

## Research Article

# System Reliability Modeling and Analysis of Distributed Networks

**Guanghui Wei** 

*ChongQing College of Electronic Engineering, Chongqing 401331, China*

Correspondence should be addressed to Guanghui Wei; 200730026@cqcet.edu.cn

Received 15 December 2021; Revised 8 January 2022; Accepted 15 January 2022; Published 31 January 2022

Academic Editor: Qiangyi Li

Copyright © 2022 Guanghui Wei. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The reliability of distributed network is researched by establishing one set of effective modeling, analysis, evaluation, and optimization method theory, which is a hot research area in the distributed network system field. The phased analysis method is utilized to study the reliability of the distributed network including the analysis and definition of the distributed network on the basis of studying its operational process. Meanwhile, task request, task scheduling, and execution phase will be also studied, and the establishment of reliability mathematical model of distributed network will be discussed to provide actual reference significance for the design of the distributed network system.

## 1. Introduction

Distributed network is a system based on a common network to provide software services. It has a high degree of cohesion and transparency. The difference between a distributed network and a common network lies in the difference of high-level software (such as an operating system). When a group of independent computers is presented to users, the distributed network provides a unified whole, just like a system. In the distributed network, there are a variety of common physical and logical resources, which can dynamically allocate tasks. The distributed physical and logical resources can exchange information through computer networks. In the distributed network, there is a distributed operating system that manages computer resources in a global way. For users, there is a layer of software middleware on top of the distributed network operating system to implement this model. The World Wide Web is such a familiar distributed network.

Different from the distributed network, the common computer network does not provide a unified model and unified software (i.e., distributed operating system). In the ordinary computer network, what users see is scattered; it is relatively independent with different hardware, different operating systems, and no unified actual machines. For users, these differences are completely possible. If a user

wants to run a program on a remote machine, he must login to the remote machine to run the program.

Distributed network is a loosely coupled system composed of multiple processors interconnected by communication lines. From the point of view of a processor in the system, the other processors and corresponding resources are remote, and only their own resources are local. The distributed network has the following four characteristics:

- (1) A distributed network is composed of multiple computers, which are geographically dispersed and can be distributed in a unit, a city, a country, or even the whole world. The function of the whole network is distributed on each node, so the distributed network has the distribution of data processing.
- (2) Each node in the distributed network contains its own processor and memory, and each has its own function of processing data. Usually, they are equal in status and have no primary and secondary distinction. They can work autonomously and can use the shared communication lines to transmit information and coordinate task processing.
- (3) A large task can be divided into several subtasks, which are executed on different hosts.
- (4) There must be a single and global process communication mechanism in the distributed network

so that any process can communicate with other processes without distinguishing between local communication and remote communication. At the same time, there should also be a global protection mechanism. There is a unified set of system calls on all machines in the network, which must adapt to the distributed network environment. Running the same kernel on all CPUs makes coordination easier.

The distributed network is constructed on the basis of WAN and shared network; its dynamic integration spans the resources of multiple organizational domains, which can provide high efficiency, humongous calculation, and data-handling capacity. However, there exist thousands of resources, services, and applications executed interactively through the distributed network platform in a distributed network environment. On account of high heterogeneity of these entities themselves, there are possible errors in these entities themselves and an interaction between entities [1]. In addition, because machine fault, network disconnection, and process termination resulted by priority use of remote host on local computing resources, the unavailability of computing nodes of the distributed network occurs commonly. Furthermore, the reliability of the distributed network computing system is also influenced. At present, the frequent occurrence of errors has become one of main obstacles that impede the steady development of the distributed network; therefore, the reliability research on the distributed network system appears to be particularly important. Recently, one good development direction of the distributed network is the desktop grid system by using free computer resources. However, desktop grid nodes have instability and great dynamics, there is great unreliability to calculate on these resource nodes [2]. The task scheduling trends to fail and impacts the efficiency of applications.

For users, the reliability of the distributed network reflects the stability of the system to provide fault-free services. The failure of the distributed network is often reflected in some form of system collapse, service failure, or result error. More specifically, fault is the real reason for the system that cannot operate normally, and error is an external manifestation of the system failure; that is, the failure or failure in the system errors essentially originates from a fault in the system. Multiple errors may occur from the same fault, and multiple faults may lead to the same error. In the distributed network, the fault may come from many reasons, such as hardware failure and software failure. Compared with the traditional parallel computing environment, the distributed network is highly heterogeneous, and the failure probability of computing resources is relatively higher. When the distributed network fails, reasonable measures should be taken to deal with the abnormal situation. It is very important to ensure the reliability of the system service. In order to ensure the correct operation of distributed networks, fault management has become an important feature of distributed networks. Common fault management technologies include fault elimination, fault prediction and avoidance, and fault tolerance. Fault elimination is a method to remove software defects in distributed networks through traditional software

