Scholar Publishing Group
Distributed Processing System
https://doi.org/10.38007/DPS.2020.010404
ISSN 2790-0916 Vol. 1, Issue 4: 25-32

# *Distributed System Simulation Application Considering Data Distribution Service*

**Pushpiita Ijanz***

*Prince Sattam Bin Abdul Aziz University, Saudi Arabia*

**corresponding author*

***Keywords:*** Data Distribution, Distributed System, Publish / Subscribe, Performance Test

***Abstract:*** Data distribution service (DDS) is a distributed real-time communication middleware specification published by object management group (OMG). This paper mainly studies the application analysis of distributed system simulation considering data distribution service. This paper first introduces the concept and composition of data distribution service, and then puts forward a distributed system data distribution model, and constructs a "bus type" data distribution model. And the performance of the data distribution model is tested. Through the performance test results, we can know that the data distribution model built in this paper can use DDS to achieve efficient remote procedure call development and operation.

## 1. Introduction

Data distribution service is a message middleware based on publish / subscribe mode, which is mainly used to provide one to many active data push service. Due to its flexibility in deployment and reliability of transmission, it has been widely used in multi-party communication scenarios with high real-time requirements such as multi-party video conference, voice intercom and unified command [1-2]. However, in the design of the existing data distribution service (DDS), there is no special processing for the data transmission process of the user, only the data is repeatedly sent to each subscriber in order to realize the data distribution process. Therefore, there is a large amount of redundant data in the DDS network, resulting in a waste of link bandwidth resources. Moreover, when the number of subscribers in the network is relatively large, the data release delay of the subscribers behind may be too long. Therefore, it is necessary to study a bandwidth saving, delay guaranteed and efficient data distribution service [3-4]. On the other hand, the traditional content delivery network (CDN) technology is mainly used to accelerate the process of Internet users accessing websites. In general, there is a large amount of redundant data in the Internet. It is precisely because of the repetitive transmission of data that the utilization of network bandwidth is greatly reduced, the transmission delay of user data is increased, and the user experience is reduced.

Therefore, by building a CDN acceleration network on the existing Internet, the user access speed can be improved and the load balance of link bandwidth can be realized [5].

At present, the widely used patterns in distributed architecture include remote procedure call (RPC), remote method call (RMI), message middleware (MOM) and stream [6]. Some scholars have compared these modes and proposed a mechanism based on the message oriented middleware system, which can effectively decouple the system, avoid congestion and increase throughput. These characteristics make the message oriented middleware have obvious advantages in asynchronous transmission scenarios. However, the scholar also pointed out two problems: on the one hand, the network bandwidth will affect the integrity and stability of message delivery [7]; On the other hand, since the push mechanism of message middleware is not real-time, the delay may affect the consistency of messages. In recent years, message queue (MQ) is a message middleware that is widely used in large-scale distributed scenarios. In terms of the more mature message queues, such as Kafka, rabbitmq, ActiveMQ and rocketmq, most of them follow the topic based publish / subscribe model [89]. Some experts compared the performance of several popular message queues at present, and compared the differences of different distributed message queues in real-time computing scenarios in terms of real-time, concurrency, reliability and scalability. The results showed that several queues showed different advantages in receiving and sending performance. However, adopting message queue can not solve the data interaction problem of distributed system once and for all [9].

In general, a single communication mode cannot meet the needs of large-scale distributed systems. A standard mechanism for request / response communication based on publish / subscribe is required to meet the needs of users for multiple communication modes. That is to study a remote procedure call mechanism based on DDS, and add a layer of service on the DDS layer to provide remote procedure call communication by using the support of the underlying basic components of DDS.

