DETECTION OF FLOODING BASED DOS ATTACK ON HADOOP DATANODE

Silky kalra, Anil Lamba

Abstract— **Denial of service (DOS) attacks present an internet –wide threat. SYN flooding is one of the DOS attack that degrades the performance of the system. It is the most powerful attack used by hackers to harm the organization. It causes service outages and loss of millions, depending on the duration of attack. SYN flood DOS attack involves sending too many SYN packets to the destination. The attack use up the resources and memory of the server. This can lead to the hang of the server machine. Dos attacks are a persistent problem for several reasons. First, they are one of the earliest attacks to perform, and they attain quick results. They are the most common attacks that an administrator with a system that's always connected to the internet can expect. Well it is an attempt to make a system or server unavailable for legitimate users and, finally, degrades the service. This is accomplished by flooding the server's request queue with fake requests. After this, server will not be able to interact with the requests of legitimate users.**

This paper focused on dos attack and how to overcome it using "HADOOP". For this here an open source tools and softwares are used. We proposed a technique to detect flooding based attack at hadoop datanode and analysed the working of hadoop distributed file system. We have also shown hadoop's effectiveness in attack scenario, discussed various motivation for deployment

Index Terms— Denial-of-Service(DOS), Datanode, Hadoop, hadoop distributed file system, Mapreduce,

I. INTRODUCTION

SYN flooding is one of the typical DoS attacks that exploit normal TCP connections between a client and a target web server by sending too many SYN packets to the destination server.this can lead to a crash or hang of server machiene. As the volume of Internet traffic increases explosively year after year, the Intrusion Detection Systems (IDSes) have faced the issue on how to assure both scalability and accuracy of analyzing the DoS attack from these huge volume of data. In recent years, several approaches have been proposed to solve this issue. Dimensionality reduction methods such as Principal Component Analysis (PCA) enables to classify large volume of traffic by separating the normal behavior from anomalies [8].

However, these schemes usually require too excessive computing cycles to apply to actual systems. Sketch-based studies focus on memory efficiency by utilizing hash tables.Though Liu et al. [9] proposed a two-level sketch approach to reduce memory consumption and searching complexity while boosting accuracy, their technique still needs sufficient memory space and complex computation. Hadoop is an open-source distributed cluster plat-form that includes a distributed system, HDFS and the programming model, MapReduce.

The Iranian Cyber Army: On December 17, 2009, attackers replaced the front page of Twitter.com with an image of the Iranian flag along with text including: "This site has been hacked by the Iranian Cyber Army." The attackers did not actually gain access to Twitter's servers, but instead changed the twitter.com domain name to point to a different IP address (the IP address of the machine hosting the "hacked by ..." page). Twitter took down its home page entirely within minutes and twitter.com remained down for a couple of hours.[8]

The attacks on the major Web sites began in early February 2000, with the first major attack being on Yahoo! On February 7 [10]. The surprise attack took the Yahoo! Site down for more than three hours. It was based on the Smurf attack, and most likely, the Tribe Flood Network technique. At the peak of the attack, Yahoo! was receiving more than one gigabit per second of data requests.

Anonymous and "Operation Titstorm" -In February, 2010, a group of people loosely connected through Internet forums calling itself "Anonymous" executed a DDoS attack against the Australian Parliament's website. The attack took down the site for two days. On the same day that Anonymous attacked the parliament's website, the group also defaced the Prime Minister's website, briefly replacing the front page with pornographic images. The attack was termed "Operation Titstorm" by its organizers, referring to a mandatory Internet filtering policy proposed by Australia's ruling party designed in part to counter pornography.

Hadoop has created a lot of interest in large-scale analytics (the MapReduce part of Hadoop). This kind of "divide and conquer" algorithm methodology has been used for numerical analysis for many years as a way of dealing with problems that were known to be bigger than the biggest machine available.

MapReduce and its open-source implementation Hadoop were originally optimized for large batch jobs such as web index construction. However, another use case has recently emerged sharing a MapReduce cluster between multiple users, which run a mix of long batch jobs and short interactive queries over a common data set

Manuscript received Jun, 2014.

