International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 03 | Mar-2018 www.irjet.net p-ISSN: 2395-0072

HHH- A Hyped-up Handling of Hadoop based SAMR-MST for DDOS Attacks in Cloud

S.Ezhilarasi[1]

[1]*Assistant Professor, Department of CSE, Velammal college of Engineering and Technology, Madurai, TamilNadu, India*

Abstract - *Hadoop is a cloud framework that supports the processing of large datasets in a distributed computing environment. Mapreduce technique is being used in hadoop for processing and generating large datasets. A key benefit of mapreduce is that it automatically handles failures and hides the complexity of fault tolerance from the user. DDoS attacks have a history of flooding the victim network with an enormous number of packets, hence exhausting the resources and preventing the legitimate users to access them. After having standard DDoS defense mechanism, still attackers are able to launch an attack. A novel scheme is proposed to detect DDoS attack efficiently by using MapReduce programming model, SAMR (Self Adaptive MapReduce) scheduling algorithm is being introduced which can find slow tasks dynamically by using the historical information recorded on each node to tune parameters. SAMR reduces the execution time when compared with existing systems.*

Key Words: **DDoS, Hadoop, Map Reduce, Cloud Computing, SAMR, MST**

1. INTRODUCTION

Mapreduce is used in cloud computing because of hiding the complexity of fault tolerance from the programmer. SAMR mapreduce scheduling technique is being developed which uses the historical information and find the slow nodes and launches backup tasks. The historical information is stored in each nodes in XML format. It adjusts time weight of each stage of map and reduce tasks according to the historical information respectively. It decreases the execution time of mapreduce job and improve the overall mapreduce performance in the heterogeneous environment. In this paper we are tuning the parameters using Minimum Spanning Tree(MST) clustering technique and then assigning tasks to each node thus improving the performance of hadoop in the heterogenous environment. With HDFS federation, multiple Namenode servers manage namespaces and this allows for horizontal scaling and performance improvements

1.1 Distributed denial of service (DDoS) attacks

DDoS attack is a distributed, large scale coordinated at-tempt of flooding the network with an enormous amount of packets which is difficult for victim network to han-dle, and hence the victim becomes unable to provide the services to its legitimate user and also the network performance is greatly deteriorated. This attack exhausts the resources of the victim network such as bandwidth, memory, computing power etc. The system which suffers from attacked or whose services are attacked is called as "primary victim" and on other hand "secondary victims" is the system that is used to originate the attack. These secondary victims provide the attacker, the ability to wage a more powerful DDoS attack as it is difficult to track down the real attacker. Denial of Service (DoS) attacks is used to consume all the resources of the target machine (victim's services) Distributed denial of service (DDoS) attack is some sort of malicious activity or a typical behavior, which cooperate the availability of the server's resources and prevents the legitimate users from using the service. DDOS attacks are not meant to alter data contents or achieve illegal access, but in that place they target to crash the servers, generally by temporarily interrupting or suspending the services of a host connected to the Internet. DOS attacks can occur from either a single source or multiple sources. Multiple source DOS attacks are called distributed denial-of service (DDOS) attacks.

A Denial of Service (DoS) attack is an attempt to make a computer resource unavailable to normal users. The Dos attacks are becoming more powerful due to bot behavior. Attack that leverages multiple sources to create the denial-of-service condition is known as The Distributed Denial of Service (DDoS) attack. DDoS attacks are big threats to internet services. HTTP flooding attack is one of the typical DDos attack, in that hosts are sending large amount of request to target website to exhaust its resources [1]. Now a day there is massive growth in internet traffic. Due to this many DDoS attack detection systems facing a problem. A Distributed Denial of service (DDos) attack can employ hundreds or even thousands of computers that have been previously flooded by HTTP GET packet.

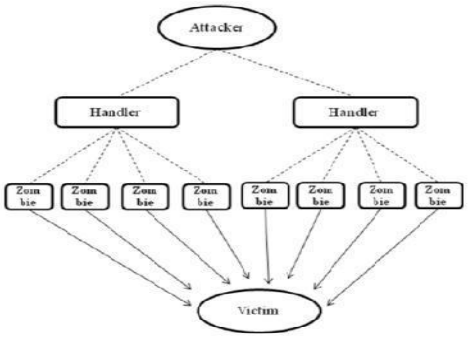

Fig.1. Architecture of DOS

1.2 Literature Survey

Hadoop defaultly uses FIFO technique in which the tasks are given priority in the order they arrived. This technique takes more response time for slower jobs when compared to faster jobs. SARS(Self-Adaptive Reduce Scheduling) can decide the start time points of each reduce tasks dynamically according to each job context, includes the job completion time. LATE (Longest Approximate Time to End) scheduling improves the execution in hadoop by finding real slow tasks. SAMR improves the execution in hadoop by finding real slow tasks

2. Detection of DDoS Using Hadoop

Hadoop provides the tools for processing vast amounts of data using the MapReduce framework and, implements the Hadoop Distributed File System (HDFS) [73,74]. It can be used to process vast amounts of data in parallel on large clusters in a reliable and fault-tolerant fashion. There are two distinct algorithms that have been proposed:

1) Counter based method: This method relies on three key parameters: time interval which is the duration during which packets are to be analyzed, threshold which indicates frequency of requests and unbalance ratio which denotes the anomaly ratio of response per page requested between specific client and server.

