International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
International Conference on Industrial Automation and Computing (ICIAC- 12-13th April 2014)

RESEARCH ARTICLE OPEN ACCESS

LOAD REBALANCING IN CLOUDS

Sushma G. Fulewar[+] , Anil N. Jaiswal[#] , Shailesh D. Kamble[*]

[+]M.tech. 2ND yeat , Department of Computer Science and Engineering, G.H.Raisoni Institute of Engineering and Technology for Women, Nagpur, India Sushmafulewar2011@gmail.com
[#] Prof. Department of Computer Science and Engineering, G.H. Raisoni Institute of Engineering and Technology for Women, Nagpur, India jaiswal.anil@gmail.com
[*]Prof. Department of Computer Technology, Yashwanatrao chawhan college of Engineering , Nagpur, India shailesh_2kin@rediffmail.com

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.

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. 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 $f \in F$ 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.

III. 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.

VI. Litreturer Review

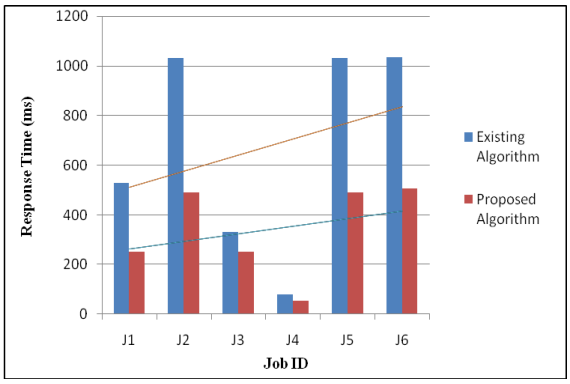
We discuss about the load balancing is implemented in the cloud computing environment to on demand resources with high availability. But the existing load balancing approaches suffers from various overhead and also fails to avoid deadlocks when there more requests competing for the same resource at a 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

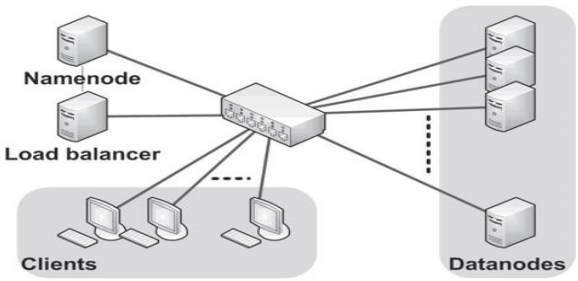
that Map Reduce tasks can be performed in parallel over the nodes. However, in a cloud computing environment, failure is the norm, and nodes may be upgraded, replaced, and added in the system. Files can also be dynamically created, deleted, and appended.

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.

PARAMETRIC COMPARISON OF LOAD BALANCING ALGORITHMS

Parameters	Round Robin	Random	Local Queue	Central Queue	Central Manager	Threshold
Overload Rejection	No	No	Yes	Yes	No	No
Fault Tolerant	No	No	Yes	Yes	Yes	No
Forecasting Accuracy	More	More	Less	Less	More	More
Stability	Large	Large	Small	Small	Large	Large
Centralized/Decentralized	D	D	D	C	C	D
Dynamic/Static	S	S	Dy	Dy	S	S
Cooperative	No	No	Yes	Yes	Yes	Yes
Process Migration	No	No	Yes	No	No	No
Resource Utilization	Less	Less	More	Less	Less	Less





VI. Existing System

However, recent experience concludes that when the number of Storage nodes, the number of files and the number of accesses to files increase linearly, the central nodes become a performance bottleneck, as they are unable to accommodate a large number of file accesses due to clients and Map Reduce applications. Thus, depending on the central nodes to tackle

the load imbalance problem exacerbate their heavy loads. Even with the latest development in distributed file systems, the central nodes may still be overloaded.

VII. PROPOSED SYSTEM:

In this paper, we are interested in studying the load rebalancing problem in distributed file systems specialized for large-scale, dynamic and data-intensive clouds. (The terms "rebalance" and "balance" are interchangeable in this paper.) Such a large-scale cloud has hundreds or thousands of nodes (and may reach tens of thousands in the future). Our objective is to allocate the chunks of files as uniformly as possible among the nodes such that no node manages an excessive number of chunks.

VII. RCHITECTURE

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. For example, consider a wordcount application that counts the number of distinct words and the frequency of each unique word in a large file. In such an application, a cloud partitions the file into a large number of disjointed and fixed-size pieces (or file chunks) and assigns them to different cloud storage nodes (i.e., chunkservers).Also we have developed word count application and word search application. 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.

DATA OWNER REGISTRATION: In this module if a owner of data(File) have to store data on a cloud server,he/she should register their details first.These details are maintained in a Database.Then he has to upload the file in a file database. The file which are stored in a database are in an encrypted form. Authorized users can only decode it.

DATA USER REGISTRATION: In this module if a user wants to access the data which is stored in a cloud server,he/she should register their details first.These details are maintained in a Database.

TTP (TRUSTED THIRD PARTY) L**:** In this module TTP has monitors the data owners file by verifying the data owner's file and stored the file in a database .Also ttp checks the CSP(CLOUD SERVICE PROVIDER),and find out whether the csp is authorized one or not.

CSP(CLOUD SERVICE PROVIDER) LOGIN**:** In this module CSP has to login first.Then only he can store the file in his cloud server.Ttp can only check the csp whether the csp is authorized csp or not.If its fake, ttp won't allow the file to store in cloud server.

VIII . CONCLUSION

This deal with the load rebalancing problem in large-scale, dynamic, and distributed file systems in clouds has been presented in this paper. Our proposal strives to balance the loads of nodes and reduce the demanded movement cost as much as possible, while taking advantage of physical network locality and node heterogeneity. Particularly, our load-balancing algorithm exhibits a fast convergence rate. The efficiency and effectiveness of our design are further validated by analytical models and a real implementation with a small-scale cluster environment.

REFERENES

[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 an-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

[6] "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.

[7] "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.

[8] "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

[9] Sagar Dhakal, Majeed M. Hayat, I Jorge E. Pezoa, Cundong Yang, and David A. Bader"

[10] 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] G. R. Andrews, D. P. Dobkin, and P. J. Downey, "Distributed allocation with pools of servers," in ACM SIGACT-SIGOPS Symp. Principles of Distributed Computing, Aug. 1982, pp. 73-83.

[12] S. Malik, "Dynamic Load Balancing in a Network of Workstation", 95.515 Research Report, 19 November, 2000.

[13] Derek L. Eager, Edward D. Lazowska , John Zahorjan, "Adaptive load sharing in homogeneous distributed systems", IEEE Transactions on Software Engineering, v.12 n.5, p.662-675, May 1986.