Resource Management Method of Distributed Parallel Database Based on Directed Graph

Yingjie Shi, Huayun Zhang, Xiang Wang, Yuchao Zhou and Shusong Jiang
China Realtime Database Co.Ltd. Building 6, big data Industrial Park, No.180, software Avenue, Yuhuatai District, Nanjing, China
Email: shiyijingjie@sgepri.sgcc.com.cn

Abstract. A distributed parallel database resource management method based on directed graph is proposed. By using content distributor based on distributed unstructured P2P network association, the high performance and stability of distributed parallel database system in dynamic changing environment are guaranteed. The problem of network congestion caused by too many redundant messages and the “barrel effect” caused by resource imbalance due to differential configuration is solved, through resource search algorithm based on directed graph lookahead, and the query node cached the resource information of two-level neighbor nodes. By adopting the Linux cgroups resource management mechanism, fully considering the multi tenant and multi factor based resource scheduling strategy, reduce resource fragments and better meet the problem of distributed parallel database storage or hot spot processing.

1. Introduction
As the core basic software of enterprise data warehouse, the Mass Parallel Processing (MPP) database provides structured data support for all kinds of analytical applications. At present, the mainstream MPP database adopts peer-to-peer (P2P) [1,2] deployment mode, realizes the system function through the mutual cooperation between nodes, makes full use of network resources and node resources, and has good autonomy, concurrency and scalability. How to manage and utilize network resources and node resources efficiently to access and calculate data is the basic key problem of MPP database application. In particular, the characteristics of P2P network, such as peer-to-peer and distributed, as well as the heterogeneity and dynamic of nodes, bring a lot of challenges and difficulties to the effective management of resources [3,4]. On the one hand, the processing efficiency and resource utilization rate of computing and storage nodes vary greatly with the hardware configuration. If the processing is not appropriate, it will easily lead to "barrel effect", which will affect the overall performance of the database; on the other hand, with the application extension of big data and cloud computing technology, the amount of user access, data change, user behavior habits and operation mode are all changing with the time. It is easy to lead to storage or hot issues in database system. In order to improve the overall operation performance of the system, MPP database must have dynamic and elastic distributed resource management ability to adjust system resources and system load [5].

At present, the research institutions at home and abroad have done a lot of research on distributed resource management technology, mainly from the aspects of resource allocation, scheduling and integration to analyze the load of balanced nodes, and achieved some research results. However, the existing load balancing methods still have some defects: some algorithms only consider the use of some system resources, which can not fully reflect the real load of nodes [6]; some algorithms are too complex to realize, which occupy more system resources and increase the load of the system [7]; some algorithms generate more data on the network when collecting resources, which occupy more data.
than others [8]. Most resource allocation strategies can not meet the needs of multiple scenarios and multiple factors. When choosing servers to allocate physical resources, the principle of space adaptation is adopted, ignoring the possibility of remaining resources being used by potential users, which is easy to form resource fragments and cause resource waste.

In view of the above problems, this paper proposes a resource management method of large-scale parallel processing database based on directed graph to solve the problem of unbalanced and unstable performance of large-scale parallel processing database caused by the differential configuration of computing and storage nodes.

2. Working Principle and Technical Scheme
The database should include multiple resource scheduling nodes and resource storage nodes, each resource scheduling node is connected with multiple resource storage nodes, each resource scheduling node contains content distributor, including the following steps: the content distributor of resource scheduling node receives the resource query request and communicates locally The lookahead algorithm based on directed graph is used for query; when the query fails, the content distributor sends the resource query request message to the neighbor node according to the directed graph, and the resource query request message is continuously forwarded in the multiple resource scheduling nodes until the required resource is found; after the resource storage node is located, the single resource storage node uses the set Linux cgroups (control groups) isolation technology [9] for resource isolation optimization.

2.1. Overall Structure
In the overall structure, according to the characteristics of peer-to-peer deployment of MPP database, common database nodes are divided into resource scheduling nodes and resource storage nodes according to resource functions, and content distributors are defined in each resource scheduling node for unified resource management and scheduling.

The whole system adopts a two-tier architecture. The upper layer is composed of resource scheduling nodes and content distributors. The unstructured P2P mode is adopted and the group is called management domain. The lower layer consists of resource storage nodes, which are called leaf nodes. A star topology structure is adopted between resource scheduling node and leaf node. One resource scheduling node is connected with multiple leaf nodes.

