Scholar Publishing Group
Distributed Processing System
https://doi.org/10.38007/DPS.2022.030408
ISSN 2790-0916 Vol. 3, Issue 4: 61-69

# *Dynamic Distributed Based on PLC Software Redundancy*

**Asiane Ullah***

*Uni de Moncton, Canada*

**corresponding author*

***Keywords:*** PCL Software Redundancy, Dynamic Distribution, Resource Management, Distributed System

***Abstract:*** At present, with the development of technology, dynamic distributed system (DDS) have been widely developed in many fields, and the maintenance requirements of the system have also undergone great changes. There are still many places worth discussing about the redundant fault-tolerant mechanism for data loss., so this paper studies the DDS based on PCL soft redundancy. The research on DDS in this paper can be divided into three parts. First, it introduces the architecture of distributed system(DS), dynamic monitoring resources, and an overview of redundant fault-tolerant mechanisms, followed by the collection nodes and resource management of DS. Design, and finally analyze the node repair collection degree and download time.

## 1. Introduction

With the rapid development of mobile Internet technology and the widespread application of various mobile devices, data is growing rapidly. How to store these massive data is one of the challenges we face today [1]. With its own advantages, the DDS has gradually developed into the preferred storage system for various types of data storage today [2]. When the dynamic distributed storage system faces the loss of disk data, it is necessary to introduce a redundant fault-tolerant mechanism to ensure that the system can work normally [3]. In order to provide users with reliable file storage services, dynamic distributed storage systems need to adopt fault-tolerant technology to improve file availability [4-5].

At present, many experts and scholars have discussed and studied DDS and redundancy strategies, and have achieved good results. For example, researchers such as Aranda LA integrated the control of transport network with distributed edge and cloud resources to deploy dynamic and efficient IoT services in order to efficiently allocate IoT analytics and use network resources [6]. Scholars such as Omar H believe that distribution feeder reconfiguration is a mixed integer nonlinear programming problem and is difficult to solve. It is necessary to use an appropriate optimization algorithm to converge to the global optimal value or find a value close to the global optimal value. are configured to extend the problem to multiple time intervals and various objective functions [7]. Researchers such as Kalra C proposed the approximate redundant fault tolerance technique, which uses approximate computing. By evaluating this technique, it was found that different applications with different approximate methods have large differences in execution time, memory footprint, and error detection ability [8]. At present, there are many researches on DDS and good research results have been achieved, but there are few researches on redundant and fault-tolerant mechanisms of DDS. Therefore, it is necessary to strengthen the research on redundant mechanisms in DDS.

This paper designs and analyzes the DDS based on PCL soft redundancy. The structure of this paper can be divided into three parts. The first part is an introduction to the relevant theoretical knowledge. The working structure of the DS can be divided into resource allocation management and resource management visualization; the second part is the design of the DDS, and the data processing is realized by designing the collection nodes, etc. and resource management functions; the third part is to analyze the system implementation.

## 2. Related Overview

### 2.1. DS Architecture

As a resource system, the management of resources is essential, which requires the resource allocation system to have certain fault tolerance, scalability and maintainability [9-10]. Lines of business access the resource allocation server and apply for resources to the system through the server [11-12]. As the main node of the resource management system, the resource server mainly completes two parts of work.

(1) Allocation and management of resources

The allocation and management of resources is developed using a distributed master-slave structure. The resource server acts as the master node and is responsible for the registration, downtime, and heartbeat detection of all slave nodes [13]. The servers in the computer rooms are mainly responsible for providing specific services for the business lines. After the business line applies for resources, the service is deployed to the server cluster to provide external services [14].

(2) Resource management visualization work

The visualization of resource management is developed using the traditional MVC architecture. Since there are currently fewer resources, and the business line is relatively fixed, there is no large number of access requirements. Therefore, the overall MVC service is deployed in the same resource allocation server, which is responsible for the logic processing and visualization of the business. work [15].

### 2.2. Dynamic Monitoring of Resources

The monitoring target of DDS is constantly changing. In order to adapt to this change, this paper involves the configuration description of monitoring target. It is completely possible to dynamically add new monitoring resources based on XML configuration. Through the description of the monitoring target resource and the analysis of the running state, it can be further abstracted. After the abstraction, it mainly includes elements: resources, indicators, states, thresholds, events and other elements [16]. The element event defines the action that needs to be triggered when the resource switches from one state to another state, that is, the event plug-in corresponding to the action [17]. The dynamic monitoring object elements are shown in Figure 1.

