Jurnal
Teknologi

Full Paper

RE-STORM: REAL-TIME ENERGY EFFICIENT DATA ANALYSIS ADAPTING STORM PLATFORM

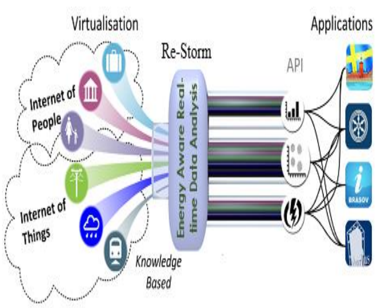
Article history
Received
4 March 2016
Received in revised form
23 June 2016
Accepted
15 September 2016

Rizwan Patan[a], Rajasekhara Babu M.[a*]

School of Computer Science and Engineering, VIT University, Vellore, Tamil Nadu, India

*Corresponding author
mrajasekharababu@vit.ac.in

Graphical abstract



Abstract

It is necessary to model an energy efficient and stream optimization towards achieve high energy efficiency for Streaming data without degrading response time in big data stream computing. This paper proposes an Energy Efficient Traffic aware resource scheduling and Re-Streaming Stream Structure to replace a default scheduling strategy of storm is entitled as re-storm. The model described in three parts; First, a mathematical relation among energy consumption, low response time and high traffic streams. Second, various approaches provided for reducing an energy without affecting response time and which provides high performance in overall stream computing in big data. Third, re-storm deployed energy efficient traffic aware scheduling on the storm platform. It allocates worker nodes online by using hot-swapping technique with task utilizing by energy consolidation through graph partitioning. Moreover, re-storm is achieved high energy efficiency, low response time in all types of data arriving speeds.it is suitable for allocation of worker nodes in a storm topology. Experiment results have been demonstrated the comparing existing strategies which are dealing with energy issues without affecting or reducing response time for a different data stream speed levels. Finally, it shows that the re-storm platform achieved high energy efficiency and low response time when compared to all existing approaches.

Keywords: Big Data, Real-Time data, Stream Computing, scheduling strategies

1.0 INTRODUCTION

1.1 Background

Big data [1] technology numerously increasing various fields and dealing various issues. It deals with two different computing areas like, (1) stream computing [2], and (2) batch computing [3] with three different processing types, (1) real time processing [4], (2) online and Interactive processing [5], and (3) batch processing [6]. In Stream computing focused to deal with both real time and online or interactive data sets. In Batch computing focusing only the batch related data sets. Using populated platforms to dealing that is Hadoop [7] in there Mapreduce [8] and HDFS [9] framework. But in consideration of an Internet of Things (IoT) [10] data aspect different in nature. Figure 1 shows the various tasks dealing the big data technology.

In storm [11] platform it is a designing for processing a data streams, and it is parallel, distributed, fault-tolerance, and low-latency. It is specifically designed for the real-time data processing for an unbounded stream.

A storm programming model [12] designing for directed acyclic graph structure called as a topology. In storm platform two different components are there Spout, Bolt respectively. Spouts and Bolts mechanism in one single cluster of multiple tasks are performing parallel on multiple physical machines. It supports a fault-tolerance by observing all worker nodes any worker node failure the user capable to re-start and reason for failure it provides and it be responsible for guaranty message passing for all worker nodes. And entire process maintaining time based operation mechanism each message passing allocating a specific time slots.

Figure 1 Big Data Fusion

The storm platform is used as a default and simple scheduling strategy of round robin algorithm for assigning a worker node across the topology. And here not considering the inter-process and inter-node traffic, it makes a major impact on performance and One more problem of the default scheduler is that it always causes Storm to use all available worker nodes in a cluster, irrespective of workload. It is identified that in case of a light workload. The operational cost such as electricity cost can be reduced significantly by combining worker nodes it consuming less energy for putting them into sleep for unused idle worker nodes. At present default scheduling strategy of storm will not be satisfying in all the conditions of the stream computing. Our approach is to modifying scheduling strategy to fulfil all the conditions.

1.2 Motivation

By our observation and clear study of the particular issue, found a specific storm platform of the currently dealing with the stream data processing it is an inefficient to solve IoT data effectively. The traffic aware scheduling through high energy efficient characteristics. Focused on both terms it is already did by the two different researcher termed t-storm and re-stream, both are did modification in scheduling aspect but different factor consider our approach is to combination both approaches as a single approach achieve high energy efficiency in traffic aware online scheduling. Moreover, improve performance big data stream computing environment for IoT generated. Label us assigning for this approach named as Re-Storm.

