IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134177

EFFICIENT RESOURCE SCHEDULING AND LOAD BALANCE USING IMPROVED ANT COLONY OPTIMIZATION ALGORITHM IN GRID COMPUTING

R. ANANTHI LAKSHMI[1], DR.S. VIDHYA[2]

Research Scholar, Department of Computer Science, KG College of Arts and Science, Coimbatore, Tamil Nadu, India[1]

Associate Professor, Department of Information Technology, KG College of Arts and Science, Coimbatore, Tamil Nadu, India[2]

Abstract: Grid computing systems are very large scale and can be used in internet sized environments with distributed machines across multiple organizations The resource scheduling system is the central component of the grid computing system. Resources in the grid are distributed, heterogeneous and autonomous**.** Scheduling in the grid environment depends on the characteristics of tasks, machines, and network connectivity. The number of jobs in the waiting line is known as the load and depending on the nature of the work it can be low, moderate, or heavy. The method of load balancing involves enhancing the computational grid systems performance so that all of the grids computing nodes are uniformly employed to the greatest extent possible to reduce execution time and increase throughput.

Keywords: Grid computing, Resource, Load Balance

I. INTRODUCTION

Grid computing is now developing into a very promising technology that makes efficient use of resource idle time. Grid's emphasis on high performance and large-scale resource sharing sets it apart from traditional distributed computing. According to Ian Foster [3], a grid is a dynamic collection of people, institutions, and resources that are shared in a flexible, safe, and coordinated manner. These groups are known as virtual organizations.

The majority of intricate problems in science, engineering, and industry require enormous resources to solve. Grid computing is seen to be the most effective way to handle these issues [4]. In addition, grid computing finds application in fields such as multi particle physics, astrophysics, bioinformatics, earthquake research, and ground water pollution. Due to the growing usage of Grid technology in various industries, a large number of developers and researchers concentrate on creating the hardware and software required for Grid design. In the field of grid computing, some of the more difficult problems are scheduling, performance prediction, and resource management [5]. It has been established that one of the NP-hard issues in parallel computing is scheduling. Grid scheduling presents unique challenges due to the heterogeneity of its operating systems and architecture.

One of the major concerns in the current Grid environment is scheduling [4]. In order to attain high performance computing, there is an increasing need for efficient scheduling. Finding the best way to divide up resources such that jobs take less time to complete and the resources are used efficiently is usually challenging. The three primary stages of grid scheduling are job execution, resource information collection, and resource discovery [6]. It has been established that selecting the optimal set of tasks and resources for the second phase is an NP-complete problem.

Users of the Grid put together a distributed application. Next the users send Grid Resource Broker their job submissions. The resource broker then sends an inquiry to the Grid Information Service to find out if resources are available and what their characteristics are. One or more information services have registered the Grid Resources. The scheduling of jobs on resources that meet the needs of the jobs is the responsibility of the resource broker following scheduling. The resource broker keeps an eye on how jobs are carried out to gathers the findings and send them back to the users [6]. Users of the Grid put together a distributed application. Next the users send Grid Resource Broker their job submissions. The resource broker then makes a request to the Grid Information Service to find out if resources are available.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134177

Static and dynamic load balancing algorithms comprise the two main classes. In static load balancing decisions are made with advance knowledge. In a static algorithm every piece of prior knowledge about the schedule is known. In contrast scheduling decisions in dynamic load balancing are made in response to the need to schedule a job for more processing. Control for a dynamic task scheduling might be dispersed or centralized.

With a centralized method all scheduling decisions are made at a single location. Each site determines its own schedule in a distributed or noncooperative method and the control is far more scalable and dependable [26].

Changing state data often is unacceptable for a dynamic load balancing method due to the significant communication overheads Improved Ant Colony Optimization Algorithm (IACO) are proposed in [3]. When a job arrives a processor uses precise data regarding the load, arrival rate and service rate of each job completion time on all of them. If the processor can complete the task with the shortest finish time and migrates the job right away to it. In the Grid-based decentralized load balancing algorithm that was suggested in [5].

In order to balance convergence speed and solution diversity and enhance optimization performance when solving large-scale optimization problems an improved ant colony optimization (IACO) algorithm based on the multi population strategy,co-evolution mechanism, pheromone updating strategy and pheromone diffusion mechanism is proposed in this paper.

In order to increase the convergence rate and prevent falling into the local optimum value and the optimization issue is split into multiple sub problem and the ants in the population are separated into elite and common ants.The optimization ability is enhanced by the pheromone update technique. A range of nearby locations are gradually affected by the pheromone released by ants at a certain point according to the pheromone diffusion process.

