Cluster Oriented Optimized Cloud Task Scheduling Strategy using Linear Programming

Mahesh S. Shinde
AISSMS COE
Pune University,
Maharashtra, India

Anilkumar Kadam
AISSMS COE
Pune University,
Maharashtra, India

ABSTRACT
Today, cloud computing has emerged as revolutionary technology in the IT industry. It has provided boost to the area of parallel and distributed computing. Cloud computing deals with on demand allocation of remotely placed computing and storage devices to the cloud users on charge per utilization basis. Cloud service provider (CSP) is responsible for the allocation of cloud resources to consumers of cloud services so that cloud consumers should get satisfied. Thus to gratify cloud users, CSP must has to schedule cloud resources so that it charges minimum amount to users in return for execution of their tasks on cloud resources. In the proposed task scheduling strategy, Fuzzy C-Means (FCM) is used as a clustering technique which is applied complementary to well known optimization technique - Linear Programming. The proposed task scheduling strategy charges minimum amount to user for execution of his tasks within the time specified by him. The results obtained by proposed task scheduling are compared with the results of existing random task scheduling method.

Keywords
Cloud computing; Fuzzy C-Means; Linear programming; task scheduling; cloud resources.

1. INTRODUCTION
Cloud computing has brought revolution in today’s internet. It has made a huge impact on the internet based services such as e-commerce, banking, social networking etc. Depending of an ownership and accessibility, cloud can be categorized into four types as public, private, community and hybrid cloud. Again cloud services used by cloud users are SaaS (Software as a Service), IaaS (Infrastructure as a service) and PaaS (Platform as a Service). SaaS provides readymade applications and tools to user, while PaaS provides platforms required for development of applications (or softwares) and IaaS includes hardware components such as processors, storage devices etc [1]. Figure 1 shows an abstract view of cloud computing environment.

Nowadays, smart phones have become inalienable part of people in the modern world. However, processing power of smart phone has limitations since it is insufficient to process large complex applications (tasks) within smart phone. Therefore for processing these complex tasks, there is need to transfer these tasks to cloud which has adequate resources that are having ability to process them [2]. Cloud task scheduler plays crucial role so that cloud resources are allocated efficiently among the cloud users according to their demands. There are common scheduling algorithms such as Random scheduling, First Come First Served scheduling, Priority based scheduling, Shortest Task First scheduling, Round Robin scheduling etc. which can be used by cloud scheduler for scheduling user submitted tasks to cloud resources [3].

2. RELATED WORK
Sokol Kosta et al. [2] have proposed a framework which has provision of transferring the smart phone’s tasks and applications over the cloud for processing so that the problem of insufficient processing power of smart phones has been resolved. But this framework has not provided any type of task scheduling on cloud.

Zixue Cheng et al. [4] have introduced an architecture having three layers as wearable devices, smart phones and cloud respectively. To mitigate the problem of insufficient processing power of wearable devices, applications or code available at the wearable devices are transferred to smart phone for execution. Again if that smart phone is unable to process, further these tasks are migrated to cloud for
3.1 Fuzzy C-Means (FCM) Algorithm

Fuzzy C-Means (FCM) Algorithm is clustering technique in which some data points from the set of data points given to the clustering may belong to two or more clusters simultaneously. In FCM clustering technique, each data point is associated with a set of values called as membership values. Each membership value represents the belongingness of that point with respect to the particular cluster. Membership value of a point with respect to particular cluster is between 0 and 1. Thus the FCM is used to determine these membership values and depending upon these values, each data point is assigned to the particular cluster [10]. This Fuzzy C-Means (FCM) algorithm can be applied using below mentioned steps:

Consider there are n elements, S = {s₁, s₂, .... sₙ} to be grouped using FCM in t number of clusters, C = {C₁, C₂, ...., Cₜ} then

1. Decide number of final clusters we want from clustering.
2. Initially define random membership value wᵢⱼ (belongingness of point sᵢ to cluster Cⱼ)
   Where, i = {1, 2, ...., n} and j = {1, 2, ...., t}
3. Repeat until the difference between membership values in successive iterations is not more than threshold, ϵ.
   a) Calculate centroid for each cluster using the formula as follows,
   \[ cⱼ = \frac{\sumₙₖ₌₁ Wᵢⱼ Sᵢ}{\sumₘₖ₌₁ Wᵢⱼ} \]
   Where, cⱼ = centroid of the jᵗʰ cluster and
   \( Wᵢⱼ \) = level of fuzziness (which decides the impact of membership value).
   b) Calculate membership value of each point with each cluster using following formula,
   \[ wᵢⱼ = \left( \frac{1}{\text{dist}(sᵢ,cⱼ)^2} \right)^\frac{1}{P - 1} \]
   \[ \sum_{q=1}^{t} \left( \frac{1}{\text{dist}(sᵢ,c_q)^2} \right)^\frac{1}{P - 1} \]
4. Thus finally we will get final centroids and corresponding clusters [10][11]

