Message Queue Optimization Model Based on Periodic Execution and Category Priority

Kunying Li¹, Xiaolian Li¹, Dexin Qiao¹, Yu Ding¹, Liling Wang¹

¹Computer Application Technology Department, PetroChina Research Institute of Petroleum Exploration & Development, Beijing 100083, China

Abstract. In recent years, the state has invested tremendous energy in the development of enterprise informatization construction and has also derived numerous high-quality enterprise informatization forms. At the same time, when the enterprise develops to a certain extent, the demand for change management, innovation mode and strategic transformation of information technology is gradually highlighted. Among them, the rapid development of information technology brings about many business systems, weak structure, difficult to adapt to system integration, information sharing and business collaboration. It is also the next important goal of enterprise information construction solutions. In order to solve the huge throughput problem caused by the integration platform of large enterprises in the face of tens of business system information data integration, the aim is to achieve three kinds of requirements for data integration, business integration and process integration of upstream business systems. Based on execution in the period range, this paper proposes a double-execution optimization algorithm for message queue with periodic execution and priority definition. The algorithm is implemented by analysing the operation behaviour of user roles, systems and time, and combining with the defined message importance tag. The experimental results show that this method can effectively filter, sort and execute a large number of message throughput accurately in a distributed working environment, and reduce the performance and delay risk caused by the complexity of information in the integrated system.

1. Introduction

At present, it is the most important direction of information construction to help traditional enterprises realize the integration of scattered systems under the background of big data environment. Especially how to achieve data integration, business integration and obtain real valuable information from different business systems becomes particularly important for various industries. Topic of conversation. In order to solve and optimize the problems of data availability, complex sources and asynchronous consistency of massive throughput data flow in large-scale integration platform, there are many targeted data flow optimization processing methods, some of which are based on real-time monitoring and processing, judging or expanding their priorities in order to reduce the priority persistence. For the purpose of line scope, some consumer objects redesign the reading of data interface and the modification of data offset interface respectively. In fact, if we can grasp the important level of message source group and message receiving group, and reserve processing space in distributed cluster environment, and process the message that users want or care about in advance, it will obviously improve the user's precise targeting and use experience in message processing. Degree. Based on the periodic permutation calculation of message queue, this paper analyses the operation behaviour of user roles, systems and time, and then
proposes a double definition based on periodic message execution and category priority by combining the pre-set label of the importance of the defined system message source and sending object. Queue optimization model is tested to simulate complex data sources with high throughput and concurrency to verify the feasibility of the model. The experimental results show that the optimized algorithm can better adapt to the actual business needs of large-scale integration platform and improve user satisfaction.

2. Research background

At present, the methods of message queue processing at home and abroad are mainly reflected in the following aspects. Optimize the message confirmation mechanism of Rabbit MQ, send the confirmation message to the producer after receiving the confirmation from the consumer, and resend the message when the message is lost[1], reduce the information recorded in the process of message confirmation, and improve the reliability of non-persistent message delivery[2] Design a message queue in shared memory. The main purpose of queue is to provide a routing method and ensure the delivery of messages. By using shared memory, data can be shared among various applications. Messages can be located by a two-level index to improve the consumption rate of messages in the case of large data[3]. Service-side high-time computing links can be synchronized with data[4]. The step-serial locking process is separated, and the asynchronous parallel processing mechanism based on multi-channel message queue middleware is introduced, and the corresponding relaxed synchronous transaction safeguard measures are provided. In view of the specific application process of HCMQM, the isolation of multi-mode message queues is realized through virtual message queue group, and the configuration file is used[5]. To manage message queue service flexibly. A new dynamic load balancing algorithm is designed, which solves the problem of poor performance of traditional algorithms in heterogeneous clusters and insufficient sensitivity to real-time load changes and improves the efficiency of message queue push[6]. Based on the existing research, this paper further records and analyses the user roles, systems and time operations, and then aggregates the optimization model by combining the defined system message source and the pre-set label of the importance of sending objects, in order to improve the execution efficiency and requirements of large-scale data throughput.

In addition, it proposes a method to solve the data consistency problem in typical application scenarios of distributed microservice by using transactional message queuing[7], so that this paper designs a distributed deployment scheme suitable for this algorithm model in the process of processing message queue, which reserves some additional resources for message body service in high concurrent emergencies. In this design, the distributed coordination technology and load balancing special allocation processing technology are used to solve the problem of reducing the sensitivity of traditional algorithms to large-scale concurrency of data in heterogeneous distributed cluster environment, and to make the algorithm model in this paper real-time and concurrent for different data sources under distributed message queue. The three aspects of performance and reliability have better performance, which provides a key modelling basis for the next queue execution.