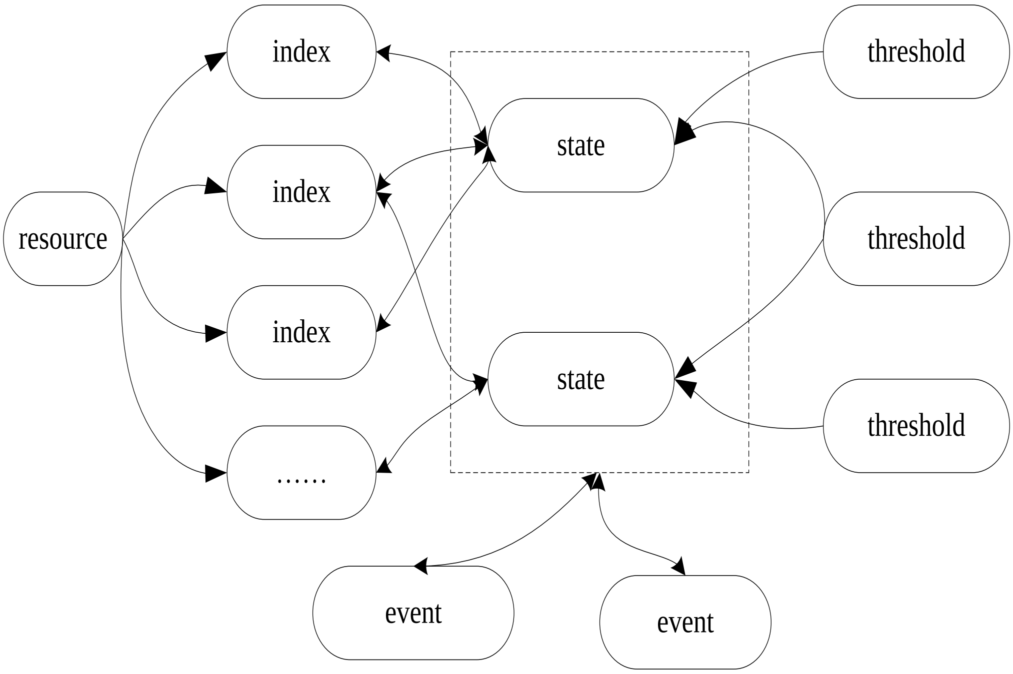


*Figure 1. Monitoring object element diagram*

### 2.3. Redundant Fault Tolerance Mechanism

Before using the distributed file system for file storage, the user blocks the files on the client side, and then distributes the block files to different storage nodes [18]. In order to ensure data availability and improve the fault tolerance of the system, the distributed file system performs redundant backup for file data by designing a redundant storage algorithm. When data is unavailable, the system uses redundant backup files to restore data. The redundancy mechanism of distributed file storage system can be divided into structural fault tolerance mechanism and data fault tolerance mechanism. Compared with the continuous data block method, the discrete data block method has lower network overhead and computational overhead, and can improve the data reading efficiency.

## 3. System Design

### 3.1. Collection Node Design

The indicator collection node undertakes the collection of the operating status information of the monitoring target, provides the system with the most basic and important monitoring data, and sends the collected data to the data aggregation processing through the agreed interface. The collection node must belong to a summary processing node. The collection node enables different operation indicator plug-ins according to the monitoring target type, so as to ensure the scalability of collection. The working mechanism of the collection node is shown in Figure 2.

