OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

A compartive study between Hungarian and Antcolony optimization algrorithm for task scheduling in the Cloud

D.Prasanth Rao
BTech, Sreenidhi Institute Of Science And Technology,Hyderabad,Telanaga,India
Corresponding Author: Mr.E.Madhukar
Associate Professor, Sreenidhi Institute Of Science And Technology,Hyderabad,Telanaga,India

ABSTRACT: *cloud computing is upcoming and evolving technologies in this era. Using cloud the capital expenditure is reduced. Many companies move towards cloud to reduce capital expenditure and operational cost. Task scheduling is the process of allocating tasks to the resources available. There are different types of task scheduling algorithms. In this paper, we use Hungarian method to allocate tasks to the resources, and compare with ant colony optimization. We also check the time and space complexities.*
KEY WORDS: *cloud computing, task scheduling, Hungarian algorithm, ant colony optimization*

Date of Submission: xx-xx-xxxx Date of acceptance: xx-xx-xxxx

I. INTRODUCTION

According to NIST, "cloud computing is a model for enabling ubiquitous, convenient on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." The essential features of cloud computing are on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service. Virtualization is process used to create various resources in a single infrastructure. It allows physical servers, storage devices and networking services to partitioned on demand.

II. CLOUD COMPUTING CHARCTERISTICS[2,3]

2.1 on-demand-self-service

consumers can access required resources and software unilaterally. They do not require human interaction with each service provider.

2.2 Broad network access

Capabilities are available over the network and are supported to dissimilar client platforms (e.g., phones, tablets, workstations).

2.3Resource pooling

The provider's computing resources are used to serve multiple consumers using a multi-tenant model.

2.4 Rapid elsaticity

Cloud is flexible and scalable enough for outward and inward commensurate with demand. To the consumers, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

2.5 Measured service

Cloud computing systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of the service. Resource can controlled, monitored and reported for both the provider and consumer of the utilized service. Only pay for what you actually used.

III. SERVICE MODELS IN CLOUD [4,5]

cloud computing has 3 service models, they are Software as a service, Platform as a service, Infrastructure as a service.

3.1 Infrastructure as a service(IaaS)

Iaas refers to online services that provide high level API's used to dereference various low level details of underlying network infrastructure.

3.2 Platform as a service(PaaS)

Cloud providers deliver a computing platform, typically including operating system, programming language execution environment, database and webserver. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers.

3.3 Software as a service(SaaS)

Cloud providers manage the infrastructure and platforms that run the applications. This eliminates the need to install and run the application on the user's own computers, which simplifies maintenance and support.

IV. DEPLOYMENT MODELS IN CLOUD

There are four types of cloud. They are private, community, public and hybrid.[1].

4.1 Public cloud

The cloud infrastructure is operated for open use by the general public. It may be owned, managed and operated by business, academic, or government organization or some combination of them.

4.2 Private cloud

The cloud infrastructure is operated solely by a single organization comprising multiple consumers.

4.3 Community cloud

The cloud infrastructure is operated by specific community of consumers from organisations that have shred concerns. It may be owned, managed and operated by one or more of the organizations in the community, a third party or some combination of them.

4.4 Hybrid cloud

The cloud infrastructure is aggregation of two or more distinct cloud infrastructures (private, public or community) that remain unique entities but are bound together that enables data and application portability..

V. TASK SCHEDULING

Task scheduling is the allocation of task to particular resources for better utilization of resources so that the execution time is reduced. Task scheduling in cloud computing is a NP-hard problem. Therefore many heuristic approaches have been proposed.[8]

VI. EXISTING ALGORITHMS

4.1 min min algorithm

A compartive study between Hungarian and Antcolony optimization algrorithm for task scheduling in the Cloud

It is based on the concept of minimum completion time. Expected completion time is calculated for each task. Task with overall minimum completion time is scheduled. [6,7]

4.2 max min algorithm

It is based on the concept of maximum completion time. Expected completion time is calculated for each task. Task with maximum completion time is allocated the resource which gives minimum completion time.[7]

4.3 Priority based algorithm

The tasks to be performed are given priority. Priority may be defined internally or externally. Internally defined priorities are make use of some measurable quantity to calculate priority. External priorities are defined using criteria beyond OS which includes significance of process.

