Scholar Publishing Group
Distributed Processing System
https://doi.org/10.38007/DPS.2022.030207
ISSN 2790-0916 Vol. 3, Issue 2: 87-94

Fault-tolerant Scheduling Algorithm for Real-time Periodic Tasks in Heterogeneous Distributed Systems Based on Discrete Logarithmic Multi-signature

Loggesh Sainuni*

Institute of IT & Computer Science, Afghanistan

**corresponding author*

Keywords: Discrete Logarithmic Multi-signature, Heterogeneous Distributed System, RT Periodic Task, Fault-tolerant Scheduling Algorithm

Abstract: Heterogeneous processors have high performance and low energy consumption, so they are widely used in various real-time(RT) systems, such as flight control systems. In a RT system, tasks that fail to meet deadlines can have catastrophic consequences. In order to avoid the situation that the system failure causes the task to miss the deadline, it is necessary to provide the system with fault tolerance. Fault-tolerant scheduling is an effective way to achieve system fault tolerance. There are two types of system faults: transient faults and permanent faults. The purpose of this paper is to study the fault-tolerant scheduling algorithm of RT periodic tasks in heterogeneous distributed systems based on discrete logarithmic multi-signature. This paper studies the transient fault tolerance problem. Existing transient fault-tolerant algorithms only consider fault-tolerance and ignore the RT nature of tasks. Therefore, considering both fault tolerance and time constraints, this paper proposes a transient fault-tolerant scheduling algorithm DB-FTSA based on time deadlines. According to the time deadline, calculate the number of fault-tolerant tasks that can provide fault tolerance, and give priority to fault tolerance of high-priority tasks. Configuration improves system reliability. At the same time, the algorithm fully considers the setup time and task execution time, making the algorithm closer to reality and more accurate. In order to overcome the problem that the traditional dynamic programming algorithm does not support a single type of sub-module, the algorithm uses active and passive sub-modules to achieve fault tolerance to reduce sub-module redundancy. It has been experimentally proven that, therefore, large tasks should be replaced with as many small tasks as possible to reduce the number of processors required.

1. Introduction

With the development of high-speed networks and high-performance computers/stations, various power distribution systems have been widely used in many safety-critical systems, such as aircraft control systems, nuclear power plant control systems, etc. The work in these programs is RT, that is, if the work is not completed by the deadline, then all the work must be completed by the deadline. They often lead to disastrous results. In order to ensure RT service even before the deadline after software/hardware failure, fault-tolerance for RT distribution systems needs to be provided to increase its reliability. High performance for various power distribution systems. In recent years, many scholars have devoted themselves to the research of programming algorithms in distributed systems. Although all these algorithms are programmed with reliability as their goal, they neither support RT operation nor fault tolerance [1-2].

In the research on the fault-tolerant scheduling algorithm of RT periodic tasks in heterogeneous distributed systems based on discrete logarithmic multi-signature, many scholars have studied it and achieved good results. The model is based on a rigorous method that can accurately analyze and predict the characteristics of RT task scheduling [3]. Ghanavati S proved that the RMS algorithm is the best algorithm in a RT task system whose execution time is equal to the deadline. Not only that, but they also linked the system utilization factor to the system efficiency and developed a formula for the minimum system utilization that can be achieved by RMS scheduling. There is a RT task system where task time equals due date. Not only that, but they also linked the system utilization factor to the system efficiency and developed a formula for calculating the minimum system utilization achievable through RMS scheduling [4].

This paper proposes a new discrete logarithmic multi-signature scheme, which improves the important generation part of Meta-ElGamal's multi-signature scheme and Schnorr's multi-signature scheme, and successfully avoids the original idea. If multiple endorsers on the network cooperate to generate each other's keys, future signature-denial attacks may succeed. This paper examines the problem of permanent fault tolerance. Existing permanent fault-tolerant algorithms just blindly back up tasks and ignore the time constraints of tasks. Therefore, this paper proposes a permanent fault-tolerant scheduling algorithm DBSA based on time deadlines. The algorithm quantifies the number of permanent failures that the system can tolerate according to the given time deadline, which avoids the situation of blindly copying tasks and missing the deadline to a certain extent.