The number of requests from a specific client to the specific URL within the same time duration is counted using the masked timestamp. The reduce function aggregates the number of URL requests, number of page re-quests, and total server responses between a client and a server. Finally values per server are aggregated by the algorithm. When the threshold is crossed and the unbalance ratio is higher than normal ratio from h, the clients are marked as attackers.

The key advantage of utilizing this algorithm is its low complexity. However the authors have indicated that the threshold value determination could be a key deciding factor in the implementation but do not offer any further information on how to determinate the value.

2) Access pattern based method: This method is based on a pattern which differentiates the normal traffic from DDoS traffic. This method requires more than two MapReduce jobs:

First job gets the access sequence to the web page between a client and a web server and computes the spending time and the bytes count for each request of the URL;

Second job finds infected hosts by comparing the access sequence and the spending time among clients trying to access the same server.

3. Proposed Model

Even though there are many DDoS solutions proposed by different researchers, literature shows that there has been no effective way proposed to defend against DDoS attacks. To Detect DDoS, Counter based and Pattern Based Algorithm are quietly famous approach in Hadoop but still the major challenges are that they still have a lot of orientation towards batch processing and because of this ad hoc query jobs are delayed.

Hadoop is open source software based on scalability, distributed and reliability concept. It is best suited for large scale *i.e.* cloud, provides optimum analyzed data by distributing cloud into multiple chunks. It uses scheduling algorithms for MapReduce.

3.1 SAMR Scheduling

To overcome the shortcoming of Hadoop scheduling SAMR scheduling was proposed. After a job is committed, SAMR splits the job into map and reduce tasks, and assigns them to a series of nodes. In the interim, it reads historical information which stored on every node and updated it after every execution. In that case, SAMR adjusts time weight of each stage of map and reduce tasks according to the historical information respectively.

As a result, it gets the progress of each task accurately and finds which tasks need backup tasks. It identifies slow nodes and classifies them into the sets of slow nodes dynamically. SAMR launches the backup tasks on the basis of information of these slow nodes and ensures that the backup tasks are not slow tasks. It gets the final results of the tasks when either slow tasks or backup tasks finish first.

The proposed model that uses SAMR Counter based algorithm that improve the efficiency as it reads historical information which stored on every node and updated it after every execution. This give more accurate Progress score and finds which task needs backup task.

This model inputs three parameters: time interval, threshold and unbalance ratio, which are stored in HDFS through packet loader. The packet collector receives IP packets from trace files on the disk, and writes them to HDFS. IP packets are stored in the binary format. The threshold and unbalanced ratios for server are passed as parameters along with the timestamp. Job starts at the client and Job Tracker running SAMR scheduler splits the job into map and reduces tasks and assigns them to a series of nodes while doing thus it also reads historical information which is stored on every node and is updated after every execution. SAMR then adjusts time weight of each stage of map and reduce tasks as per the historical information respectively. Thus, it gets the progress scores of each task accurately and finds which of the tasks need backup tasks to run and also identifies the slow nodes and classifies them into the sets of slow nodes dynamically.

It gets the final results of the fine-grained tasks when either slow tasks or backup tasks finish first. The map task generates keys to classify the requests and response HTTP messages. Then, the reduce task summarizes the HTTP

request messages and marks the abnormal traffic load by comparing it with the threshold. The map task generates keys to classify the requests and response HTTP messages. Then, the reduce task summarizes the HTTP request messages and marks the abnormal traffic load by comparing it with the threshold. The results are saved back to HDFS.

