Scholar Publishing Group
Distributed Processing System
https://doi.org/10.38007/DPS.2021.020106
ISSN 2790-0916 Vol. 2, Issue 1: 75-82

Real-time Fault-tolerant Task Scheduling Design of Ant Colony Algorithm for Solving Distributed Systems

Additya Kummar[*]

Shabakeh Pardaz Azarbaijan, Iran

[*]*corresponding author*

Keywords: Ant Colony Algorithm, Distributed System, Real-time Fault Tolerance, Task Scheduling

Abstract: With the increasing popularity of computer applications, the scope of real-time distributed systems is expanding. Real-time systems have strict real-time and reliability requirements, and ensuring the real-time performance and reliability of real-time systems has become an urgent problem to be solved in real-time system research. The purpose of this article is to solve the real-time fault-tolerant task scheduling design of distributed systems by ant colony algorithm. A global optimization of checkpoint interval and fault-tolerant priority search algorithm are proposed to solve the established optimization model and further reduce the number of tasks. The response time improves the schedulability of the system. In order to ensure the efficient use of resources, an adaptive resource adjustment strategy is used to adjust the number of virtual machines. The practicability of the algorithm is verified by scheduling simulation of 9100 task sets through simulation tests. The results show that with the increase of the number of task failures, the ant colony algorithm can significantly reduce the overhead of computing resources, and the reduction range is about 95%. Compared with FTSA, the communication overhead is More communication resources can be saved. Compared with FTSA, the ant colony algorithm can save more than 98% of the overhead of communication resources.

1. Introduction

The basic concept of fault tolerance is to increase the reliability of the system by adding redundancy, but this redundancy will generate additional system costs, which requires the use of resources as much as possible while ensuring fault tolerance. In addition, under the technical constraints of real-time tasks, cloud computing systems need computing resources with sufficient performance to ensure the immediate execution of tasks [1]. However, increasing performance means increasing efficiency, which requires reducing system consumption while maintaining performance. Therefore, it is the key of this paper to improve system efficiency and equipment efficiency while adapting to system operating load and system fault tolerance requirements [2].

To solve the problem of real-time fault-tolerant task scheduling in distributed systems, a lot of research has been done in the academic circles. Aziza H brings mission planning to cloud computing. This problem is notoriously difficult to solve because it has many points. The main function of your model is to estimate the time required to perform certain tasks in the cloud, thereby reducing processing costs. A genetic approach to modeling and solving operational design problems in cloud computing is presented. Therefore, a decision support system based on CloudSim was developed. In terms of cost-effectiveness, the obtained results show that their method has significant advantages over previous programming methods. Timeline results are also shorter than other algorithms in terms of completion time [3]. Krishnadoss P proposed a solution that uses production time and cost as key constraints for the optimization problem. Combining two algorithms, Cuckoo Search Algorithm (CSA) and Object-Based Learning (OBL), a new hybrid algorithm called Cuckoo Search Algorithm (OCSA) was created to solve the above problems. Compared with other task scheduling algorithms, the proposed OCSA algorithm has significant improvement. The proposed work is simulated in the cloudsim programming environment, and the simulation results show the effectiveness of the proposed work in reducing cost and construction time parameters. Compared with other existing algorithms such as particle swarm optimization (PSO) and genetic algorithm (GA), the obtained results are better [4]. Jawade P B discussed the research of different task scheduling algorithms under distributed computing conditions. This review clearly demonstrates the different techniques used for career planning. In addition, security-based operational planning activities are also analyzed. Analyze the performance evaluation of different operational planning strategies [5]. In summary, although there are many studies on fault-tolerant programming of autonomous multitasking distributed real-time systems, the combination of distributed scheduling and local real-time processor programming still needs further discussion.

To study the new requirements and characteristics of distributed real-time systems in the current application field, and to study the impact of these requirements and characteristics on the system model. The algorithm is improved and an efficient real-time fault-tolerant scheduling algorithm is designed. Scientific scheduling can not only effectively deal with emergency situations such as resource input and output, network failure and resource failure, crash fault tolerance, system capacity improvement and service quality assurance in the cloud environment, but also improve the utilization of resources in the cloud, and the cloud environment can fully exploit resources diversity, play to the individuality of resources, and use resources in the best possible way. In addition, the complex and changeable cloud computing environment makes the study of programming problems full of challenges. It is very important to understand the programming direction in the cloud computing environment.