3. Queue optimization algorithm with periodic execution plan and priority definition

Due to the increasing integration of platform business systems and the different requirements of importance, sensitivity and synchronization speed of different business systems, message queuing algorithms need to provide differentiated satisfaction for data from different sources. In the process of data processing, the system needs to know which data are most urgently needed by users. At present, the optimized scheme is to process the message body itself preliminarily. Periodic information pre-processing time arrangement calculation is the key link, and it is also the most common and efficient way. Processing the later data can effectively solve the sequential execution problem[8]. However, in the actual execution process, the user’s concerns and the importance of message body are difficult to be expressed by this method. Simple time series pre-processing calculation does not solve the complex logic execution problems caused by the expanding range of message queue data.

3.1 Periodic Execution Allocation
Therefore, the key premise of the algorithm is to arrange the data according to the time sequence. Following are the basic computational time allocation methods used in practice:

### 3.1.1 Resources Calculate:
RC represents the idle rate of queues in the entire resource environment. If \( \text{total}_r \) of total resources and \( \text{used}_r \) of used resources is set, then the message queue vacancy rate at time \( t \) node can be defined as follows:

\[
RC_t = \frac{\text{total}_r - \text{used}_r}{\text{total}_r} \tag{1}
\]

The denominator \( \text{total}_r \) in the formula represents the total number of nodes owned by the current time node \( t \), and the molecule represents the number of remaining available nodes under the current time node.

### 3.1.2 Time allocation plan:
When a message body enters a message queue, according to the current timestamp and the estimated execution time calculated from resources, a label identifying the sequence of the same batch is added to the data, which can better represent the distribution characteristics of the batch of data.

Setting up a message queue, the total number of loaded batches is \( m \), the total number of messages contained in each batch is \( \sum_{i=1}^{m} n_i = 1 \), and the system can set each message to allow the session duration to be \( T \). At this time, the same batch of data will be executed one after another, so the queue ranking \( Q_m \) is set according to the time, which is used to ensure the consistency of data. According to the prediction rule, the total consumption time \( TW \) is:

\[
TW_i = \sum_{i=1}^{n} Q_m \times \max_{T,T>0} \tag{2}
\]

Based on the above two calculation rules, an execution rule plan TAP suitable for system resource allocation can be generated. The execution efficiency decreases inversely with the number of running messages, and the frequency of start-up execution nodes increases directly with the time consumed. Finally, TAP value can be expressed by log function, and periodic ranking plan can be obtained.

\[
TAP_{t,i} = \log_{RC_t} TW_i \tag{3}
\]

The TAP generated by the above method provides a queue execution reference value for each batch of data messages. Combining with the early warning index and reserved resource pool, it helps the system allocate reasonably under overload and improves the efficiency of resource utilization.

### 3.2 An Optimized Model Based on the Definition of Message Category Importance
According to the \( TAP_{t,i} \) computing rules, this paper proposes an optimization model based on the definition of message category importance. By setting a series of environmental attributes such as user role, system, system message source and the importance of sending object, the importance of message data in each priority system is calculated. In this way, the execution of message queue is closer to the evaluation criteria oriented by the importance of information environment, and further optimized in data streaming.

According to the selection range of the super parameters, each group of super parameters is sent to the queue. Each node obtains the message from the queue and then trains the model in parallel and verifies the accuracy\[9]. Finally, the model with the highest accuracy is selected to calculate the data set to be predicted. The principal implementation of the model is as follows: Firstly, when platform integrates docking system, it defines important level for each application system. Secondly, each user will have his own role attribute, which is used to distinguish different groups of users, and the importance of each role is different. Then the platform will define the process section according to which each message belongs. Points define corresponding important values, and the importance of the recipient of the final message data is also considered in the whole evaluation process.
3.2.1 Assuming that the evaluation model is $\text{AssesModel}$. The importance of a system is vector $\text{Sys}_x$, the current user role is defined as vector $\text{Role}_m$, the importance of the process link of the current message is $\text{Proc}_y$ and the role of the message receiver is defined as vector $\text{NRole}_n$.

$$\text{AssesModel} = \frac{\text{ass} (\text{Sys}_x, \text{Role}_m, \text{Proc}_y, \text{NRole}_n)}{TW_i}$$  \hspace{1cm} (4)

Under the calculation of the above model, the evaluation value given by the matrix is quoted with the total time consumed by the batch of messages. The value calculated by the model will approach 0 as the time consumed increases. At the same time, if the importance of the matrix increases continuously, the result of the model will also approach 1.

3.2.2 The final evaluation criteria are:

a) If the message data to be executed is found to be about to reach the full value when computing resource $\text{TAP}_{ti}$, all message data waiting for queuing will be evaluated by $\text{AssesModel}$ evaluation model. If the evaluation value is greater than the set reserve resource access standard, the batch message will be sent to priority execution after load balancing adjustment.

b) If the value of $\text{TAP}_{ti}$ is below the standard line, then the message groups to be queued will be prioritized by the $\text{AssesModel}$ model and put into the queue according to the evaluation value.