***Silky kalra**, computer science and engineering , Haryana engineering collage, ., Yamunanagar, India, 9996378451*
***Anil Lamba**, computer science and engineering, Haryana engineering collage, Yamunanagar, India, 9416344059,*

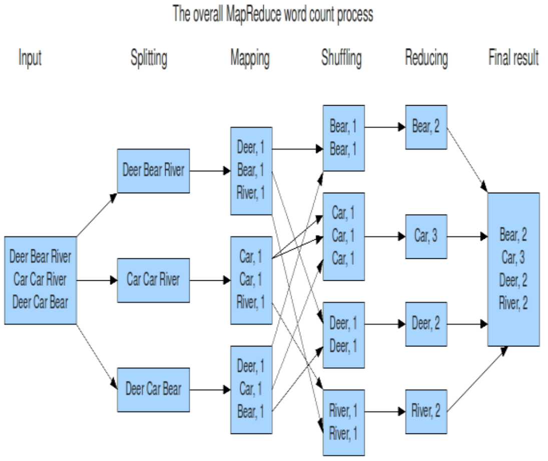


Figure 1: Mapreduce word count process

As shown in fig1 MapReduce is an elegant way of structuring this kind of algorithm that isolates the analyst/programmer from the specific details of managing the pieces of work that get distributed to the available machines, as well as an application architecture that doesn't depend on any specific structuring of the data.

II. RELATED STUDY

In [1], Prashant Chauhan, Abdul Jhummarwala , Manoj Pandya in December, 2012 provided an overview of Hadoop. This type of computing can have a homogeneous or a heterogeneous platform and hardware. The concept of cloud computing and virtualization has derived much momentum and has turned a more popular phrase in information technology. Many organizations have started implementing these new technologies to further cut down costs through improved machine utilization, reduced administration time and infrastructure costs. Cloud computing also confronts challenges. One of such problem is DDoS attack so in this paper author will focus on DDoS attack and how to overcome from it using honeypot. For this here open source tools and software are used. Typical DDoS solution mechanism is a single host oriented and in this paper focused on a distributed host oriented solution that meets scalability.

In [2], Jin-Hyun Yoon, Ho-Seok Kang and Sung-Ryul Kim, in 2012, proposed a technique called "triangle expectation" is used, which works to find the sources of the attack so that they can be identified and blocked. To analyze a large amount of collecting network connection data, a sampling technique has been used and the proposed technique is verified by experiments.

In [3], B. B. Gupta, R. C. Joshia, Manoj Misra, in 2009, the main aim of this paper is First is to demonstrate a comprehensive study of a broad range of DDoS attacks and defense methods proposed to fight with them. This provides a better understanding of the problem, current solution space, and future research scope to fight down against DDoS attacks. Second is to offer an integrated solution for entirely defending against flooding DDoS attacks at the Internet Service Provider (ISP) level.

In [4], Yeonhee Lee, Youngseok Lee, in 2011 proposed a novel DDoS detection method based on Hadoop that implements an HTTP GET flooding detection algorithm in MapReduce on the distributed computing platform.

In [5], Matei Zaharia, Dhruba Borthakur, Joydeep Sen Sarma, Khaled Elmeleegy, Scott Shenker, Ion Stoica, in April 2009, provided an overview of Sharing a MapReduce cluster between users. It is attractive because it enables statistical multiplexing (lowering costs) and allows users to share a common large data set. They evolved two simple techniques, delay scheduling and copy-compute splitting, which improve throughput and response times by factors of 2 to 10. Although we concentrate on multi-user workloads, our techniques can also increase throughput in a single-user, FIFO workload by a factor of 2.

In [6], Radheshyam Nanduri, Nitesh Maheshwari, Reddy Raja, Vasudeva Varma, **in 2011**, proposed an approach which attempts to hold harmony among the jobs running on the cluster, and in turn minify their runtime. In their model, the scheduler is made reminful of different types of jobs running on the cluster. The scheduler tries to assign a task on a node if the incoming task does not affect the tasks already running on that node. From the list of addressable pending tasks, our algorithm pick out the one that is most compatible with the tasks already running on that node. They bring up heuristic and machine learning based solutions to their approach and attempt to maintain a resource balance on the cluster by not overloading any of the nodes, thereby cutting down the overall runtime of the jobs. The results exhibit a saving of runtime of around 21% in the case of heuristic based approach and approximately 27% in the case of machine learning based approach when compared to Yahoo's Capacity scheduler.