2. Research on Real-time Fault Tolerant Task Scheduling of Distributed System

2.1. Distributed System

Distributed system is a kind of control system based on data communication technology through the cooperation of multi-processors to achieve a common goal, using the advantages of screen control and direct computer control [6-7]. From the hardware point of view, each node is autonomous, but from the software point of view, the user treats the entire system as a host and performs unified programming [8-9].

2.2. Ant Colony Algorithm

The Ant Colony (ACO) algorithm is an analog optimization algorithm that simulates the intelligent behavior of foraging ant colonies. It mimics the foraging behavior of real insect communities in forests and releases chemicals called pheromones along the foraging path [10-11]. By looking at this element, all insects can choose the shortest distance from the insect colony to the food, and finally get the overall optimal solution. At present, many researchers have successfully used ant colony algorithm to solve "NP-hard" problems, color matching problem [12-13].

2.3. Task Scheduling Algorithm

In the real-time task scheduling process, three constraints are involved: resource constraints, time constraints and priority constraints.

Accounting tasks in real-time systems are subject to strict time constraints that must be met to achieve desired goals. For tasks in real-time systems, this period of time is called deadline, which represents the time when the task was last completed [14-15].

For tasks, the available computing resources include CPU, memory, hard disk, bandwidth, etc. To reduce system load and resource performance, the task scheduling process is not limited by the number of resources [16-17].

In complex applications, there are often priority constraints between tasks, that is, tasks are not completely independent, and some tasks must depend on the execution results of other tasks [18].

3. Design and Experiment of Ant Colony Algorithm for Solving Real-time Fault-tolerant Task Scheduling of Distributed Systems

3.1. Model Design Based on the Ant Colony Algorithm Part

(1) Predict the execution time of subtasks

Calculate the predicted execution time Vm Time required for each virtual machine computing resource to complete all subtask sequences assigned to it by the system, as shown in formula (1):

$$vmTime(VM_j) = \sum_{i=1}^{n} Time(T_i, VM_j) \quad (i \in [1, N]) \tag{1}$$

Among them, time (T_i, VM_j) represents the time required for the ith subtask T_i allocated to the virtual machine VM_j to complete the execution.

(2) Predictive execution capability

EV_j calculation method is shown in formula (2):

$$EV_j = pe\_num_j * pe\_mips_j \tag{2}$$

where pe_mips_j represents the processing speed of each VM_j processor.

Therefore, based on the above assumptions and descriptions, the task scheduling problem in cloud computing can be defined as how to reasonably allocate the sub-task sequence submitted by the user to the computing resources of each virtual machine, so that the time and cost required for the completion of the task can meet the needs of the user as much as possible.

3.2. Fault-tolerant Design of Distributed Systems

(1) Fault-tolerant design of real-time distributed system

Real-time planning not only needs to ensure the correctness of the logical relationship between tasks, but also needs to respond to various events and complete the corresponding tasks within the specified time. According to statistics, the probability of a computer failure in a real-time environment is dozens of times that of the computer room. Therefore, reliable technologies must be used in distributed computing systems to truly realize their potential benefits. Using fault-tolerant technology, especially redundancy technology, in real-time distributed computing system is an important way to improve system reliability.

(2) Network fault tolerance technology

In this system, all nodes are connected to both buses at the same time, but only one bus is active and the other is inactive.

3.3. Experimental Environment and Parameter Settings

This simulation test verifies the practicability of the algorithm by scheduling simulation on 9100 task sets, each task scheduling consists of 10 real-time periodic tasks (tasks in the task set can be expanded to any number). The computer configuration used in the simulation experiment is: Intel(R) i7-4710MQCPU, 16G memory, 1000G hard disk.

4. Analysis and Research on Real-time Fault-tolerant Task Scheduling of Distributed System

4.1. Reliability Comparison

Under the same system configuration conditions, the reliability comparison relationship between ant colony algorithm and FTSA with the change of the number is known in Table 1.

Table 1. Reliability of task set algorithms with different number of tasks

Number of tasks	Ant Colony Algorithm/%	FTSA algorithm/%
30	94.36	90.12
40	93.45	90.08
50	92.69	89.69
60	91.36	88.97
70	90.78	88.28

It can be seen from Table 1 that the reliability of ant colony algorithm is greater than that of FTSA, indicating that ant colony algorithm can meet the reliability requirements. The dynamic number of task replicas during task scheduling can also achieve higher reliability.