Not considering issues in a Big Data stream computing environment is energy efficiency and proper resource scheduling strategies. For especially processing IoT applications generated data aspect. The both aspects it is effecting more on the performance scaling. Achieving higher energy efficiency, it is a great impact on the whole big data environment. In current scenario different type of techniques are there to be scaled the energy efficiency without affecting response time. Internet of Things increasing and creating huge amounts of real-time data it is a big challenging task in IT industry. Stream computing is suitable the fastest and most efficient solution to get valuable information from big data. Moreover, many data streams from different data sources may form a mixture of the data types that may be incompatible. During the data stream processing, the data fusion phase is very complex, and is completely different from the process of the composition of the massive data streams. In the current stream computing environments, service level objectives and high performance are considered as the main issues, while energy efficiency is ignored. At the time of traffic means data rate increase also platform may capable to handle their tasks. Proper allocation of their worker nodes it is major issue in storm platform.

1.3 Contributions

A summarized workflow of the paper as follows:
The issues propagating and dealing with the IoT data. It is generated data form will be very different to tackle big data stream computing. Issues associated with phasing difficulties while handling IoT Data is

1. Categorizations with definitions of the data stream graph, stream computing and their architecture. Mathematical relations, proof related to the high energy efficiency, low response time, and resource utilization with various traffic levels of a stream in big data stream computing on storm platform.
2. A study presented on the current working nature of the storm platform and categorizing problems facing with default scheduling strategy.
3. Classifying factors considered in a T-storm system to solving. A different traffic levels of streams to arranging efficiently their worker nodes online. By using hot-swapping techniques on top of scheduling strategy.
4. In Re-stream they are trying to improve, the energy efficiency re-streaming their stream graphs to schedule their resources utilize properly throw words achieving high energy efficiency with low response time in big data stream computing.
5. Together are individual modified scheduling strategies but in different logics they are consider to overcome this two different issues one is traffic aware online consolidation and second is improving energy efficiency.
6. Moreover, adapt both different techniques in one single modified scheduling approach towards high energy efficiency, low response time with different traffic levels of streams
7. Finally achieved high energy efficiency, low response time with different traffic levels of

streams in stream computing is called as re-storm.

1.4 Paper Outlines

Outline for paper as follows first part Contain, discussing related work. Second part Contain, problem statement, third part Contain, proposed system. Fourth part contain, experimental setup, fifth part contain, Simulation Setup. Sixth part contain, theoretical evaluation process. Seventh part contains, study and discussion of results and Final part contain, Conclusion.

1.5 Related Work

In real time data IoT data sets are very critical fashion. The real-time data through big data techniques facing difficulty fulfilling. First it is Not satisfactory factors are listed out for literature to see gap and challenges for big data stream computing environment. Existing approaches they are used to full filling their gap and their logics.

In M. Zaharia *et al*. [13], developed advanced a Discretized Streams (D-Stream) on top of Spark platform, using streams for a fault tolerances aspect they focused on fastest recovery without the overhead of duplication the needs among this state and older data observable at a fine granularity explicitly for fast, repeatedly sub-second, recovery from stragglers and faults, in superior scales on data stream computing environments. In this only concentrating the fault tolerances.

In F. Chen *et al*. [14], in this article considered the problem of together scheduling all three levels of a MapReduce process. They presented outlined several heuristics and approximation algorithms to resolve the joint scheduling problem.

1.5.1 Taking Into Consideration Of Performance Issue

In J. Xu *et al*. [15], developed a T-Storm on top storm platform, using this trying to controlling the speed of traffic of a high speed streams by using the hot-swapping technique in traffic aware online resource scheduling algorithm designed to assigning a task in an effective manner however reducing a worker node in a storm system. T-Storm achieves more effective to distribute tasks and moreover improved performance.

In A. M. Aly *et al*. [16], in this introduced a reformed MapReduce architecture model, it allowed data to be pipelined among operators. This covers the MapReduce programming model beyond batch processing, and can decrease processing time and expand system utilization for batch jobs with various sizes of data.

In M. Dusi *et al*. [17], author introduced to the Blackmon technique similar to the s4, storm but focusing on web based content sensing data.

Social media connecting with smart devices and sensing huge amount of streaming data increasing double by two to three years, this is also one type of IoT data because connected with GPS temperature etc. devices. It is only suitable needs of the one type of big data only data forms is changing in different sectors as well as changing the traffic levels.