2. Research on Fault-Tolerant Scheduling Algorithm of RT Periodic Tasks in Heterogeneous Distributed Systems Based on Discrete Logarithmic Multi-Signature

2.1. Heterogeneous RT Systems

Tasks in these systems often have serious consequences if they fail to execute within a given time constraint. RT systems play an important role in various aspects, such as defense, aerospace, automatic control processing. [5-6].

Hard RT system: In safety-related fields, such as aerospace, military, nuclear industry, etc., the time constraints of the system must be satisfied. If the time limit is exceeded, a major safety accident such as an airplane crash may occur, causing serious damage. Loss of life and property. Therefore, when designing this type of system, it is necessary to ensure that its time constraints and functional requirements are met.

Soft RT system: The requirements for response time are not high. Only when the response time exceeds a certain threshold value, will the correctness of the system be affected. For example, the typical soft RT system "VOD system", even if the data delay causes the picture distortion, it will not have a great impact on the user, and there will be no major life-threatening accidents. This paper studies hard RT systems. Compared with general-purpose computers, RT systems have several important characteristics[7-8]:

Reliability: Reliability is a very important indicator for measuring RT systems. In RT applications, even minor faults during operation may cause unexpected consequences. Therefore, some measures need to be taken to ensure that the application can still operate normally in the event of failure, such as adopting a redundant configuration to make the application fault tolerant.

Predictability: Predictable performance predicts the execution completion time of RT tasks. Since RT systems are time-critical, predictability is also an important performance requirement in RT systems.

Figure 1 shows a schematic diagram of a RT system. The system receives the current state of the system through sensors or receives commands through the man-machine interface, and then executes the corresponding control algorithm to obtain control commands within a certain driving time [9-10].

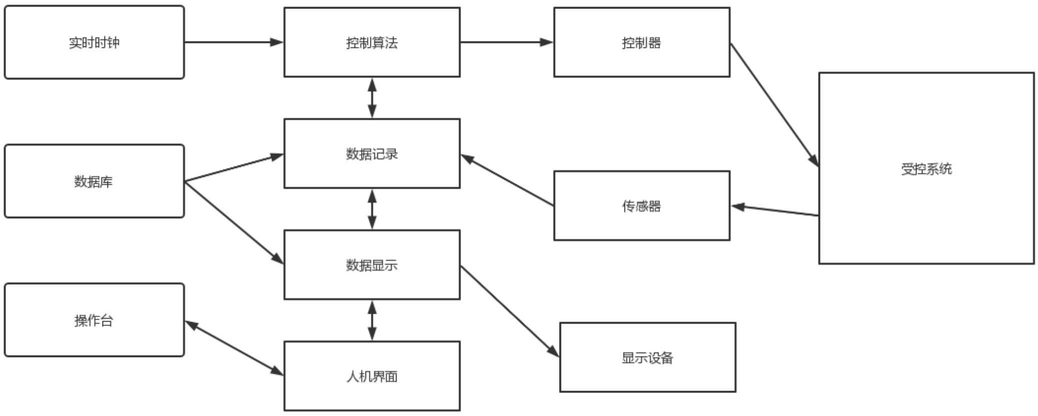

Figure 1. A typical RT system

2.2. RT Scheduling

Static priority scheduling assigns different priorities based on some static property of tasks. After the priority of the task is determined, its priority will not be changed during execution. The dynamic priority algorithm refers to dynamically calculating the priority of the task according to the dynamic attribute of the task during the task running process. Static priority scheduling belongs to offline scheduling, that is, the priority of the task has been determined before it runs. Dynamic priority scheduling is an online scheduling method. When a new task arrives or a task is completed, the priority of the unexecuted task will change [11-12].