testing and verification methods to ensure service reliability. This method is applicable in the early days of system release. Fault prediction and avoidance is to predict the occurrence of faults and take reasonable measures to prevent them in the process of job scheduling. It requires system designers to have the professional knowledge to deal with various types of faults. Fault tolerance is a kind of ability that the system can still correctly perform functions in case of failure. Fault tolerance technology in the distributed network environment is mainly divided into two categories: active fault tolerance is to take measures to restore the normal function of the system after a fault occurs. The common active fault-tolerant technologies include checkpoint technology and active replication technology. Typical passive fault tolerance uses multiple replica voting to avoid errors caused by faults.

Although fault management technology plays an important role in improving the reliability of the distributed network and ensuring the performance of services, the use of fault management technology will inevitably increase the power consumption of a data center, bring additional operating costs to distributed network service providers, and reduce the return on investment. For example, in 2020, Google's annual power consumption is 140 billion kwh, and the annual power consumption of Microsoft Data Center is  $6 \times 10^8$  kwh, which needs to pay 36 million US dollars [3]. According to the current 35% growth rate [4], by 2022, the power consumption of data centers in the United States will reach  $2.55 \times 10^{11}$  kwh. At that time, the United States may need to build 21 power stations to meet the power demand of the data center. Thus, reducing the energy consumption of data center has become another challenge for distributed network service providers. Therefore, while ensuring the reliability of the distributed network, how to reduce the energy consumption of data centers is also a problem that must be considered when designing fault management technology.

Because the nodes involving in desktop computing have instability and dynamics, when the scheduling and computation are executed on unstable resources, partial task execution may fail to complete [5]. By the reason of failure of some task computing, the entire application program cannot be executed, and the efficiency can be reduced. Consequently, the research results obtained through the analysis of sensibility of reliability parameters in the distributed network system in this paper can provide reference bases for how to improve system service reliability when designing the distributed network system.

## 2. Relevant Research on the Reliability of Distributed Network

With the development of the global distributed network system, the information providers are increasingly distributed and varied. Under this diverse and resilient environment, reliability becomes a problem that must be considered. On account of a loose coupling characteristic between

distributed network resources, reliability and high performance have become two design objectives of the distributed network application inherently [6]. On the one hand, the scale of the distributed network can increase the resource failure probability rapidly, and the reliability has become one key technology for successful configuration of the distributed network; on the other hand, with the increase of the distributed network scale, the loose coupling between distributed network resources results in performance decline, which cannot support process communication, and synchronous and combined scheduling functions required in scalable distributed network applications effectively. Thus, the performance has become one of key issues of distributed network computing [7]. Therefore, the reliability of the distributed network is one of the major issues to be resolved by distributed network computing. Thereby, only the distributed network that supports reliability engineering can obtain important application and significant application value in the research and commercial field.

Inherently, the distributed network environment is unreliable [8]. According to literature [9], one failure detection service and one flexible error handling framework can be served as the reliability mechanism of the distributed network. In a distributed network environment, numerous resources have a low utilization rate; these unoccupied resources can be used to execute task backups and ensure that at least one task backup can be successfully executed. Hence, there are numerous research studies on reliability task scheduling based on backup. According to literature [10], the concept of trust can be integrated into distributed network resource management, and a trust model that integrates safety impact into the scheduling algorithm is put forward. A credit model is widely applied to the P2P distributed network and distributed secure resource access, namely any resources have certain reputation value, which can reflect reliability standard of resources of which it includes the model of centralized credibility, distributed credibility, and authorization model [11]. According to literature [12], a security-driven task scheduling algorithm based on the distributed network trust model and trust utility function is put forward, while this scheduling algorithm throws away directly for tasks that do not meet trust demand. Under the actual distributed network environment, the security requirements of the task and the reliability rating of the nodes may not match when scheduling events occur. These tasks shall select delay scheduling rather than abandonment directly so as to improve the reliability of the scheduling algorithm [13]. A secure reliability scheduling algorithm proposed on the basis of distributed network in this paper will schedule the tasks that cannot meet the security needs temporarily in the next scheduling until the tasks can be performed successfully. Thus, the success rate of task scheduling can be improved. The reliability research of the distributed network is mainly divided into a user-centered method, architecture-based method, and state-based method. In the state-based approach, various errors are divided into different levels in reference [14], and these layers are mapped to different Markov states. The reliability of the system is predicted by queuing theory, graph theory,

and Bayesian analysis. In reference [15], based on the transmission time model, the execution time of the program in the distributed system is evaluated, and the corresponding Markov state is defined with the time constraint information to calculate the reliability. Dong et al. [16] searched the minimum resource spanning tree by considering the factors of resources, elements, working time, nodes, and links, and used conditional probability to calculate the reliability of the system.