Tentative results show that SAMR significantly decreases the time of execution up to 25% com-pared with Hadoop's scheduler

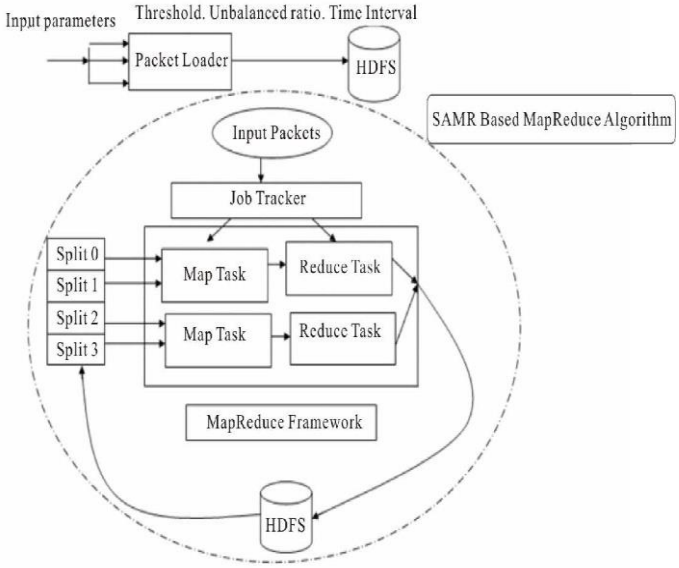

Fig.2. SAMR scheduling

3.2 Theoretical Foundation

The framework provides a default partitioning function but the user is allowed to override this function by a custom partitioning. The locations of these buffered pairs on the local disk are passed back to the master. The master then forwards these locations to the reduce workers. When a reduce worker is notified by the master about these locations, it uses remote procedure calls to read the buffered data from the local disks of mapworkers.

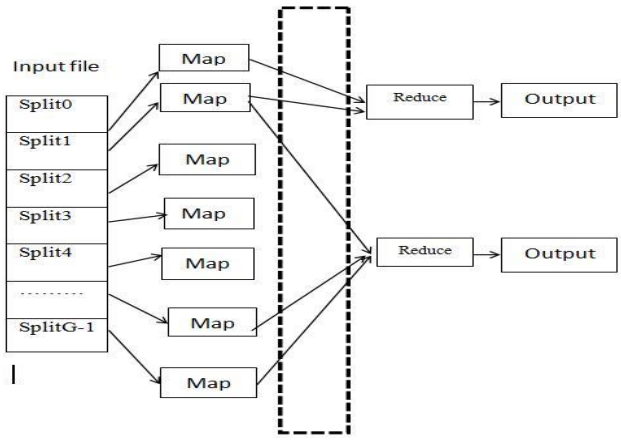

Fig.3 Map Reduce Framework

3.3 MST Methodology

The SAMR technique uses the historical information that is being stored in each node and using that information it finds the real slow tasks. Then it maps the slow tasks and reduces the slow tasks. In this paper we use the Minimum Spanning Tree (MST) clustering technique to tune the parameters in the historical information and finding the slow tasks very accurately. The proposed Minimum Spanning Tree (MST) algorithm can solve even the most difficult clustering issues. It requires the number of clusters that we are going to use in our process. The algorithm finds k centroids, one for each cluster. Depending on the location of the centroid the result will vary.

