International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 12, December - 2013

A Survey on Load Rebalancing in Clouds

Ms. Sushma G. Fulewar
M-Tech Scholar
Dept. Of CSE
GHRIETW,NAGPUR

Mr. Anil N. Jaiswal
Associate Professor
Dept. Of CSE
GHRIETW, NAGPUR

Mr. S. D. Kamble
Associate Professor
Dept. Of CSE
YCCE ,NAGPUR

ABSTRACT

In distributed file systems, nodes simultaneously serve computing and storage functions, a file is partitioned into a number of chunks allocated in distinct nodes so that application data processing tasks can be performed in parallel over the multiple nodes[1]. Here every node plays the same role and perform same computation equally distributed by the master computer. Distributed file system will give access to common data storage to every node [2]. Every node has a responsibility to perform given task and give acknowledgement to master computer where master computer having responsibility to provide appropriate output to the user. [3] Here it is considered that every client will work properly but there is no fix assurance for it. If any node fails to perform his task and goes down then its master's responsibility to re-distribute the task to nodes and get it done. Here we are proposing the re balancing and redistribution of data to be processed to available node using a DFS over cloud computer. Designed redistribution scheme will be implemented on multiple network machines and data storage server will be accessed through network file system.

Index Term – load balance, clouds.

I.INTRODUCTION

Distributed file systems are key building blocks for cloud computing applications based on the Map Reduce programming Paradigm. In such file systems, nodes simultaneously serve computing and storage functions; a file is partitioned into a number of chunks allocated in distinct nodes so that Map Reduce tasks can be performed in parallel over the nodes. However, in a cloud computing environment, failure is the norm, and nodes may be upgraded, replaced, and added in the system. Files can also be dynamically created, deleted, and appended. This results in load imbalance in a distributed file system; that is, the file chunks are not distributed as uniformly as possible among the nodes. Emerging distributed file systems in production systems strongly depend on a central node for chunk reallocation. This dependence is clearly inadequate in a large-scale, failure-prone environment because the central load balancer is put under considerable workload that is linearly scaled with the system size, and may thus become the performance bottleneck and the single point of failure. In this paper, a fully distributed load rebalancing algorithm is presented to cope with the load imbalance problem. Our algorithm is compared against a centralized approach in a production system and a competing distributed solution presented in the literature. The simulation results indicate that our proposal is comparable with the existing centralized approach and considerably outperforms the prior distributed algorithm in terms of load imbalance factor, movement cost, and algorithmic overhead. The performance of our proposal implemented in distributed file system.. Load algorithm is presented to cope with the load imbalance problem. Our algorithm is compared against a centralized approach in a production system and a competing distributed

II. RELATED WORK

A novel load balancing algorithm to deal with the load rebalancing problem in large-scale, dynamic, and distributed file systems have been presented in this paper. This compare with the centralized algorithm in the Hadoop HDFS production system and dramatically outperforms the competing distributed algorithm in terms of load imbalance factor, movement cost, and algorithmic overhead.[1] The efficiency and effectiveness of the design are further validated by analytical models and a real implementation with a small-scale cluster environment.[3] The evaluation of the proposed approach will be done in terms of the response time and also by considering the hop time and wait time during the migration process of the load balancing approach to avoid deadlocks[4] This paper presents a concept of Cloud Computing along with research challenges in load balancing. It also focus on merits and demerits of the cloud computing. Major thrust is given on the study of load balancing algorithm, followed by a comparative survey of these abovementioned algorithms in cloud computing with respect to stability, resource utilization, static or dynamicity, cooperative or non-cooperativeness and process migration.

III. LOAD REBALANCING PROBELM

Whenever we consider distributed file system consisting of a group of server V in a cloud, where the relation of V is |V | = n. Typically, n can be one thousand, ten thousand, or more. In the system, a number of files are stored in the n group of servers. First, denote the set of files as F.Any file f ∈ F is partitioned into a number of disjointed, fixed-size groups denoted by C. For example, each group has the same Size, 64 Mbytes, in Hadoop HDFS [2]. Second, assume that the load of a server is proportional to the number of sets or groups hosted by the server. Third, we consider failure to be the norm in such a distributed system, and the groups of servers may be upgraded, replaced and added in the system. Moreover, the files in F may be arbitrarily created, deleted, and appended. The net effect results in file chunks not being uniformly distributed to the groups of servers.

