism is placed under tight constraint of QOS(Quality of Service) parameters. These parameters ensures the validity of the result with minimum errors. This type of load shedding is called as value-based QoS. This technique is defined in section -B.

B. Related Work

In [19], authors explain the problem of load shedding using queueing theory. They used little's law to formulate the problem. Unlike load shedding, eviction is a process of removing the data from cache using same principal as LRU algorithm but with advancements. The cost of eviction is lower system recall. The study includes several eviction strategies like:

1) *Random eviction* deletes the data from cache according to uniform distribution.
2) *First In First Out*(FIFO) deletes the data which in turn deletes the oldest data in cache.
3) *Least Recently Used* (LRU) deletes the data which is oldest but with a condition that whenever it is used it will swap its place from last to first therefore it extends FIFO.
4) *Garbage Collection* data is deleted from the cache based on the relevancy of data according to context of a query.

One important question that was answered by authors was
Why not LRU in this context?
LRU is not used, in this context, as an eviction strategy because of the nature of the algorithm. Substantial reason for the same is that the data which is still in use but had stayed long in the cache will be removed. Two real world data sets were taken into consideration SRBench and ViSTA-TV. ViSTA-TV consists of TV viewers data consisting of userlog (consisting data of which user is watching which channel and what time user switches to another channel) and EPG streams(which channel is showing which program and at what time). Data of EPG is stored in cache and a two-way join over userlog and EPG is taken to find which user is watching which program . EPG is initialized with some of the popular channels users are watching. When a new entry from join comes it first checks the entry which has been there for long in cache. It then removes the entry from cache and place new entry. This is an issue, as it can remove the entry from cache, which is still relevant.

The solution to the above problem in LRU is resolved by adding score to each entry in cache. Whenever a new entry comes and it is still there in cache, then the score of the entry is elevated by one. Deprecation function is executed on all the entries of the cache. The item is removed from the cache, whenever the score goes below a certain threshold. This mechanism helps solve the problem which was there in LRU.

Some of the limitations those exist in the current study are that it is only limited to join queries, deprecation function used in the study uses static decrements i.e. by 1, it is not tested for production systems.

Authors of [20] proposed SEaMLeSS(SElf Managing Load Shedding for data Stream management systems), an improvement over Aurora and CTRL, which had improvements on following factors:

1) delay target issue was taken into consideration as one of the main improvements over Aurora[8]
2) the proposed system can work well with complex queries like joins.

[8]Aurora Project http://cs.brown.edu/research/aurora/

TABLE I. STORM AND SPARK STREAMING COMPARISON

	Storm	Spark Streaming
Developed By	Twitter	AMP Lab Berkeley
Processing	One record at a time	Small batches
API's support	Java, Python and many more	Scala, Java and Python
Latency	sub second	few seconds
Batch framework Integration	No	Yes
Implementation Language	Clojure	Scala
Fault Tolerance	Atleast once for every record	Exactly once with no duplicates
Commercial support	NA	Databricks
Use in Production	Twitter, Groupon, Fullcontact	Yahoo, Databricks
Distribution Support	Hortonworks Hadoop Data Platform	Cloudera and MapR

3) the proposed system can automatically adjust headroom factor, which was the problem in CTRL.

In SEaMLeSS delay estimation scheme use CTRL model. It uses queued load method, which is used to estimate the response time of the query network. It is calculated, at a time interval *k*, by multiplying length of queue of each operator with load coefficient. This gives the total load on the queue, which then is used to determine when to shed load. By evaluating the performance against Aurora and CTRL SEaMLeSS outperforms in response time.

Loadstar[21] defines the load shedding in mining as opposed to managing data streams. Loadstar defines the strategy of load shedding dependent on two quality of decision (QoD) measure. QoD metric is used measure uncertainty at the time of data classification (feature classification) when exact data to measure is not available due to load shedding. Markov model is used predict the feature distribution in the data stream, thus making decision using predicted value and QoD metrics. The two schemes which are used for defining load shedding are defined as under:

1) With the use of predicted distribution for prediction of feature values in the next time unit.
2) Prediction model for feature values using Markov model.