In [7], Dongjin Yoo, Kwang Mong Sim, in 2011, compare contrasting scheduling methods, evaluating their features, strengths and weaknesses. For settlement of synchronization overhead, two categories of studies; asynchronous processing and speculative execution are addressed. For delay scheduling in Hadoop, Quincy scheduler in Dryad and fairness constraints with locality improvement are addressed.
In[8] Dileep Kumar Gupta, Abhishek Mishra et all, discussed the availability problem in the existing framework for e-Governance and also provide a better solution to solve availability problem in future framework for e--governance in cloud computing. They have proposed here a new modified Model by adding one filtering module in the existing algorithm. Basically, DoS attacks are used for two purposes. First is to consume the resources and second is to consume the bandwidth of network. In both cases, either resources or bandwidth of network are scarce. The most difficult part to defend against DoS attack is that, how to differentiate between normal traffic and malicious traffic? DoS attack has two solutions. It blocks the
packets either from the port numbers or by the IP addresses. When blocking of packets is done by port number then it will

block all the packets coming from the particular port. For example, if we allow TCP packets to come into the network so that all UDP packets will drop and we cannot confirm that all TCP packets are coming from authenticated user so that we have used IP filtering mechanism to protect DoS attack.

.

III. PROPOSED FRAMEWORK

We have observed the effectiveness of hadoop in different attack scenario. Hadoop consist of two core components: the job management framework that handles the map and reduce tasks and hadoop distributed file system. We introduce the syn flooding attack with the help of code attached to hadoop and then captured it with wireshark. Datanode of HDFS receives the blocks of data and deletes the flooded blocks and a fair scheduler for better job management in which small adhoc query jobs can be executed with periodic jobs (for monitoring) in parallel that prevents the degradation in performance of distributed file system.

IV. METHODOLOGY

Step1: Flooding on hadoop datanode

Step2: Capturing the live traffic

Step3: copying that file to hadoop user

Step4: job assignment

Step5: map and reduce task

Step6: delete the flooded blocks of data

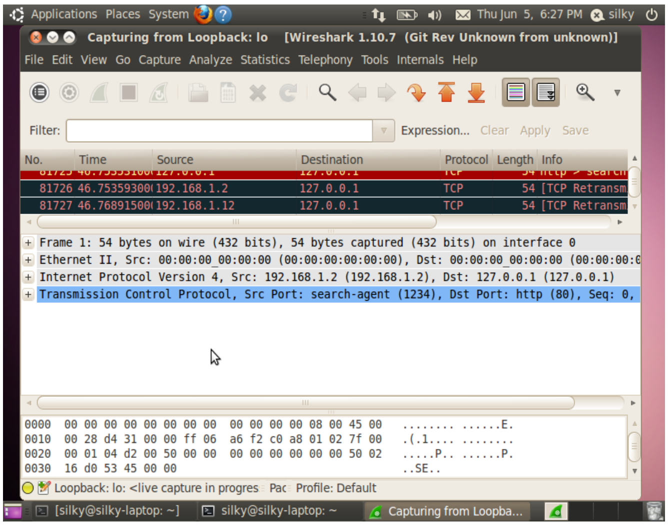
Step7: Collecting results

V. IMPLEMENTATION

We added a code for flooding at local host in c

```
int main (void)
{

    int s = socket (PF_INET, SOCK_RAW, IPPROTO_TCP);
    char datagram[4096] , source_ip[32];
    struct iphdr *iph = (struct iphdr *) datagram;
    struct tcphdr *tcph = (struct tcphdr *) (datagram + sizeof
(struct ip));
    struct sockaddr_in sin;
    struct pseudo_header psh;

    strcpy(source_ip , "192.168.1.2");

    sin.sin_family = AF_INET;
    sin.sin_port = htons(80);
    sin.sin_addr.s_addr = inet_addr ("127.0.0.1");
```

And then captured the traffic using wireshark . and then pass it to hadoop datanode.