The reliability analysis, modeling, and performance index of application based on the distributed network are challenging because the failure of any resource in the distributed network system may cause the whole application running in this environment to crash. In reference [16], the characteristics of hardware fault are analyzed in detail, and the prediction index of hardware fault is preliminarily analyzed. Wang et al. [17] proposed a trust evaluation model, which can effectively reconfigure various distributed network resources and distribute them to multiple user requests. In reference [18], errors were classified into eight categories, and then they were classified into two phases: service request and service execution.

In this paper, the research on the reliability modeling technique of the distributed network system is of great significance. Firstly, the reliability modeling analysis of the distributed network system in mission profile for performing user request is studied. The fault tolerance features of the distributed network system itself are also considered in this process. Then, the corresponding reliability model is also set up by stages. Finally, algorithm emulation and analysis, and conclusion are conducted for the established model.

### 3. Reliability Definition of Distributed Network

According to the definition of reliability in reliability engineering theory, the reliability of the distributed network can be defined as follows: the distributed network can complete the user request under given conditions and in set time [20]. The basic reliability depends on the performance and reliability of the system structure and all parts, and the task reliability relies on specific task profile [21]. Since the particularity of distributed network system type, the task reliability is mainly discussed in this paper. For convenience of calculation and expression, the distributed network system shall be assumed as follows:

- (1) The request arrival of the user service is random, the time interval of request arrival is an independent identically distributed random variable, and the mean value of probability density function is  $f$ .
- (2) The request arrival distributed network system of the user service is divided into  $m$  ( $r > 1$ ) subtasks later, of which  $m$  is a random variable, its distribution is  $p\{r = i\} = g_i$  ( $i \geq 1$ ), and the mean value is  $E(r) = g$ .
- (3) The scheduling system contains  $M$  ( $M \geq 1$ ) schedulers; each scheduler has different processing capacities and a same structure. The time required by

scheduled tasks is the random variable  $Y$  that obeys index distribution; its mean value is  $E(R) = \mu_r$ .

- (4) The task queue capacity of tasks scheduling system in the distributed network system is  $N$ , and  $N \geq M$ .
- (5) The failure rate of computational nodes and communication link is a positive constant; its failure recovery time is an independent identically distributed random variable.
- (6) Each computational node can deal with a subtask at the same time.

The tasks of the distributed network system refer to providing services including computing, storage, and application, and handle with the user service request within the shortest time. Then, it will submit the results to users [22]. The reliability of the distributed network system is shown in the formula below:

$$P(t) = (1 - p_c^{(n)}) P_r \{T_s^{(n)} < t\} \quad (1)$$

In the above formula,  $R(t)$  means the reliability of the distributed network system, namely the probability of service request submitted by the users can be completed in given time  $t$ ;  $n$  refers to the number of subtasks decomposed when the service request submitted by users is executed actually;  $p_c^{(n)}$  refers to the blocking probability of service request submitted by users;  $T_s^{(n)}$  refers to service response time, which is the time from request submitting of users to final execution.

#### 4. Reliability Modeling of Distributed Network System

In essence, the distributed network integrates parallel computing and virtualization technology as a whole, which is a new calculation mode of the information system that develops and evolves constantly. It is characterized by significant scalability, flexible configuration, acquisition as required, and accountability [23]. Meanwhile, it is more dependent on the network; hence, its reliability involves more factors. I strive to make an abstract description on the reliability of the whole system in this paper and provide a reliability model for the system. However, as for the next level of the system, if it contains the reliability of a single computational node to control or configure software and hardware servers, the detailed exposition shall not be provided, and the existing research results can be adopted directly. The staged modeling of the distributed network system shall be carried out in this paper. First, the user service request arrives at the task scheduling system, that is, the task request stage; the next is the subtask scheduling process of the scheduling system for service requests, which is referred to as the scheduling stage; the final stage is the execution stage from the beginning of the subtask to the completion of the subtask.