2.3. Failure Model

In this paper, it is assumed that the number of failures on each processor pj obeys a Poisson distribution, and the expected number of failures on processor pj in the time interval t is λj. Different processors have different λ values, so the failures in the time interval t The distribution can be represented by formula (1).

$$f\left(k,\lambda_j\right)=\frac{\lambda_j^k e^{-\lambda_j t}}{k!} \tag{1}$$

Among them, k represents the number of failures in the time interval t. The reliability of task vi refers to the probability of successful execution of task vi. Calculated by formula (2) [13-14]:

$$R_i = f\left(0,\lambda_j\right)=e^{-\lambda_j w_{ij}} \tag{2}$$

From this formula, the system reliability for executing N tasks can be calculated[15-16]:

$$R=\prod\nolimits_{i=1}^{N} R_i \tag{3}$$

3. Research and Design Experiment of Fault-Tolerant Scheduling Algorithm for RT Periodic Tasks in Heterogeneous Distributed Systems Based on Discrete Logarithmic Multi-Signature

3.1. Experimental Environment

Simulation experiments are carried out from two aspects. First, DYFARS is compared with similar algorithms in terms of schedulability and reliability. Then analyze the influence of each parameter of the algorithm on the performance of the algorithm. The simulation program is completed with VC6.0, and the hardware platform is Pentium 4 1.7G CPU with 512M SDRAM[17-18].

3.2. Experimental Design

This paper mainly conducts experimental analysis on the fault-tolerant scheduling algorithm in this paper. First, it compares the two fault-tolerant algorithms and analyzes their comparison in task reliability. The second is to study the number of processors required by different task sets, and analyze the optimal task set allocation.

4. Experimental Analysis of Fault-Tolerant Scheduling Algorithm for RT Periodic Tasks in Heterogeneous Distributed Systems Based on Discrete Logarithmic Multi-Signature

4.1. Algorithm Comparison

This paper first compares DYFARS with DRFTBOA in terms of reliability cost. This paper selects various task sets of different sizes for the two algorithms to compare the reliability cost. The experimental data is shown in Table 1.