4.2. Comparison of Computing Resource Occupancy

Table 2 shows the change of the computing resource occupancy cost.

Table 2. Comparison of computing resource usage of algorithms with different failure times

Number of task failures	Ant Colony Algorithm/*10^4	FTSA algorithm/*10^4
1	0.58	0.59
2	0.78	1.48
3	1.28	2.28
4	1.49	2.59

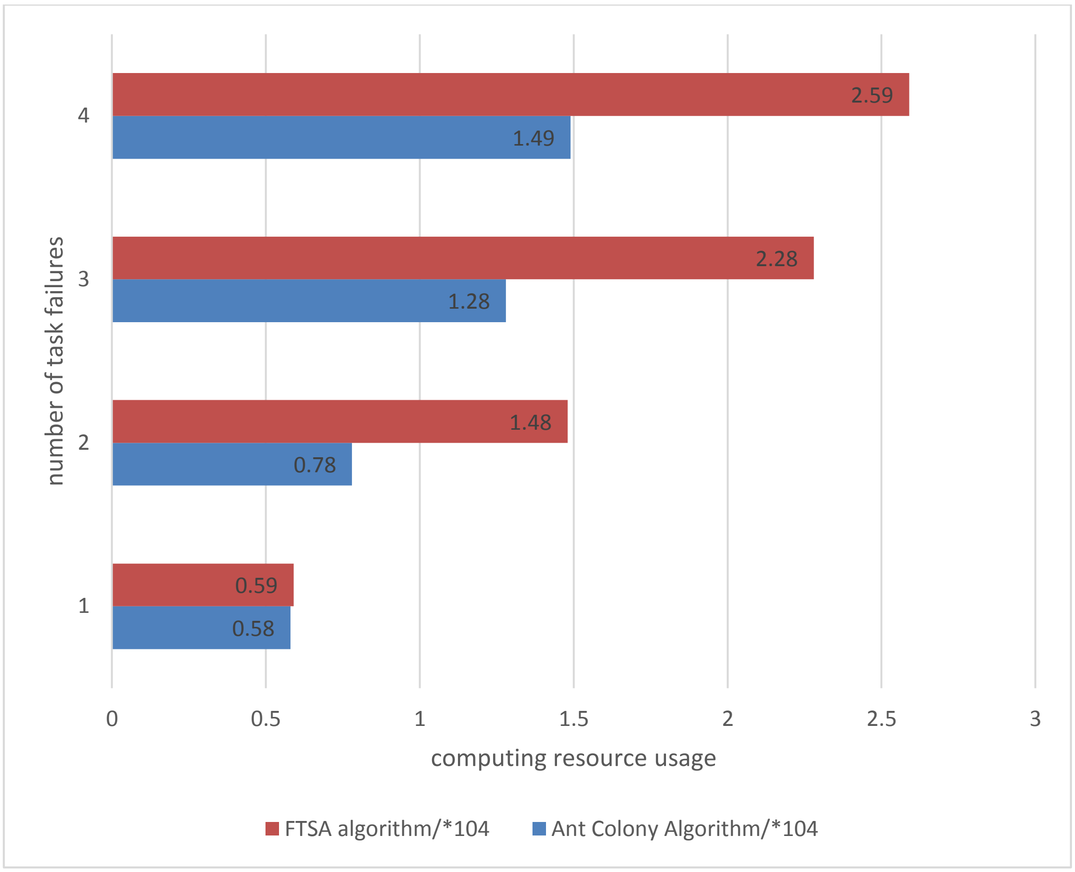

Figure 1. Computational resource occupancy results of the algorithm with different failure times

It can be seen from Figure 1 that when the number of task failures is 2, the computing resource occupation of the ant colony algorithm is 0.78 *10^4, the computing resource occupation of FTSA is 1.48*10^4; when the number of task failures is 3, the computing resource occupation of ant colony algorithm is 1.28*10^4, and the computing resource occupation of FTSA is 2.28*10^4; when the number of task failures is 4, ant The computing resource occupation of the swarm algorithm is 1.49*10^4, and the computing resource occupation of FTSA is 2.59*10^4. Therefore, compared with FTSA, the ant colony algorithm can significantly reduce the overhead of computing resources by about 95%.

4.3. Comparison of Communication Resource Occupancy

The number of tolerable continuous errors in the algorithm is shown in Table 3.