1.5.2 Taking Into Consideration Of Energy Issue

In D. Sun *et al*. [18], developed a Re-Stream on top storm platform, using this trying to reallocating their streams directed acyclic graphs and dividing as a two segments, one critical vertex and non-critical vertex based strategies throw words reducing energy efficiency to managing their stream graphs. Performance aware dynamic scheduling of critical vertices algorithm.

In K. Kanoun *et al*. [19], in this article evolving a low power many core architecture model for big data stream computing with integrating able to adapt scalable, low-power and reconfigurable cores for reducing energy with different memory levels. This is very expansive to modify hardware equipment and reconfigure with the IcyFlex4 Core for achieving high performance.

The above all are trying fulfill their needs using different platforms and computing engines. Fully not considering overall needs of IoT generated real time sensing data using big data. Partially fulfill very few techniques, so seeing this literature find a gap on different type of organization sensing different traffic levels of data with using different types of sensor devices.

1.6 Problem Statement

To develop novel performance scaling management policies that improves efficiency of real-time big data Fusion using Big Data Stream Computing Environment for real time online data handling with over traffic regulation.

2.0 METHODOLOGY

2.1 Proposed System

Re-Storm platform it is design to adapting the Storm platform. It is open source platform to dealing streaming data. In this energy aware issues are not considered. Our proposed Model as energy efficiency performance scaled platform. Overview of Re-Storm process workflow is shown in Figure 2.

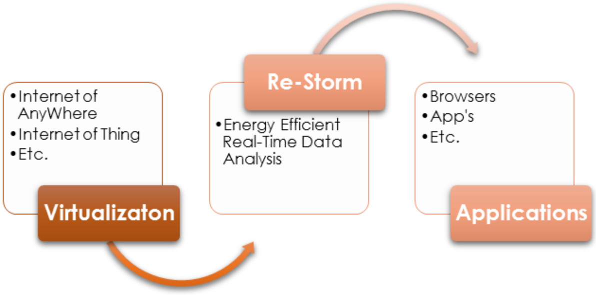


Figure 2 Re-Storm Processing Overview

To controlling the Stream computing platform in big data while computing IoT data. It is facing difficulty so search for further gap filling to overcome this point. In Stream Computing Storm [11] platform is open sourced with fulfilling the maximum needs of existing streaming data aspect. In this one using scheduling strategy is Round Robin [20] as by default one. It is not satisfying the data traffic is very high in the point processing slow and energy consumption very for that purpose many failures. Devices generated data. Process flow in internally in the storm platform shown in Figure 3.

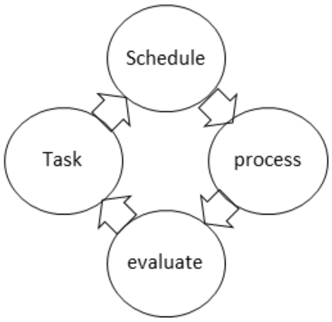


Figure 3 internal workflow of storm

In the time of computing real-time data, three different type of data sensing devices are there in IoT (1) event driven communication X. Meng *et al*. [21], (2) periodical communication E. A. Billard *et al*. [22], (3) Communication on request. For processing this kind of data it is want for more computation power. Modifying scheduling approach passion to meet the different type of data sets handling very effective way. Modified algorithm mention below algorithm 1, in this algorithm mention calculating there time complexity using. For the farther computation steps mention in algorithm itself with double slash.

The below algorithm 1 is working with three strategies. First is optimizing their stream graphs by using DVFS (Dynamic Voltage Frequency Scaling) [23] approaches to reduce energy. Second is Hot-swapping [24] technique for online rescheduling their worker nodes. Third is assigning both one and two approaches to scaling their performance.

