ISSN:0975-9646

Optimized Load Balancing using Linear Programming

Prof.B.C. Julme
Department of Computer Engineering
Pune Vidhyarthi Griha's ,COET
Pune, India

Spruha Shevade, Pranita Shinde, Shweta Lokhande, Sneha Suryawanshi
Department of Computer Engineering
Pune Vidhyarthi Griha's ,COET
Pune, India

Abstract: Cloud computing technology provides shared computer resources to other client computers and devices on demand. Various remote devices can access and share resources when required using cloud computing. This technology is growing rapidly because of its numerous advantages. It decreases storage requirements, avoid expenditure on hardware devices, provides security, allows access from any location and so on.
The resources are allocated to consumers by cloud service provider (CSP). CSP needs to schedule the resources efficiently such that the tasks are processed in minimum cost and time. Due to the ever increasing amount of load on the cloud server, it is essential to balance the load by offloading it on various task processors. For this purpose, we are using two algorithms: K-Means clustering and Linear Programming. The task is divided into clusters using K-Means algorithm and clusters are optimally distributed to volunteers using Linear Programming.
The processing time required in the proposed scheme is compared with processing time required in random scheduling.

Keywords: Cloud computing, load balancing, K-Means clustering, Linear Programming

I. INTRODUCTION

Today, there is a great need of smart systems in human life. These smart systems are available to humans in wide range making human life comfortable. Some of these smart systems lack processing power to run large applications. Thus, there is a need to jump to cloud computing which is responsible to service many web based technologies. Cloud computing provides shared computer processing resources and data to computer and other devices on demand. Most cloud computing services fall into three broad categories. They are SaaS (Software As A Service), PaaS (Platform As A Service) and IaaS (Infrastructure As A Service). SaaS provides readymade applications to users. In PaaS, applications are developed by cloud users and other platform supports are provided by CSP. But in IaaS, CSP only provides hardware components and all other layers of cloud infrastructure are managed by cloud users. There are three different ways to deploy cloud computing resources: Private cloud (resources are used by a single organization), Public cloud (operated by a third party-cloud service provider) and Hybrid cloud (combine public and private). In this paper, we are using private cloud. Since, cloud computing provides tremendous benefits to users, number of users are increasing day by day. So to serve their requests, cloud server performs an important task, called scheduling.

In this paper, we propose a system in which users submit tasks to cloud. Cloud will schedule those tasks to available volunteers depending upon their capabilities. For scheduling, cloud server uses K-Means as clustering algorithm and Linear programming as optimization algorithm. Depending upon the amount of data processed, cloud will pay the volunteers. The processing which is performed by volunteers is grey scaling of images.

II. RELATED WORK

Sokol Kosta et al. [1] have proposed a system which can transfer mobile phone's tasks and applications on cloud. This reduces problem of insufficient processing power of mobile phones. But this system does not provide task scheduling on cloud.

Zixue Cheng et al. [2] have introduced 3-layered architecture: wearable devices, mobile devices and cloud. To solve problem of insufficient power of wearable devices, the application codes available at wearable devices are transferred to smart phone. If smart phone are unable to process, they are transferred to cloud. But this system has no task scheduling implementation.

Ning Liu et al. [3] have provided architecture for energy conservation by using proper task scheduling. Tasks are allocated to cloud resources in the decreasing order of their processing capacity so that resource having high processing capacity gets highest number of tasks. In this manner, user provided set of tasks can be executed with minimum number of cloud resources, which in turns minimizes energy consumption.

Sivadon Chaisiri et al. [4] have provided a scheduling algorithm for minimization of total resource provisioning cost. Resource reservation strategy is less costlier than on-demand resource provision mechanism but needs complex prediction of future resource requirement. This scheduling algorithm properly resolves the tradeoffs between reservation and on-demand resource provision strategies.

Jignesh Lakhani et al. [5] introduced framework to minimize the communication overhead. This scheduling algorithm sends all the inter-dependent tasks in a batch to the corresponding cloud resource rather than sending each task separately.

R. Vijayalakshmi et al. [6] provided task scheduling framework that provides minimization of execution time required for user submitted tasks. To achieve this objective of time minimization, priorities of tasks are considered. Task that having highest priority is scheduled to processor with high processing power. But in this system, cost minimization problem is not considered.

AV. Karthick et al. [7] have provided task scheduling based on three queues using Tri Queue Scheduling (TQS). Main goal of this scheduling was to maximize the utilization of available resources and improve the performance of system. Dynamic time slice is used to execute the tasks in small, medium and large sized queues.