The leaf node is composed of resource storage nodes, corresponding to the calculation node or data node of MPP database. The leaf node group is called the processing domain. The leaf node is used to store the actual data and resources, and send its own CPU, I/O and network resource information to the content distributor. Leaf nodes also adopt peer-to-peer deployment mode, each node has no impact on each other, so it has high dynamic flexibility, good scalability, and satisfies the MPP database peer-to-peer computing node deployment structure.

C/S mode is used to communicate between the content distributor and the leaf node, which ensures low network bandwidth consumption and high search speed when searching resources in the management and processing domain.

![Figure 1. Overall structure.](image-url)
2.2. Functional Structure of Content Distributor

The content distributor is a resource scheduling service deployed in each resource scheduling node. The services are associated through the decentralized unstructured P2P network. Because there is no control of the central node, it has high scalability. The content distributor stores the topology resource information of leaf nodes, the file index information and the basic information of files. In the process of node state change (such as online or offline), the system does not need to transfer the node state information, and resources do not need to change its storage location. The structure of the content distributor is shown in Figure 2.

![Figure 2. Structure of content distributor](image-url)

2.3. Mpp Database Resource Management Process

1) Resource stored procedure

The resource storage node is responsible for the partition of storage files, and regularly sends its own resource performance information to the content distributor. The resource storage node has the following functions:

- Ability information upload: this function regularly sends performance information such as CPU, I/O, network, etc. to the content distributor as the basis of how to select the database computing node for the content distributor.
- File fragmentation storage: this function will ensure that file fragmentation and other relevant information are stored on the local disk.
- File fragment upload: this function ensures to find the corresponding file fragment in the local disk after receiving the request from the content distributor, and then upload the calculation result to the content distributor.

In the resource scheduling node, the content distributor stores the CPU, I/O and network resource information of the leaf node through the resource information table. And through unstructured P2P directed graph network association, to ensure high reliability.

2) Resource acquisition process

When querying resources, first send a request to the content distributor. After receiving the request, the content distributor searches for the free storage node of the corresponding table through the resource search algorithm, obtains the required file fragments through the resource storage node, reorganizes the files, and then sends the reorganized files to the query node to complete the distributed query of data resources.

The steps to obtain resources are as follows.

Step 1: initialize the resource storage node, set the CPU and I/O parameters related to cgroups, and set the priority and upper limit of related resources.

Step 2: the MPP database connection request is sent to the content distributor for resource query.
Step 3: when the content distributor receives the resource query request, it first queries through the lookahead resource search algorithm based on the directed graph locally. If successful, the query result will be returned.

Step 4: if the query fails, the content distributor will send the query request to the neighbor nodes according to the directed graph. In this way, the request information will be forwarded in the whole upper unstructured P2P network until the required resources are found.

Step 5: after locating the resource storage node, the cgroups isolation technology is used to optimize the resource isolation for a single resource storage node, so as to optimize the calculation efficiency of a single computing node in data writing, data query and other aspects.

2.4. Lookahead Resource Searching Algorithm Based on Directed Graph
The main body of the resource searching algorithm of the content dispenser adopts the breadth first searching algorithm, which should meet the characteristics of as many resource factors as possible, no single point failure, high searching speed and good scalability. The design idea is as follows:

1. Based on the lookahead algorithm [10], the routing information of the two-level neighbor nodes of the query node is cached by using the directed graph. In the process of resource query, the query message is propagated forward to solve the problem of single point failure.
2. The algorithm of breadth first search is adopted to meet the requirement of high search speed.
3. Considering CPU, disk I/O and other resource factors, we can reflect the real load of nodes.
4. The search algorithm makes full use of the resource information table stored in each content distributor, reduces the redundant information produced by resource searching. Cut down the consumption of network, and improves the scalability of the system.

The network topology of lookahead resource searching algorithm based on directed graph is directed graph, and messages between nodes are forwarded in the way of directed graph, which avoids the network storm caused by infinite message forwarding and reduces the network bandwidth consumption.

Definitions:
- Direct neighbor: the node pointed to in the network topology;
- Indirect neighbor: the node whose direct neighbor points to in the network topology;
- Parent node: the node pointing to this node in the network topology.

