IJARES/July 2015/Volume-3/Issue-7/1976-1980 **ISSN 2347- 9337**

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN ENGINEERING AND SCIENCE

IMPACT OF DISTRIBUTED SYSTEMS IN MANAGING CLOUD APPLICATION

N.Vijaya Sunder Sagar[1], M.Dileep Kumar[2], M.Nagesh[3], Lunavath Gandhi[4]

[1]Associate Professor & HOD, Dept of CSE, Ashoka Institute of Engineering and Technology, Hyderabad, T.S, India

[2,3]Assistant Professor, Dept of CSE, Ashoka Institute of Engineering and Technology, Hyderabad, T.S, India

[4]M.Tech, Dept of CSE, Ashoka Institute of Engineering and Technology, Hyderabad, T.S, India

ABSTRACT:

Modern distributed file systems within clouds depend on central nodes to handle the metadata information concerning file systems in addition to balancing of loads concerning storage nodes on basis of metadata. Even with most recent development within distributed file systems, central nodes might still be overloaded. In our work we study load rebalancing difficulty within distributed file systems specialized in support of extensive, vibrant as well as data-intensive clouds and such an extensive cloud has several huge number of nodes. The most important aim of our work is to design an algorithm of load rebalancing to reallocate chunks of files consistently between the nodes such that no node controls an extreme number of chunks and chunks are distributed as equivalently as possible and minimizing movement cost to the extent that feasible. For the most part of existing solutions of Load-balancing algorithms are considered lacking in view of movement cost as well as node heterogeneity and can set up important continuation network traffic to DHTs. A novel algorithm of load-balancing was put forward to deal with the difficulty of load rebalancing in important, dynamic, as well as distributed file systems in clouds. The projected system not only takes benefit of physical network vicinity within the reallocation of file chunks to decrease the movement cost however also exploits

competent nodes to get better the general system performance.

Keywords: Load rebalancing, File system, Distributed file systems, Load-balancing, File chunks, Heterogeneity.

1. INTRODUCTION:

Strategies of key enabling for clouds consist of Map Reduce programming, virtualization and so on which give emphasis to scalability [1]. Hence clouds can be outsized in scale, and comprise entities can at random fail and join while sustaining system consistency. Load balance between storage nodes is considered as an important function in cloud system. In a load-balanced cloud, the resources are be well provisioned, by maximizing the performance of applications of MapReduce. The most important aim of our work is to distribute chunks of files consistently between the nodes such that no node controls an extreme number of chunks. When number of storage nodes, number of files as well as the number of accesses to files increases linearly, the central nodes turn out to be a performance blockage, since they are incapable to hold a large number of file accesses due to clients as well as MapReduce applications. Consequently depending on central nodes to undertake the load imbalance difficulty worsen their heavy loads. Even with most recent development within distributed file systems, central nodes might still be overloaded. In our work we study load rebalancing difficulty within distributed file systems specialized in support of extensive, vibrant as well as data-intensive clouds and such an extensive cloud has several huge number of nodes [2][3].

2. METHODOLOGY:

Distributed file systems are important building blocks in support of applications of cloud computing on basis of MapReduce programming concept. In such file systems, nodes simultaneously make available storage functions; a file is partitioned into several chunks which are allocated in distinctive nodes so that tasks of MapReduce are carried out in parallel over the nodes. Modern distributed file systems within clouds depend on central nodes to handle the metadata information concerning file systems in addition to balancing of loads concerning storage nodes on basis of metadata. Our intention is to decrease

network traffic which is caused by rebalancing loads of nodes as much as likely to exploit the network bandwidth obtainable to normal applications. A novel algorithm of load-balancing was put forward to deal with the difficulty of load rebalancing in important, dynamic, as well as distributed file systems in clouds. Offloading the task of load rebalancing towards storage nodes was recommended by having the storage nodes to stabilize their loads instinctively and this get rid of confidence on central nodes. The storage nodes are prearranged as network basis distributed hash tables (DHT); to determine a file chunk can merely refer to speedy key lookup in distributed hash table. DHT facilitate nodes to self-organize and fix while continuously recommending lookup functionality in dynamism of node, simplify managing of system [4]. For the most part of existing solutions of Load-balancing algorithms are considered lacking in view of movement cost as well as node heterogeneity and can set up important continuation network traffic to DHTs. The projected system not only takes benefit of physical network vicinity within the reallocation of file chunks to decrease the movement cost however also exploits competent nodes to get better the general system performance. By leveraging distributed hash tables, we put forward an algorithm of load rebalancing as shown in fig1 in support of distributing file chunks as equivalently as possible and minimizing movement cost to the extent that feasible. The algorithm which was put forward operates within a distributed method in which nodes carry out their load-balancing tasks separately devoid of synchronization and moreover strives to stabilize loads of nodes and decrease demanded movement cost to the extent that possible, while taking benefit of node heterogeneity and physical network locality.