LOAD REBALANCING

Load balancing is the process of distributing the load among various resources in any system. Thus load need to be distributed over the resources in cloud-based architecture, so that each resources does approximately the equal amount of task at any point of time. Basic need is to provide some techniques to balance requests to provide the solution of the application faster. To deal with the load imbalance problem, in this study we advocate off loading the load rebalancing task to storage nodes by having the storage nodes balance their loads spontaneously.

IV. CENTRAL QUEUE ALGORITHM

Central Queue Algorithm works on the principle of dynamic distribution. It stores new activities and unfulfilled requests as a cyclic FIFO queue on the main host. Each new activity arriving at the queue manager is inserted into the queue. Then, whenever a request for an activity is received by the queue manager, it removes the first activity from the queue and sends it to the requester. If there are no ready activities in the queue, the request is buffered, until a new activity is available. If a new activity arrives at the queue manager while there are unanswered requests in the queue, the first such request is removed from the queue and the new activity is assigned to it.

When a processor load falls under the threshold, the local load manager sends a request for a new activity to the central load manager. The central load manager answers the request immediately if a ready activity is found in the *process-request queue*, or queues the request until a new activity arrives.

Few factors for load balancing algorithms:

a) Cost effectiveness: Overall improvement in system performance at a reasonable cost.

b) Scalability and flexibility: Algorithm must be scalable and flexible enough to allow such changes to be handled easily.

c) Priority: Priority must be decided first, algorithm itself for better service. Service provision for all the jobs regardless of their origin.

V. LITRETURER REVIEW

We discuss about the load balancing is implemented in the cloud computing environment to on demand resources with high availability. But the existing load balancing approaches suffers from various overhead and also fails to avoid deadlocks when there more requests competing for the same resource at a time when there are resources available are insufficient to service the arrived requests Another approach was proposed by This describes the autonomous and distributed load-balancing policy that can dynamically reallocate incoming external loads at each node. This adaptive and dynamic load balancing policy is implemented and evaluated in a two-node distributed system **[2]** Latter on in This describes the autonomous and distributed load-balancing policy that can dynamically reallocate incoming external loads at each node. This adaptive and dynamic load balancing policy is implemented and evaluated in a two-node distributed system **[3]** nodes simultaneously serve computing and storage functions; a file is partitioned into a number of chunks allocated in distinct nodes so that Map Reduce tasks can be performed in parallel over the nodes. However, in a distributed computing environment, failure is the norm, and nodes may be upgraded, replaced, and added in the system. Distributed file systems (DFS) are key building blocks for cloud computing applications based on the MapReduce programming paradigm. In such file systems, nodes simultaneously serve computing and storage functions, a file is partitioned into a number of chunks allocated in distinct nodes so that Map Reduce tasks can be performed in parallel over the nodes. However, in a cloud computing environment, failure is the norm, and nodes may be upgraded, replaced, and added in the system. Files can also be dynamically created, deleted, and appended. This results in load imbalance, that is, the file chunks are not distributed as uniformly as possible in the nodes. Although distributed load balancing algorithms exist in the literature to deal with the load imbalance problem, emerging DFSs in production systems strongly depend on a central node for chunk reallocation. This dependence is clearly inadequate in a large-scale, failure-prone environment because the central load balancer is put under considerable workload that is linearly scaled with the system size, and may thus become the performance bottleneck and the single point of failure. In this paper, we illustrate and define the load rebalancing problem in cloud DFSs. We advocate file systems in clouds shall incorporate decentralized load rebalancing algorithms to eliminate the performance bottleneck and the single point of failure. Simulation results for a potential distributed load balancing algorithm are illustrated. The performance of our proposal implemented in the Hadoop distributed file system is also demonstrated. Distributed file systems are key building blocks for cloud computing applications based on the Map Reduce programming paradigm. In such file systems, nodes simultaneously serve computing and storage functions; a file is partitioned into a number of chunks allocated in distinct nodes so that MapReduce tasks can be performed in parallel over the nodes. However, in a cloud computing environment, failure is the norm, and nodes may be upgraded, replaced, and added in the system. Files can also be dynamically created, deleted, and appended. This results in load imbalance in a distributed file system; that is, the file chunks are not distributed as uniformly as possible among the nodes. Emerging distributed file systems in production systems strongly depend on a central node for chunk reallocation. This dependence is clearly inadequate in a large-scale, failure-prone environment because the central load balancer is put under considerable workload that is linearly scaled with the system size, and may thus become the performance bottleneck and the single point of failure. In this paper, a fully distributed load rebalancing algorithm is presented to cope with the load imbalance problem. Our algorithm is compared against a centralized approach in a production system and a competing distributed solution presented in the literature. The simulation results indicate that our proposal is comparable with the existing centralized approach and considerably outperforms the prior distributed algorithm in terms of load imbalance factor, movement cost, and algorithmic overhead. The performance of our proposal implemented in the Hadoop distributed file system is further investigated in a cluster environment.

