INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

LARGE DATA MANAGEMENT USING BITDEW TECHNOLOGY

RADHIKA BANG[1], PROF. N. M. KANDOI[2]

1. Department of Computer Science Engineering, SSGMCE, Amravati University, India.
2. Faculty of Computer Science Engineering Department, SSGMCE, Amravati University, India.

Accepted Date: 05/03/2015; Published Date: 01/05/2015

Abstract: In recent years social media gaining lot of popularity and almost every user is sharing different contents in the form of images, video, text, etc. every day. This is creating a lot media repository and in near future it may require to search content from this repository and not from the typical string based data search. Major issue with media repository is that one can't search the exact content based on string query search in real time. It requires a lot of pre-processing and indexing on this media content repository. Proposed system is considering this problem and trying to provide a media search on the basis of media internal properties like color. Main component involved in research is parallel processing of huge resources using load balancer, indexing of media repository and search engine for searching content from indexed table and the media repository.

Keywords: Load Balancing, Repository, Files / Image Repository, Indexing/Querying: Service Level Agreement

Corresponding Author: MS. RADHIKA BANG

Access Online On:

www.ijpret.com

How to Cite This Article:

Radhika Bang, IJPRET, 2015; Volume 3 (9): 694-699

PAPER-QR CODE

INTRODUCTION

1.1 BACKGROUND:

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. For Internet services, the load balancer is usually a software program that is listening on the port where external clients connect to access services. The load balancer forwards requests to one of the "backend" servers, which usually replies to the load balancer. This allows the load balancer to reply to the client without the client ever knowing about the internal separation of functions. It also prevents clients from contacting back-end servers directly, which may have security benefits by hiding the structure of the internal network and preventing attacks on the kernel's network stack or unrelated services running on other ports.

Some load balancers provide a mechanism for doing something special in the event that all backend servers are unavailable. This might include forwarding to a backup load balancer, or displaying a message regarding the outage. Load balancing gives the IT team a chance to achieve a significantly higher fault tolerance. It can automatically provide the amount of capacity needed to respond to any increase or decrease of application traffic.

It is also important that the load balancer itself does not become a single point of failure. Usually load balancers are implemented in high-availability pairs which may also replicate session persistence data if required by the specific application.

1.2 AIM AND OBJECTIVE:

Our objective is to distribute the task among the nodes on the basis of its resource utilization such that no node computes any excessive task and user can get task’s output effectively as fast as possible. So our aim is to reduce the overhead of master node by rebalancing the load in the case failure detection. Another major aim is to develop the indexing technique for media content searching.

2. PROPOSED WORK:

The key issue in searching large media repository increase the system’s overhead. To resolve this problem load balancing technique is used where the proper utilization of computing resources get done. It can be done by partitioning a task according to load distribution strategies. Here we propose the effective load balancing technique for media repository where user can search for image from large File/Image Repository. This task get done in Processing module where master distributes, assigns the task to the slaves and task's result is stored in indexed table in Indexing module. According to indexing, fetch image set or URL of image files as final result from File/Image Repository.

2. LITERATURE SURVEY:

In paper [1] In this paper, an efficient scheme to ensure fair load distribution in such P2P systems by utilizing proximity information is explained. Nodes having higher capacity carry more loads. Proximity information is utilized to guide load balancing such that loads are assigned and transferred between physically close heavy nodes and light nodes, thereby minimizing the load transferring overhead and allowing load balancing to perform more efficiently.

MapReduce plays a key role in Cloud Computing. In this paper[2], they develop a web-based graphic user interface for ordinary users to utilize MapReduce without the real programming. MapReduce decreases the complexity of the distributed programming and is easy to be developed on large clusters of common machines. So, users do not need to install Hadoop in their computers.

This paper[3] presents *Chord*, a distributed lookup protocol that addresses location of the node that stores a desired data item. Chord provides support for a operation: given a key, it maps the key onto a node. Data location can be easily implemented on top of Chord by associating a key with each data item, and storing the key/data pair to which the key maps. Chord adapts efficiently as nodes join and leave the system, and can answer queries even if the system is continuously changing. Resulting *Chord* is scalable: Communication cost and the state maintained by each node scale logarithmically with the number of Chord nodes.

This paper[4], they presents various load balancing schemes in different cloud environment based on requirements specified in Service Level Agreement (SLA). On one hand static load balancing scheme provide easiest simulation and monitoring of environment but fail to model heterogeneous nature of cloud. On the other hand, dynamic load balancing algorithm are difficult to simulate but are best suited in heterogeneous environment of cloud computing. Distributed nature of algorithm provides better fault tolerance. Dynamic load balancing techniques in distributed environment provide better performance.