3.2 Linear Programming

Linear programming is famous optimization approach used in business. Linear programming deals with the linear equations that represent the system under consideration. For applying linear programming, the system is represented in three parts, namely, objective function (equation of particular parameter which is to be optimized), constraints (equations that represent some limits while optimizing the objective function) and non-negativity constraints (list of parameters which can’t be negative). A typical representation of linear programming equations is as follows [12]:
Maximize (or Minimize),

\[ z = \sum_{j=1}^{n} a_j x_j \]  \hfill (3)

Subject to:

\[ \sum_{j=1}^{n} b_{ij} x_j \]  \hfill (4)

for \( i = 1, 2, 3, \ldots, m \)

Non-negativity constraints:

\[ x_j \geq 0 \]  \hfill (5)

for \( j = 1, 2, 3, \ldots, n \)

where,

\( a, b \), and \( c \) are set of real numbers and \( x \) are set of variables.

In this way, the objective of linear programming is to find out the value of \( x \) in such a way that objective function will get optimized (minimized or maximized) without violating any constraint. For solving the linear equations, generally Simplex algorithm is used. A Simplex algorithm is as follows [13]:

Step 1: Construct linear equations in standard form for given problem.

Step 2: Transform linear equations into slack form in order to convert inequality constraints into equality constraints by adding slack variables.

Step 3: Calculate a basic solution by making all the non-basic variables (variables present on right hand side) to 0 and then finding the values of all basic variables (variables present on the left hand side).

Step 4: Rewrite the set of equations and reform linear equations until basic solution changes in each iteration and gives greater objective value as compared to previous iteration. Following steps are carried out for this goal:

1) Select a non-basic variable having positive coefficient in objective function.

2) Until no constraint is violated, expand the value of this chosen non-basic variable.

3) Then change this non-basic variable to basic variable and thereby existing basic variable to non-basic variable.

Step 5: Until all coefficients in the objective function become negative, step 3 and 4 are repeated.

Step 6: Now required objective value is obtained just by putting final basic solution in original linear equations.

### 3.3 System Architecture

Proposed system presents cluster oriented optimized cloud task scheduling strategy using linear programming algorithm. The proposed system has used computers and smart phones as cloud resources which are allocated to process submitter’s tasks by charging different costs to user according to their processing power. Generally, resources having high processing power (computers) charges more cost to user as compared to resources that having low processing power (smart phones). But as computers having high processing power, they execute tasks or applications in less time than smart phones. Therefore it is important to schedule submitter’s tasks in such a way that it optimizes cost charged to submitter (cloud user) and time required to execute these tasks. Architecture of this proposed task scheduling strategy is depicted in figure 2. As shown in architecture, cloud consumer (submitter) submits group of images to cloud for processing (grey scaling). Cloud scheduler receives this batch of images from submitter and applies Fuzzy C-Means (FCM) in order to bundle them together according to their sizes into three groups, namely, small, medium and large. Out of these, small and large sized images are as it is given to smart phone (having low processing power) and computer (having high processing power) respectively for processing. And on remaining medium images, linear programming is applied to determine whether a particular image should be given to smart phone or computer for processing as shown in figure 2.

**Figure 2: Proposed cluster oriented task scheduling using Linear Programming**

As linear programming is to be applied on group of medium sized images, firstly there is need to express this task scheduling system 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,

\[ z = \sum_{i=1}^{n} c_{1i} x_{1i} + \sum_{j=1}^{m} c_{2j} x_{2j} \]  \hfill (6)

Subject to:

\[ \sum_{i=1}^{n} t_{1i} x_{1i} + \sum_{j=1}^{m} t_{2j} x_{2j} \leq t_{\text{max}} \]  \hfill (7)
\[
\sum_{i=1}^{n} x_{1i} + \sum_{j=1}^{m} x_{2j} = x_{\text{total}} \quad (8)
\]

Non-negativity constraints:
\[x_{1i}, x_{2j} \geq 0 \quad (9)\]

Where,
\[c_{1i} = \text{cost per pixel for } i^{\text{th}} \text{ smart phone}\]
\[c_{2j} = \text{cost per pixel for } j^{\text{th}} \text{ computer}\]
\[x_{1i} = \text{total number of pixels processed by } i^{\text{th}} \text{ smart phone}\]
\[x_{2j} = \text{total number of pixels processed by } j^{\text{th}} \text{ computer}\]
\[t_{1i} = \text{processing time per pixel at } i^{\text{th}} \text{ smart phone}\]
\[t_{2j} = \text{processing time per pixel at } j^{\text{th}} \text{ computer}\]
\[t_{\text{max}} = \text{maximum time allowed to process the entire batch of images}\]
\[x_{\text{total}} = \text{total number of pixels in the medium sized images}\]

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 \(t_{\text{max}}\).