II. RELATED WORK

A QoS-based predictive Max-Min, Min-Min Switcher algorithm for grid work scheduling is presented by Singh.M et al. [1]. Before scheduling the next job, the algorithm determines which of the two algorithms—QoS based Min-Min or QoS based Max-Min—is appropriate based on the heuristic applied. The impact of non-dedicated resource properties on grid job execution times has also been taken into account. The programme forecasts the performance of non-dedicated resources based on historical data regarding the completion of tasks.

An algorithm to schedule tasks within grid resources and a model to illustrate grid architecture have been provided by Yagoubi B et al. [2]. The method attempts to equitably allocate the grid environment's burden among the grid resources. The overall makespan of the system does not always reduce, even while the method employed here and other comparable tactics that aim to achieve load balancing within grid resources can increase the throughput of the entire grid environment.

Load balancing techniques have been extensively researched in conventional distributed systems consisting of homogeneous and dedicated resources. However, these algorithms won't function well in grid architecture due to its autonomy, scalability, and heterogeneity [3]. This increases the difficulty of the load balanced scheduling algorithm for grid computing and makes it a fascinating subject for numerous academics.

The non-traditional algorithms are different from the traditional conventional algorithms in that they yield optimal solutions quickly. For every grid computing system there is not a single optimal scheduling algorithm. An alternative is to choose the right scheduling algorithm based on the machines, tasks, and network heterogeneity in a particular grid setting [4].

In recent years numerous enhanced ACO algorithms have been presented by specialists and scholars for the purpose of studying the algorithm. A modified continuous technique of ACO paired with a differential evolution method was developed by Coelho et al. [1].

For the time and space assembly line balancing problem, eight multi-objective ACO algorithms were proposed by Rada-Vilela et al. [2]. A multi-objective ACO method was suggested by He and Ma [3] to investigate non-redundant linear sensor network design issues. A co-evolution continuous ACO method was presented by Juang et al. [4] to solve accuracy-oriented fuzzy system design issues.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134177

III. PROPOSED METHODOLOGY

The ACO algorithm is made up of several repetitions. Several ants build comprehensive solutions in each iteration by utilizing heuristic knowledge and the accumulated experiences of earlier ant populations. The pheromone trail, which is left behind on the component parts of a solution, is used to symbolize these gathered experiences. Depending on the problem to be solved, the pheromone may be deposited on the connections or the component parts of a solution. The following describes the pheromone update rule technique.

A Resource is selected based on the current work load and fault rate of the resources, and before forwarding the job to the selected resource, a job dispatcher dispatches the jobs to the checkpoint handler. On receiving the job from the job dispatcher, the checkpoint handler gets the number of successful jobs completed and number of failed jobs of the resource from the fault index handler on which it is scheduled and sets the number of time to checkpoint and the checkpoint interval based on the failure rate of the resource.

The checkpoint handler then submits the job along with the checkpoints to the selected resource. The checkpoint repository receives partially executed result of a task from a Grid resource in the intervals specified by the checkpoint handler. It maintains Grid tasks and their checkpoint table which contains information of partially executed tasks by the Grid resources. For a particular job, the checkpoint repository discards the result of the previous checkpoint, when a new value of the checkpoint result is received. When a particular job is completed, its entity will be removed from the checkpoint result table.

Algorithm 1: Pseudocode for the *AntZ* algorithm (ACO)

A user submits a Job to its local resource node
An Ant is created and invoked in response to the users requests, the job is delivered to the ant.
For each Job {
(A)Ant starts to move from node to node for a number of steps searching for the best suitable node (lightest loaded node).
For each step taken by the ant
{ (i) Pheromone Laying
Ant collects load information of the node it is visiting and adds to its history
Updates the load information table in the visiting node
(ii) Decision Making
Look up Load table information of nodes
Random selection with a probability of mutation factor
}//end for
(B) Ant deliver job to a particular resource and dies
}//End for

IV. RESULTS AND DISCUSSION

1.THE TRANSITSION RULES

An ant is a simple computational agent in the ACO algorithm. It iteratively constructs a solution for the problem at hand. At each iteration of the algorithm, each ant moves from a state r to state s, corresponding to a more complete intermediate solution. The k^{th} ant from state r to state s is selected among the unvisited states memorized in according to the following formula

$$s = \arg\max_{u \epsilon J^k} [\tau_i(r,u)^{\alpha} \cdot \eta(r,u)^{\beta}] \text{ if } q \leq q_0 \text{ (Exploitation)} \quad (1)$$

The trail level represents a posteriori indication of the desirability of that move. Trails are updated usually when all ants have completed their solution increasing or decreasing.