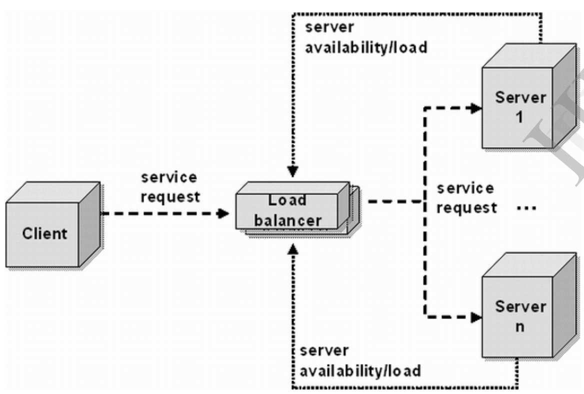


Fig: client/server/load

CLIENT

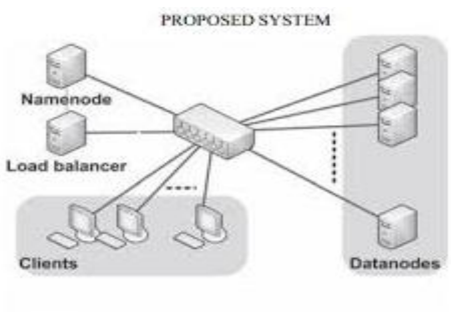
A **client** is a piece of computer hardware or software that accesses a service made available by a server. The server is often (but not always) on another computer system, in which case the client accesses the service by way of a network.

LOAD BALANCER

Load balancing is a computer networking method for distributing workloads across multiple computing resources, such as computers, a computer cluster, network links, central processing units or disk drives. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any one of the resources. Using multiple components with load balancing instead of a single component may increase reliability through redundancy. Load balancing is usually provided by dedicated software or hardware, such as a multilayer switch or a Domain Name System server process.

VI. PROPOSED WORK

Our proposal strive to balance the load and reduce the overhead While execution of provided task is expected that proposed system should allow the user to provide a task to master server and master server will distribute the task equally to available nodes and while execution if any node get fails then system should properly redistribute the task pending by failure node.



Implementation of data distributer sharing system

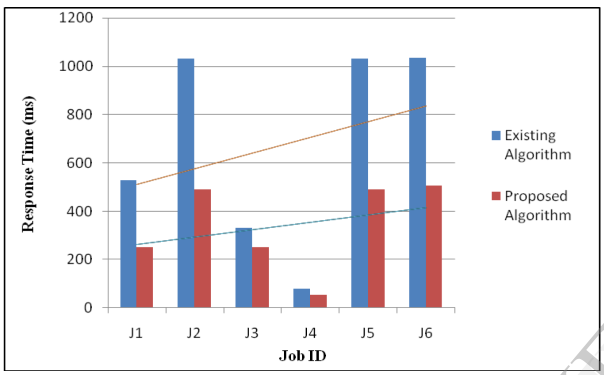


Fig. Comparison of Algorithm

VII. CONCLUSION

Implementing distributed processing can reduce overheads and it makes the proper utilization of multiple systems rather than implementing supercomputing processor proposed system can use normal lower configuration PC system to complete the task and even input task is not dependent on the single system so it reduces the risk of failure. As the system is based on master slave terminology we can extend to dynamic role to every system. By the time of failure any client system can become master system and fulfill the user requirement and handle rest of the process which can be called as backup server or backup maser system.

REFERENCE

1. "*The Load Rebalancing Problem in Distributed File Systems* "IEEE INTERNATIONAL CONFERENCE ON CLUSTER COMPUTING JUNE 2012.