## 2. Data Distribution Service in Distributed System

### 2.1. Concept and Composition of Data Distribution Service

DDS is a publish / subscribe message middleware, which provides a one to many real-time communication architecture and realizes information exchange in a distributed environment. DDS is regulated by the object management organization and defines a set of comprehensive quality of service (QoS) policies to achieve the matching of data transmission standards [10]. All participant entities in the DDS communication architecture, as well as the read and write operations of the participant entities, need to bind a special QoS policy. The user can set different QoS policies for each participant entity and operation according to different communication environments, so as to realize the control of communication quality in the transmission process. However, DDS relies on the QoS policy guarantee to provide high reliability and flexibility. At present, the data communication service established by using DDS middleware has been widely used in many real-time multi-party communication application scenarios such as multi-party video conference, multi-party voice intercom, unified command and operation [11-12]. DDS is mainly used in three design and application environments, including:

(1) Application scenarios with high user delay requirements;

(2) An application scenario in which multiple communication parties participate in communication;

(3) Application scenarios with real-time changes in the communication network architecture;

The DDS standard divides the functions of data distribution services into two layers: data centric publish subscriber (DcpS) and local data reconstruction layer (DLRL) [13]. Among them, the DcpS layer is the core function of the entire data distribution service. Its main role is to realize the transmission of data between publishers and subscribers, and ensure the reliability and effectiveness of the transmission process, that is, to realize the one to many transmission process of data. The DLRL layer is based on the DcpS layer. Through hardware devices and logic, all local data are shared with remote users to realize remote users' reading of local data [14]. At present, the three most widely used DDS implementation products in the market are opendds, rtidds and opensplice DDS. Their implementation processes, supported systems and implemented programming languages are different, but all have realized the functions of DcpS layer [15].

### 2.2. Distributed System Data Distribution Model

When building the data distribution model, its distributed structural characteristics should be considered, and the point-to-point model is not applicable to the distributed system [16]. The client / server model and the distributed object model can be used as the data distribution model of the distributed system, but they have the following disadvantages: both are models with a central node, and the failure of the central node will lead to the failure of the entire system. Both require the client and the server to establish a stable data communication link through the request response mechanism. The request packet sent by the client and the response packet sent by the server consume a certain amount of network resources, and it takes time to establish the communication link. Both of them ensure the reliability of data distribution while sacrificing real-time. The data distribution of the two depends on the establishment of a stable data link. The client needs to know the address of the server before it can send the request packet. From the perspective of space, the two are closely coupled. The server needs to reply the confirmation data packet before the link can be established. Therefore, the client and the server are online at the same time. In terms of time, the two are closely coupled [17]. Before the data link is established, even if there is data to be distributed, the data distribution is blocked. From the perspective of data flow, the two are closely coupled.

The publish / subscribe model supports one-to-one, one to many and many to many communication modes. Combined with the distributed structural characteristics, the publish / subscribe model can be used as a data distribution model. In the publish / subscribe model, the roles of participants are flexible and there are no privileged nodes. The exit and joining of any node will not affect other data distribution processes in the system, and the functional subsystem realizes "plug and play". The data information interaction based on the publish / subscribe model does not need to establish a stable link, improves the real-time and efficiency of data information interaction, and can solve the demand of "control oriented" data information for hard real-time. The most prominent advantage of the publish / subscribe model is that it can realize complete decoupling of time, space and data flow [18]. In the distributed system, the decoupling characteristic of time can enable the newly added functional subsystem to obtain the required historical data; The decoupling property of space allows nodes to communicate anonymously without establishing a stable data link, reducing the network resources occupied by redundant information caused by the establishment of a link; The data flow decoupling avoids the data transmission process from being blocked, and improves the efficiency of data communication to a certain extent. The advantages of fully decoupled publish / subscribe model make the whole unmanned boat system have better expansibility, flexibility and maintainability. The data distribution models of distributed systems are mostly distributed object models and publish / subscribe models. Publish / subscribe model is the first choice of data distribution model in distributed system.

Based on the architecture of distributed system and the data distribution principle of publish / subscribe model, this paper proposes a "bus type" data distribution model. The model follows the participant role and data communication mechanism of the publish / subscribe model. "Bus type" means that all functional subsystems of the system are connected to the "data bus" and realize data information interaction. The "bus type" data distribution model is shown in Figure 1.