Algorithm 1 Energy aware Real-Time Data Analysis

Input: Real Time Data sets as source
Output: Computed analyzed Data with High performance

Begin
Get Real-Time Data sets
Process
Compute No. of executers
E= (No. of Requests DAG+1)
// Not exceeded $\gamma\left(\frac{N_s}{K}\right)$
Compute No. of worker nodes
N_w= (number of worker nodes in cluster * number cores per worker node) - (number of acker tasks)
// Calculate by N_w
Compute Current Load each Executer
l_i= (E/N_w)*100
//calculate by l_i

Loop for i=1 to No. of iteration for schedule
// for traffic aware consolidation using factor for compute is X
Update "Q" permitting to the X_{ij}
Partition by α% of giving to the performance ratio.
Send data to the worker nodes
Computing load by using the formula (1)
Increment i;
loop end
Computing energy for each node measure by (5)
Return traffic aware schedule allocation factor is X
Computing Energy for total system measure by (4)
Computing performance by (6)
End

2.2 Simulation Setup

For analysis our model determination is frame a simulation environment. Hardware requirement are used to creating simulation environment. Intel i3 processor, 16 GB RAM, 512 Mbps speed network connectivity, using 4 core machines 10 for testing our model. All the modified scheduling approaches applied on top of the Ischeduler of the storm platform. Software requirement using for the computing the results, storm 0.10.0, Ubuntu server Version 14.01, java 1.8.25, zookeeper 3.4.0, python 3.0. Deploying modified energy efficient self-scheduling algorithm monitoring their results are observing on the StormUI. The values are taken by the particular task is given below Table 1.

Table 1 Producing experimental value

No.	Bounds	Values
1	Monitoring load and Estimation period	50 sec.
2	Coefficient estimation (a)	0.5
3	Schedule fetching period p(s_f)	30 sec.
4	Schedule generation period p(s_g)	500 sec.
5	Each Experiment Running Time E_{RT}	1500 sec.

For the above values are taken for calculating the performance metric. Storm is used to estimating a results StormUI. For synchronizing the worker nodes using the NTP protocol D. Mills [25] standards. Average time bound are considering is 10 minutes. And total rest of mechanism as like on default strategy of storm. Focusing on the two different aspect to getting for accurate results. One is a StormUI for monitoring the values of minutes' pulse and also worker nodes count. Second one is a NTP protocol is getting there results seconds pulse with very accurate for using to monitor and as well as traffic level scaling.

2.3 Theoretical Evaluation Process

For usage of the mathematical calculation process followed by simplifying the streaming graph. To computing a stream graphs, process is given below

2.3.1 Categorization of Stream Computing

A continues sequence of data sets is called stream. Infinite sequence of data sets is called continues streams, more than one stream processed at a same time is called parallel streams. A program used to process continues data streams is called stream computing. Generally, stream computing processing by the Data stream graph (DSG) is derived by the Directed Acyclic Graph (DAG). Below definitions providing correct measurable view of DSG.

2.3.2 Definitions

DSG is considered as DAG, is a resultant by the DAG. Every G is continuing with two parameters G=(V_G, E_G), in this V_G is a vertices of the group and E_G is an Edges of group. And sub graph G_s, $\forall V_i \epsilon V_{G_s}$ than $\forall V_G \epsilon V_{G_s}$, path it is going route of the DAG. It is a direction of an (V_s, V_e) if $S \neq E$ the starting point and ending point same the graph is not a directed graph and it is mostly indicating null node. Topological Sort (TS) is another characteristic graph the graph not containing any cycle's formations.it must be a DAG. DAG means it is Topology sort order. Partitioning a graph, it is considering by the DAG based on a TS us splitting the vertices of a graph. Partition graph (G_P) is a partitioning vertices based on the topology sort $G_P = \{G_{P_1}, P_{P_2}, P_{P_3} \ldots\}$ for each partition will having $G_{P_1} = \{v_1, v_2, v_3, \ldots\} \in G, G_{P_2} = \{v_1, v_2, v_3, \ldots\} \in G$. It is subgroup containing $\forall i \neq j, i,j \in \{1, n\}$ then $G_{P_i} \cap G_{P_j} = \emptyset$ $\cup_1^n G_{P_i} \neq V_G$

$$G = (V_G, E_G)$$

Each Vertex and edge containing some tuples

$$V_G = (id_v, f_v, c_v i_v, o_v)$$
$$E_G = (id_e, c_e)$$
$$P(V_s, V_e)$$

Where

V_G- $\{V_1, V_2, V_3, \ldots \ldots \ldots V_i\}$= Graph Vertices,

E_G- $\{E_1, E_2, E_3, \ldots \ldots \ldots E_j\}$=Graph Edges,

V_s, V_e= Stating vertex and Ending vertex,

id_v, f_v, c_v, i_v, o_v = vertices identification, function, computation cost, input data stream, output data stream.

id_e, C_e= identification of directed edge, communication cost of directed edge.