2. *"Load Rebalancing for Distributed File Systems in Clouds"* Hung-Chang Hsiao, Member, IEEE Computer Society, Hsueh-Yi Chung, Haiying Shen, Member, IEEE, and Yu-Chang Chao IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 24, NO. 5, MAY 2013.

3. *"Implement A Reliable and Secure Cloud Distributed File System"* Fan-Hsun Tseng Chi-Yuan Chen Li-Der Chou and Han-Chieh Chao 12 IEEE INTERNATIONAL SYMPOSIUM ON INTELLIGENT SIGNAL PROCESSING AND COMMUNICATION SYSTEMS (ISPACS 2012) NOVEMBER 4-7, 2012

4. **"***Execution analysis of load balancing algorithms in cloud computing Environment*" Soumya Ray and Ajanta De Sarka rINTERNATIONAL JOURNAL ON CLOUD COMPUTING: SERVICES AND ARCHITECTURE (IJCCSA),VOL.2, NO.5, OCTOBER 2012

5. "*Performance evaluation of a cloud based load Balancer severing pareto traffic*" Ayman G. Fayoumi JOURNAL OF THEORETICAL AND APPLIED INFORMATION TECHNOLOGY OCTOBER 2011. VOL. 32 NO.© 2005 - 2011 JATIT & LLS. ALL RIGHTS RESERVED

6. Fan-hsun tseng , Chi-Yuan Chan, LI-Der Chou, Han Chieh Chau." *Implement A*

Reliable and Secure Cloud Distributed File System "2012 IEEE INTERNATIONAL SYMPOSIUM ON INTELLIGENT SIGNAL PROCESSING AND COMMUNICATION SYSTEMS (ISPACS 2012) NOVEMBER 4-7, 2012

7. "*Towards a Load Balancing in a Threelevel Cloud Computing Network",* Shu-Ching Wang, Kuo-Qin Yan, Wen-Pin Liao, ShunSheng Wang, 2010 IEEE,pp.108.

8. "*LBVS: A Load Balancing Strategy for Virtual Storage*", Hao Liu, Shijun Liu, Xiangxu Meng, Chengwei Yang, Yong Zhang, 2010 INTERNATIONAL CONFERENCE ON SERVICE SCIENCES, PP. 257-262.

9. "*Enhanced Load Balancing Approach to Avoid Deadlocks in Cloud* " Rashmi. K. Suma. V vaidehi MS *pecial Issue of* INTERNATIONAL JOURNAL OF COMPUTER APPLICATIONS (0975 – 8887) ON ADVANCED COMPUTING AND COMMUNICATION TECHNOLOGIES FOR HPC APPLICATIONS - ACCTHPCA, JUNE 2012

10. Sagar Dhakal, Majeed M. Hayat, I Jorge E. Pezoa, Cundong Yang, and David A. Bader*" Dynamic Load Balancing in Distributed Systems in the Presence of Delays: A Regeneration-Theory Approach"* IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 18, NO. 4, APRIL 2007"

11 Bhasker Prasad Rimal, Eummi Choi, Lan Lump (2009) "*A Taxonomy and Survey of Cloud Computing System*", 5TH INTERNATIONAL JOINT CONFERENCE ON INC, IMS AND IDC, IEEE EXPLORE 2527AUG 2009.

12 Bhathiya, Wickremasinghe.(2010)"*Cloud Analyst: A Cloud Sim-based Visual Modeller for Analysing Cloud Computing Environments and Applications"*

13 C.H.Hsu and J.W.Liu(2010) "*Dynamic Load Balancing Algorithms in Homogeneous Distributed System,"* PROCEEDINGS OF THE 6THINTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS.

14 Rajkumar Buyya and Karthik Sukumar "*Platforms for Building and Deploying Applications for Cloud Computing",* CSI Communication, pp. 6-11, 2011.

15 .R.Buyya, R. Ranjan AND R. N. Calheiros, "*Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities*", PROCEEDINGS OF THE 7TH HIGH PERFORMANCE COMPUTING AND SIMULATION CONFERENCE (HPCS 2009, ISBN: 978-1-4244-4907-1, IEEE PRESS, NEW YORK,USA), JUNE 21 - 24, 2009.