4. EXPERIMENTAL SETUP

Private cloud is used for a demonstration of the proposed system. One smart phone is used as a cloud user (task submitter) which submits the bunch of images for processing to the task scheduler of this cloud. As shown in figure 3, number of smart phones and computers are used as cloud resources for executing the user submitted tasks (batch of images). Smart phone with Android 4.3 operating system and 1 GHz dual core processor has been used as task submitter. Task submitter module in smart phone of cloud user is developed using eclipse (ADT bundle). The computer with 4 GB RAM, 1 TB hard disk and 2.6 GHz processor is used as a task scheduler which use GlassFish server. Scheduler module has executed on this computer so that it acts as scheduler (CSP) of system. Image processing (gray scaling of images) logic is implemented on number of computers and smart phones which were used as cloud resources. For the sake of evaluation of the proposed task scheduling, the charge of processing for smart phone and computer resources are fixed to five and ten units per thousand pixels respectively.

5. EXPERIMENTAL RESULTS

The evaluation of proposed task scheduling has been done by comparing the results (in terms of charges user required to pay to CSP) obtained by submitting same set of images to random task scheduling and proposed task scheduling. As linear programming has been used in the proposed system for task scheduling, cost required for the processing of input group of images depends on the user submitted time (time constraint). As computer resource has high processing capability than that of smart phone resource, time required for computer to process submitted tasks is shorter than smart phone while cost required is high as compared to smart phone. Therefore the proposed system tries to optimize time and cost depending on the user permitted time bound. Results of proposed scheduling strategy have been obtained under tight time bound (giving less time for execution) and relaxed time bound (allowing more time for execution) provided by user. Following table 1 shows the comparison of results obtained by random task scheduling and proposed task scheduling strategy (in tight and relaxed time bound).

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Number of images</th>
<th>Charges in random task scheduling</th>
<th>Charge in proposed task scheduling strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Tight time bound</td>
<td>Relaxed time bound</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3290</td>
<td>3150</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>6720</td>
<td>5810</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>9130</td>
<td>9060</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>10940</td>
<td>10780</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>15280</td>
<td>13560</td>
</tr>
</tbody>
</table>

Charge (cost) graph of random task scheduling and proposed task scheduling system under tight and relaxed time bound is shown in Figure 4.

![Figure 3: Experimental setup for proposed system](image-url)

![Figure 4: Cost graph of Random and proposed task scheduling (under tight and relaxed time bound)](image-url)
Graph in the Figure 4 shows that the cost required to execute the user submitted tasks is optimized on the basis of user specified time constraint for task execution. Execution time comparison in case of tight and relaxed time bound of the proposed system is given in Table 2. These values of processing time (in seconds) of number of images are obtained when user specifies tight and relaxed time bound for processing of these images.

### Table 2: Execution time comparison proposed system under tight and relaxed time bound

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Number of images</th>
<th>Time required to process images in the proposed task scheduling strategy (in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Tight time bound</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>45</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>67</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>85</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>110</td>
</tr>
</tbody>
</table>

Comparison of execution (processing) times of number of images under the user submitted tight and relaxed time bounds is shown by graph as shown in Figure 5. Now by comparing Figure 4 and Figure 5, it is observed that if user gives tight time bound for execution of images (when execution time decreases) results in increase in cost (charge) of processing. And if user allowed time bound is larger (under relaxed time bound) then charges required to pay to the cloud service provider are low. Therefore proposed task scheduling strategy allows user to control the execution of his tasks on cloud resources by giving time bounds based on his requirement that whether he want to process his tasks in minimum time or minimum cost.

**Figure 5: Execution time graph under tight and relaxed time bound of proposed system**

### 6. Conclusion

Today, cloud computing is becoming more popular which has resulted in increasing number of cloud users day by day. Therefore the role of task scheduler is very important to manage shared resources among cloud consumers in order to attain proper resource utilization, load balancing, user satisfaction by cost and time optimization, energy conservation etc. This paper has introduced cluster oriented optimized cloud task scheduling strategy using linear programming approach with an objective of cost and time optimization. As shown in results, using the proposed task scheduling strategy, one can minimize charges that user required to pay in return for processing his task on cloud resources within the time limit supplied by him.

### 7. Future Scope

The proposed system has used the linear programming approach for the task scheduling to achieve the objective of cost (charges) minimization under time user specified time bound. Depending upon user’s requirement, it is also possible to design linear programming by reversing the parameters for objective function and constraints as time and cost respectively. Also some more number of parameters like resource utilization, communication cost, energy consumption etc. can be used while constructing linear programming equations to make scheduling strategy more extensive.

### 8. References