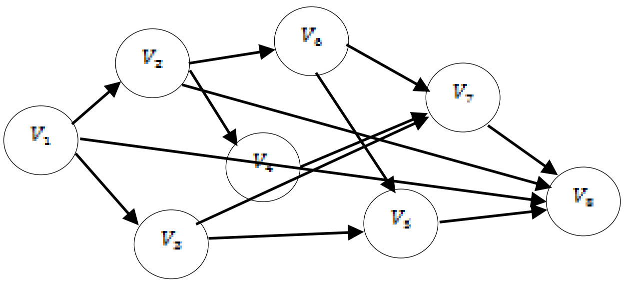


Figure 4 Example DAG

In the above graph the graph is a stat's at V_1 and ends V_2 it is not containing a circles and it ϵG $\{V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8\}$ and in this one sub graph are assume for example $\{V_1, V_3, V_5, V_8\} \in G$ and path is two paths in this above graph $\{V_1, V_2, V_4, V_6, V_7, V_8\}$ and $\{V_1, V_3, V_5, V_8\} \in G$. the above graph moreover TS. The workflow nature it is telling order of a storm execution flow. Each task depending upon an another task it won't be executing parallel and worker node assigning a particular task it is relative to their task strength in that time achieve more simulative results. For example, DAG shown in Figure 4.

Below are provide appropriate equations to calculate an energy efficiency, response time and traffic aware energy aware scheduling strategies S. Zhuravlev *et al.* [26] and their mathematical relations among the energy efficient Rizwan *et al.* [27] with response time, energy efficient inter-node processing, energy efficient traffic consolidation and worker node assigning with low response time.

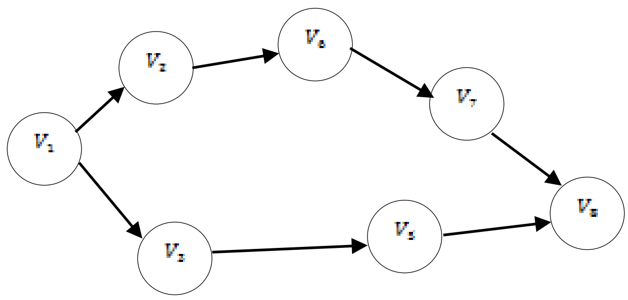
The mathematical formulas and their proofs, calculation energy and response time for DAG

Executers assigning a slot computing by using formula (1)

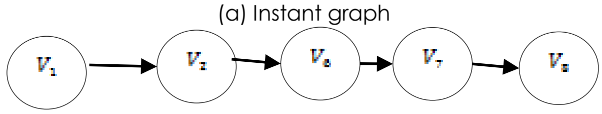
$$s^* = \text{argmin}_{q \in Q} \sum_{i'=1}^{N_e} (r_{\pi(i')i} \sum_{w(s) \neq w(q)} (x_{i'j}) + (r_{i'\pi(i)})) \quad (1)$$

2.3.3 Scheduling Strategies For Data Stream Graphs

In commonly allocation of resources is done by the appropriate scheduling aspects like array format, graph format, tree format etc. different type of mechanism follows different types scenarios to allocate their resources. In concentrating the various scheduling strategies are there for solving Data Stream Graph corresponding Directly Acyclic Graph. Previously are discoursed by the DAG and their role in stream and related issues. Applying scheduling schemas example on top Figure 4 it modifies shown in Figure 5.



(a) Instant graph



(b) Logical graph

Figure 5 Example Stream DAG splinter structures

Finding energy consumption with response time as factors, and traffic consolidation computed by Eq. 2

$$E_{cn_i} = \int_{t_0}^{t_{n-1}} P_{cn_i}(\mu_i(t))\,dt \quad (2)$$

Where computation node is "cn_i", computation node Energy is "E_{cn_i}", computation node Power is "P_{cn_i}", node indicator is "i", $[t_0, t_{n-1}]$ time interval is "t".