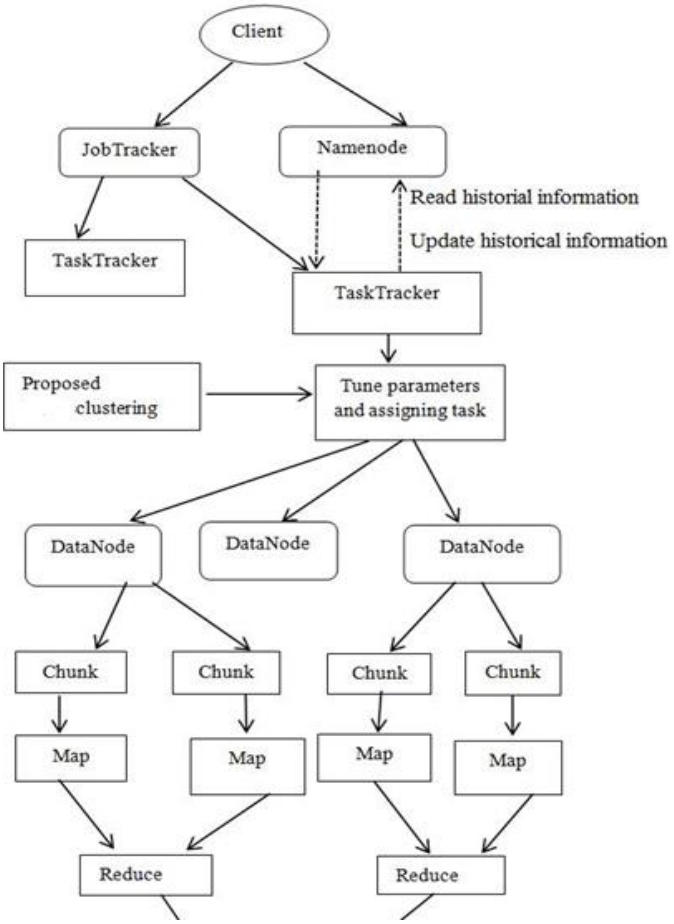

Fig.4 MapReduce Implementation

3.4 SAMR Algorithm

Step 1: Start procedure

Step 2: input: Key/Value pairs

output: Statistical results

Step 3: read historical information

Step 4: tune parameters using proposed k-means clustering

Step 5: Find slow tasks

Step 6: Find slow tasktrackers

Step 7: Launch back up tasks

Step 8: Using the results update the historical information

Step 9: End procedure

Table -1: Comparison of existing with proposed systems

S.NO	ALGORITHM	ADVANTAGE	DISADVANTAGE
1	First In First Out(FIFO) scheduling	Reduces response time due to speculative execution. Works well in case of short jobs.	Uses fixed threshold for selecting tasks to reexecute. Can't identify which tasks to be reexecuted on fast nodes correctly.
2	Self-adaptive Reduce scheduling(SARS)	Reduces completion time. Decrease respo nse time	Only focuses on reduce process.
3	Longest approximate time to end(LATE) scheduling	Robustness to heterogeneit y. Address the problem of how to robustly maximize performan ce.	Does not compute remaining time for tasks correctly and can't find real slow tasks. Poor performance due to the static manner in computing the progress of the tasks.
4	Self-adaptive mapreduce(SAM R) scheduling	Decreases the execution time of map reduce heterogeneo us environment.	Does not find the slow jobs accurately.

3. CONCLUSION

It discusses the history the of DDoS attacks along with some major incidents to provide a better understanding and gravity of the problem. The paper includes latest techniques such as Hadoop along with other available techniques for prevention and detection of distributed denial of service attacks so that a comprehensive solution can be developed with several detection layers to trap the intrusion keeping in mind the limitations of these prevention and detection techniques. In this paper I propose a method to improve the efficiency of the map reduce scheduling algorithms. It works better than existing map reduce scheduling algorithms by taking less amount of computation and gives high accuracy. I used the proposed Minimum spanning Trees (MST) clustering algorithm together with the Self-Adaptive MapReduce(SAMR) algorithm. SAMR reduces the execution time by 25% when compared with FIFO and 14% when compared with LATE.

REFERENCES

[1] T. Kitten, "DDoS: Lessons from Phase 2 Attacks," 2013.

[2] S. Zargar, J. Joshi and D. Tipper, "A Survey of Defense Mechanisms against Distributed Denial of Service (DDoS) Flooding Attacks," Communications Surveys & Tutorials, IEEE, Vol. PP, No. 99, 2013, pp. 1-24

[3] "CERT Advisory: SYN Flooding and IP Spoofing At-tacks," CERT® Coordination Center Software Engineer-ing Institute, Carnegie Mellon, 2010. http://www.cert.org/advisories/CA-1996-21.html

[4] CERT, "Tech Tips: Denial of Service Attacks," CERT® Coordination Center Software Engineering Institute, Car-negie Mellon, 2010. http://www.cert.org/tech_tips/denial_of_service.html

[5] R. Mackey, "'Iranian Cyber Army' Strikes Chinese Web-site," New York Times Lede Blog, 2011.

[6] DDoS-for-Hire Service Is Legal and Even Lets FBI Peek in, Says a Guy with an Attorney," 2012. http://www.ddosdefense.net

[7] J. Kirk, "Mt. Gox under Largest DDoS Attack as Bitcoin Price Surges," 2013. http://www.computerworld.com/s/article/9238118/Mt._G ox_under_largest_DDoS_attack_as_bitcoin_price_surges

[8] "Mstream Distributed Denial of Service Tool (Zombie Detected) (DdosMstreamZombie)," 2013. http://www.iss.net/security_center/reference/vuln/dd os-m stream-zombie.htm

[9] N. McAllister, "GoDaddy Stopped by Massive DDoS Attack," 2012. http://www.theregister.co.uk/2012/09/10/godaddy_d dos_ attack/

[10] D. Kravetz, "Anonymous Unfurls 'Operation Titstorm'," Wired Threat Level Blog, 2010.

[11] K. Zetter, "Lazy Hacker and Little Worm Set off Cyber-war Frenzy," 2009. http://www.wired.com/threatlevel/2009/07/mydoom/

[12] L. Greenemeier, "Estonian Attacks Raise Concern over Cyber "Nuclear Winter"," Information Week, 2007. http://www.informationweek.com/estonian-attacks-raise-c oncern-over-cybe/199701774

BIOGRAPHIES

She received B.E(CSE) from Raja College of Engineering and Technology, Madurai, M.E(CCE) from Pavendar Bharathidasan College of Engineering and Technology, Trichy, M.B.A(ISM) form Bharathiyar University, Coimbatore and PGDB from Bharathiyar University, Coimbatore. She is currently working as Assistant Professor in department of CSE in Velammal College of Engineering and Technology, Madurai. She has published various papers in four different International journals and published papers at eight International conferences and sixteen National conferences.