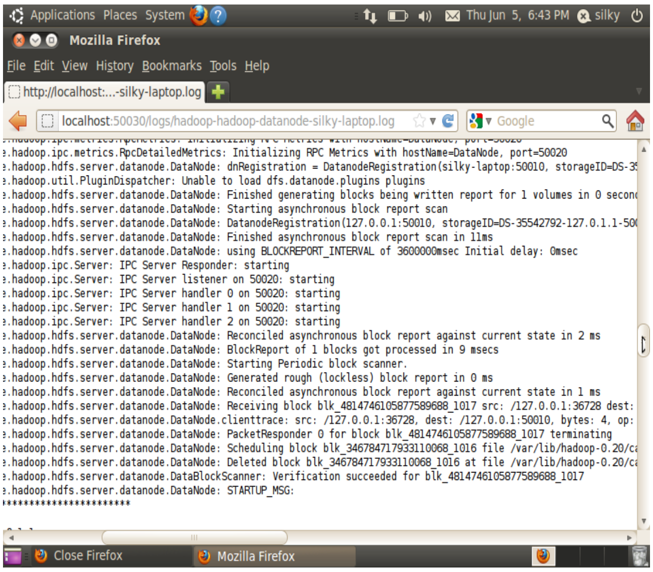
Applications Places System
Thu Jun 5, 6:27 PM silky
Capturing from Loopback: lo [Wireshark 1.10.7 (Git Rev Unknown from unknown)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
Loopback: lo: <live capture in progres Pac Profile: Default
[silky@silky-laptop: ~]
silky@silky-laptop: ~
Capturing from Loopba...

```
No.   Time         Source          Destination   Protocol Length Info
81725 46.75353100( 127.0.0.1       127.0.0.1     TCP          54 http > search
81726 46.75359300( 192.168.1.2     127.0.0.1     TCP          54 [TCP Retransm
81727 46.76891500( 192.168.1.12    127.0.0.1     TCP          54 [TCP Retransm

+ Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
+ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:0
+ Internet Protocol Version 4, Src: 192.168.1.2 (192.168.1.2), Dst: 127.0.0.1 (127.0.0.1)
+ Transmission Control Protocol, Src Port: search-agent (1234), Dst Port: http (80), Seq: 0,

0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ........ ......E.
0010  00 28 d4 31 00 00 ff 06  a6 f2 c0 a8 01 02 7f 00   .(.1.... ........
0020  00 01 04 d2 00 50 00 00  00 00 00 00 00 00 50 02   .....P.. ......P.
0030  16 d0 53 45 00 00                                  ..SE..
```

Figure 2: Shows the capturing of traffic

VI. RESULTS

Datanode of hadoop distributed file system receives the blocks of data and deletes the flooded blocks.