4.3 FCFS algorithm

What comes first is handled first, the next request in line executes once the one before it completes.

VII. HUNGARIAN ALGORITHM

The Hungarian method is a combinatorial optimization algorithm that solves the assignment problem in polynomial time and which anticipated later primal-dual methods. It was developed and published in 1955 by Harold Kuhn, who gave the name "Hungarian method" because the algorithm was largely based on the earlier works of two Hungarian mathematicians: Dénes Kőnig and Jenő Egerváry[14]. James Munkres reviewed the algorithm in 1957 and observed that it is (strongly) polynomial. Since then the algorithm has been known also as the Kuhn–Munkres algorithm or Munkres assignment algorithm. The time complexity of the original algorithm was $O(n^4)$, however Edmonds and Karp, and independently Tomizawa noticed that it can be modified to achieve an $O(n^3)$ running time.[13]

Algorithm

1. Subtract the smallest number in **each row** from every number in the row. This is called a row reduction. Enter the results in a new table.
2. Subtract the smallest number in **each column** of the new table from every number in the column. This is called a column reduction. Enter the results in another table.
3. Test whether an optimum assignment can be made. You do this by determining the minimum number of lines needed to cover all zeros. If the number of lines equals the number of rows, an optimum assignment is possible. In that case, go to step 6. otherwise go on to step 4.
   i. Tick all the unassigned rows.
   ii. If ticked **row has zero**, then tick the corresponding column.
   iii. If a ticked **column has an assignment**, then tick the corresponding row.
   iv. Repeat steps (ii) and (iii) till no more ticking is possible.
   v. Draw lines through unticked rows & ticked columns.
4. If the number of lines is less than the number of rows, modify the table in this way:
   i. **Subtract** the smallest uncovered number from every uncovered number in the table.
   ii. **Add** the smallest uncovered number to the numbers at intersections of covering lines.
   iii. Numbers crossed out but not at intersections of cross-out lines carry over unchanged to the next table.
5. Repeat the steps 3 and 4 until an optimal table is obtained.
6. Make the assignments. Begin with rows or columns with only one zero. Match items that have zeros, using only one match for each row and each column. Cross out both the row and the column after the match.[12]

Step 1:

$$\begin{bmatrix} 90 & 75 & 75 & 80 \\ 35 & 85 & 55 & 65 \end{bmatrix}$$

125	95	90	105
45	110	95	115

Step 1:

$$\begin{bmatrix} 15 & 0 & 0 & 5 \\ 0 & 50 & 20 & 30 \\ 35 & 5 & 0 & 15 \\ 0 & 65 & 50 & 70 \end{bmatrix}$$

Step 2:

$$\begin{bmatrix} 15 & 0 & 0 & 0 \\ 0 & 50 & 20 & 25 \\ 35 & 5 & 0 & 10 \\ 0 & 65 & 50 & 65 \end{bmatrix}$$

Step 3 (i):

$$\begin{bmatrix} 15 & \boxed{0} & 0 & 0 \\ \boxed{0} & 50 & 20 & 25 \\ 35 & 5 & \boxed{0} & 10 \\ 0 & 65 & 50 & 65 \end{bmatrix}$$

Step 3 (ii):

$$\begin{bmatrix} 15 & \boxed{0} & 0 & 0 \\ \boxed{0} & 50 & 20 & 25 \\ 35 & 5 & \boxed{0} & 10 \\ 0 & 65 & 50 & 65 \end{bmatrix}$$