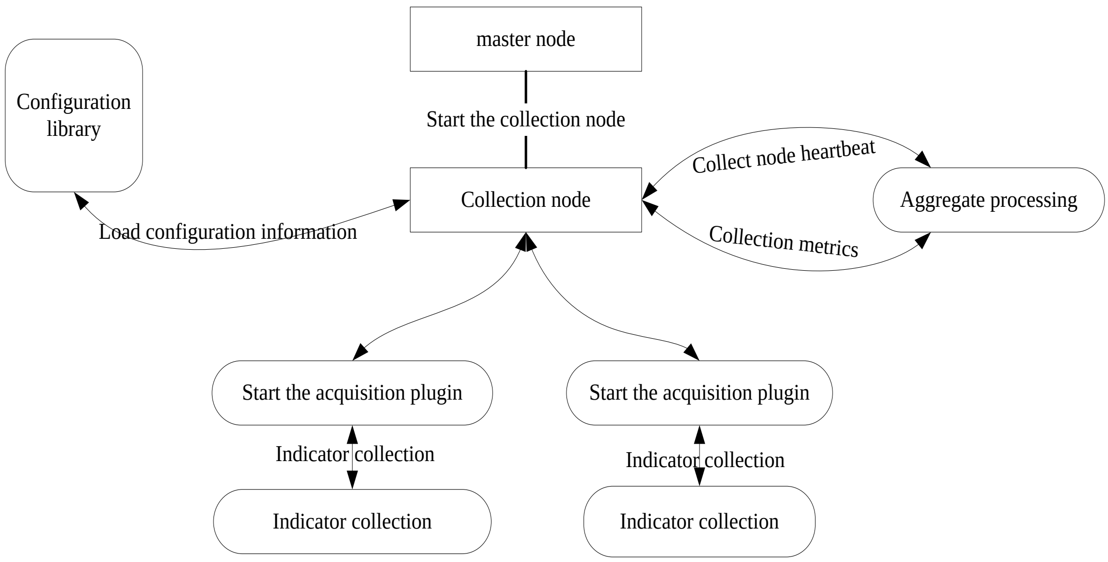


*Figure 2. Acquisition node flow chart*

The procedure of the collection node is to start the summary processing node first, and then start the collection node to obtain the type and configuration information of the monitoring target. Then the collection node loads the corresponding plug-in, and the plug-in collects the running indicators according to the monitoring list of the collection node. The collection plug-in returns the collected plug-in to the collection node, and the collection node sends it to the data aggregation processing unit through the agreed interface.

## 3.2. Controller Design

Considering that there are N dynamic monitoring resource event elements in a DDS, the equation of event plug-in i is formula (1):

$$d_i(r) = Sd_i(r) + Kz_i(r) + Gq_i(r), \quad i = 1,\dots,N \tag{1}$$

In formula (1),$d_i(r) \in R^n$ is the state of the event, and $z_i(r) \in R^m$ is the control input to be designed. S, K and G are system matrices with appropriate dimensions, $q_i(r) \in R^s$ represents external disturbance, and its equation is as follows:

$$\dot{q}_i(r) = Aq_i(r) \tag{2}$$

where $A \in R^{S*S}$ is a known constant matrix.

Through the design of DDS resource controller, the problem of suppressing disturbed dynamic resources in DS is solved, and a new dynamic event triggering control scheme is completed at the same time. Compared with the existing triggering mechanism, the number of event triggering is reduced.

## 3.3. Resource Management Design

The resource management module is the core module of the system, mainly composed of four sub-modules: resource application module, IP pool management module, equipment procurement pool management module and resource operation module. The overall structure is shown in Figure 3.