Table 3. Occupation of communication resources at different times of failure

Number of task failures	Ant Colony Algorithm/*10^5	FTSA algorithm/*10^5
1	1.63	1.98
2	1.64	4.36
3	3.36	6.87
4	4.53	10.98

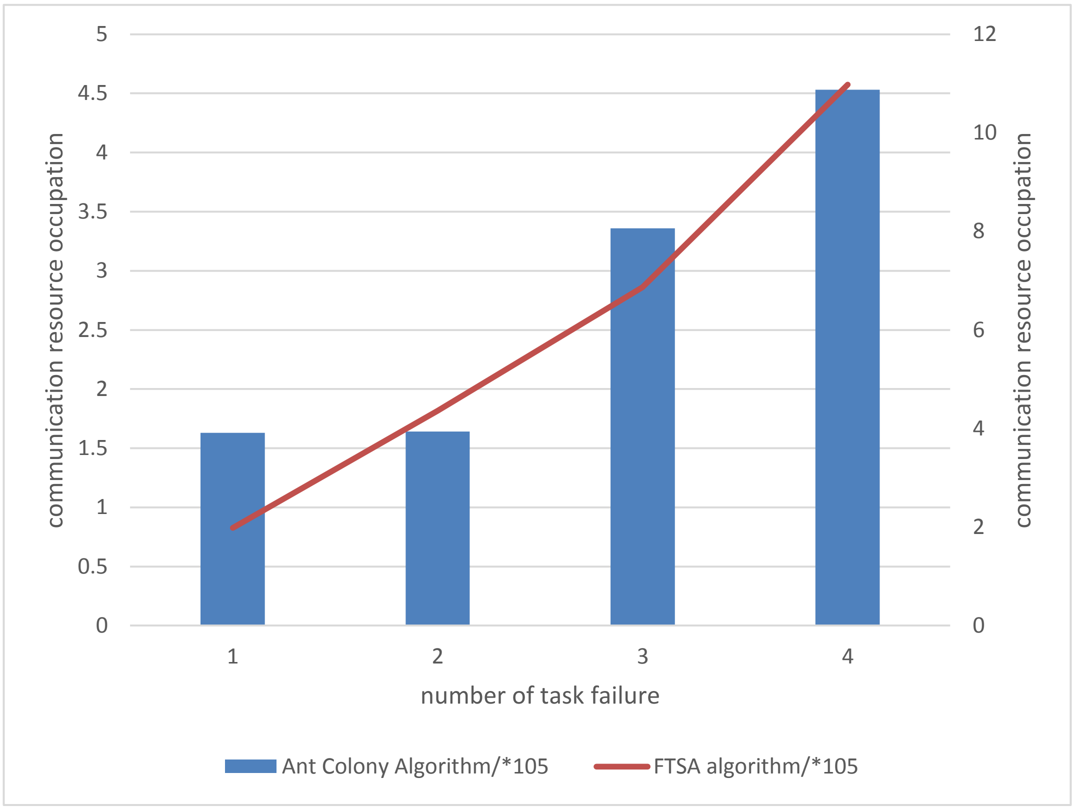

Figure 2. Comparison of communication resource occupancy with different failure times

It can be known from Figure 2 that when the task failures is 1, the communication resource occupation of the ant colony algorithm is 1.63*10^5, and the communication resource occupation of FTSA is 1.98*10^5; when the number of task failures is 2, the communication resource occupation of the ant colony algorithm is 1.64 *10^5, the communication resource occupation of FTSA is 4.36*10^5; when the number of task failures is 3, the communication resource occupation of the ant colony algorithm is 3.36*10^5, and the communication resource occupation of FTSA is 6.87*10^5; when the number of task failures is 4, the ant The communication resource occupation of the group algorithm is 4.53*10^5, and the communication resource occupation of FTSA is 10.98*10^5. With the increase of ε, the communication overhead of ant colony algorithm can save more communication resources than FTSA.

5.Conclusion

According to the application requirements of current real-time systems, machine-based real-time fault-tolerant programming algorithms, and for a large number of real-time distributed tasks in practical applications, this paper studies the theoretical programming and algorithm design and implementation. In the task scheduling of distributed systems, how to deal with computer node failures and task execution failures can meet the QoS requirements of users to the greatest extent, and at the same time maintain a relatively balanced system load of computing resources in urgently needed tasks. Carry out research on fault tolerance of cloud task scheduling, which is of great significance for maximizing user service quality requirements, maximizing economic benefits and improving resource utilization.