Table 1. Comparison between DYFARS and DRFTBOA on reliability costs

	20	40	60	80	100
DYFARS	9	12	twenty three	28	31
DRFTBOA	13	twenty four	42	51	56

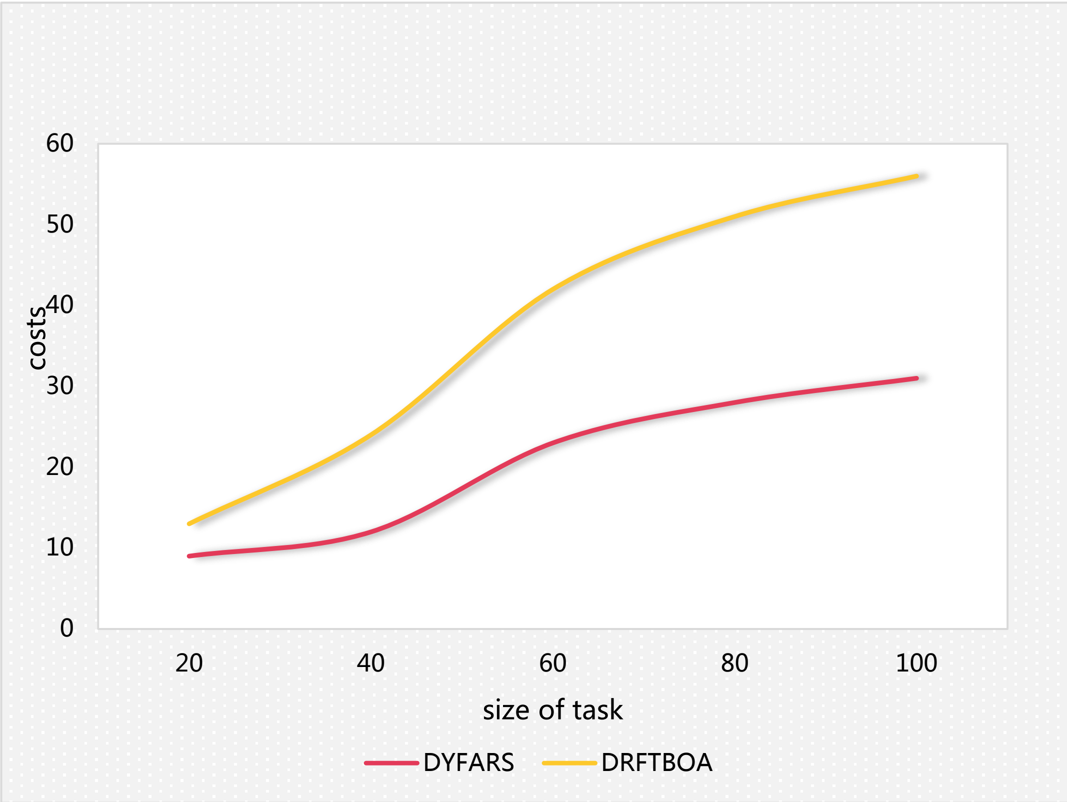

Figure 2. Reliability cost comparison of the two algorithms under different task sets

It can be seen from Figure 2 that under different task sets, the reliability cost of DYFARS algorithm is lower than that of DRFTBOA algorithm, so DYFARS algorithm is better than DRFTBOA algorithm in reliability, so this paper chooses DYFARS algorithm as the algorithm of this system.

4.2. The Relationship between the Number of Processors and the Utilization of Task Resources

In order to study the relationship between the minimum number of processors and the resource utilization rate of tasks in the system of this paper, this paper selects different task sets and the minimum number of processors required under different resource utilization rates. The experimental data is shown in Table 2. shown.

Table 2. The Relationship between the task granularity and the number of processors

	10	11	12	13
0.1-0.2	30	34	38	40
0.05-0.1	twenty four	26	29	30
Minimum number of processors	twenty one	twenty two	twenty three	25