**4.1. Task Request Phase.** The service request shall be submitted by users to the task scheduling system (divided into  $m$  subtasks). The task scheduling system ( $S$  schedulers) is responsible for receiving subtasks. According to the above

assumption, this stage is a multiservice queuing system. From the perspective of the system, it will reach a stable state after operating a period of time, namely the sum of the number of subtasks waiting in the task queue and the number of scheduled subtasks in the scheduling system is  $k$ , which can be called as the  $k$  state. The steady-state probability  $P_k$  is shown in the formula below:

$$P_k = \frac{\alpha S}{\min(k, S)g} \sum_{i=0}^{k-1} g_i \sum_{j=k-1}^{n-i} p_j, 1 \leq k \leq N, \quad (2)$$

where  $p_j$  refers to the arrived service request;  $j$  refers to the probability of ( $j=1, 2, 3, \dots, N$ ) tasks in the system,  $\alpha = g/tS\mu_r$ ;  $p_0 = 1 - \sum_{k=1}^N P_k$ ;  $N$  is the task queue capacity.

It is assumed that the service request arrival abides by Poisson distribution  $P_i$ , which is expressed as follows:

$$P_i = \sum_{j=0}^N P_j P_{i,j}, 0 \leq j \leq N, \quad (3)$$

where  $p_{i,j}$  refers to the one-step transition probability.

If the scheduling system queue of the current distributed network system owns enough long space to contain subtask of user service request, all the subtasks enter into the queue. Otherwise, they shall be blocked, resulting in the failure of the whole user service request. Therefore, the blocking probability  $P_B$  of user request in the task scheduling system is shown in the formula below:

$$P_B = \sum_{i=n-j+1}^{\infty} g_i \sum_{j=0}^N P_j. \quad (4)$$

Therefore, the blocking probability  $p_B^{(m)}$  of user service request which has been separated  $m$  subtasks is shown in the formula below:

$$P_B^{(m)} = \sum_{j=0}^{N-m} P_j. \quad (5)$$

**4.2. Dispatching Stage.**  $m$  subtasks enter into scheduling system; if  $m \leq S$  and  $j(0 \leq j \leq S-m)$  subtasks are in waiting queue, then  $m$  subtasks can be disposed by scheduler immediately; otherwise, at least one must be waiting. When  $N(m) = j$  ( $j=0, 1, \dots, N-m$ ) subtasks in the queue, the waiting time of user service request  $m$  is  $T_w^{(m,j)}$ . It means that the time from  $m$  subtask entering into task waiting queue to the last subtask starts scheduling. When  $m \leq S$  and  $S-m < j \leq N-m$ , the last subtask can start scheduling before the previous  $j - (S-m)$  subtasks complete the scheduling; hence,  $T_w^{(m,j)}$  complies with Gamma distribution, whose order is  $j-S+m$  and parameter is  $\mu_r^S$ .

When  $m$  subtasks enter into the scheduling system totally, the scheduling system is the scheduling  $Y$  subtask in  $m$  subtasks, of which  $Y$  is a discrete random variable. Its probability is shown in the formula below:

$$q_k^{(m)} = \sum_{m=k} \left[ \prod_{j=1}^k \left(\frac{j}{S}\right)^{n_j} \cdot \prod_{j=1}^{k-1} \left(1 - \frac{j}{S}\right) \right], 1 \leq k \leq \min(m, S). \quad (6)$$

$T^{(k)}$  is the time of the  $m$  subtask from entering into the scheduling system to all subtasks when user service request

finishes scheduling.  $T_s^{(m,j)}$  means the time from the  $m$  subtask entering into the scheduling system under the condition of  $N(m)=j$  ( $j=0,1,\dots,N-m$ ) to all the subtasks finishing the scheduling completely,  $T_s^{(m,j)} = T_W^{(m,j)} + T^k$ . According to Gamma distribution and  $q_k^{(m)}$  probabilistic nature, the probability density function is shown in the formula below:

$$f_{T_s^{(i,j)}}(t) = \sum_{k=1}^{\min(m,s)} q_k^{(m)} \cdot f_{T_W^{(m)}}(t) \otimes f_{T_k}(t). \quad (7)$$

The task scheduling can be conducted as a certain task scheduling algorithm in the scheduling system. The distribution mode of the subtask on the node can be expressed with the subtask configuration matrix ( $W$ ). As for the element  $W_{ik}$  of  $W$ , if the  $i$  subtask is distributed on  $k$  processing node,  $W_{ik}=1$ ; otherwise,  $W_{ik}=0$ . It is assumed that the processing time of subtask  $i$  under the configuration of task assignment matrix  $W$  is  $T_{SPT}^{(w)}$ , namely, the time from the node  $j$  starts to receive the subtask  $i$  to the completion time of subtask; recording  $t_{i,j}$  is the processing time of the subtask  $i$ , as shown in the formula below:

$$t_{i,j} = \frac{W_{pi}}{PS_j}, \quad (8)$$

where  $W_{pi}$  refers to the workload of subtask  $i$ ;  $PS_j$  denotes the processing speed of node  $j$ .