Funding

This article is not supported by any foundation.

Data Availability

Data sharing is not applicable to this article as no new data were created or analysed in this study.

Conflict of Interest

The author states that this article has no conflict of interest.

References

[1] Hojati M. A greedy heuristic for shift minimization personnel task scheduling problem. Computers & Operations Research, 2018, 100(DEC.):66-76. https://doi.org/10.1016/j.cor.2018.07.010

[2] Ghanavati S, Abawajy J H, Izadi D. An Energy Aware Task Scheduling Model Using Ant-Mating Optimization in Fog Computing Environment. IEEE Transactions on Services Computing, 2020, PP(99):1-1.

[3] Aziza H, Krichen S. Bi-objective decision support system for task-scheduling based on genetic algorithm in cloud computing. Computing, 2018, 100(2):65-91. https://doi.org/10.1007/s00607-017-0566-5

[4] Krishnadoss P, Jacob P. OCSA: Task Scheduling Algorithm in Cloud Computing Environment. International Journal of Intelligent Engineering and Systems, 2018, 11(3):271-279. https://doi.org/10.22266/ijies2018.0630.29

[5] Jawade P B, Sai K D, Ramachandram S. A Compact Analytical Survey on Task Scheduling in Cloud Computing Environment. International Journal of Engineering Trends and Technology, 2021, 69(2):178-187.

[6] Ahmad S, Malik S, Kim D H. Comparative Analysis of Simulation Tools with Visualization based on Realtime Task Scheduling Algorithms for IoT Embedded Applications. International Journal of Grid and Distributed Computing, 2018, 11(2):1-10.

[7] Chrysafiadi K. Improving task scheduling by using a fuzzy reasoner. Intelligent Decision Technologies, 2020, 14(2):1-6. https://doi.org/10.3233/IDT-190110

[8] Saleem U, Liu Y, Jangsher S, et al. Mobility-Aware Joint Task Scheduling and Resource Allocation for Cooperative Mobile Edge Computing. IEEE Transactions on Wireless Communications, 2020, PP(99):1-1.

[9] Lord S A, Ghasabsaraei M H, Movahedinia M, et al. Redesign of stormwater collection canal based on flood exceedance probability using the ant colony optimization: study area of eastern Tehran metropolis. Water Science and Technology, 2021, 84(4):820-839. https://doi.org/10.2166/wst.2021.273

[10] Soheili S, Zoka H, Abachizadeh M. Tuned mass dampers for the drift reduction of structures with soil effects using ant colony optimization. Advances in Structural Engineering, 2021, 24(4):771-783.

[11] Abdolhosseinzadeh M, Alipour M M. Design of experiment for tuning parameters of an ant colony optimization method for the constrained shortest Hamiltonian path problem in the grid networks. Numerical Algebra, Control, Optimization, 2021, 11(2):321-332. https://doi.org/10.3934/naco.2020028

[12] Sadiq A T, Raheem F A, Abbas N. Ant Colony Algorithm Improvement for Robot Arm Path Planning Optimization Based on D Strategy. International Journal of Mechanical & Mechatronics Engineering, 2021, 21(No. 1):96-111.*

[13] Al-Amyal F, Hamouda M, L Számel. Torque Quality Improvement of Switched Reluctance Motor Using Ant Colony Algorithm. Acta Polytechnica Hungarica, 2021, 18(7):129-150.

[14] Kanso B, Kansou A, Yassine A. Open Capacitated ARC routing problem by Hybridized Ant Colony Algorithm. RAIRO - Operations Research, 2021, 55(2):639-652. https://doi.org/10.1051/ro/2021034

[15] Deol E. Hadoop Job Scheduling Using Improvised Ant Colony Optimization. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 2021, 12(2):3417-3424.

[16] Gayetri D, Nalini T. Optimizing Automated Programming Contracts with Modified Ant Colony Optimization. Indian Journal of Computer Science and Engineering, 2021, 12(1):226-238.

[17] Reshma M, Thomas N, Varghese S M. Dynamic Path Finding using Ant Colony Optimization. International Journal of Recent Technology and Engineering, 2021, 9(5):134-138. https://doi.org/10.35940/ijrte.E5210.019521

[18] Kanso B, Kansou A, Yassine A. Open Capacitated ARC routing problem by Hybridized Ant Colony Algorithm. RAIRO - Operations Research, 2021, 55(2):639-652. https://doi.org/10.1051/ro/2021034