The specific component is Power consumption by computer is network, memory, Disk storage, CPU. Applying DVFS approach for power consumption of i^{th} computation node computing by Eq. (3) according to the Eq. 2

$$P_{cn_i}(\mu_i(t))\begin{cases} P^{idl}_{cn_i} = \alpha . P_{cn_i}(MAX), X \in [0,1] \\ P^{spd}_{cn_i} = (1-\alpha). P_{cn_i}(MAX). \mu_i(t),\ \mu_i(t) \in [0,1] \end{cases} \quad (3)$$

Theorem-1: for computing by Eq. 4 an E_{sys} is energy consumption by system. For big data stream computing environment.

$$E_{sys} = \sum_{i=0}^{num-1} E_{cn_i} \quad (4)$$

Theorem-2: for computing by Eq. 5 an E_{cn_i} is energy consumption, $[t_0, t_{n-1}]$ is time intervals, divided in to $[t_0, t_1, \ldots, t_{n-1}]$ by formula proof [18]

$$E_{cn_i} = \left(\alpha . P_{cn_i}(MAX)\right).(t_{n-1} - t_0) + \left((1 - \alpha). P_{cn_i}(MAX)\right)\sum_{k=1}^{n-1}\left(\mu_{i_{k-1}}, [(t]_k, t_{k-1})\right) \quad (5)$$

Where E_{cn_i} is according to the Eq. 4, measure the energy consumption while the total power consumed [28] by the each computing node of the data storage, for directly measured by the fully utilized state.

For Estimating the performance using the formula Eq. 6

$$PF_j = w \times \frac{\frac{1}{T_j}}{\left(\sum_{vn_j \in s}[\frac{1}{T_j}]\right)} \quad (6)$$

3.0 RESULTS AND DISCUSION

The testing source input and applying testing strategies M. Rajasekhara Babu *et al.* [29] in this article applied different testing strategies. collected data sets in the CityPulse data set [30] collection web forum for source input for our technique analysis. It contains different real time data sets are having to open source access. And addition to add the real time additional tools for creating a regulation environment. To deliver an information with irrelevant traffic ranges on storm platform. The results for accuracy are divided in to three different categories of streams tuple ranges are considered. One the range it should be initial stage of tuple range 0-100. In this aspect are considering values variant manner but range are taken it is a constant one tuples. Figure 6 Show the tuple range will be 0-100 and are submitting their tuples with different traffic mediums to testing their accuracy y = $0.0003x^2 + 0.0343x + 13$ $R^2 = 0.9923$ for storm platform and modified re-storm platform y = $4.8273x^{0.3222}$ $R^2 = 0.6348$.

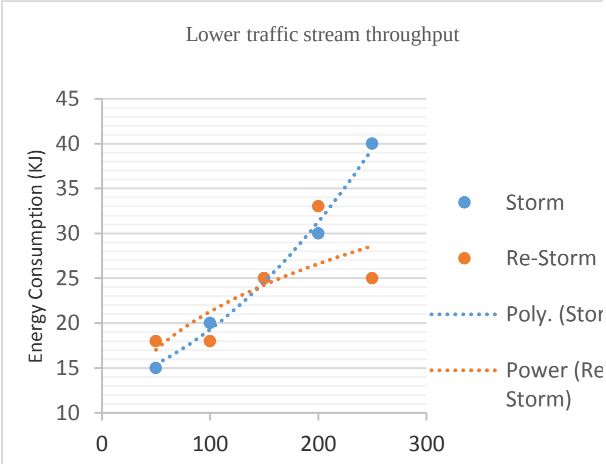


Figure 6 Data stream load will be minor state tuples generation range 0-100

Second medium is the range it should be average of tuple range 100-250. In this aspect are considering values variant manner but range are taken it is a constant one tuples. Figure 7 Show the tuple range will be 100-250 and 0are submitting their tuples with different traffic mediums to testing their accuracy y = $29.434e^{0.0017x}$ $R^2 = 0.9829$ for storm platform and modified re-storm platform y = $-0.0003x^2 + 0.189x + 15.943$ $R^2 = 0.9189$.