**4.3. Subtask Execution Stage.** Since the distributed network system often adopts a mass of reliability techniques, the actual processing time of subtask (recorded as  $T_{i,j}$ ) is not always equal to  $t_{i,j}$ ; it may include fault recovery time after node failures. It is assumed that  $TR_j^{(k)}$  is the failure recovery time for calculating the  $k$  times ( $k=1, 2, \dots$ ) of node  $j$ . Each failure recovery time needed by the node is an independent identically distributed random variable, so the total recovery time is needed when calculating node  $j$  within  $(0, t]$  time. The formula is expressed as follows:

$$TR_{(j)}(t) = \sum_{k=1}^{N_j(t)} TR_j^{(k)}, \quad (9)$$

where  $TR_{(j)}(t)$  is a compound Poisson process.  $N_{(j)}(t)$  means that the total number of failures happened within  $[0, t]$  node; its probability is

$$p_r\{N_j(t) = k\} = \frac{(\lambda_j t)^k}{k!} \exp(-\lambda_j t), k = 0, 1, 2, \dots, \quad (10)$$

where  $\lambda_j$  means the failure rate of node  $j$ , which serves to Poisson's distribution.

If  $N_j(t_{i,j}) = k$  ( $k=1, 2, \dots$ ), then  $TR_j(t_{i,j})$  complies with the Gamma distribution, whose order is  $k$  and morphological parameter is  $\mu_j$ . Its probability density is shown in formula (11), and the probability density of  $T_{i,j}$  can be calculated via formula (12):

$$F_{RT_j(T_{i,j})}(t) = \sum_{k=1}^{\infty} \gamma(\lambda_j t_{i,j} k) \frac{z(k, \mu, t)}{(k-1)!}, t \geq 0. \quad (11)$$

$$f_{T_{i,j}}(l) = \frac{1}{k} \sum_{k=1}^{\infty} kr(\lambda_j t_{i,j}, k) r(\mu_{jk_{i,j}}, k), l \geq t_{i,j}. \quad (12)$$

In formula (12), in the processing of computational node on subtask  $j$ , the data exchange may exist in computational nodes. Therefore, the failure and recovery of communication link between nodes must be considered. It is assumed that  $\varphi(i)$  is a collection of a communication link needed for processing subtask  $i$  and  $S_{ik}$  is the communication time needed on a communication link  $k$  for processing subtask  $i$ . The calculation of  $S_{ik}$  is shown in the formula below:

$$S_{i,k} = \frac{c_{ik}}{bw_k}, i = 1, 2, \dots, n, k \in (i), \quad (13)$$

where  $c_{ik}$  means the data size transferred through the link  $k$  during the processing task  $i$ ;  $bw_k$  refers to the communication link  $k$  bandwidth.

Since communication link failure and the failure process, the actual communication time of subtask  $i$  may be not same with the ideal communication time on the communication link  $k$ . It is assumed that  $S_{ik}$  is the actual communication time of subtask  $i$  on communication link  $k$ , which is similar to the computing method of scheduling time, as shown in the formula below:

$$S_{ik} = s_{ik} + SR_k(s_{ik}), \quad (14)$$

$$X_k(s_{ik}) = 1, 2, \dots, n, k \in (i),$$

where  $X_k(t)$  refers to the failure rate of the communication link within the time of  $(0, t)$ ;  $SR_k(S_{ik})$  refers to the total recovery time needed by the communication link  $k$  within the time of  $(0, t)$ ;  $f_{s_{ik}}(l)$  means the probability density function of the actual communication time  $S_{ik}$ , as shown in the formula below:

$$f_{s_{ik}}(l) = \frac{1}{\varepsilon_{ik}} \sum_{n=1}^{\infty} nt(r_k s_{ik}, n) t(\varphi_k \varepsilon_{ik}, n), l \geq s_{ik}, \quad (15)$$

where  $\varphi_k$  refers to the recovery rate of the communication link,  $\varepsilon_{ik} = 1 - s_{ik}$ .

The mean value of the actual communication time  $S_{ik}$  is shown in the formula below:

$$A[S_{ik}] = \frac{(\varphi_k + r_k) s_{ik} - z(r_k s_{ik}, l)}{\varphi_k}, \quad (16)$$

where  $r_k$  means the failure rate of the communication link; the occurrence of the failure rate of communication link obeys Poisson's distribution.

The processing time  $T_{SPT}^{(w)}$  of the subtask  $i$  is the sum between actual processing time  $T_i^{(w)}$  and actual communication time  $S_i$ :  $T_{SPT}^{(w)} = T_i^{(w)} + S_i$ . Its probability density function is shown in the formula below:

TABLE 1: Node parameter calculation table.