Step 3(iii):

$$\begin{bmatrix} \cancel{15} & \cancel{\boxed{0}} & \cancel{0} & \cancel{0} \\ \boxed{} & & & \end{bmatrix}$$

$$\begin{matrix} 0 & 50 & 20 & 25 \\ 35 & 5 & \boxed{0} & 10 \\ 0 & 65 & 50 & 65 \end{matrix} \Leftarrow$$

Step 4:

$$\begin{bmatrix} 15 & 0 & 0 & 0 \\ 0 & 50 & 20 & 25 \\ 35 & 5 & 0 & 10 \\ 0 & 65 & 50 & 65 \end{bmatrix}$$

Step 4:

$$\begin{bmatrix} 35 & 0 & 0 & 0 \\ 0 & 30 & 0 & 5 \\ 55 & 5 & 0 & 10 \\ 0 & 45 & 30 & 45 \end{bmatrix}$$

Step 5 &6:

$$\begin{bmatrix} 40 & \boxed{0} & 5 & 0 \\ 0 & 25 & 0 & \boxed{0} \\ 55 & 0 & \boxed{0} & 5 \\ \boxed{0} & 40 & 30 & 40 \end{bmatrix}$$

Step 5&6:

$$\begin{bmatrix} 40 & 0 & 5 & \boxed{0} \\ 0 & 25 & \boxed{0} & 0 \\ 55 & \boxed{0} & 0 & 5 \\ \boxed{0} & 40 & 30 & 40 \end{bmatrix}$$

Assignment pairs are indicated by the positions of the marked zeros in the cost matrix. If C(i,j) is a starred zero, then the task associated with row i is assigned to the element virtual machine with column j. Makespan is known as the completion time of all jobs. It is calculated by adding corresponding $C_{i,j}$'s.

Therefore there are two solutions for the given assignment problem.

SOLUTION 1:

Task 1 assigned to VM 2

Task 2 assigned to VM 4

Task 3 assigned to VM 3

Task 4 assigned to VM 1

Makespan=75+65+90+45=275

SOLUTION 2:

Task 1 assigned to VM 4

Task 2 assigned to VM 3

Task 3 assigned to VM 2

Task 4 assigned to VM 1

Makespan=80+55+95+45=275

VIII. ANT COLONY OPTIMIZATION

The complex social behaviors of ants have been much studied, and now scientists are finding that these behavior patterns can provide models for solving difficult combinatorial optimization problems.

Ants communicate to one another by laying down pheromones along their trails, so where ants go within and around their ant colony is a stigmergic system. In many ant species, ants walking from or to a food source, deposit on the ground a substance called pheromone. Other ants are able to smell this pheromone, and its presence influences the choice of their path, that is, they tend to follow strong pheromone concentrations. The pheromone deposited on the ground forms a pheromone trail, which allows the ants to find good sources of food that have been previously identified by other ants.[9]

$$p_{ij} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{h \in s}[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}$$

Where τ is the pheromone, η is the inverse of the distance between the two nodes. This probability determines ant k at node i to choose node j such p[i][j] has the greatest probability.

Ant System was the first ACO algorithm to be proposed in the literature. Its main characteristic is that the pheromone values are updated by all the ants that have completed the tour. The pheromone update for τij, that is, for edge joining cities i and j, is performed as follows:[10,11]

$$\tau_{ij}(t) = (1-\rho).\tau_{ij}(t-1) + \sum_{k=1}^{m} \Delta \tau_{ij}^{k}$$

where ρ is the evaporation rate, m is the number of ants, and $\Delta\tau kij$ is the quantity of pheromone per unit length laid on edge (i,j) by the kth ant:

$\Delta\tau^{k}_{ij}$ = $\{$ Q/L_k if ant k used edge (i,j) in its tour

0 otherwise

where Q is a constant and Lk is the tour length of the kth ant.

To solve our assignment probability, we choose $\alpha=1$ and $\beta=1$. $\tau0$,the initial amount of pheromone is also equal to 1.we assume that there is no evapouration of pheromone, therefore ρ is equal to 1.