The content distributor forwards resource query request messages as follows:

Step 1: find the direct neighbor node that meets the conditions. The conditions should be: the parent node of the direct neighbor node is not the forwarding node of the resource query request message, and the direct neighbor node is not the forwarding node of the resource query request message;

Step 2: after getting the direct neighbor nodes that meet the conditions, find the nodes that meet the conditions in the direct neighbor nodes of these direct neighbor nodes. The conditions that should be met are: the node is not the forwarding node of the resource query request message and the node is not the direct neighbor node of the forwarding node;

Step 3: if the indirect neighbor node is the direct neighbor node of multiple direct neighbor nodes, it will only forward once.

2.5. Resource Isolation Management Based on Cgroups
Cgroups is used for resource management mechanism at operating system level for leaf nodes. Multi tenant and multi factor based resource scheduling strategies are fully considered to limit, record and process groups (such as CPU, I/O, etc.) used by process groups, effectively reduce resource fragmentation, so as to make full use of system resources.

CPU resource isolation: configure the priority parameter of resource pool and cpu_percent parameter implements CPU resource isolation and priority sharing, as shown in Table 1. Where, priority is the priority parameter, cpu_percent is the relative weight. When a task in a cgroups is idle, or the CPU time is not fully used, the remaining time is collected into a public unused CPU pool. Other cgroups are allowed to borrow CPU time from this pool. This shows that when the CPU is full,
it will be allocated in proportion. When the CPU is idle, it will be allocated on demand, and resources can be shared in groups.

Table 1. CPU resource configuration parameters

| parameter   | explain                                                                 |
|-------------|--------------------------------------------------------------------------|
| priority    | Resource pool priority, value range \{1 | 2 | 3 | 4 | 5 | 6 | 7 | 8\}, 1 represents the highest priority, 8 represents the lowest, default value is 1 |
| cpu_percent | Resource pool CPU usage limit, integer, range [1, 100], required parameter. |

I/O resource isolation: use the blkio subsystem of cgroups to control and monitor the I/O access of block devices. Through weight and weight_device parameters of device set the weight division, which is internally implemented by the Linux kernel full fair queue I/O scheduler. This policy allows setting the weight of cgroups. Each cgroup has a percentage of all I/O operations. At the same time, configure the write_bps_device and throttle.read_bps_device parameter limits the maximum disk I/O read / write rate of the resource pool, as shown in Table 2.

Table 2. I/O resource configuration parameters

| parameter                | explain                                                                 |
|--------------------------|--------------------------------------------------------------------------|
| blkio.weight             | Specifies the relative weight of I/O access, value range [100, 1000].     |
| blkio.weight_device      | The relative weight of I/O access of the specified device that cgroups can access. The value range is [100, 1000]. |
| blkio.write_bps_device   | Maximum write bandwidth of resource pool disk, unit: MB / s, parameter type: integer, value range [1, 879609302207] |
| blkio.throttle.read_bps_device | Maximum read bandwidth of resource pool disk, unit: MB / s, parameter type: integer, value range [1, 879609302207] |

3. Specific Implementation Mode
The topology of lookahead resource lookup algorithm based on directed graph is shown in Figure 3. The direct neighbor node of node V is node U and node X, the indirect neighbor node is node X and node Y, and the parent node of the direct neighbor node is node U and node Y.

Figure 3. Topology of lookahead resource searching algorithm based on directed graph

The information that node V needs to save is shown in Figure 4:

Figure 4. Example of resource information saved by nodes
As shown in Figure 5, during the execution of the lookahead resource lookup algorithm based on the directed graph, suppose X is the initiating node of the resource query request message, and N contains the required resource.

**Figure 5.** Execution process of lookahead resource lookup algorithm based on directed graph

The specific search process includes the following steps:

Step 1: at the beginning, node X queries its own resource information list and its own cached neighbor resource information list, and fails to find the required resources. According to the rules, it forwards the query resource request message to indirect neighbor node V. Node V does not store the required resources. Direct neighbor nodes of V include X, U, W1, W2, W3, W4.

Step 2: perform the following operations on the direct neighbor node of node V: node X is the initiating node of the resource query request message, which need no process, and node U is the direct neighbor node of X, which need no process too. Therefore, only the W1, W2, W3, W4 nodes need to be queried, and these four nodes do not store the required resources, so forward the resource query request to the direct neighbor nodes of these four nodes seeking information.

Step 3: W1 has no direct neighbor node and does not need to be forwarded.

