

A Review on HADOOP MAPREDUCE-A Job Aware Scheduling Technology

Silky Kalra¹, Anil lamba²

1 M.Tech Scholar, Computer Science & Engineering, Haryana Engineering College Jagadhari, Haryana,

2 Associate Professor, Department of Computer Science & Engineering, Haryana Engineering College Jagadhari, Haryana

ABSTRACT:

Big data technology remodels many business organization perspective on the data. conventionally, a data framework was like a gatekeeper for data access. such frameworks were built as monolithic “scale up”, self contained appliances. Any added scale required added resources, which often exponentially multiplies cost. One of the key approaches that have been at the center of the big data technology landscape is Hadoop. This research paper includes detailed view of various important components of Hadoop, job aware scheduling algorithms for mapreduce framework, various DDOS attack and defense methods.

KEYWORDS: *Distributed Denial-of-Service (DDoS), Hadoop, Job aware scheduling, Mapreduce,*

I. INTRODUCTION

MapReduce is currently the most famous framework for data intensive computing. MapReduce is motivated by the demands of processing huge amounts of data from a web environment. MapReduce provides an easy parallel programming interface in a distributed computing environment. Also MapReduce deals with fault tolerance issues for managing multiple processing nodes. The most powerful feature of MapReduce is its high scalability that allows user to process a vast amount of data in a short time. There are many fields that benefit from MapReduce, such as Bioinformatics, machine learning, scientific analysis, web data analysis, astrophysics, and security. There are some implemented systems for data intensive computing, such as Hadoop. An open source framework, Hadoop resembles the original MapReduce.

With increasing use of the Internet, Internet attacks are on the rise. Distributed Denial-of-Service (DDoS) in particular is increasing more. There are four main ways to protect against DDoS attacks: attack prevention, attack detection, attack source identification, and attack reaction. DDoS attack is one such threat which is distributed form of Denial of Service attack in which service is consumed by an attacker and legitimate user can not use the service. DDoS attack is one such threat which is distributed form of Denial of Service attack in which service is consumed by attacker and legitimate user can not use the service. We can find a solution against DDoS attack, but they are based on a single host and lacks performance so here Hadoop system for distributed processing is used.

II. HADOOP

The GMR(Google map reduce) was invented by Google back in their earlier days so they could usefully index all the rich textural and structural information they were collecting, and then present meaningful and actionable results to users. MapReduce(you map the operation out to all of those servers and then you reduce the results back into a single result set), is a software paradigm for processing a large data set in a distributed parallel way. Since Google’s MapReduce and Google file system (GFS) are proprietary, an open-source MapReduce software project, Hadoop, was launched to provide similar capabilities of the Google’s MapReduce platform by using thousands of cluster nodes[1]. Hadoop distributed file system (HDFS) is also an important component of Hadoop, that corresponds to GFS. Hadoop consists of two core components: the job management framework that handles the map and reduces tasks and the Hadoop Distributed File System (HDFS). Hadoop’s job management framework is highly reliable and available, using techniques such as replication and automated restart of failed tasks.

2.1 Hadoop cluster architecture

A small Hadoop cluster includes a single master and multiple worker nodes. The master node consists of a JobTracker, TaskTracker, NameNode and DataNode. A slave or *worker node* acts as both a DataNode and TaskTracker, though it is possible to have data-only worker nodes and compute-only worker nodes. These are normally used only in nonstandard applications. Hadoop requires [Java Runtime Environment \(JRE\)](#) 1.6 or higher. The standard start-up and shutdown scripts require [Secure Shell](#) to be set up between nodes in the cluster.

In a larger cluster, the HDFS is managed through a dedicated NameNode server to host the file system index, and a secondary NameNode that can generate snapshots of the namenode's memory structures, thus preventing file-system corruption and reducing loss of data.

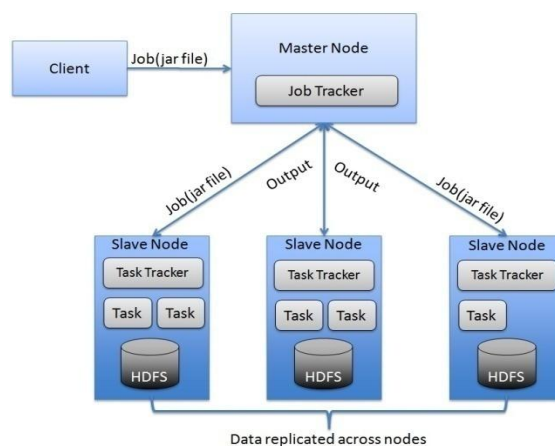


Figure 1. shows Hadoop multinode cluster architecture which works in distributed manner for MapReduce problem.

Similarly, a standalone JobTracker server can manage job scheduling. In clusters where the Hadoop MapReduce engine is deployed against an alternate file system, the NameNode, secondary NameNode and DataNode architecture of HDFS is replaced by the file-system-specific equivalent.

2.2 Hadoop distributed file system

HDFS stores large files (typically in the range of gigabytes to terabytes^[13]) across multiple machines. It achieves the reliability by replicating the data across multiple hosts, and hence does theoretically not require RAID storage on hosts (but to increase I/O performance some RAID configurations are still useful). With the default replication value, 3, data is stored on three nodes: two on the same rack, and one on a different rack.