Both the schemes are used in Loadstar for classifying the multiple data streams at the time of system overload. The process involves following steps:

1) Multiple data streams are input to load shedding scheme. At the time of system overload feature prediction is used as defined above schemes.
2) Data is them fed to data preparation and analysis block, which cleanses the data.
3) Data after cleaning is then fed to feature predictor block and data stream classifier, which then outputs all the features values from all the streams. If a data is dropped, a feature prediction model predicts the feature based on the historical values. It inserts the predicted value in the output generated.

For experimentation both synthetic and real-life data are used to study the performance of Loadstar. Comparison was done with Naive algorithm, in which loads are shed equally likely. It was seen that Loadstar had lower error rates than Naive algorithm under different levels of overload.

Load shedding in [22] also takes Aurora as the base of the study. The study proposes two methods (operators) random drop and semantic drop. Authors treats this as an optimization problem. The problem defined considers three fundamental decisions i.e., when to shed the load, where to shed the load and how much to shed the load. The decision what and where depends upon a load shedding data structure Load Shedding Road Map(LSRM). This plan consists of three parameters on which drop is executed. The parameters on which drop is executed are cycle savings coefficients, drop insertion plan and percent delivery cursor. After executing the plan LSRM is used for both random load shedding and semantic load shedding.

Aurora[23], which is studied by many researchers in their study is a new DSMS to cater data streams. The load shedding methodology was adopted had greedy approach. This approach identifies a smallest negative slope, in a QoS graph, and traverses till it gets another slope. The data between these slopes are treated as an opportunity of removing the data stream tuples. Removal of the data between the range is achieved by inserting a dropbox. An operator which eliminates all the data which comes inside the box. The consideration of inserting the dropbox depends on the fact that there is minimum decrease in the overall QoS. The process if inserting drops is repeated until the load on the resources comes under a threshold. Another approach that is used is semantic load shedding. This method use filtering process. I this method Aurora use value-based QoS, by building a histogram of the frequency of values. To filter a lowest frequency interval(also called as lowest-utility interval) is used.

In [24], the problem of load shedding is formulated as an optimization problem with the objective function to minimize the inaccuracy of the query results, subject to constraint that input data rate does not exceed the throughput. The approach authors used is random sampling. The class of problem for which random sampling is considered, is sliding window aggregate queries. The algorithm finds the probability of tuple to be considered or discarded. The loss of the tuple, which is discarded, is overcome by scaling the aggregate for an unbiased approximation. Whenever overload happens, load shedders are inserted in many places in query plan. The aim is to reduce the maximum relative error among all the queries.

CONCLUSION

Big data era had opened many fields of research. One of the research area is Volume. When voluminous data enters a system the system can get overwhelmed and throughput of the system starts decreasing. Volume also affects systems which deals with realtime data streams. The data in streams are continuous and upheavals regularly enters the stream which overwhelms the system computing. In order to overcome this limitation DSMS (Data Steam Management Systems) use concept of load shedding. Load shedding is a procedure to drop the incoming packets, from a data stream, based on a strategy. Some of the strategies have been discussed in section -B. Majority of researchers used random method to delete tuples. [19] defines the problem using queueing theory and defines the approach as eviction. The approach use LRU with advancements and has been proved for join queries. [20] proposed SEaMLeSS as an improvement over Aurora and CTRL, which works well with all types of queries. [21] proposed the solution by using Markov model and probability. These two QoD measures are used for feature prediction in continuous data streams. [22] used two methods random load shedding and semantic drop. Drop is executed on three parameters cycle saving, drop insertion plan and percent delivery cursor by inserting drop using LSRM plan. [23] proposed the use of semantic load shedding by filtering tuples. This is done building a histogram containing the frequency of range of values. Load is shed by selecting the lowest interval. [24] uses random sampling to shed load. This is done using probability of processing and discarding the tuple. In this paper, we studied different approaches for Load shedding, which is one of the methods used to reduce volume. We also gave brief introduction to current realtime stream processing engines those provide a good reference of current technologies and approaches in research and industry.

REFERENCES

[1] Google, "Google Trends," http://www.google.co.in/trends/explore#q=%2Fm%2F0296q2&date=today%2012-m&cmpt=q, 2014, [Online; accessed 28-October-2014].

[2] C. Lefevre, "Lhc: the guide, cds. cern. ch/record/1092437," 2008.