3. AN OVERWIEW OF PROJECTED SYSTEM:

The most important aim of our work is to design an algorithm of load rebalancing to reallocate chunks of files consistently between the nodes such that no node controls an extreme number of chunks and chunks are distributed as equivalently as possible and minimizing movement cost to the extent that feasible. The chunk servers in the projected system are prearranged as a DHT network; specifically each chunk server executes a DHT procedure [5]. A file within the system is partitioned into several

fixed-size chunks in which each chunk contain an exceptional chunk handle which is named with a globally recognized hash. Every chunk server also contains an exceptional ID.function. The hash function returns a distinctive identifier in support of a given file's pathname string as well as a chunk index. In the proposed system chunk servers self-configure as well as self-heal for the reason of their arrivals, departures, as well as failures, shortening the system management. To decrease the lookup latency, we can implement modern DHTs. The projected system not only takes benefit of physical network vicinity within the reallocation of file chunks to decrease the movement cost however also exploits competent nodes to get better the general system performance. Typical DHTs assurance that if a node departs, then its locally hosted chunks are dependably moved to its successor; if a node joins, subsequently it distributes the chunks whose IDs instantaneously lead the joining node from its successor to supervise. The projected system heavily depends on arrival as well as departure operations of node to move about file chunks between nodes. The DHT system is apparent to the metadata managing in the projected system. Load-balancing algorithm displays a speedy convergence rate. The efficiency as well as effectiveness of projected system is additionally validated by analytical representations as well as an actual implementation with a small-scale cluster setting. The algorithm moreover decreases algorithmic overhead which is introduced to DHTs to the extent that possible and is assessed all the way through computer simulations and moreover reveals a fast convergence rate.

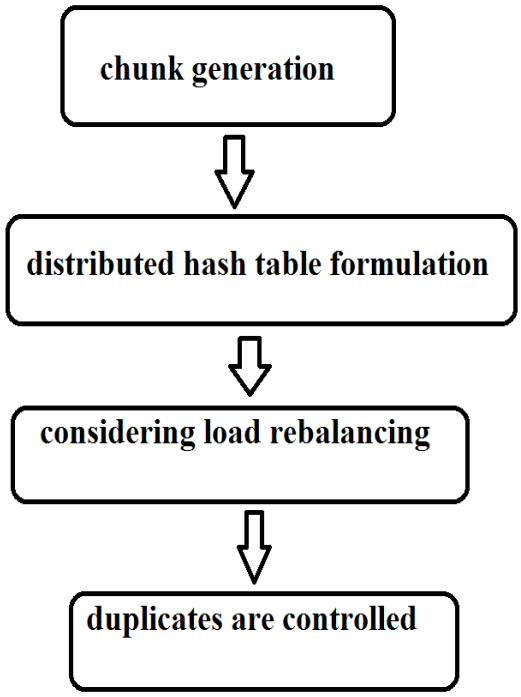

Fig1: An overview of Load Rebalancing Process

4. CONCLUSION:

Load balance between storage nodes is considered as an important function in cloud system. In a load-balanced cloud, the

resources are be well provisioned, by maximizing the performance of applications of MapReduce. Even with most recent development within distributed file systems, central nodes might still be overloaded. For the most part of existing solutions of Load-balancing algorithms are considered lacking in view of movement cost as well as node heterogeneity and can set up important continuation network traffic to DHTs. In our work we study load rebalancing difficulty within distributed file systems specialized in support of extensive, vibrant as well as data-intensive clouds and such an extensive cloud has several huge number of nodes [6]. Our intention is to decrease network traffic which is caused by rebalancing loads of nodes as much as likely to exploit the network bandwidth obtainable to normal applications. A novel algorithm of load-balancing was put forward to deal with the difficulty of load rebalancing in important, dynamic, as well as distributed file systems in clouds. By leveraging distributed hash tables, we put forward an algorithm of load rebalancing in support of distributing file chunks as equivalently as possible and minimizing movement cost to the extent that feasible. The algorithm which was put forward operates within a distributed method in which nodes carry out their load-balancing tasks separately devoid of synchronization and moreover strives to stabilize loads of nodes and decrease demanded movement cost to the extent that possible, while taking benefit of node heterogeneity and physical network locality. The projected system heavily depends on arrival as well as departure operations of node to move about file chunks between nodes.

REFERENCES

[1] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "BCube: A High Performance, Server-Centric Network Architecture for Modular Data Centers," Proc. ACM SIGCOMM '09, pp. 63-74, Aug. 2009.

[2] H. Abu-Libdeh, P. Costa, A. Rowstron, G. O'Shea, and A. Donnelly, "Symbiotic Routing in Future Data Centers," Proc. ACM SIGCOMM '10, pp. 51-62, Aug. 2010.

[3] S. Surana, B. Godfrey, K. Lakshminarayanan, R. Karp, and I. Stoica, "Load Balancing in Dynamic Structured P2P Systems," Performance Evaluation, vol. 63, no. 6, pp. 217-240, Mar. 2006.

[4] S. Iyer, A. Rowstron, and P. Druschel, "Squirrel: A Decentralized Peer-to-Peer Web Cache," Proc. 21st Ann. Symp. Principles of Distributed Computing (PODC '02), pp. 213-222, July 2002.

[5] I. Raicu, I.T. Foster, and P. Beckman, "Making a Case for Distributed File Systems at Exascale," Proc. Third Int'l Workshop Large-Scale System and Application Performance (LSAP '11), pp. 11- 18, June 2011.

[6] M. Jelasity, S. Voulgaris, R. Guerraoui, A.-M. Kermarrec, and M.V. Steen, "Gossip-Based Peer Sampling," ACM Trans. Computer Systems, vol. 25, no. 3, Aug. 2007.