III. SYSTEM ARCHITECTURE

The system flow consists of following three components:

1. Submitters
2. Cloud server
3. Volunteers (Laptop and mobile)

The task submitter (PC client) will submit a set of images to the server. The images should be of variable sizes. The cloud server will get the images and it will apply two algorithms. Firstly, K-Means clustering which will form clusters of images depending on their size. i.e. we will have three clusters namely, small, medium and large. Now, the server will apply linear programming algorithm which will allocate the clusters to appropriate volunteers. The processing power of laptop and mobile is different. The algorithm will allocate the clusters to the task processors in such a way that the task will be processed in minimum time and cost. The volunteers will receive the task from cloud server. They will process the images i.e., perform the gray scale operation and will send the processed images back to server. Finally, we will get the processed images on our client PC. The main objective of the system is that the tasks are assigned to processors according to users' requirements such as minimum cost and time. This is done by using linear programming which dynamically allocates jobs to the volunteers.

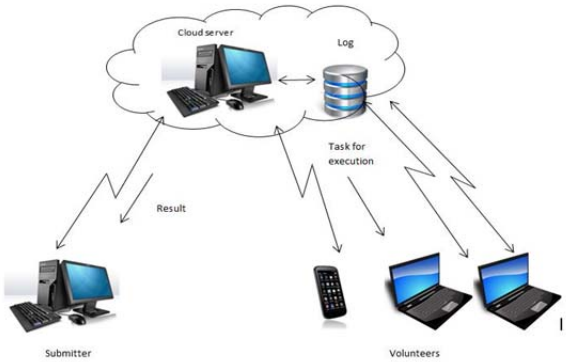


Fig.1 System Architecture

a) **Client (Submitter):** Client is a web based application. The users can register through client and login into the system. Client app allows user to select images for uploading onto the server for image processing. The final processed images can be viewed on the client.

b) **Server:** The server is Glassfish web server. The images uploaded by the client are sent to the server. Clusters of the images are formed according to their size using K-means algorithm. Linear programming is used to divide the clusters optimally among all volunteers.

c) **Volunteers:** The volunteers perform gray scale operation on the set of images. The processed images are sent back to the server.

A. K-MEANS CLUSTERING ALGORITHM

K-means clustering algorithm is an unsupervised learning algorithm. It is used to form groups of data which have some common characteristics. The groups are called as clusters. In K-means clustering, K denotes number of clusters.

We use K-means algorithm to divide set of images into three clusters based on their size. As a result we get 3 clusters: small, medium and large. The nearest cluster center is determined by Euclidean distance between observation and cluster centers.

1) *Algorithm Steps:*

Let $D = \{d_1, d_2, d_3, \ldots\ldots, d_n\}$ be the set of data points and $C = \{c_1, c_2, \ldots\ldots, c_m\}$ be the set of centers.

1) Select '*m*' cluster centers randomly from set of data points.

2) Calculate the distance between each data point and cluster centers.

3) Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers.

4) Recalculate the new cluster center using:

$$C_i = (1/m_i) \sum_{j=1}^{m} d_i$$

where, 'm_i' represents the number of data points in i^{th} cluster.

5) Recalculate the distance between each data point and new obtained cluster centers.

6) If no data point was reassigned then stop, otherwise repeat from step 3).

B. LINEAR PROGRAMING ALGORITHM:

Linear programming algorithm is applied on cluster of medium sized images. The task scheduling system is

represented in the form of linear equations required for linear programming. Proposed system is represented in the form of linear equations as follows:
Objective function:
Minimize,

$$z = \sum_{i=1}^{n} c_{1i}x_{1i} + \sum_{j=1}^{m} c_{2j}x_{2j}$$

Subject to:

$$\sum_{i=1}^{n} t_{1i}x_{1i} + \sum_{j=1}^{m} t_{2j}x_{2j} \leq t_{max}$$

$$\sum_{i=1}^{n} x_{1i} + \sum_{j=1}^{m} x_{2j} = x_{total}$$

Non-negativity constraints:

$x_{1i}, x_{2j} \geq 0$

Where,
c_{1i} = cost per pixel for i^{th} smart phone
c_{2j} = cost per pixel for j^{th} computer
x_{1i} = total number of pixels processed by i^{th} smart phone
x_{2j} = total number of pixels processed by j^{th} computer
t_{1i} = processing time per pixel at i^{th} smart phone
t_{2j} = processing time per pixel at j^{th} computer
t_{max} = maximum time allowed to process the entire batch of images
x_{total} = total number of pixels in the medium sized images
Thus, goal is to minimize the total cost as represented in objective function within the minimum or specified time. Now, we use Simplex algorithm to solve this linear problem to find values of x_{1i} and x_{2j}. Putting these values in the objective function gives the total minimum cost required to process the batch images within user specified time t_{max}.