Subtask	Failure efficiency of node (s-1)	Recovery rate of node (s-1)	Theoretical execution time (s)
a	$\lambda_3 = 0.05$	$\eta_3 = 0.06$	t13 = 200
b	$\lambda_6 = 0.08$	$\eta_6 = 0.08$	t36 = 180
c	$\lambda_9 = 0.08$	$\eta_8 = 0.07$	t39 = 300

$$f_{T_{ST}^{i(w)}}(t) = f_{T_i^{(w)}}(t) \otimes f_{s_i}(t). \quad (17)$$

When the completion time of user service request including  $n$  subtasks is recorded as  $T_{ST}^{(n)}$ , namely the time from beginning of processing the first task of computational node to the completion of all subtasks. During the execution process, the execution of subtasks cannot be independent totally; it may exist the constraint on execution sequence or data communication exchange, which needs to consider the completion time  $T_{ST}^{(n)}$  of subtasks and the constrained relationship between subtasks.

For instance, taking  $n=3$  (namely three subtasks) user service request as an example, subtask  $b$  needs regarding output data of subtask  $a$  as input data. After the execution of subtask  $b$  is completed, it can be input with subtask  $c$ , and the task is completed finally. Under this circumstance, the completion time of subtask is  $T_{ST}^{(n)} = \max(T_{ST}^a, T_{ST}^b, T_{ST}^c)$ , of which the probability density function of  $T_{ST}^{(n)}$  is

$$f_{T_{ST}^{(n)}}(t) = f_{T_{ST}^a}(t) \otimes f_{T_{ST}^b}(t) + f_{T_{ST}^c}(t) \cdot \int_0^1 f_{T_{ST}^a}(t) \otimes f_{T_{ST}^b}(x) dx. \quad (18)$$

In summary, the response time of the entire system on user service request is  $T_{ST}^{(m)} = T_S^{(m)} + T_{ST}^{(m)}$ ; its probability density is  $f_{T^{(m)}}(t) = f_{T_S^{(m)}}(t) \otimes f_{T_{ST}^{(m)}}(t)$ , and finally, the reliability of the distributed network system can be obtained as shown in the formula below:

$$R(t) = \left(1 - P_B^{(m)} \Pr\{T_{ST}^{(m)} < t\}\right) = \left(1 - P_B^{(m)}\right) \int_0^1 f_{T_{ST}^{(m)}}(x) dx. \quad (19)$$

## 5. Algorithm Simulation and Analysis

To analyze and evaluate the proposed evaluation method of the reliability model machine of distributed network system, this paper takes user service request of three subtasks (namely  $n=3$ ) as an example to simulate. It is assumed that subtask  $b$  needs to regard the output data of subtask  $a$  as input; after the execution of subtask  $b$  is finished, it can be input together with subtask  $c$ , and the tasks are completed finally.

User service request arrival is a Poisson process. In  $\lambda_a = 1 (s^{-1})$ , it is assumed that the entire distributed network has three schedulers, namely  $S=3$ , and the service rate of each scheduler is  $\eta_r = 0.5 (s^{-1})$ . For the division of subtask of user service request, the distributed network system obeys uniform distribution, namely  $g_i = 0.1 (i=1,2,3)$ . The failure rate and recovery rate of the computational node and communication link are generated by adopting random

sampling and obeying the uniform distribution of [0.001, 0.1]. The summary of specific parameters in simulation is shown in Tables 1 and 2.

When the user service request arrival follows a Poisson distribution, and the division way for subtask is determined in the system, for instance, the division of the distributed network system for user service request obeys uniform distribution, and the blocking probability of service request becomes a certain quantity. Under this circumstance, the service dependability of the distributed network system is directly related to the service response time. Meanwhile, the service response time of the distributed network system is influenced by the sojourn time of user service request and the completion time of subtask. The service request dwell time is directly related to the number of schedulers in the distributed network system. The completion time of subtasks is affected by the failure rate and recovery rate of computing nodes in the distributed network system. Therefore, the analysis of the probability function distribution curve of the change of service request residence time to the number of modulators and the probability distribution curve of service response time to the change of calculation node failure rate can be used as an aspect of evaluating the service reliability of the distributed network system and provide a reference for the design of the distributed network system.

The probability density function of the sojourn time of user service  $T_S^{(3)}$  is shown in Figure 1 ( $S=3$  curve). The probability density function of service response time  $T_{ST}^{(3)}$  is shown in Figure 2 (failure rate is 0.08 curve).