The HDFS file system includes a so-called *secondary namenode*, which misleads some people into thinking ^[citation needed] that when the primary namenode goes offline, the secondary namenode takes over. In fact, the secondary namenode regularly connects with the primary namenode and builds snapshots of the primary namenode's directory information, which the system then saves to local or remote directories. These checkpoint images can be used to restart a failed primary namenode without having to replay the entire journal of file-system actions, then to edit the log to create an up-to-date directory structure. Because the namenode is the single point for storage and management of metadata, it can become a bottleneck for supporting a huge number of files, especially a large number of small files. HDFS Federation, a new addition, aims to tackle this problem to a certain extent by allowing multiple name-spaces served by separate namenodes.

An advantage of using HDFS is data awareness between the job tracker and task tracker. The job tracker schedules map or reduce jobs to task trackers with an awareness of the data location. For example: if node A contains data (x,y,z) and node B contains data (a,b,c), the job tracker schedules node B to perform map or reduce tasks on (a,b,c) and node A would be scheduled to perform map or reduce tasks on (x,y,z). This reduces the amount of traffic that goes over the network and prevents unnecessary data transfer.

2.3 Hadoop V/S Dbms

However, no compelling reason to choose MR over a database for traditional database workloads

MapReduce is designed for one-off processing tasks

- Where fast load times are important
- No repeated access. It decomposes queries into sub-jobs, schedules them with different policies.

DBMS	HADOOP
In a centralized database system, you've got one big disk connected to four or eight or 16 big processors. But that is as much horsepower as you can bring to bear	In a Hadoop cluster, every one of those servers has two or four or eight CPUs. You can run your indexing job by sending your code to each of the dozens of servers in your cluster, and each server operates on its own little piece of the data. Results are then delivered back to you in a unified whole.
To DBMS researchers, programming model doesn't feel new	Hadoop MapReduce is a new way of thinking about programming large distributed systems
Schemas: DBMS require them	Schemas: - MapReduce doesn't require them - Easy to write simple MR problems - No logical data independence

Table1. Comparison of Approaches

III. DISTRIBUTED DENIAL OF SERVICE ATTACK

A denial-of-service attack (DoS attack) is an attempt to make a computer resource unavailable to its intended users[1]. Distributed denial-of-service attack (DDoS attack) is a kind of DoS attack where attackers are distributed and targeting a victim. It generally consists of the concerted efforts of a person, or multiple people to prevent an Internet site or service from functioning efficiently or at all, temporarily or indefinitely.

3.1 How DDoS Attack works

DDoS attack is performed by infected machines called bots and a group of bots is called a botnet. This bots (Zombie) are controlled by an attacker by installing malicious code or software which acts as per command passed by an attacker. Bots are ready to attack any time upon receiving commands from the attacker. Many types of agents have scanning capability that permit to identify open port of a range of machines. When the scanning is finished, the agent takes the list of machines with open port and launches vulnerability-specific scanning to detect machines with un-patched vulnerability. If the agent found a machine with vulnerability, it could launch an attack to install another agent on the machine.

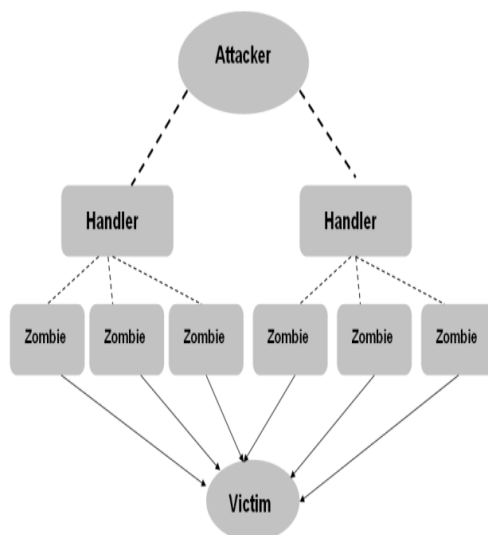


Fig 2. shows DDoS attack architecture, explaining its working mechanism.

IV. FAIR SCHEDULER

The core idea behind the fair scheduler is to assign resources to jobs such that on average over time, each job gets an equal share of the available resources. This behavior allows for some interactivity among Hadoop jobs and permits greater responsiveness of the Hadoop cluster to the variety of job types submitted.

The number of jobs active at one time can also be constrained, if desired, to minimize congestion and allow work to finish in a timely manner. To ensure fairness, each user is assigned to a pool. In this way, if one user submits many jobs, he or she can receive the same share of cluster resources as all other users (independent of the work they have submitted).

V. 5. 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 minimize their runtime. In their model, the scheduler is made mindful 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 picks 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.

VI. CONCLUSION AND FUTURE WORK

Traditional scheduling methods perform very poorly in mapreduce due to two aspects

- running computation where the data is
- Dependence between map and reduce task .

The Hadoop accomplishment creates a set of pools into which jobs are placed for selection by the scheduler. Each pool can be assigned shares to balance the resources across jobs in pools. By default, all pools have same shares, but we can configure accordingly to provide more or fewer shares depending upon the job type.

REFERENCES

- [1] Prashant Chauhan, Abdul Jhummarwala, Manoj Pandya, "Detection of DDoS Attack in Semantic Web" International Journal of Applied Information Systems (IJ AIS) – 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 Elmelegy, 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: IIT/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] Mirkovic and P. Reiher, A Taxonomy of DDoS Attack and DDoS Defense Mechanisms, ACM SIGCOMM CCR, 2004
- [9] H. Sun, Y. Zhaung, and H. Chao, A Principal Components Analysis-based Robust DDoS Defense System, IEEE ICC, 2008
- [10] Y. Lee, W. Kang, and Y. Lee, A Hadoop-based Packet Trace Processing Tool, TMA, April 2011
- [11] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters Commun. ACM, 51(1):107– 113, 2008.
- [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, pages 53:1–53:12, Piscataway, NJ, USA, 2008. IEEE Press.