In order to avoid cluster overload and mass data execution at the same time affecting the execution efficiency, time protection measures are used in this paper. When a message is sent, it is necessary to consider the ability of the server cluster itself to withstand, including disaster preparedness, high availability and high performance. When the data execution time is too long and no response is returned, it will be moved out of the execution queue and added to the temporary queue, waiting for the resource to execute again when it is free.

4. Results and Analysis

4.1 Test environment and methods

Because this test uses RabbitMQ to test and deploy, and start the persistence queue, the performance testing tool uses Performance Runner. The following is the basic configuration table of the test environment.

| Table 1. Performance Test Environment Configuration |
|-----------------------------------------------|
| Configuration item           | parameter    |
| CPU                          | 8 Core       |
| Memory                       | 8G           |
| Hard disk                    | 500G         |
| operating system            | Windows Server2012 R |
| RabbitMQ                     | V 3.6.10    |
| PerformanceRunner            | V 1.1.4      |

This test will deploy three virtual machines as cluster environment, start different number of persistent queues in each virtual machine, will link a total of 30 different business systems, there are 15 producers, 18 consumers, 1 channel in each producer connection, 2 channels in each consumer connection. The number of queues is less than and so on. Bind to the same persistent exchange switch at 20:00; bind to two persistent exchange switches when the number of queues is more than 20. Record the normal message before and after optimization and the message sending rate whose evaluation is higher than the standard value. Test each system data source for 20 minutes, test five times to get the average.
The calculation method of message sending rate: the total number of bytes needed to send a message queue / the time needed to send a message (seconds).

5. Analysis of test results
In this test, as shown in Figure 1, the number of messages in the message queue is classified and compared according to the total number, normal and over-standard messages. It can be seen the increasing of data sources in the business system, the data streams exceeding the standard values are also increasing. When the fourth business data source accesses the platform, the data streams exceeding the standard values are increasing. The data entering the special message queue began to rise continuously, indicating that the effect of the evaluation model has begun to highlight at this time; when the 10th service data source accesses, it has tended to rise steadily; when the 25th time, it needs special processing, it is close to the platform message.

![Figure 1](image)

Figure 1. Number comparison of message categories.

From the above figure, it can be seen that deepening of business, more and more complex logic and diversified role definition, the data users need to care about is becoming more and more timely, and the performance and efficiency of the message queue processing mechanism is becoming stronger and stronger. By using the optimized method, the data that need real-time processing can be processed in time through reserve resources, and the data in the common queue can be prioritized to ensure that the common message data can be executed according to the expectations of users.

6. Conclusion
This paper describes in detail the use of a double-definition queue optimization model based on periodic message execution and category priority, which solves the problem of message queue execution in the face of complex business and guarantees the real-time and efficiency of users in key data transmission. At the same time, in the process of executing the optimized model to process message queue, due to the insufficient reserve resources in the early stage, the messages that need to be processed first exceed the capacity of reserve resources, which leads to the delay of messages in the queue. Therefore, the control of load balancing for nodes is adjusted accordingly, which reduces the pipeline of common nodes and expands the message data that needs to be processed first. Therefore, it is the next research direction to dynamically adjust different indicators with the help of large data computing and intelligent analysis technology, so that the system can automatically adjust the queue access value and resource allocation.

REFERENCES
[1] XU Zhen, JIAO Wen-Bin. Optimization of Tiny Message Acknowledgement in RabbitMQ. COMPUTER SYSTEMS APPLICATIONS, 2018, 27(3): 252-257.
[2] Yang W, Liu X, Zhang L, et al. Big Data Real-Time Processing Based on Storm. 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (Trust Com). 2013
[3] Zhou C. Design and implementation of communication model based on improved ActiveMQ [D]. University of JiLin. 2017.

[4] YANG Haitao, ZHANG Chuanbin, RUAN Zhenjiang, et al. Asynchronous parallel optimization of large-scale cloud sync-collection data systems. Computer Engineering and Applications, 2017, 53(2):88-97.

[5] He Shuangyuan. Research on and Application of Message Queue Model Under High Concurrency[D]. Wuhan University of Technology. 2015.

[6] Hou Weiguang. Research and implementation of key technologies internet massive traffic information collection and transmission[D]. Beijing University of Posts and Telecommunications. 2019.

[7] ZHANG Jie, MAN Shu-Guang, LIU Kai, ZHOU Li-Jun. Data Consistency Method Based on Message Queuing [J]. Computer Systems & Applications, 2019, 28(09):185-189. Conference Name: ACM Woodstock conference

[8] C. Sadfi, B. Penz, C. Rapine, J. Blazewicz, P. Formanowicz. An improved approximation algorithm for the single machine total completion time scheduling problem with availability constraints. European Journal of Operational Research. 2005

[9] NAN Dong-liang WANG Wei-qing WANG Hai-yun, Optimization of LightGBM hyper-parameters based on message queuing [J]. Computer Engineering & Science, 2019, 41(08): 1360-1365.