As shown in Figures 1 and 2, the mean value of  $T_S^{(3)}$  is 6.82, and the mean value of  $T_{ST}^{(3)}$  is 1339.29.

Figures 1 and 2 shows the influence of system parameter of the distributed network system on the property of distributed network system, namely the impact of sojourn time and response time of user service request. Thus, it reflects that the reliability of the distributed network system is sensitive to the parameters. These results can provide a reference for designing the distributed network system. It can be seen from Figure 1 that the probability density function of user service request has morphologic change with the change of the number of scheduler, namely the number of scheduler is in inverse proportion to the mean sojourn time of user service request. When  $S=3, 2, 1$ , the mean value of  $T_S^{(3)}$  is 9.63, 6.27, and 5.01. Similarly, the changing curve of response time for the distributed network system when the failure rate of computational node 4 changes is shown in Figure 2. It can be seen from Figure 2 that the failure rate of computational node 4 is in direct proportion to the response time of the distributed network system, namely, the reliability of service also decreases. When  $\lambda = 0.03, 0.06, 0.12$ , the mean value of  $T_{ST}^{(3)}$  is 1276, 1365, and 1503, respectively.

TABLE 2: Communication link parameters.

Subtask	Failure efficiency of node ( $s-1$ )	Recovery rate of node ( $s-1$ )	Theoretical execution time (s)
a	$\lambda_3 = 0.03$	$\eta_3 = 0.05$	t13 = 270
b	$\lambda_4 = 0.06$	$\eta_4 = 0.07$	t36 = 280
c	$\lambda_2 = 0.12$	$\eta_2 = 0.09$	t38 = 300
	$\lambda_5 = 0.2$	$\eta_5 = 0.06$	t38 = 500

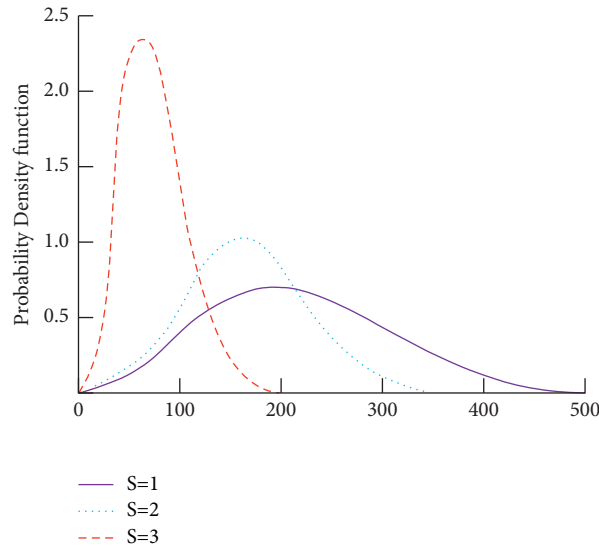


FIGURE 1: Probability density curve of service request sojourn time.

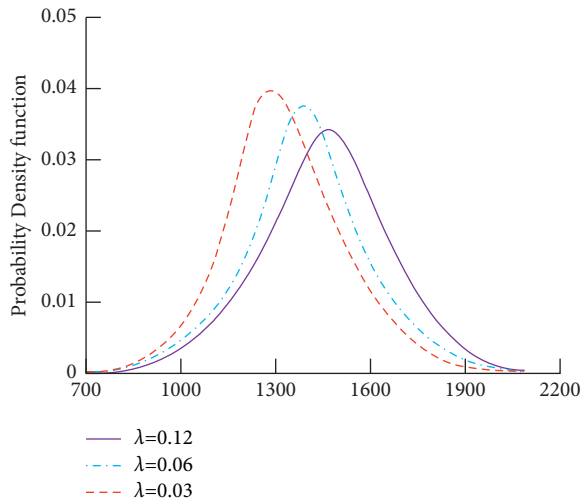


FIGURE 2: Probability density curve of service response time.

## 6. Conclusion

The issue on the reliability of distributed network system is studied in this paper. The concept, modeling, and the appraisal procedure of the reliability of distributed network system are put forward. Finally, the simulated analysis of the last concrete example is carried out, indicating that the model of reliability of distributed network system proposed in this paper is valid. The model and evaluation method put forward in this chapter can provide practical reference

significance for the design of distributed network system. While ensuring the reliability of the distributed network, how to reduce the energy consumption of data center is also an issue that must be considered when designing fault management technology, which will be the direction of follow-up research together with cloud computing scheduling. [19].

## Data Availability

The labeled dataset used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The author declares no conflicts of interest.

## Acknowledgments

This work was partially supported by the scientific research planning project of the Fifth Council of the Chinese Society For Technical and Vocational Education, “Research on the construction of double qualified teachers under the “1 + X” certificate system” (No. 2020B0162).