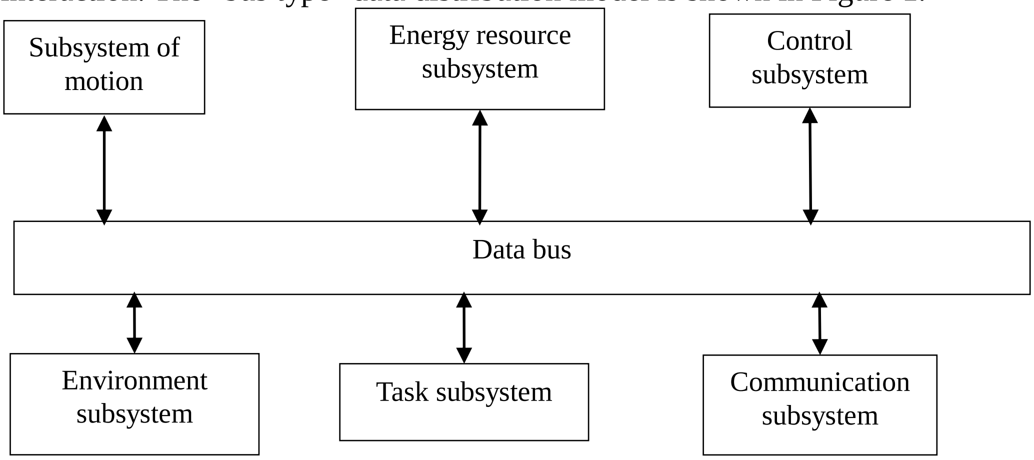


*Figure 1. Distributed data distribution model*

The "bus type" data distribution model is essentially a publish / subscribe model. The publisher and the subscriber do not need to establish a stable data link and realize anonymous communication. As a participant in the "bus type" data distribution model, the functional subsystem does not care about the number of other functional subsystems, address details and data information sources, and only publishes and subscribes data.

The data distribution service provides communication function services for data information interaction among participants, aiming to determine the details of all participants, match participants who intend to interact with data information, and notify corresponding participants to interact with data information. It is mainly divided into three aspects:

(1) Automatically discover endpoints. Through the automatic discovery algorithm, the data distribution service automatically discovers the number of participants in the network, as well as the address and other details, in preparation for matching the publisher and the subscriber.

(2) Match publisher and subscriber. Through the matching algorithm and participant details, the data distribution service matches the publisher and subscriber interested in interactive data information, which is equivalent to establishing a virtual link for the two. The establishment of the matching relationship is not subject to time constraints, space constraints and data flow constraints.

(3) Notification. The data distribution service notifies the functional subsystem by itself. After the matching relationship is established, the publishing module or subscription module in the functional subsystem is activated. At this time, the function subsystem will publish or subscribe data.

## 3. Data Distribution Performance Test

This paper tests the remote procedure call communication mechanism based on DDS and its influence on information integration management software. The remote procedure call communication mechanism based on DDS extends the service discovery matching algorithm on the basis of the original DDS, and adds the information filtering link. Therefore, the added link can be tested, and the time consumption of each link can be taken as the measurement index.

### 3.1. Delay Test

The communication delay reflects the speed of message processing and transmission of the DDS prototype system. In the distributed system, the client, server and other endpoints are not necessarily clock synchronized, and the single transmission delay is short, so it is difficult to directly measure, and the measured data may have a large error. Therefore, the paper adopts the method of multiple round trips of data, and finally calculates the average value.

### 3.2. Throughput Test

The test observes the change of client throughput by changing the data size, and compares it with DDS and socket under the same communication conditions.

## 4. Analysis of Performance Test Results

### 4.1. Analysis of Delay Test Results

*Table 1. Test for different data sizes*

| | 2KB | 4KB | 16KB | 64KB | 128KB |
|---|---|---|---|---|---|
| DDS | 1.24ms | 1.79ms | 2.04ms | 5.87ms | 11.43ms |
| Socket | 1.17ms | 1.63ms | 1.89ms | 5.21ms | 10.26ms |

As shown in Table 1, the test results show that the communication delay of DDS increases with the increase of the data size of the transmitted information, and the time delay is slightly larger than the socket, and the difference is small.

As shown in Figure 2, the number of clients in the test increased from 1 to 30. From the test results, it can be seen that the communication delay of DDS and socket increases steadily with the increase of the number of clients (Publishers), and the communication delay of DDDs is higher than that of socket. When the number of clients is small, the communication delay between socket and DDS is not much different. When the number of clients increases, the service binding time increases, and the information to be filtered increases greatly. The information filtering time is the main reason for the increase in delay. Therefore, when the number of clients increases, the information filtering time also increases, making the delay gap larger.