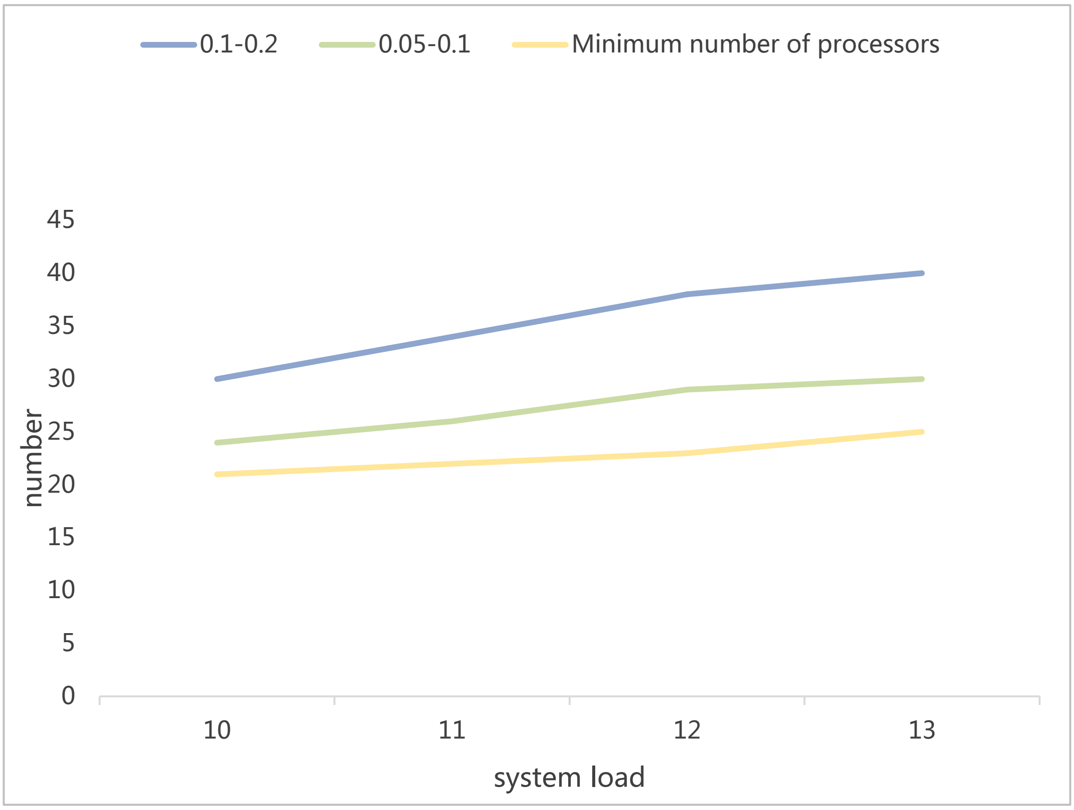

Figure 3. Change in the number of processors under different task granularity

As can be seen from Figure 3, the higher the sensitivity of the task, the more instructions are required. This is because when a processor has many idle resources, it can satisfy tasks using low-performance resources, but it cannot satisfy tasks using high-performance resources, such as a processor at 0.15 of the resource, if the service resource is used and allocated as 0.1, it can be allocated, if the use is 0.2, it is not satisfied and has been written. Therefore, large tasks should be replaced with as many small tasks as possible to reduce the number of processors required.

5. Conclusion

With the rapid development of science and technology, RT systems have been widely used in industrial, commercial and military fields. A key element of a RT system is that the system must be predictable in time, and the system must ensure that each RT task is completed within the deadline. At the same time, because RT systems are widely used in many safety-critical fields, if the RT performance of RT tasks in these systems cannot be guaranteed, it will cause catastrophic consequences. Therefore, it is necessary to provide a certain fault-tolerant capability for the RT system, so as to improve the reliability of the system. At present, RT systems have been widely used in distributed systems, especially heterogeneous distributed systems. In order to improve the schedulability, reliability and task acceptance rate of distributed RT systems, this paper studies RT fault-tolerant scheduling algorithms for homogeneous and heterogeneous distributed systems. In this paper, a transient fault-tolerant scheduling algorithm based on time deadline is proposed by using discrete logarithmic multi-signature algorithm. Existing transient fault-tolerant algorithms only perform fault tolerance on tasks and ignore the RT nature of tasks. Aiming at transient faults in the system, this paper takes the time deadline as the main constraint, calculates the tasks that the system can provide fault tolerance, and then gives a task allocation scheme including backup copies. The experimental data also confirmed the effectiveness of the algorithm. A permanent fault-tolerant scheduling algorithm based on time deadline is proposed. Existing permanent fault-tolerant algorithms blindly back up tasks so that the schedule length exceeds the deadline. Aiming at the permanent failures in the system, this paper quantifies the number of permanent failures that the system can tolerate according to time constraints, and allocates all tasks and their copies. The experimental results show that the algorithm can effectively improve the system reliability on the basis of meeting the time deadline.

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] Roy A, Aydin H, Zhu D. Energy-aware primary/backup scheduling of periodic RT tasks on heterogeneous multicore systems. Sustainable Computing: Informatics and Systems, 2021, 29(2):100474. https://doi.org/10.1016/j.suscom.2020.100474

[2] Kada B, Kalla H. A Fault-Tolerant Scheduling Algorithm Based on Checkpointing and Redundancy for Distributed RT Systems. International journal of distributed systems and technologies, 2019, 10(3):58-75.