In general, the kth ant moves from state r to state s with the probability $p_k(r, s)$

The basic ACO algorithm, IACO algorithm and the proposed are executed 10 times respectively. The indexes of maximum value, minimum value, average value and variance of the 10 results are used to describe and compare the experiment results. The experiment results are shown in Table I.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

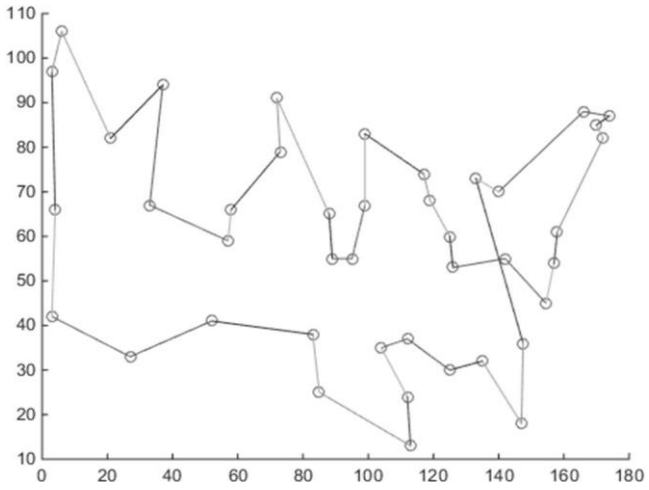
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134177

Here the optimal value represents the obtained best value. The maximum value represents the obtained maximum value in the simulation test of 10 times, the minimum value represents the obtained minimum value in the simulation test of 10 times, the average value represents the average value of 10 times. The variance represents the variance between the maximum value and the minimum value.

TABLE 1:THE TEST OF THE PARAMETER

Algorithms	**ACO**	**IACO**
Ants(k)	30	30
Pheromone factor(α)	1	1
Heuristic factor(β)	5	5
Volatility coefficient(ρ)	0.1	0.1
Pheromone amount(Q)	100	100
Initial concentration($\tau_{ij}(0)$)	1.5	1.5
Maximum iterations(T)	200	200



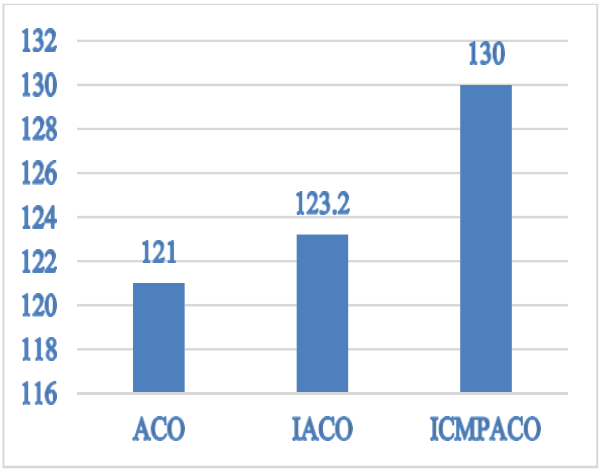


FIGURE 1.THE OPTIMAL PATH OF IACO FIGURE 2:COMPARISON OF ACO &IACO

IACO algorithm has better optimization ability than the basic ACO algorithm and the IACO algorithm in solving these TSP standard instances. From the average value the IACO algorithm can also obtain the best average value. It shows that the optimization performance of the IACO algorithm is more obvious advantage.

It can be seen that the variance of the optimization performance of the proposed IACO algorithm in solving the the smallest value.

V. CONCLUSION AND FUTURE WORK

The optimization performance of the IACO algorithm is compared with the basic ACO algorithm and the IACO algorithm in solving the traveling salesmen problem and gate assignment problem. The proposed IACO algorithm can obtain the best optimization value in solving these TSP standard examples and the gate assignment problem.

It can assign 132 flights to 20 gates and the assigned efficiency reaches 83.5%, and fast obtain the ideal gate assignment result.

Therefore the proposed IACO algorithm takes on better optimization ability and stability than the ACO algorithm. Because the IACO algorithm exists the longer computation time in solving complex optimization problem the IACO algorithm need to further be studied in order to reduce the time complexity.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134177

REFERENCES

[1]. Luo J, Chen HL, Zhang Q, Xu YT, Huang H, Zhao XH. An improved grasshopper optimization algorithm with application to financial stress prediction. AppliedMathematical Modelling, 2018, 64:654-668.

[2]. Ajeil FH, Ibraheem IK, Azar AT, Humaidi AJ. Grid-Based Mobile Robot Path Planning Using Aging-Based Ant Colony Optimization Algorithm in Static and Dynamic Environments. *Sensors*. 2020:1880.https://doi.org/10.3390/s20071880.

[3]. Sharifipour H, Shakeri M, Haghighi H. Structural test data generation using a mimetic ant colony optimization based on evolution strategies. Swarm and Evolutionary Computation, 2018,40:76-91.

[4]. Zhong YG, Ai B. A modified ant colony optimization algorithm for multi-objective assembly line balancing. Soft Computing, 2017,21(22): 6881-6894.