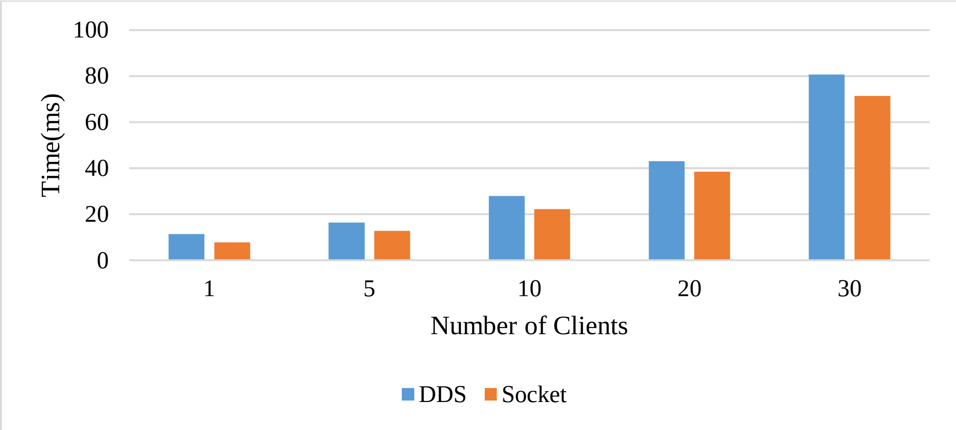


*Figure 2. Latency of different clients*

## 4.2. Throughput Test

*Table 2. Test for different data sizes*

| | 2KB | 4KB | 16KB | 64KB | 128KB |
|---|---|---|---|---|---|
| DDS | 2237 | 2196 | 1179 | 524 | 15 |
| Socket | 2814 | 2503 | 1348 | 557 | 13 |

As shown in Table 2, the throughput of the DDS is smaller than that of the socket, and when the data size gradually increases, the number of requests completed by the client (that is, the number of requests sent and replies received) gradually decreases. This is because the system has limited ability to process information and can only process a certain amount of data per unit time. Therefore, when the data becomes large, the information received by the client to complete the request will be correspondingly reduced.

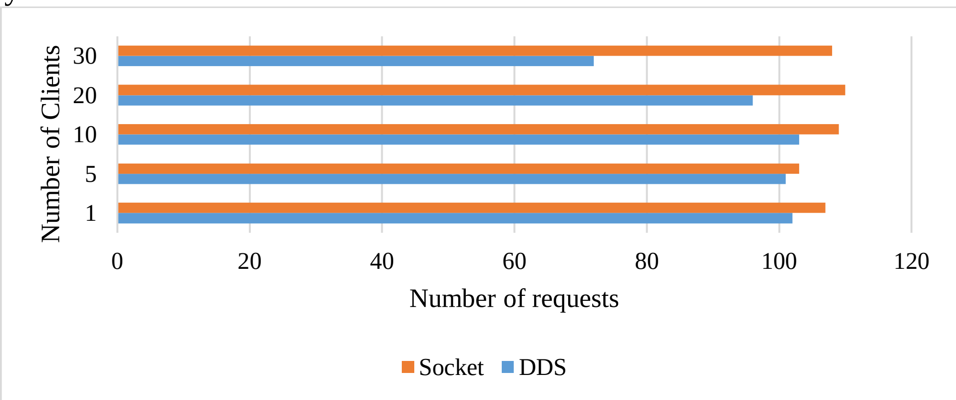


*Figure 3. Server-side throughput testing*

As shown in Fig. 3, it can be seen from the throughput test results that in most cases, the throughput of DDS and sockets keeps stable when the number increases, and the overall throughput of sockets is greater than that of DDS. When the number of clients is greater than 15, the throughput of the DDS server is significantly reduced.

## 5. Conclusion

This paper tests the DDS system, verifies the correctness of the remote procedure call communication mechanism based on DDS, and compares the data transmission success rate of DDS before and after the improvement of the service discovery matching algorithm. From the perspective of performance, the service binding time is tested, and the impact of the improved service discovery and matching time on the system communication delay is analyzed according to the service discovery and matching time of the information integration management software; The information filtering overhead is tested to analyze the impact of the information filtering mechanism on the communication delay. In addition, the delay and throughput of DDS are compared with socket under the same conditions. The test results show that the remote procedure call communication mechanism based on DDS can use the DDS publish / subscribe middleware to realize the communication mechanism of remote procedure call, and greatly utilizes the communication advantages of the underlying DDS and extends the communication mechanism of DDS.