In the paper [5], a novel load balancing protocol, Earliest Completion Load Balancing (ECLB) that monitors the distribution of system load and network latency in an attempt to dynamically create and maintain an evenly loaded system, even in the case of heterogeneous nodes or heavy load conditions. In heterogeneous environments, the completion of transactions before their deadlines depends upon both the network latency and load on each node. In general, it outperforms other load balance strategies regardless of the environment. ECLB has shown to match the performance of basic, greedy load based protocols in situations where load is the most important factor in decision making.

In this paper[6], The Master Ant Colony algorithm has proposed. The Master ACO achieve better optimal solution for load balancing with vanishing node concept. The load balancing is solved by removing unoptimized path. The execution time reduced when all nodes in average state.

In this paper[7], they address the mis-sequencing problem by introducing a three-stage load-balancing switch architecture enhanced with an output load-balancing mechanism. It achieves high forwarding capacity and provides transmission delay. For this nested load-balancing scheme is used.

Advantages: Scalability, Bounded Delay, 100% Throughput, Low Average Delay. The output load-balancing mechanism is critical to the performance of the 3SLB switch, it produces finite queue lengths.

3. PROPOSED SYSTEM:

3.1 PROPOSED APPROACH:

I. We proposing Load balancer for media repository where user ask query for images through search engine. This search carried on multi client-server application where master get query as input hence increase in load. For resolving this problem, we use effective load balancing algorithm, accordingly master distributes and assigns task to the slaves and get result. Using output from processing module, we create and update indexed table. According to indexing, image set or URL will fetch from file/image repository as final result for user.

3.2 PROPOSED ARCHITECTURE:

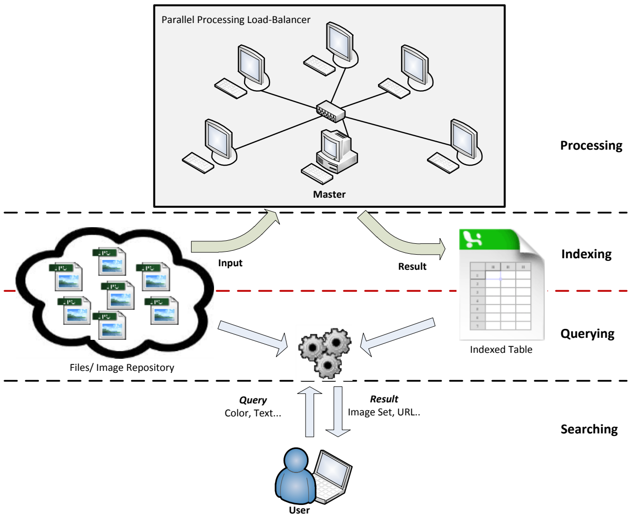

Fig. 1: System Architecture

Fig.1 shows the basic block diagram of the project. Project contains following modules.

Files / Image Repository: Image repository is a large set of irrelevant images or files where we can search for required images.

Processing: Processing is based on multi-client server application building with GUI using socket programming. When task of searching is issued, File/Image Repository is required as input to the processing module and is assigned to a node, called as Master and Master distributes the task on the basis of its status (i.e. resource availability or PC hardware statistics) to other nodes, called as clients to reduce the load on the system.

Task distribution consist of:

1. File size calculation
2. File list distribution
3. Complete task progress monitoring
4. Failure detection
5. Task reassigning

Indexing/Querying: Processing provides its result to Indexing to get sequencing of images and it increases image counter so that user can directly get index of image and accordingly fetch image or URL of images from File/Image Repository.

Search Engine: User gives input as query about colour or text through system to search engine. Search of files or image set on the basis of colour level identification and text. User get output as image set or URL of image files.

5. REFERENCES:

1. Yingwu Zhu and Yiming Hu,"Towards Efficient Load Balancing in Structured P2P Systems", In Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS'04),IEEE 2004

2. Jin-Ming Shih, Chih-Shan Liao and Ruay-Shiung Chang, "Simplifying MapReduce Data Processing" 2011 Fourth IEEE International Conference on Utility and Cloud Computing.

3. Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek And Hari Balakrishnan,"Chord: A Scalable Peer-To-Peer Lookup Protocol For Internet Applications", In IEEE/ACM transactions on networking, vol. 11, no. 1, February 2003

4. Mayanka Katyal , Atul Mishra , " A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment", International Journal of Distributed and Cloud Computing Volume 1 Issue 2 December 2013

5. Waqar Haque, Andrew Toms and Aaron Germuth, "Dynamic Load Balancing in Real-Time Distributed Transaction Processing", In IEEE 2013 16th International Conference on Computational Science and Engineering

6. Anamika Jain and Ravinder Singh, "An Innovative Approach of Ant Colony Optimization for Load Balancing in P2P Grid Environment ", In IEEE 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)

7. Yan Cai, Xiaolin Wang, Weibo Gong And Don Towsley, "A Study On The Performance Of A Three-Stage Load-Balancing Switch" IEEE/ACM Transactions On Networking, Vol. 22, No. 1, February 2014.