[3] Chauhan N K, Tyagi I, Kumar H, et al. Tasks Scheduling Through Hybrid Genetic Algorithm in RT System on Heterogeneous Environment. SN Computer Science, 2022, 3(1):1-13. https://doi.org/10.1007/s42979-021-00959-0

[4] Ghanavati S, Abawajy J, Izadi D. Automata-based Dynamic Fault Tolerant Task Scheduling Approach in Fog Computing. IEEE Transactions on Emerging Topics in Computing, 2020, PP(99):1-1.

[5] Baital K, Chakrabarti A. Dynamic Scheduling of Real-Time Tasks in Heterogeneous Multicore Systems. Embedded Systems Letters, IEEE, 2019, 11(1):29-32. https://doi.org/10.1109/LES.2018.2846666

[6] Bouyahia O, Abdallah A, Yazidi A, et al. Fault Tolerant Fuzzy Logic Control of a 6-Phase Induction Generator for Wind Turbine Energy Production. Electric Power Components and Systems, 2021, 49(8):756-766.

[7] Kumar N, Mayank J, Mondal A. Reliability Aware Energy Optimized Scheduling of Non-Preemptive Periodic Real-Time Tasks on Heterogeneous Multiprocessor System. IEEE Transactions on Parallel and Distributed Systems, 2020, 31(4):871-885.

[8] Sharma R, Nitin N, Alshehri M A R, et al. Priority-based joint EDF–RM scheduling algorithm

for individual real-time task on distributed systems. The Journal of Supercomputing, 2021, 77(1):890-908.

[9] Abohamama A S, El-Ghamry A, Hamouda E. Real-Time Task Scheduling Algorithm for IoT-Based Applications in the Cloud–Fog Environment. Journal of Network and Systems Management, 2022, 30(4):1-35. https://doi.org/10.1007/s10922-022-09664-6

[10] Senapati D, Sarkar A, Karfa C. PRESTO: A Penalty-aware Real-time Scheduler for Task Graphs on Heterogeneous Platforms. IEEE Transactions on Computers, 2021, PP(99):1-1.

[11] Younes A, Salah A B, Farag T, et al. Task Scheduling Algorithm For Heterogeneous Multi Processing Computing Systems. Journal of Theoretical and Applied Information Technology, 2019, 97(12):3477-3487.

[12] Zhuo R, Chen Y, Peng J, et al. A novel spectral-temporal Bayesian unmixing algorithm with spatial prior for Sentinel-2 time series. Remote Sensing Letters, 2022, 13(5):522-532.

[13] Lakhan A, Li X. Mobility and Fault Aware Adaptive Task Offloading in Heterogeneous Mobile Cloud Environments. ICST Transactions on Mobile Communications and Applications, 2019, 5(16):159947. https://doi.org/10.4108/eai.3-9-2019.159947

[14] Behera L. A fault-tolerant time-triggered scheduling algorithm of mixed-criticality systems. Computing, 2022, 104(3):577-599.

[15] Rao K P, Ramamurthy D V, Reddy N S, et al. Integrated simultaneous scheduling of machines, automated guided vehicles and tools in multi machine flexible manufacturing system using symbiotic organisms search algorithm. Journal of Industrial and Production Engineering, 2022, 39(4):317-339.

[16] Ren P, Hong J C. Research on Hybrid Real-Time System Scheduling Algorithm for Multiprocessor Environment. Journal of Physics: Conference Series, 2021, 1732(1):012069 (7pp).

[17] Hervé, Samba Aimé, Aurelien Y T, Leandre N N, et al. Improved Networked Iterative Learning Fault-tolerant Control Algorithm for Systems with Time-delays, Random Packet Losses, Limited Communication and Actuator Failure. International Journal of Control, Automation and Systems, 2022, 20(7):2425-2433. https://doi.org/10.1007/s12555-021-0179-9

[18] Bukhari S S H, Younus M U, Jaffri Z U A, et al. Scientific Workflow Makespan Minimization in Edge Multiple Service Providers Environment. Wireless Personal Communications, 2022, 125(4):3187-3203. https://doi.org/10.1007/s11277-022-09704-z