## 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] Peppanen J, Reno M J, Thakkar M, et al. Leveraging AMI Data for Distribution System Model Calibration and Situational Awareness. IEEE Transactions on Smart Grid, 2017, 6(4):2050-2059.*

*[2] Krishnan V, Bugbee B, Elgindy T, et al. Validation of Synthetic U.S. Electric Power Distribution System Data Sets. IEEE Transactions on Smart Grid, 2020, PP(99):1-1.*

*[3] Mishra, Nilamadhab, Chang, et al. Sensor data distribution and knowledge inference framework for a cognitive-based distributed storage sink environment. International Journal of Sensor Networks, 2018, 26(1):26-42.*

*[4] Franczyk A. Sensitivity analysis of the specific heat and thermal conductivity values on pore pressure and temperature distribution within embankment. Running algorithm of simulator of wireless distributed measurement-control system with rule based processing, 2017,*

*63(3):86-89.*

*[5] Jhala K M, Natarajan B, Pahwa A, et al. Stability of Transactive Energy Market-based Power Distribution System under Data Integrity Attack. IEEE Transactions on Industrial Informatics, 2019, PP(99):1-1.*

*[6] Vikulov E O, Denisov O V, Denisova L A. Data distribution system: preparation of server stations data. Journal of Physics Conference Series, 2018, 1050(1):012097.*

*[7] Takehiro, NAGATO, Takumi, et al. Distributed Key-Value Storage for Edge Computing and Its Explicit Data Distribution Method. IEICE Transactions on Communications, 2020, E103.B(1):20-31.*

*[8] Mishra N, Chang H T, Lin C C. Sensor data distribution and knowledge inference framework for a cognitive-based distributed storage sink environment. International Journal of Sensor Networks, 2018, 26(1):26.*

*[9] Titov M, G Záruba, De K, et al. A study of the applicability of recommender systems for the Production and Distributed Analysis system PanDA of the ATLAS Experiment. Journal of Physics Conference Series, 2018, 1085(4):042028.*

*[10] Tekinerdogan B, T Çelik, Ömer Köksal. Generation of Feasible Deployment Configuration Alternatives for Data Distribution Service Based Systems. Computer Standards & Interfaces, 2018, 58(MAY):126-145.*

*[11] Ma C, Menke J H, Dasenbrock J, et al. Evaluation of energy losses in low voltage distribution grids with high penetration of distributed generation. Applied Energy, 2019, 256(Dec.15):113907.1-113907.15.*

*[12] Khalil I, Ahmad I, Almazroi A A. Energy Efficient Indivisible Workload Distribution in Geographically Distributed Data Centers. IEEE Access, 2019, 7(1):82672-82680.*

*[13] M Jiménez-López, F Torres-González, JL Gutiérrez-Rivas, et al. A Fully Programmable White-Rabbit Node for the SKA Telescope PPS Distribution System. IEEE Transactions on Instrumentation and Measurement, 2018, 68(2):632-641.*

*[14] Chernetsova E, Shishkin A. An Algorithm for Processing Data Received from a Distributed Monitoring Network Consisting of Several Sensors. IOP Conference Series Materials Science and Engineering, 2020, 1079(6):062062.*

*[15] Thi Q, Bobineau C, Debusschere V, et al. CPDE: A methodology for the transparent distribution of centralized smart grid programs. IEEE Transactions on Parallel and Distributed Systems, 2020, PP(99):1-1.*

*[16] Gil, Katz, Pablo, et al. Distributed Binary Detection With Lossy Data Compression. IEEE Transactions on Information Theory, 2017, 63(8):5207-5227.*

*[17] A D J P, B S G, C N C C. Interesting association rule mining with consistent and inconsistent rule detection from big sales data in distributed environment - ScienceDirect. Future Computing and Informatics Journal, 2017, 2( 1):19-30.*

*[18] Majumdar A, Agalgaonkar Y, Pal B C, et al. Centralized Volt-Var Optimization Strategy Considering Malicious Attack on Distributed Energy Resources Control. IEEE Transactions on Sustainable Energy, 2017, PP(99):1-1.*