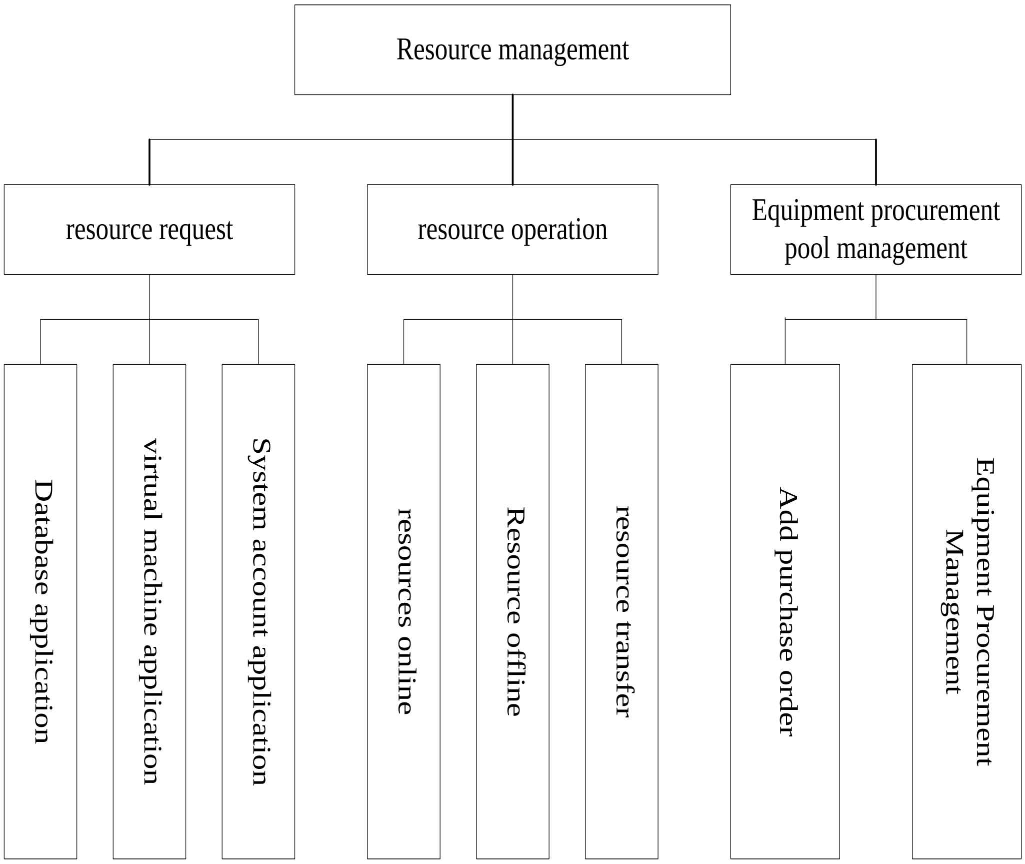


*Figure 3. Overall structure of resource management*

Resource application, as the core part of resource management, completes the related work of various resource applications. The input data of this module is the application information filled in by the line-of-business users on the page, and the output data is the content of the line-of-business work order. The whole process involves three roles, namely the line of business, DBA, and DBA supervisor, of which the line of business is responsible for initiating application resources, DBA is responsible for the preliminary review of the application, and decides to pass the application, reject or submit the application to the superior according to the DBA's own authority, resource characteristics and application type. The DBA is responsible for high-level resource applications, mainly responsible for processing the applications submitted by the DBA, and decides to approve or reject the application according to the characteristics of the resource.

## 4. System Implementation

### 4.1. Node Repair Selectivity

The node repair selectivity refers to the number of repair sets when a node fails, that is, the number of repair schemes. The number of node repair sets can affect the fault tolerance of the system. If a data storage node is lost, the number of repair sets of the FR code is mainly related to the repetition degree s and the node storage size g. If the repetition degree of the constructed FR code is s=2, then there is only one accurate repair scheme when the node fails. If the repetition degree s of the constructed FR code is greater than 2, there are multiple options for repairing the faulty node, and they can be repaired from other s-1 surviving nodes. When the storage size of the FR is g, there are (s- 1) g repair sets. FIG. 4 is a graph showing the relationship between the number of repair sets and the repetition degree of the FR code, where g is equal to 3. It can be seen from Figure 4 that the repair selectivity increases with the increase of data block repetition.

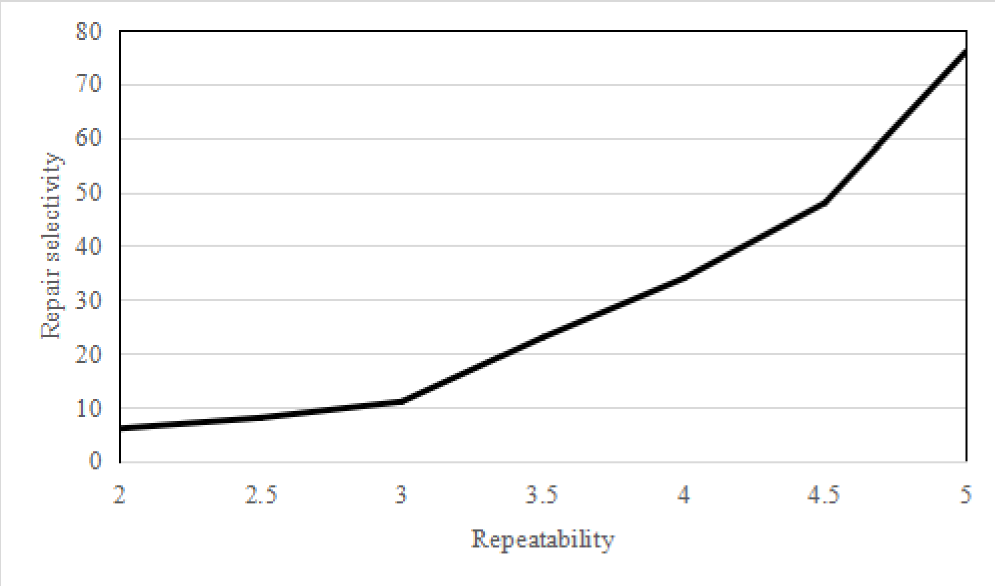


*Figure 4. The relationship between repair selectivity and repeatability*

## 4.2. Download Time Analysis

*Table 1. Average timeline for different phases*

| | Continuous data chunking method | Discrete Data Blocking Methods |
|---|---|---|
| Data upload | 32s | 35s |
| Data download | 96s | 99s |
| Data recovery | 52s | 54s |

