Study on the Resource Allocation Optimization in Cloud Computing
Based on the Hybrid Optimization Algorithm

Yue-jin ZHOU*
School of Management and Engineering, Nanjing University, Nanjing, China
*Corresponding author

Keywords: Cloud computing, Resource allocation, Hybrid optimization algorithm, Allocation model.

Abstract. The methods of solving resource allocation are mainly heuristic algorithms which could not solve resource allocation problems in cloud computing. The hybrid optimization algorithm is studied to solve the problem. There are many different hybrid optimization algorithms. Our research hopes to find a simple and effective method. We select a combination optimization algorithm of the genetic and ant colony algorithms. In the early phase of this algorithm, with the help of the wide range search capabilities of the genetic algorithm, it finds a better solution; in the later stage of this algorithm, with the help of positive feedback and efficiency of the ant colony algorithm, it finds the optimal solution. In addition, the two algorithms convergence conditions and the way of how to make the better solution of the genetic algorithm translate into the initial pheromone distribution provisions of the ant colony algorithm are set up. At last, the algorithm was realized with a simulation environment, and a specific example was made by comparative analysis to verify the correctness and effectiveness of the algorithm.

Introduction

With the development and innovation of computer technology, computer mode has entered the period of cloud computing from PC, client/server, and grid computing [1, 2]. Cloud computing is an integrator of above three modes [3, 4, 5, 6]. With the development of cloud computing technology, the problem of the resources allocation optimization must be solved. This paper provides a hybrid optimization algorithm to solve this problem. The reasonable resource allocation can reduce operational cost and increase marketing competition for the cloud computing operators. On the other hand, the reasonable resource allocation can meet customer dynamic demands and promote service quality for customers. In general, the reasonable resource allocation can improve operational efficiency and ensure fairness and effectiveness.

Literature Review

The resource allocation of cloud computing belongs to large scale and multi-tasks scheduling issue that is NP-hand problem. The resource allocation of grid computing has been researched widely for the past years. It set up the base for the resource allocation of cloud computing.

Liu et al. [7] provided a genetic algorithm to solve the issue of the resource allocation optimization. This algorithm is called the improved genetic algorithm based on chromosome encoded mode and fitness function since characteristics of dynamic, hetero-geneous and large-scale tasks need to be processed in cloud computing environment. The simulation results proved that the improved algorithm performed better than GA and Sufferage method in regard to performance and quality of service.

Hua et al. [8] provided an updated ant colony algorithm in term of the framework of Map/Reduce. The evaluation standard of computing resource was defined firstly. These resources include projected executive time, network bandwidth, network delay, and so on. The disposal time that the computing resource completed the task was forecasted. Finally the optimized computing resource was found by this algorithm.
Liu et al. [9] focused on solving the optimization problem of the cloud computing’s service cluster resource schedule and loading balance. They presented a cloud computing resource schedule strategy in term of Modified Particle Swarm Optimization (MPSO) algorithm. In order to increase the search efficiency and effectiveness in global and local search, and to prevent falling into local optimal, the introduced dynamic multi-group collaboration mechanism and the reverse of the flight of mutation particles to the Particle Swarm Optimization (PSO) algorithm were introduced. With extending the cloud computing simulation platform CloudSim to test the algorithm, the results proved that this algorithm is effective, and the operation efficiency is very high.

Wei et al. [10] provided a game-theoretic method of fair resource allocation. With cloud computing services becoming dynamic and more numerous, resource supply turns to be more and more challenging. A QoS constrained resource allocation problem was solved in their research, A Binary Integer Programming method was used to solve the independent optimization. And an evolutionary mechanism was provided, which changes multiplexed strategies of the initial optimal solutions for different participants with minimizing their efficiency losses. The algorithms within the evolutionary mechanism considered both optimization and fairness. It was proved that Nash equilibrium always exists if the problem of the resource allocation game has feasible solutions.

Chanhan et al. [11] provided a QoS Guided Weighted Mean Time Min-Min Max-Min Selective heuristic for QoS based task scheduling. The algorithm takes single QoS parameter as requirements of tasks for deciding the match level between resources and tasks. The method also takes into account the performance of resources scheduling. Authors have appraised the algorithm in GridSim for various task scenarios. The simulation results were taken and compared with other heuristics like Min-Min, Max-Min, QoS Guided Min-Min, Weighted Mean Time Min-Min Max-Min Selective. The obtained results of the proposed method were better for makespan, resource utilization and resource load balancing than other mentioned methods above.