At first iteration ,

Ant1:

$$P^1_{11} = \frac{1/90}{1/90+1/75+1/75+1/80} = 0.22$$

$$P^1_{12} = \frac{1/75}{1/90+1/75+1/75+1/80} = 0.26$$

$$P^1_{13} = \frac{1/75}{1/90+1/75+1/75+1/80} = 0.26$$

$$P^1_{14} = \frac{1/80}{1/90+1/75+1/75+1/80} = 0.24$$

Ant 2

$$P^2_{21} = \frac{1/35}{1/35+1/85+1/55+1/65} = 0.38$$

$$P^2_{22} = \frac{1/85}{1/35+1/85+1/55+1/65} = 0.15$$

$$P^2_{23} = \frac{1/55}{1/35+1/85+1/55+1/65} = 0.24$$

$$P^2_{24} = \frac{1/65}{1/35+1/85+1/55+1/65} = 0.20$$

Ant 3

$$P^3_{31} = \frac{1/125}{1/125+1/95+1/90+1/105} = 0.20$$

$$P^3_{32} = \frac{1/95}{1/125+1/95+1/90+1/105} = 0.26$$

$$P^3_{33} = \frac{1/90}{1/125+1/95+1/90+1/105} = 0.28$$

$$P^3_{34} = \frac{1/105}{1/125+1/95+1/90+1/105} = 0.24$$

Ant 4

$$P^4_{41} = \frac{1/45}{1/45+1/110+1/95+1/115} = 0.43$$

$$P^4_{42} = \frac{1/110}{1/45+1/110+1/95+1/115} = 0.18$$

$$P^4_{43} = \frac{1/95}{1/45+1/110+1/95+1/115} = 0.20$$

$$P^4_{44} = \frac{1/115}{1/45+1/110+1/95+1/115} = 0.17$$

At second iteration,

$$\eta = \begin{bmatrix} 1 & 2 & 1 & 1 \\ 2 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 2 & 1 & 1 & 1 \end{bmatrix}$$

Ant 1

$$P^1_{21} = \frac{1/35*2}{1/35*2+1/55+1/65} = 0.62$$

$$P^1_{23} = \frac{1/55}{1/35*2+1/55+1/65} = 0.20$$

$$P^1_{24} = \frac{1/65}{1/35*2+1/55+1/65} = 0.17$$

Ant 2

$$P^2_{12} = \frac{1/75*2}{1/75+1/75*2+1/80} = 0.50$$

$$P^2_{13} = \frac{1/75}{1/75+1/75*2+1/80} = 0.25$$

$$P^2_{14} = \frac{1/80}{1/75+1/75*2+1/80} = 0.23$$

Ant 3

$$P^3_{12} = \frac{1/75*2}{1/90+1/75*2+1/80} = 0.68$$

$$P^3_{11} = \frac{1/90}{1/90+1/75*2+1/80} = 0.13$$

$$P^3_{14} = \frac{1/80}{1/90+1/75*2+1/80} = 0.18$$

Ant 4

A compartive study between Hungarian and Antcolony optimization algrorithm for task scheduling in the Cloud

$$P^4_{12} = \frac{1/75*2}{1/75*2+ 1/75+1/80} =0.50$$

$$P^4_{13} = \frac{1/75}{1/75*2+ 1/75+1/80} =0.25$$

$$P^4_{14} = \frac{1/80}{1/75*2+ 1/75+1/80} =0.23$$

At third iteration,

$$\eta = \begin{bmatrix} 1 & 5 & 1 & 1 \\ 3 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 2 & 1 & 1 & 1 \end{bmatrix}$$