C. GRAY SCALE ALGORITHM:

Steps:
1. Obtain the R,G,B values of a pixel of image.
2. Calculate average of all RGBs to get a single gray value
   GS= (R+G+B)/3
3. Replace original RGB value with new gray value.

D. *MATHEMATICAL MODEL*

System S is defined as:
S = { I ,O , Fn, S, F, Serv, Tasksub, Vol}
Where,
I = Set of Inputs
O = Set of Outputs
Fn = Set of Functions
S = Success

F = Failure

I = Input images = $\{I_1, I_2, I_3, \ldots\ldots I_n\}$
where I is infinite set of images.

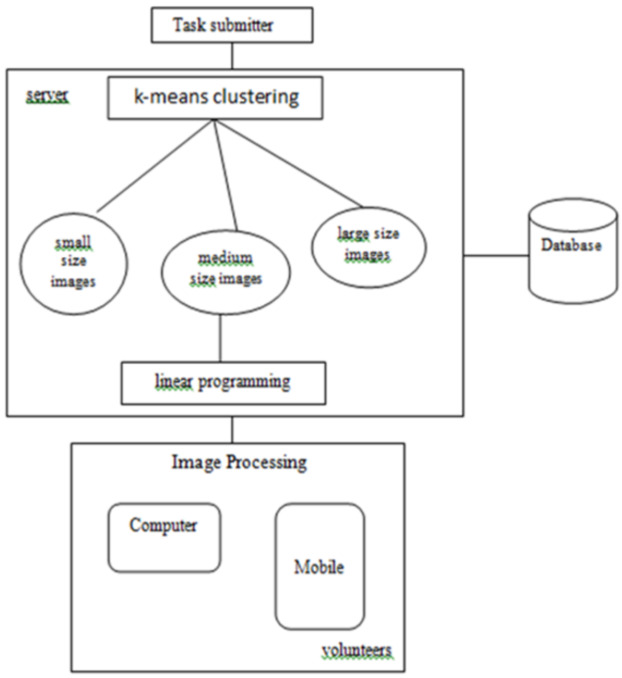
Serv= {serv}
where, Serv is set of finite servers.

Tasksub={Tasksub}
where, Tasksub is finite set of task submitters.

Vol=$\{vol_1, vol_2, vol_3, \ldots\ldots vol_n\}$
where , Vol is infinite set of volunteers.

Fn={register(), login(), submitTask(), cluster(), distribute(), process(), submitRes(), reply()}

register(): Users can register into system.
login(): Registered users can log in the system
submitTask(): Selected images are sent to the server.
cluster(): Clusters of images are formed according to the sizes.



IV. CONCLUSION

In cloud computing load balancing is hot area for research. In this paper we have proposed task scheduling using K-Means and Linear Programming approach. Main objective of the system is to schedule task provided by submitter to volunteers in such a way that it minimises cost and time required for processing. This is achieved by cluster oriented and optimised distribution of task among processors. Comparative study of results obtained by the proposed system and the results of random scheduling shows that this system is more efficient.

REFERENCES

[1] Sokol Kosta, Andrius Aucinas, Pan Hui, Richard Mortier and Xinwen Zhang, "ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading", Proceedings IEEE INFOCOM, IEEE 978-1-4673-0775-8/12, PP: 945953, 2012.

[2] Zixue Cheng, Peng Li, Junbo Wang and Song Guo, "Just-in-Time Code Offloading for Wearable Computing", IEEE Transactions on Emerging Topics in Computing, 2015.

[3] Ning Liu, Ziqian Dong and Roberto Rojas-Cessa, "Task Scheduling and Server Provisioning for EnergyEffcient Cloud-Computing Data Centers", IEEE 33rd International Conference on Distributed Computing Systems Workshops, IEEE 978-0-7695-5023-7/13, PP: 226-231, 2013.

[4] Sivadon Chaisiri, Bu-Sung Lee and Dusit Niyato, "Optimization of Resource Provisioning Cost in Cloud Computing", IEEE TRANSACTIONS ON SERVICES COMPUTING, IEEE 1939-1374/12, PP: 164-177, 2012.

[5] Jignesh Lakhani and Hitesh A. Bheda, "An Approach to Optimized Resource Scheduling using Task Grouping in Cloud", International Journal of Advanced Research in Computer Science and Software Engineering, ISSN:2277 128X, Volume 3, Issue 9, September 2013, PP: 594-599.

[6] R. Vijayalakshmi and Mrs. Soma Prathibha, "A novel approach for task scheduling in cloud", 4th ICCCNT, IEEE-31661, July 4-6, 2013.

[7] AV. Karthick, Dr. E. Ramaraj and R. Kannan, "An Efficient Tri Queue Job Scheduling using Dynamic Quantum Time for Cloud Environment", International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), IEEE 978-1-46736126-2/13, 2013.