Abrishami et al. [12] provided a new QoS-based workflow scheduling algorithm in term of a novel concept called Partial Critical Paths (PCP), which tries to minimize the cost of workflow execution while meeting a user-defined deadline. This algorithm was designed to recursively schedule the partial critical paths ending at previously scheduled tasks. The simulation results of the algorithm showed that its performance was very hopeful.

Xu et al. [13] firstly provided an algorithm of job scheduling based on Berger model. In this process, the algorithm sets up dual fairness constraint. The constraint 1 was to classify user tasks by QoS preferences, and to launch the general expectation function under the classification of tasks to restrain the fairness of the resources in selection process. The constraint 2 was to define resource fairness justice function to calculate the fairness of the resources allocation. Authors have updated and expanded simulation platform CloudSim, and have carried out the job scheduling algorithm. The simulation results showed that their designed algorithm can effectively complete the user tasks and presents better fairness.

Since characteristics of dynamic, heterogeneous and large-scale tasks need to be processed as well as considering QoS in cloud computing environment it is required to develop newer and more effective algorithms to solve resource allocation problem.

**Modelling**

**Structure Modelling of Resource Allocation**

The resource allocation process of cloud computing is to distribute cloud resource to the required customers by cloud service providers through Internet. Therefore, we suppose that there are n independent tasks which should be distributed to m resources (m<n). The ultimate objective is to minimize the task disposal cost.
The resource allocation process of cloud computing involves resource, service, and task. They cooperate and complete resource allocation and management in accordance with the certain reciprocal rules.

Mathematical Modeling of Resource Allocation

We suppose the tasks have been dealt with the rule of “FIFS”. The issue of the task graph G resource allocation is defined $S(G)$, the goal is to find the min time that all sub-tasks are carried out. The time includes ProcessTime and CommuTime. The task graph G takes the task map to each disposal unit and assigns the start time of each task. The task $x_1$ is assigned the disposal unit $y_1$, the task start time is defined as $StartT(x_1,y_1)$, the closed time as

$$FinishT(x_1,y_1) = StartT(x_1,y_1) + ProcessTime(x_1,y_1)$$

(1)

$$totaltime(S) = \max\{FinishT(x,y), x \in X, y \in Y\}$$

(2)

The goal is to find $\min\{totaltime(S)\}$

In order to compute $\min\{totaltime(S)\}$, we make the following definitions:

**Definition 1**: When the task node $x_j$ has no the parent-node, i.e., $\text{prec}(j) = \emptyset$ and $x_j$ is assigned to the disposal unit $y_i$, it has

$$FinishT(x_j,y_i) = ProcessTime(x_j,y_i)$$

(3)

**Definition 2**: When the task nodes $x_i, x_j$ locate in the disposal units $y_2,y_1$, $x_j$ is the direct pioneer $x_j$ and $x_j$ has lower priority than $x_k$, it has:

$$FinishT(x_j,y_i) = \max\{\text{Finish}(x_j,y_i) + ProcessTime(x_j,y_i) +\text{CommuTime}(x_i,x_j),\text{Finish}(x_i,y_i) + ProcessTime(x_j,y_i)\}$$

(4)

**Definition 3**: When the task nodes $x_i, x_j$ locate in the same disposal units $y_i$, and $x_i$ is the direct pioneer of $x_j$, it has:

$$FinishT(x_j,y_i) = \text{Finish}(x_i,y_i) + ProcessTime(x_j,y_i)$$

(5)

Design and Implement of the Hybrid Optimization Algorithm

**Design Idea**

Two kinds of algorithms, the genetic algorithm and the ant colony algorithm are combined and the algorithms are updated in order to form a new algorithm for the resource allocation of cloud computing.

**Design of Critical Point**

The key of the hybrid algorithm is to design the algorithm change point. In our algorithm design, the number of the min/max iteration of the genetic algorithm is set in advance. The minimum evolution rate of the progeny of the genetic algorithm is set based on the statistic evolution rate of the progeny. Within the iteration interval if the evolution rate of the progeny of continuous $\text{Gene}_{\text{die}}$ is less than the minimum evolution rate, the genetic algorithm could be ended and go to the ant colony algorithm. The other steps of the algorithm are similar with the genetic and ant colony algorithms.
Definition of the Objective Function

The objective function of the genetic algorithm is defined as follows:

\[
\min \text{Resource} (e) = \frac{A \text{processsteme}(e) + B \text{netdelay}(e)}{C \text{bandwidth}(e)}
\]  

(6)

\[
\text{s.t } \begin{cases} 
\text{processsteme}(e) < \text{maxT} \\
\text{bandwidth}(e) > \text{minB} \\
\text{netdelay}(e) < \text{maxN}
\end{cases}
\]

(7)

Among them, maxT, minB, and maxN present max disposal time, min bandwidth, and max network delay respectively. A, B, and C mean the weights of the three constraint conditions respectively.