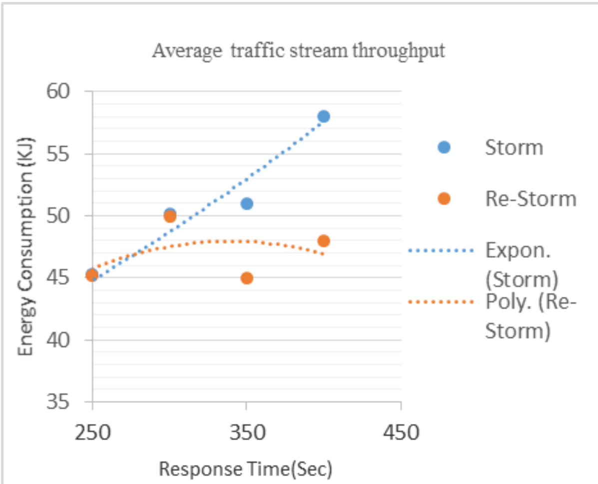


Figure 7 Data stream load will be average tuples generation range 100-250

Third medium is the range it should be maximum stage of tuple range 250-above. In this aspect are considering values variant manner but range are taken it is a constant one tuples. Figure 8 Show the tuple range will be 250-above and are submitting their tuples with different traffic mediums to testing their accuracy y = $25.407e^{0.0021x}$ $R^2 = 0.9185$ for storm platform and modified re-storm platform y = $-0.0004x^2 + 0.3607x - 34.457$ $R^2 = 0.8664$.

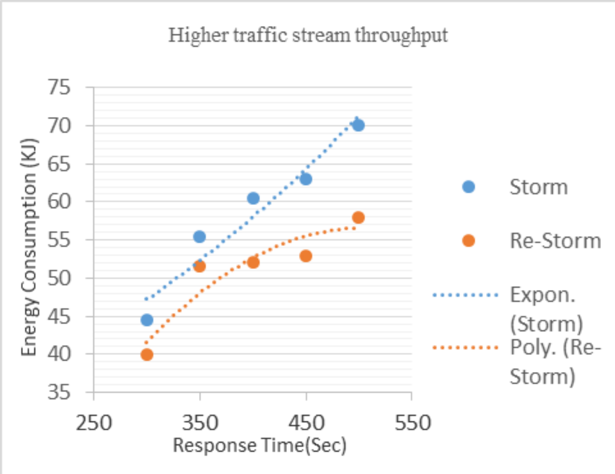


Figure 8 Data stream load will be average tuples generation range 250-above

4.0 CONCLUSION

In this paper, proposed an Energy Efficient Traffic aware resource scheduling and Re-Streaming Stream Structure on replacing their default scheduling strategy of storm entitled as re-storm. Online real-time data is very dissimilar arriving rate and it doesn't have constant traffic sensing medium. It was discussed what are the importance are there for optimizing stream graphs and energy efficiency needs for big data stream computing environment. Our approach considered two factors for improving their performance efficiency. First is modifying their scheduling strategy with irrelevant traffic stream support and second is optimizing their graphs using critical path elimination. To maintain a constituency for different traffic medium data. Moreover, upgrading 20-30 % efficiency in stream computing. Finally, it creates a huge impact on overall all big data stream computing environment.

References

[1] H. Mohanty. 2015. Big Data: An Introduction. Big Data SE – 1. vol. 11. H. Mohanty, P. Bhuyan, and D. Chenthati, Eds. Springer India. 1-28.

[2] D. Sun, G. Zhang, W. Zheng, and K. Li, 2015. *Key Technologies for Big Data Stream Computing*. Big Data, Chapman and Hall/CRC. 193-214.

[3] J. Li, Z. Bao, and Z. Li. 2015. Modeling Demand Response Capability by Internet Data Centers Processing Batch Computing Jobs. *Smart Grid, IEEE Transactions on*. 6(2): 737-74.

[4] D. Bhattacharya. 2013. Analytics On Big Fast Data Using A Real Time Stream Data Processing Architecture. 34.

[5] Interactive or Online Processing. 2015. Wikispaces. [Online]. Available: http://dis-dpcs.wikispaces.com/3.3.5+Batch,+Online+%26+real+time+Processing. [Accessed: 05-Nov-2015].

[6] V. Beal. 2015. Batch Processing, Webopedia. QuinStreet Inc. [Online]. Available: http://www.webopedia.com/TERM/B/batch_processing.html. [Accessed: 05-Nov-2015].

[7] J. P. Verma, B. Patel, and A. Patel. 2015. Big Data Analysis: Recommendation System with Hadoop Framework. *2015 IEEE Int. Conf. Comput. Intell. Commun. Technol*. 92-97.

[8] M. Bhandarkar. 2010. Mapreduce Programming With Apache Hadoop. Parallel & Distributed Processing (IPDPS). *2010 IEEE International Symposium on*. 1.

[9] What is Hadoop Distributed File System (HDFS)? - Definition from WhatIs.com. [Online]. Available: http://searchbusinessanalytics.techtarget.com/definition/Hadoop-Distributed-File-System-HDFS. [Accessed: 07-Feb-2016].