Ant 1

$$P^1_{33} = \frac{2/90}{2/90+ 1/105} =0.7$$

$$P^1_{34} = \frac{1/105}{2/90+ 1/105} =0.3$$

Ant 2

$$P^2_{33} = \frac{2/90}{2/90+ 1/105} =0.7$$

$$P^2_{34} = \frac{1/105}{2/90+ 1/105} =0.3$$

Ant 3

$$P^3_{41} = \frac{1/45*2}{1/45*2+ 1/115} =0.83$$

$$P^3_{44} = \frac{1/115}{1/45*2+ 1/115} =0.16$$

Ant 4

$$P^4_{23} = \frac{1/55}{1/55+ 1/65} =0.54$$

$$P^4_{24} = \frac{1/65}{1/55+1/65} =0.45$$

TOTAL COST OF PATHS TRAVELLED BY ANTS

ANT 1: 75+35+90+115 = 315

ANT 2: 35+75+90+115 = 315

ANT 3: 90+75+45+65 = 275

ANT 4: 45+75+55+105 = 280

Therefore ant 3 has travelled in minimum path. Taking the assignments done by ant3:

Task 3 assigned to VM 3

Task 1 assigned to VM 2

Task 4 assigned to VM 1

Task 2 assigned to VM 4

Makespan=75+65+90+45=275

IX. CONCLUSION

In this work we have successfully applied the Ant colony System and Hungarian algorithm to the Quadratic Assignment Problem (QAP). The results obtained using Hungarian algorithm are better than the results obtained using ACO in some cases especially during multiple solutions. Ant colony algorithms being a probabilistic method converge to a single solution. As ant system is based on behaviour of ants, stagnation becomes a problem. Hungarian is an efficient method for solving assignment problems in polynomial time.

REFERENCES

[1] Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. A taxonomy and survey of cloud computing systems. “Networked Computing and Advanced Information Management”, International Conference on, 0:44–51, 2009.

[2] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009..

[3] F.M. Aymerich, G. Fenu, and S. Surcis. An approach to a cloud computing network. “Applications of Digital Information and Web Technologies”, 2008. ICADIWT 2008., pages 113 –118, August 2008.

[4] Grossman, R. L. The case of cloud computing,proc. Of IEEE ,Educational Activities Department,Piscataway, NJ, USA vol.11,Issue 2,pp 23-27,March,2009.

[5] B. Grobauer, T. Walloschek, and E. Stöcker, “Understanding Cloud Computing Vulnerabilities,” 2011 IEEE Security and Privacy, pp. 50-57, DOI= March/April 2011.

[6] Nidhi Bansal, Amitab Maurya, Tarun Kumar, Manzeet Singh, Shruti Bansal, “Cost performance of QoS Driven task scheduling in cloud Computing” Third International Conference on Recent Trends in Computing (ICRTC 2015) Published by Elsevier B.V. © 2015.

[7] K. Etminani, and M. Naghibzadeh, "A Min-min Max-min Selective Algorithm for Grid Task Scheduling,"The Third IEEE/IFIP International Conference on Internet, Uzbekistan, 2007..

[8] Huang Q.Y., Huang T.L.,“An Optimistic Job Scheduling Strat egy based on QoS for Cloud Comput ing”, IEEE Int ernat ional Conference on Intelligent Comput ing and Integrated Systems (ICISS), 2010, Guilin, pp. 673-675, 2010.

[9] “An Introduction to Ant Colony Optimization”,Marco Dorigo and Krzysztof Socha IRIDIA, Universit´e Libre de Bruxelles, CP 194/6, Av. Franklin D. Roosevelt 50, 1050 Brussels, Belgium http://iridia.ulb.ac.beApril 30, 2007,pp.6-7.

[10] M. Dorigo, L M Gambardella “Ant colonies for the traveling salesman problem”. BioSystems, 1997

[11] “An Ant Colony Optimization Algorithm for Solving Travelling Salesman Problem” ,Krishna H. Hingrajiya, Ravindra Kumar Gupta, Gajendra Singh Chandel.

[12] R. A. Pilgrim, Munkres' Assignment Algorithm. Modified for Rectangular Matrices, Course notes, Murray State University.

[13] Bruff, Derek, The Assignment Problem and the Hungarian Method (matrix formalism).

[14] Mordecai Golin. Dept. of Computer Science, Hong Kong UST. “Bipartite Matching & the Hungarian Method”