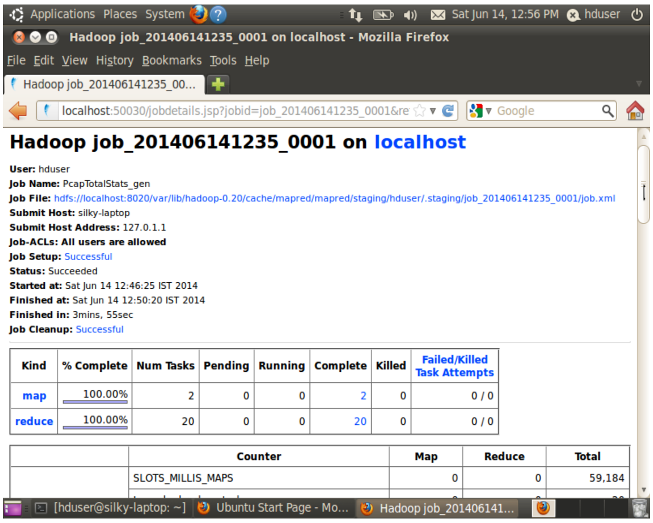
Applications Places System
Thu Jun 5, 6:43 PM silky
Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:...-silky-laptop.log
localhost:50030/logs/hadoop-hadoop-datanode-silky-laptop.log
Google
Close Firefox
Mozilla Firefox

```
e.hadoop.ipc.metrics.RpcDetailedMetrics: Initializing RPC Metrics with hostName=DataNode, port=50020
e.hadoop.hdfs.server.datanode.DataNode: dnRegistration = DatanodeRegistration(silky-laptop:50010, storageID=DS-3
e.hadoop.util.PluginDispatcher: Unable to load dfs.datanode.plugins plugins
e.hadoop.hdfs.server.datanode.DataNode: Finished generating blocks being written report for 1 volumes in 0 secon
e.hadoop.hdfs.server.datanode.DataNode: Starting asynchronous block report scan
e.hadoop.hdfs.server.datanode.DataNode: DatanodeRegistration(127.0.0.1:50010, storageID=DS-35542792-127.0.1.1-50
e.hadoop.hdfs.server.datanode.DataNode: Finished asynchronous block report scan in 11ms
e.hadoop.hdfs.server.datanode.DataNode: using BLOCKREPORT_INTERVAL of 3600000msec Initial delay: 0msec
e.hadoop.ipc.Server: IPC Server Responder: starting
e.hadoop.ipc.Server: IPC Server listener on 50020: starting
e.hadoop.ipc.Server: IPC Server handler 0 on 50020: starting
e.hadoop.ipc.Server: IPC Server handler 1 on 50020: starting
e.hadoop.ipc.Server: IPC Server handler 2 on 50020: starting
e.hadoop.hdfs.server.datanode.DataNode: Reconciled asynchronous block report against current state in 2 ms
e.hadoop.hdfs.server.datanode.DataNode: BlockReport of 1 blocks got processed in 9 msecs
e.hadoop.hdfs.server.datanode.DataNode: Starting Periodic block scanner.
e.hadoop.hdfs.server.datanode.DataNode: Generated rough (lockless) block report in 0 ms
e.hadoop.hdfs.server.datanode.DataNode: Reconciled asynchronous block report against current state in 1 ms
e.hadoop.hdfs.server.datanode.DataNode: Receiving block blk_4814746105877589688_1017 src: /127.0.0.1:36728 dest:
e.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /127.0.0.1:36728, dest: /127.0.0.1:50010, bytes: 4, op:
e.hadoop.hdfs.server.datanode.DataNode: PacketResponder 0 for block blk_4814746105877589688_1017 terminating
e.hadoop.hdfs.server.datanode.DataNode: Scheduling block blk_3467847179331l0068_1016 file /var/lib/hadoop-0.20/c
e.hadoop.hdfs.server.datanode.DataNode: Deleted block blk_3467847179331l0068_1016 at file /var/lib/hadoop-0.20/c
e.hadoop.hdfs.server.datanode.DataBlockScanner: Verification succeeded for blk_4814746105877589688_1017
e.hadoop.hdfs.server.datanode.DataNode: STARTUP_MSG:
**********************
```

Figure 3: Logs of datanode

Here job details are shown:

- Job_id is assigned to each job
- It maps and reduce tasks


Applications Places System
Sat Jun 14, 12:56 PM hduser
Hadoop job_201406141235_0001 on localhost - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Hadoop job_201406141235_00...
localhost:50030/jobdetails.jsp?jobid=job_201406141235_0001&re
Google
[hduser@silky-laptop: ~]
Ubuntu Start Page - Mo...
Hadoop job_201406141...

Hadoop job_201406141235_0001 on localhost

User: hduser
Job Name: PcapTotalStats_gen
Job File: hdfs://localhost:8020/var/lib/hadoop-0.20/cache/mapred/mapred/staging/hduser/.staging/job_201406141235_0001/job.xml
Submit Host: silky-laptop
Submit Host Address: 127.0.1.1
Job-ACLs: All users are allowed
Job Setup: Successful
Status: Succeeded
Started at: Sat Jun 14 12:46:25 IST 2014
Finished at: Sat Jun 14 12:50:20 IST 2014
Finished in: 3mins, 55sec
Job Cleanup: Successful

Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	100.00%	2	0	0	2	0	0 / 0
reduce	100.00%	20	0	0	20	0	0 / 0

	Counter	Map	Reduce	Total
	SLOTS_MILLIS_MAPS	0	0	59,184

Figure 4: Shows map and reduce tasks and their completion

Completion graphs of both map and reduce tasks. Here we have 2 map tasks and 20 reduced tasks. It has completed both the map tasks and all reduce tasks.