[10] E. Benkhelifa, M. Abdel-Maguid, S. Ewenike, and D. Heatley. 2014. The Internet Of Things: The Eco-System For Sustainable Growth. Computer Systems and Applications (AICCSA), 2014 IEEE/ACS. *11th International Conference on*. 836-842.

[11] Storm (event processor). Wikipedia. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Storm_(event_processor)&redirect=no. [Accessed: 25-Nov-2015].

[12] Storm (software) - Wikipedia, the free encyclopedia. [Online]. Available: https://en.wikipedia.org/wiki/Storm_(software). [Accessed: 07-Feb-2016].

[13] M. Zaharia, T. Das, H. Li, T. Hunter, S. Shenker, and I. Stoica. 2013. Discretized Streams: Fault-Tolerant Streaming Computation At Scale. *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*. 423-438.

[14] F. Chen, M. Kodialam, and T. V Lakshman. 2012. Joint Scheduling Of Processing And Shuffle Phases In Mapreduce Systems. *INFOCOM, 2012 Proceedings IEEE*. 1143-1151.

[15] J. Xu, Z. Chen, J. Tang, and S. Su. 2014. T-Storm: Traffic-aware Online Scheduling in Storm. *IEEE Int. Conf. Distrib. Comput. Syst*.

[16] A. M. Aly, A. Sallam, B. M. Gnanasekaran, W. G. Aref, M. Ouzzani, and A. Ghafoor. 2012. M3: Stream Processing on Main-Memory MapReduce. *Icde*. 8: 1253-1256.

[17] M. Dusi, N. D'Heureuse, F. Huici, A. Di Pietro, N. Bonelli, G. Bianchi, B. Trammell, and S. Niccolini. 2012 Blockmon: Flexible And High-Performance Big Data Stream Analytics Platform And Its Use Cases. *NEC Tech. J.* vol. 7(2): 102-106.

[18] D. Sun, G. Zhang, S. Yang, W. Zheng, S. U. Khan, and K. Li. 2015. Re-Stream: Real-Time And Energy-Efficient Resource Scheduling In Big Data Stream Computing Environments. *Inf. Sci. (Ny)*. Mar. 2015.

[19] K. Kanoun, M. Ruggiero, D. Atienza, and M. Van Der Schaar. 2014. Low Power and Scalable Many-Core Architecture for Big-Data Stream Computing. *Medianetlab.Ee. Ucla.Edu*. 20: 468-473.

[20] Round-robin scheduling. Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Round-robin_scheduling. [Accessed: 20-Nov-2015].

[21] X. Meng and T. Chen. 2013. Event-Driven Communication For Sampled-Data Control Systems. *Am. Control Conf. (ACC), 2013*. 1: 3002-3007.

[22] E. A. Billard and J. C. Pasquale. 1993. Effects of Periodic Communication on Distributed Decision-Making. *IEEE Int. Conf. Syst. Man, Cybern*.

[23] E. Le Sueur and G. Heiser. 2010. Dynamic Voltage And Frequency Scaling: The Laws Of Diminishing Returns. *Proc. 2010 Int. Conf. Power aware Comput. Syst*. 1-8.

[24] Hot swapping. Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Hot_swapping#References. [Accessed: 02-Nov-2015].

[25] D. Mills. 1985. Network Time Protocol (NTP). Rfc958.

[26] S. Zhuravlev, J. C. Saez, S. Blagodurov, A. Fedorova, and M. Prieto. 2013. Survey of Energy-Cognizant Scheduling Techniques. *IEEE Trans. PARALLEL Distrib. Syst*. 24(7): 1447-1464.

[27] Rizwan Patan, Rajasekhara Babu. 2015. M. A Study Analysis of Energy Issues In Big Data. *Intrnational J. Appl. Eng. Res*. 10(6): 15593-15609.

[28] Rajasekhara Babu M., Krishna P. V. and Khalid. 2013. A Framework For Power Estimation And Reduction In Multi-Core Architectures Using Basic Block Approach. *Int. J. Commun. Networks Distrib. Syst. Inderscience Enterp. Ltd*. 10(1): 40-51.

[29] Rajasekhara Babu M. and A. J. B. Alok N. Bhatt. 2013, Automation Testing Software that Aid in Efficiency Increase of Regression Process. *Recent Patents*

Comput. Sci. 6(2): 107-114.
[30] CityPuls. CityPulse Dataset Collection. [Online]. Available: http://iot.ee.surrey.ac.uk:8080/datasets.html. [Accessed:10-Nov-2015].