Step 4: W2's direct neighbor nodes are W1 and Y1, W1 is the direct neighbor node of node V, no need to forward, node Y1 needs to forward.

Step 5: the direct neighbor nodes of W3 are Y1 and Y2, node Y1 has been forwarded once, no need to forward again, Y2 needs to forward.

Step 6: the direct neighbor nodes of W4 are Y3 and N, both of which meet the forwarding conditions and need to be forwarded. Finally, find the required resources in node N.

In this process, the forwarded message only passes through nodes X, V, Y1, Y2, Y3 and N. The dotted line in Figure 5 shows the forwarding route of the resource query request message.

The priority and percentage of its dynamic resource pool based on cgroups can be represented by a two-dimensional table. See Table 3 for typical case configurations. There are three resource priorities in horizontal direction, from high to low, level 1, level 2 and level 3. In vertical direction, according to the internal functions of MPP database cases, they are divided into six resource groups, namely super pool, OLTP group, OLAP group, load pool, development pool and other pool. Examples of setting according to the function priorities are as follows:

**Table 3.** Typical resource pool priority configuration

| Resource group   | Level1 | Level2 | Level3 |
|------------------|--------|--------|--------|
| Super pool       | 90%    |        |        |
| OLTP pool        | 50%    |        |        |
| OLAP pool        | 20%    |        |        |
| Load pool        | 10%    |        |        |
| Develop pool     |        | 50%    |        |
| Other pool       |        |        | 50%    |
4. Beneficial Effects
Through the distributed unstructured P2P network association based content distributor, using directed graph based resource search algorithm and Linux Cgroups resource management mechanism and other methods, fully consider multi tenant and multi factor based resource scheduling strategy, solve the problem of network congestion caused by too many redundant messages, reduce the "barrel effect" caused by resource imbalance due to differential configuration, reduce resource fragments, and ensure the high performance of large-scale parallel processing database system in dynamic changing environment. It is stable and has a good application effect.

5. Conclusion
In this paper, a large-scale parallel processing database resource management method based on directed graph is proposed according to the challenges brought by the characteristics of peer-to-peer and distributed network of MPP database system and the heterogeneity and dynamics of nodes to the effective management of resources. Its working principle, technical scheme, implementation process and beneficial effects are described in detail. The feasibility of this method is verified by a specific example, which effectively solves the bottleneck problems of network and resources in the process of resource management, has a certain practical value.

6. Acknowledgements
This paper is supported by the key technology research project of measurement data sharing service (524623200001).

7. References
[1] Shan Wang, Xinyu Zhang, Yulan Liu. Peer-to-Peer Research on data sharing[J]. Computer application and software, 2003, 20(11):1-4.  
[2] Yangqun Li, Mei Zhou. A resource management method of wot based on P2P: China, CN201810818741.9[P]. 2018-12-28.  
[3] Zilong He, Ningjiang Chen, Ruwei Huang, Guihao Jia, Xiang Li. TenantCPUMan: Multi tenant dynamic CPU resource adjustment based on load analysis[J]. Computer application and software, 2016, 33(12):11-14,19.  
[4] Zhenyu Li. Research on Key Technologies of P2P network resource management[D]. BeiJing: Institute of computing technology, Chinese Academy of Sciences, 2009.  
[5] Ling Zhang. Research on load balancing technology of distributed real-time database[D]. BeiJing: University of Chinese Academy of Sciences, 2013.  
[6] Qiao Sun, Lei Zhao, Buqiao Deng, etc. Low cost distributed database load balancing technology[J]. Computer application, 2017, 37(z1):240-242.  
[7] Jun He, Wei Xiong, Luo Chen, etc. Research and implementation of dynamic load balancing based on Database Cluster[J]. Microcomputer and Application, 2011, 30(2):68-71.  
[8] Lun Tang, Run Liang, Ya Zhang, Qianbin Chen. Research on load balancing algorithm based on POMDP load sensing in heterogeneous dense network[J]. Journal of College of Electronic Engineering, 2019, 8(5):87-93.  
[9] Xing Chen. Research on resource management strategy of cloud database based on CGroup[D]. Wu Han: Huazhong University of science and technology, 2016.  
[10] Ming Lin; Rong Chen; Jun S. Liu. Lookahead Strategies for Sequential Monte Carlo[J]. Statistical science: A review journal of the Institute of Mathematical Statistics, 2013, 28(1):69-94.