## References

- [1] D. Shi, “Duplex is adopted to improve the reliability of the system in the distributed network,” *Computing Technology and Automation*, vol. 18, no. 3, pp. 148–151, 1999.

- [2] K. Chen and Bo Liu, "Reliability evaluation model for grid node using fuzzy logic," *Ji Suan Ji Yu Xian Dai Hua*, vol. 217, no. 9, pp. 27–34, 2019.
- [3] Q. Asfandyar, *Powere-Demand Fouting in Massive Geo-distributed systems*, MIT Press, Cambridge, MA, 2010.
- [4] P. Andres, *The Sony Pictures Hack, explained* [EB/OL]-12-18)[2019-04-01], 2014.
- [5] W. Tian, "Reliability and operation model analysis of distributed computer network structure," *Computer Knowledge and Technology*, vol. 10, no. 35, pp. 8390-8391, 2014.
- [6] W. Jing, Z. Wu, and H. Liu, "Multiole DAGs dynamic workflow reliability scheduling algorithm in a cloud computing system," *Journal of Xidian University*, vol. 43, no. 2, pp. 83–89, 2016.
- [7] Y. Zheng, "Problems and challenges of software reliabilities on distributed systems," *Computer and Information Technology*, vol. 22, no. 6, pp. 29–32, 2014.
- [8] C. Yang, J. Xiong, and L. I. Dong, "Research on reliability model for distributed system," *Computer Engineering*, vol. 38, no. 3, pp. 51–53, 2017.
- [9] F. Yang and Y. Shi, "Research of evaluation algorithm for operational reliability of power grid based on distributed computing," *Industry and Mine Automation*, vol. 40, no. 8, pp. 22–25, 2014.
- [10] J. Feng, X. Tang, and W. Wang, "Reliability verification method of numerical simulation based on grid independence and time independence," *Journal of Shihezi University:Natural Science*, vol. 35, no. 1, pp. 52–56, 2017.
- [11] C. Yan, Z. Hu, and Li Xi, "Reliability-cost optimization scheduling model and algorithm in grid," *Computer Science*, vol. 40, no. 3, pp. 136–150, 2013.
- [12] C. Yan, Z. Hu, and L Xi, "Reliable scheduling algorithm for time-constrained grid workflow," *Journal of Chinese Computer Systems*, vol. 33, no. 7, pp. 1528–1532, 2012.
- [13] H. Zhu, L Ting, and J. Yan, "Research on a distributed task scheduling algorithm based on dynamic load balancing," *High Tech Communication*, vol. 24, no. 12, pp. 1261–1269, 2014.
- [14] D. Dong, *Study on Prediction and Early Warning of Gas Concentration Based on Measured Data of Safety Monitoring System*, Xi'an University of science and technology, Xi'an, China, 2012.
- [15] D. Dong, S. Qu, and H. Wang, "Gas concentration early warning method based on correlation analysis of monitoring data," *Industry and Mine Automation*, vol. 41, no. 6, pp. 1–4, 2015.
- [16] D. Dong, S. Qu, and H. Wang, "Characteristics analysis and pretreatment of mine gas monitoring data," *Industry and Mine Automation*, vol. 41, no. 9, pp. 1–4, 2015.
- [17] A. Wang, G. Zhang, and F. Liu, "Research and application of EM algorithm," *Computer Technology and Development*, vol. 19, no. 9, pp. 108–110, 2009.
- [18] H. Zhang, *EM Algorithm and its Application*, Shandong University, Jinan, 2014.
- [19] Z. Ru and anli Gao, "Application of EM algorithm in parameter estimation of incomplete data," *Journal of Nanjing Institute of Technology Natural Science Edition*, vol. 6, no. 4, pp. 9–12, 2008.
- [20] W. Deng, "Improved reliability dynamic level for grid scheduling algorithm," *Computer & Digital Engineering*, vol. 42, no. 6, pp. 927–929, 2014.
- [21] Y. Zhang, Y. Liu, and Y. Chen, "Research on grid computing and its reliability," *Journal of Mudanjiang University*, vol. 19, no. 2, pp. 102–106, 2018.
- [22] M. Qiu and X. Gui, "Study of a fault-tolerant grid framework for dependable grid computing," *Microelectronics & Computer*, vol. 22, no. 7, pp. 99–106, 2005.
- [23] S. Bawa, "Modified Min-Min Heuristic for Job Scheduling Based on Qos in Grid Environment," in *Proceedings of the 2013 2nd International Information Management in the Knowledge Economy International Conference*, pp. 166–171, IEEE, Chandigarh, India, December 2013.