[3] M. J. Shaw, C. Subramaniam, G. W. Tan, and M. E. Welge, "Knowledge management and data mining for marketing," *Decision support systems*, vol. 31, no. 1, pp. 127–137, 2001.

[4] M. Mathioudakis and N. Koudas, "Twittermonitor: trend detection over the twitter stream," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 1155–1158.

[5] IBM, "IBM big data and information management," http://www-01.ibm.com/software/data/bigdata/, [Online; accessed 28-October-2014].

[6] D. Tam, "Facebook processes more than 500 TB of data daily," http://www.cnet.com/news/facebook-processes-more-than-500-tb-of-data-daily/, 2012, [Online; accessed 28-October-2014].

[7] Gartner, "Big Data," http://www.gartner.com/it-glossary/big-data/, [Online; accessed 28-October-2014].

[8] Y. Demchenko, P. Grosso, C. de Laat, and P. Membrey, "Addressing big data issues in scientific data infrastructure," in *Collaboration Technologies and Systems (CTS), 2013 International Conference on*. IEEE, 2013, pp. 48–55.

[9] Y. Demchenko, C. de Laat, and P. Membrey, "Defining architecture components of the big data ecosystem," in *Collaboration Technologies and Systems (CTS), 2014 International Conference on*, May 2014, pp. 104–112.

[10] X. Liu, N. Iftikhar, and X. Xie, "Survey of real-time processing systems for big data," in *Proceedings of the 18th International Database Engineering & Applications Symposium*. ACM, 2014, pp. 356–361.

[11] W. Yang, X. Liu, L. Zhang, and L. T. Yang, "Big data real-time processing based on storm," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*. IEEE, 2013, pp. 1784–1787.

[12] B. Institute, "Big Data Survey EUROPE," www.pmone.com/fileadmin/user_upload/doc/study/BARC_BIG_DATA_SURVEY_EN_final.pdf, 2014, [Online; accessed 28-October-2014].

[13] M. Barlow, *Real-Time Big Data Analytics: Emerging Architecture*. " O'Reilly Media, Inc.", 2013.

[14] Apache, "Storm," https://storm.apache.org/documentation/Powered-By.html, [Online; accessed 15-December-2014].

[15] M. Zaharia, T. Das, H. Li, S. Shenker, and I. Stoica, "Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters," in *Proceedings of the 4th USENIX conference on Hot Topics in Cloud Ccomputing*. USENIX Association, 2012, pp. 10–10.

[16] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2002, pp. 1–16.

[17] A. Jain, E. Y. Chang, and Y.-F. Wang, "Adaptive stream resource management using kalman filters," in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. ACM, 2004, pp. 11–22.

[18] C. Olston, J. Jiang, and J. Widom, "Adaptive filters for continuous queries over distributed data streams," in *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. ACM, 2003, pp. 563–574.

[19] S. Gao, T. Scharrenbach, and A. Bernstein, "The clock data-aware eviction approach: Towards processing linked data streams with limited resources," in *The Semantic Web: Trends and Challenges*. Springer, 2014, pp. 6–20.

[20] T. N. Pham, P. K. Chrysanthis, and A. Labrinidis, "Self-managing load shedding for data stream management systems," in *Data Engineering Workshops (ICDEW), 2013 IEEE 29th International Conference on*. IEEE, 2013, pp. 70–76.

[21] Y. Chi, H. Wang, and P. Yu, "System and method for load shedding in data mining and knowledge discovery from stream data," Feb. 17 2009, uS Patent 7,493,346. [Online]. Available: http://www.google.com/patents/US7493346

[22] N. Tatbul, U. Çetintemel, S. Zdonik, M. Cherniack, and M. Stonebraker, "Load shedding in a data stream manager," in *Proceedings of the 29th international conference on Very large data bases-Volume 29*. VLDB Endowment, 2003, pp. 309–320.

[23] D. J. Abadi, D. Carney, U. Çetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, and S. Zdonik, "Aurora: a new model and architecture for data stream management," *The VLDB JournalThe International Journal on Very Large Data Bases*, vol. 12, no. 2, pp. 120–139, 2003.

[24] B. Babcock, M. Datar, and R. Motwani, "Load shedding techniques for data stream systems," in *Proc. Workshop on Management and Processing of Data Streams*. Citeseer, 2003.