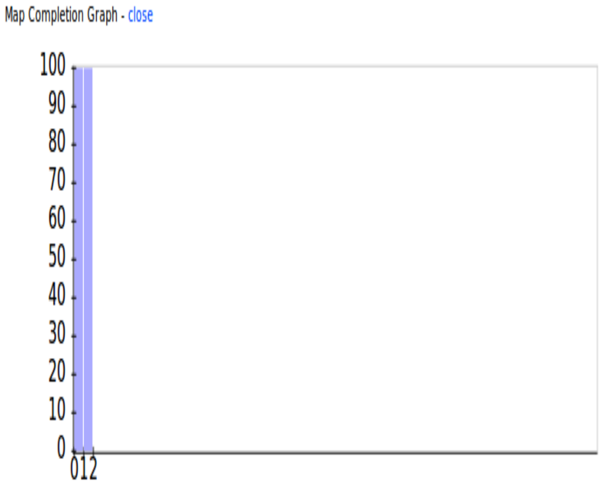


Figure 5: Map completion graph

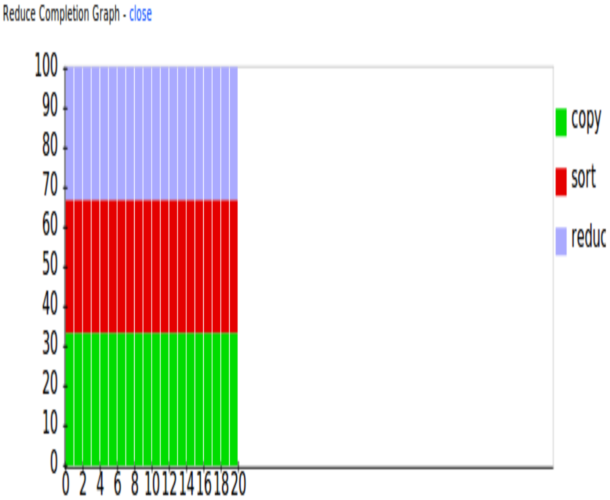


Figure 6: Reduce completion graph

VII. Conclusion

Detection of DoS attack is focused in this paper by Hadoop based DoS detection model. Flooding based DoS attack involves large number of packets sent to the hadoop within a short span of time. We have justified it with wireshark. We use a distributed detection system to efficiently detect these attacks at an early stage. MapReduce technique of Hadoop is used for distributing the analysis task to idle workers in the Hadoop cluster and gets that job done efficiently and accurately.

Appendix

A. *List Of Abbreviations*

Acronym	Description
HDFS	Hadoop distributed file system
DOS	Denial of service

B. *TOOLS USED FOR IMPLEMENTATION*

Hadoop
Wireshark

Acknowledgment

While preparing this article I got unlimited support and spirit from my family and friends. I am highly grateful to the Principal, Haryana Engineering College, Jagadhri for providing me the opportunity to carry out my research. I express my deep sense of gratitude to my Research Supervisor Er. Amit lamba, for his expert guidance, stimulating discussions as well as continued impetus throughout the period of this work. I would like to thank wholeheartedly Er. Anil Lamba, Chairperson, Department of CSE, HEC Jagadhri for encouraging solicited advice and all kind of help extended during the period of this work. I also express my gratitude to Er. Suveg Moudgil for his valuable and precious help during this research.
Finally, I am thankful to all those people who are directly or indirectly related with this work. Above all I pay my regards to the Almighty for always having his hand on me.

REFERENCES

[1] Prashant Chauhan, Abdul Jhummarwala, Manoj Pandya, “Detection of DDoS Attack in Semantic Web” International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 4– No.6, December 2012 – www.ijais.org, pp. 7-10

[2] Jin-Hyun Yoon, Ho-Seok Kang and Sung-Ryul Kim, Division of Internet and Media, Konkuk University,Seoul, Republic of Korea h2jhyoon@gmail.com, hsriver@gmail.com, kimsr@konkuk.ac.kr , pp. 200-203

[3] B. B gupta. , Joshi, R. C. and Misra, Manoj(2009), 'Defending against Distributed Denial of Service

Attacks: Issues and Challenges', Information Security Journal: A Global Perspective, 18: 5, 224 — 247

[4] Yeonhee Lee, Chungnam National University, Daejeon, 305-764, Republic of Korea, yhlee06@cnu.ac.kr