Table 1 shows the timetables spent on data uploading, data downloading and recovery for the continuous data block method and the data discrete data block method at different stages. The overall performance of the discrete data block method and the continuous data block method are compared. The experiment tests the speed of the two methods in data upload, download and recovery. Among them, data recovery refers to the efficiency measured when one data block fails. As can be seen from Table 1, in the case of no data block failure, since the discrete data block method can avoid the jump of the magnetic head during data reading and writing, and improve the throughput rate, the data of the discrete data block method The write rate and read rate are slightly higher than the continuous block method. In the case of data block failure, the read speed of the discrete data block method is still higher than that of the continuous block method. Therefore, regardless of whether there are invalid data blocks, the efficiency of discrete data block method is higher than that of continuous block method.

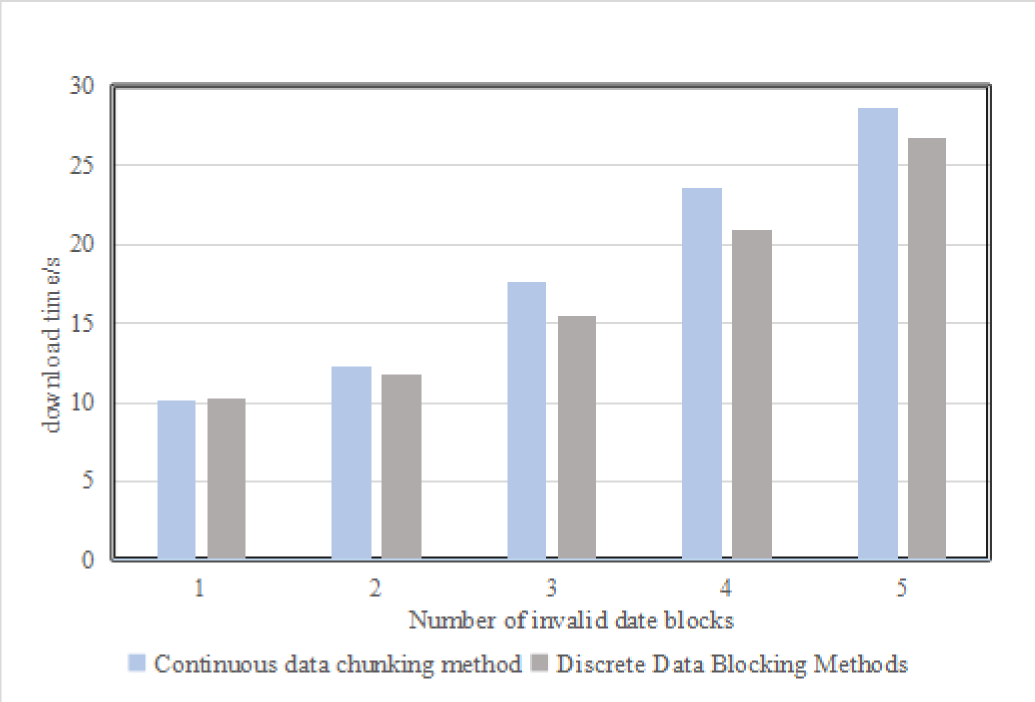


*Figure 5. Download time comparison chart*

Figure 5 shows the change trend of data download time with the increase of the number of failed data blocks for different block methods. When the read file size is 1GB and the number of invalid data blocks changes from 0 to 4, the download time of the continuous block method is 10.12s, 12.35s, 17.58s, 23.56s, 28.63s. The data download time of the discrete data block method is 10.23s, 11.78s, 15.48s, 20.93s, 26.78s, respectively. It can be seen that with the increase of the number of invalid data blocks, the download speed of the discrete data block and continuous block methods both decrease, and the speed advantage of the discrete data block method is more and more obvious. For the same number of invalid data blocks, the download efficiency of discrete data block method is higher than that of continuous block method. This is because with the increase of the number of invalid data blocks, the amount of data that needs to be downloaded by the continuous data block method increases, resulting in a more significant rate drop. However, in the discrete data chunking method, the amount of data downloaded is basically equal to the amount of data requested, and the performance is degraded because the number of decoding increases.