The fitness function must compute the max value. While our objective function is to solve the min value. It is necessary to change the fitness function. We define the fitness function as follows:

\[
f(i) = \sum_{k=1}^{M} \left( \frac{\text{totaltime}(k) - \text{Min} + 1}{M \times [\text{totaltime}(i) - \text{Min} + 1]} \right)^2 
\]

\[i = 1, 2, ..., M\]

(8)

Among them, M presents the number of population, totaltime(i) means the max completion time of the task assignment in ith chromosome of the present population, Min indicates min value of the totaltime in all the chromosomes of the present one.

Algorithm Implementation and Experiment Analysis

Visual Basic language was used to develop a simulation environment and the hybrid algorithm was implemented under the WIN7 environment. A specific example was made by the comparative analysis to verify the effectiveness of the algorithm. The simulation parameters come from reference [14]. The three kinds of algorithms comparison was made in terms of the scheduling time, the number of iterations, and loading balance. We suppose 20-100 tasks and 8 resources.

Figure 1 indicates the task curve of the three algorithms within 20-100 task nodes. We find the genetic algorithm has rapider convergence rate in the task interval [20, 40] from the Table 1. But with the number of iterations increase its convergence rate becomes more and more slow. The performance of the ant colony algorithm is opposite to the genetic algorithm. At the start the convergence rate is slow because of the less initial pheromone. With the pheromone increase the convergence rate is faster and faster. The hybrid algorithm is the best than the other two algorithms.

![Figure 1. Time-task curve of the three algorithms.](image)
Table 1. Comparison of the task allocation time for the three algorithms.

<table>
<thead>
<tr>
<th>Task nodes</th>
<th>Ant</th>
<th>Genetic</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>123</td>
<td>68</td>
<td>12</td>
</tr>
<tr>
<td>30</td>
<td>368</td>
<td>290</td>
<td>210</td>
</tr>
<tr>
<td>40</td>
<td>998</td>
<td>867</td>
<td>305</td>
</tr>
<tr>
<td>50</td>
<td>1869</td>
<td>2014</td>
<td>1120</td>
</tr>
<tr>
<td>60</td>
<td>2569</td>
<td>4001</td>
<td>1508</td>
</tr>
<tr>
<td>70</td>
<td>3678</td>
<td>6224</td>
<td>2007</td>
</tr>
<tr>
<td>80</td>
<td>6450</td>
<td>11238</td>
<td>3048</td>
</tr>
<tr>
<td>90</td>
<td>9980</td>
<td>19568</td>
<td>4239</td>
</tr>
<tr>
<td>100</td>
<td>15458</td>
<td>32395</td>
<td>6012</td>
</tr>
</tbody>
</table>

Figure 2 indicates the number of iterations to find the optimum solution for the three algorithms. With the task-nodes increase the number of iterations is on the rise for the three algorithms but the hybrid algorithm has the less number of iterations.

Figure 2. The number of iterations to find the optimum solution.

Figure 3 indicates resource load balance for the three algorithms. The load is in wave state. Comparatively the load of the hybrid algorithm is relatively balanced.

Figure 3. Resource load balance.

Conclusion

With the number of tasks increase the advantages of the hybrid algorithm are more and more obvious. So it is specially used in the environment of cloud computing in which so many tasks need to be dealt with. On the other hand, its load is relatively balanced.

This paper provides a new, hybrid algorithm to solve the problem of the resource allocation in the environment of cloud computing. The algorithm is more effective than the genetic and ant colony algorithms. But the research may be improved in the following aspects. In the design of the algorithm
more solutions could be tried to get much better features such as the ending conditions of the genetic algorithm, the entering time of the ant colony algorithm, the solution of the genetic algorithm how to transfer the initial solution of the ant colony algorithm, and so on. On the other hand, the single objective to minimize the completion time (makespan) is not enough. Other objectives such as energy consumption [15], service of quality, et al also should be considered to form a problem of multi-objectives optimization. In the implementation of the algorithm, the program was developed to verify the effectiveness of the algorithm. The cloud simulation platform CloudSim could be used as the algorithm test to get much better results.

Acknowledgement

This research was supported by National Natural Science Foundation of China (Grant No. 71571095).

References