[5] Matei Zaharia, Dhruba Borthakur, Joydeep Sen Sarma,Khaled Elmeleegy_ Scott Shenker,Ion Stoica , “Job Scheduling for Multi-User MapReduce Clusters” ,University of California, Berkeley , Facebook Inc _Yahoo! Research Electrical Engineering and Computer Sciences University of California at Berkeley Technical Report No. UCB/EECS-2009-55 http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-55.html April 30, 2009

[6] Radheshyam Nanduri, Nitesh Maheshwari, Reddy Raja, Vasudeva Varma,” Job Aware Scheduling Algorithm for MapReduce Framework” by In 3rd IEEE International Conference on Cloud Computing Technology and Science Athens, Greece. Report No: IIIT/TR/2011/-1, Centre for Search and Information Extraction Lab,International Institute of Information Technology,Hyderabad - 500 032, INDIA, November 2011, pp. 724-729

[7] Dongjin Yoo, Kwang Mong Sim , A comparative review of job scheduling for mapreduce Multi-Agent and Cloud Computing Systems Laboratory, School of Information and Communication, Gwangju Institute of Science and Technology (GIST), Gwangju, IEEE CCIS2011, 978-1-61284-204-2/11/$26.00 ©2011 IEEE, pp.353-358

[8] Robert Mackey, “'Iranian Cyber Army' Strikes Chinese Website,” New York Times Lede Blog, January 12, 2010, accessed October15,2010,http://thelede.blogs.nytimes.com/2010/01/12/iranian-cyber-army-strikes-chinese-site/.

[9] David Kravetz, “Anonymous Unfurls 'Operation Titstorm',” Wired Threat Level Blog, February 10, 2010, accessed October 15, 2010, http://www.wired.com/threatlevel/2010/02/anonymous-unfurls-operation-titstorm/

[10] Jose Nazario, “Politically Motivated Denial of Service Attacks.”

[11] Mirkivic and P. Reiher, A Taxonomy of DDoS Attack and DDoS Defense Mechanisms, ACM SIGCOMM CCR, 2004

[12] Jaideep Dhok, Nitesh Maheshwari, and Vasudeva Varma.Learning based opportunistic admission control algorithm for mapreduce as a service. In ISEC ’10: Proceedings of the 3rd India software engineering conference, pages 153–160. ACM, 2010

[13] Richard O. Duda, Peter E. Hart, and David G. Stork. Pattern Classification (2nd Edition). Wiley-Interscience, edition, November 2000

[14] Geoffrey Holmes Bernhard Pfahringer Peter Reutemann Ian H. Witten Mark Hall, Eibe Frank. The weka data mining software. SIGKDD Explorations, 11(1), 2009.

[15] Hadoop Distributed File System. http://hadoop.apache.org/common/docs/current/hdfs design.html.

[16] Fair Scheduler. http://hadoop.apache.org/common/docs/r0.20. 2/fair scheduler.html.

[17] Capacity Scheduler. http://hadoop.apache.org/common/docs/r0.20.2/capacity scheduler.html.

[18] Quan Chen, Daqiang Zhang, Minyi Guo, Qianni Deng, and Song Guo. Samr: A self-adaptive mapreduce scheduling algorithm in heterogeneous environment. In Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on, 29 2010.

[19] J. Polo, D. Carrera, Y. Becerra, V. Beltran, J. Torres, and E. Ayguade and. Performance management of accelerated mapreduce workloads in heterogeneous clusters. In Parallel Processing (ICPP), 2010 39th International Conference on, pages 653 –662, 2010.

[20] Aameek Singh, Madhukar Korupolu, and Dushmanta Mohapatra. Server-storage virtualization: integration and load balancing in data centers. In Proceedings of the 2008 ACM/IEEE conference on Supercomputing, SC ’08, pages53:1–53:12, Piscataway, NJ, USA, 2008. IEEE Press.

Silky kalra
Persuing M.Tech (CSE), Haryana engineering collage, Jagadhri, Haryana, India . Her research interest areas include Big Data, Wireless Networks, Security.

Anil Lamba
M.Tech (CSE), Associate Professor, Haryana engineering collage, Jagadhri, Haryana, India . His research areas include Wireless Networks, Mobile IP.