## 5. Conclusion

This paper studies the DDS based on PCL soft redundancy technology. This paper first introduces the relevant overview, then the system design part, and finally the system implementation part. In the system implementation part, the repair selection is found by analyzing the node repair selectivity. The degree increases with the increase of the number of repair sets. By analyzing the data download time, it is found that the discrete data block method is faster than the continuous data block data download time, so the efficiency of the discrete data block method is higher than that of the continuous block method.. There are many deficiencies in the research of

DDS in this paper, which need to be improved, but it is worth to study the DDS based on PCL soft redundancy.

## 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] Alghareb F S, Ashraf R A, Demara R F. Designing and Evaluating Redundancy-Based Soft-Error Masking on a Continuum of Energy versus Robustness. IEEE Transactions on Sustainable Computing, 2018, 3(3):139-152.*

*[2] Benites L, Benevenuti F, Oliveira A, et al. Reliability Calculation With Respect to Functional Failures Induced by Radiation in TMR Arm Cortex-M0 Soft-Core Embedded Into SRAM-Based FPGA. IEEE Transactions on Nuclear Science, 2019, 66(7):1433-1440.*

*[3] Oz I, Arslan S. A Survey on Multithreading Alternatives for Soft Error Fault Tolerance. ACM Computing Surveys (CSUR), 2019, 52(2):1-38.*

*[4] Kang X, Ma R. Power Supply Redundancy Design of Aircraft's Electric Braking Electro-Mechanical Actuation System. Xibei Gongye Daxue Xuebao/Journal of Northwestern Polytechnical University, 2018, 36(1):110-116.*

*[5] Kim D, Kwon J, Han S, et al. Deep Full-Body Motion Network for a Soft Wearable Motion Sensing Suit. IEEE/ASME Transactions on Mechatronics, 2019, 24(1):56-66.*

*[6] Aranda L A, A Sánchez-Macián, Maestro J A. An Algorithmic-Based Fault Detection Technique for the 1-D Discrete Cosine Transform. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2020, 28(5):1336-1340.*

*[7] Omar H, Khan O. PRISM: Strong Hardware Isolation-based Soft-Error Resilient Multicore Architecture with High Performance and Availability at Low Hardware Overheads. ACM Transactions on Architecture and Code Optimization, 2021, 18(3):1-25.*

*[8] Kalra C, Previlon F, Rubin N, et al. ArmorAll: Compiler-based Resilience Targeting GPU Applications. ACM Transactions on Architecture and Code Optimization, 2020, 17(2):1-24.*

*[9] Hussain T, Muhammad K, Ullah A, et al. Cloud-Assisted Multiview Video Summarization Using CNN and Bidirectional LSTM. IEEE Transactions on Industrial Informatics, 2020, 16(1):77-86.*

*[10] Asem P. Base resistance of drilled shafts in soft rock using in situ load tests: A limit state approach. Soils and Foundations, 2019, 59( 6):1639-1658.*

*[11] Nishiyama H, Takamuku S, Iiyama A, et al. Dynamic Distribution of Chemical States of Water inside a Nafion Membrane in a Running Fuel Cell Monitored by Operando Time-Resolved CARS Spectroscopy. The Journal of Physical Chemistry C, 2020, 124(36):19508-19513.*

*[12] Kolodyazhny S Y, Nekrasov A I. Structural and dynamic distribution patterns of oil fields in*

*the central part of the Volga-Ural anteclise. Geodynamics & Tectonophysics, 2020, 11(1):123-140.*

*[13] Jiang M, Sun J, Obadi M, et al. Effects of ultrasonic vacuum drying on the drying kinetics, dynamic moisture distribution, and microstructure of honey drying process:. Food Science and Technology International, 2021, 27(5):426-440.*

*[14] Alonso-Mora J, Samaranayake S, Wallar A, et al. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proc Natl Acad Sci U S A, 2018, 114(3):462-467.*

*[15] Peng J, Edgar T F, Eldridge R B. Dynamic Process Intensification of Binary Distillation via Periodic Operation. 2018, 58(12):2671-2680.*

*[16] Ditmarsch H V, Ruan J, Verbrugge R. Sum and Product in Dynamic Epistemic Logic. Journal of Logic & Computation, 2018, 18(4):563-588.*

*[17] Chung H M, B.A. Loomis, Smith D L. Swelling and structure of vanadium-base alloys irradiated in the dynamic helium charging experiment. Journal of Nuclear Materials, 2019, 233(1):466-475.*

*[18] Causse M, Dalguer L A, Mai P M. Variability of dynamic source parameters inferred from kinematic models of past earthquakes. Geophysical Journal International, 2018(3):1837-1837.*