ISSN : 0974 - 7435 Volume 10 Issue 22

2014

BioTechnology

An Indian Journal

FULL PAPER

BTAIJ, 10(22), 2014 [13720-13726]

Study of the design and realization of database connection sharing technology based on the distributed connection management system

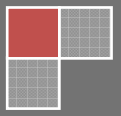
Juan Du, Jianchun Cao
Information Technology Engineering Department, Yellow River Conservancy Technical Institute, Kaifeng, 475004, (CHINA)

ABSTRACT

When the database is connecting, due to certain restrictions, it will affect the multiple programs to share data, and mainly expressing in that some programs takes up too much the database connection. And some programs cannot acquire database connection service, appearing unbalanced performance bottlenecks. The key point is how to solve the problem that various programs acquire effective allocation of their own needs. The method adopted is mainly through distributed connection management system to realize, it is mainly based on JDBC connection pool, which can achieve a higher level of connection sharing technical solutions. The research mainly studies the complexities of each node when carrying out share connection under distributed connection management system, in order to realize the optimal condition of dynamic load balancing; And how to add the new node to meet the high load system in the process of dynamic management, while reduce the node when the utilization rate of the load is low. Then allocation algorithm is studied, proposing the priority allocation strategy; The overall scheme of distributed connection management system are introduced, including its core module and implementation method of the structure, and also by introducing a new design to ensure that the connection pool system under the circumstance of high concurrency operate steady. Finally, the whole test system is verified, mainly on the distribution of the priority, dynamic adjustment and dynamic load balancing, etc.

KEYWORDS

Database connection; Sharing technique; Distributed connection management system; JDB connection pool.



INTRODUCTION

Database connection sharing technology can be said to be the crucial factors of website response speed, the current internet website has been involved in every aspect of life, more and more people use the website. And the corresponding speed of website is extremely important to keep the user. According to the data, the response time increases a second, it will lead to loss of 20% of the users. Figure 1 shows the basic structure of the general website application. Besides, modular service design is the main way to solve the difficulty of development in the current network application system[1], thus controlling the changes in application service content. But this kind of design can cause application services deployed in single machine or cluster server, namely, multiple applications share a single database at the same time. At this time, the database performance bottlenecks will lead to the emergence of serious problems, and it mainly due to the limited resources of database, when there is a connection online at full capacity, some application service will not be able to obtain a connection, and some application service takes up too much of the database connection. Therefore, how to realize the database connection sharing technology running efficiently has very practical value.

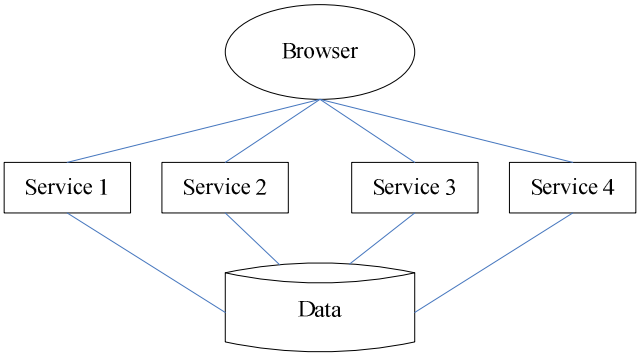


Figure 1 : The basic structure of the network applications

THEORETICAL BASIS AND RESEARCH STATUS

Research status

For the moment, the main problem in database connection sharing technology is how to solve the database connection limitation, reduce the performance bottleneck, making each application service obtain the database connection evenly. Currently there are two kinds of common database connection sharing scheme. One is realizing through the database connection pool built on the data application services; the other is building a database connection pool between the application services. The difference between the two is that the realization of former is in a single application, while the latter is in the public database connection pool.

The first scheme adopts the JAVA language, and database connection pool is realized on the JDBC[2], the main construction schemes are as follows:

(1) JDBC connection pool adopts C3P0 open source code[3], it is issued with Hibernate and is the database connection pool that used. It has the basic function of the general database connection pool, and the operation is stability and reliability.
(2) Open source BoneCP connection pool has been adopted, and the development kit is only 40k, while the development kit of CP3 reaches 600 k. The disadvantage is that the JDBC driver needs to be loaded, and besides, there are also other problems like inflexible and so on.
(3) The default database connection pool of Tomcat data source is DBCP, it has the normal function of the connection pool, but it can't test the failure of database connection pool.
(4) As the driver of the JAVA SQL, Proxool can provide many types of driver package. It is a mature configurable connection pool with very good performance of transplantation and high transparency.

The implementation of the second database connection sharing technology scheme needs to build database middleware, at present, more general schemes are as follows:

(1) Mysql database middleware Oceanbase in Alibaba can well implement the shared database connection, and it is also very effective for controlling the table base of relational database.
(2) The Atlas is developed by 360 companies, and its data connection sharing is implemented mainly in application code level, the major characteristics include read/write splitting, load balancing and depots table.

The basic theory

JDBC

JDBC is mainly composed of JAVA language classes and interfaces[4], it is mainly used to execute SQL statements including JAVA API. If JAVA program adopts JDBC, it can access to different types of databases easily, and can complete the establishment and management of the database connection. The database connection pool adopted in this study is built

using the JDBC, it has four drive types: JDBC – ODBC Bridge, the local API driver, network protocol driver and local protocol driver. JBDC API set a series of classes in order to express the metadata in database connection, the driver and database. See the specific in Figure 2:

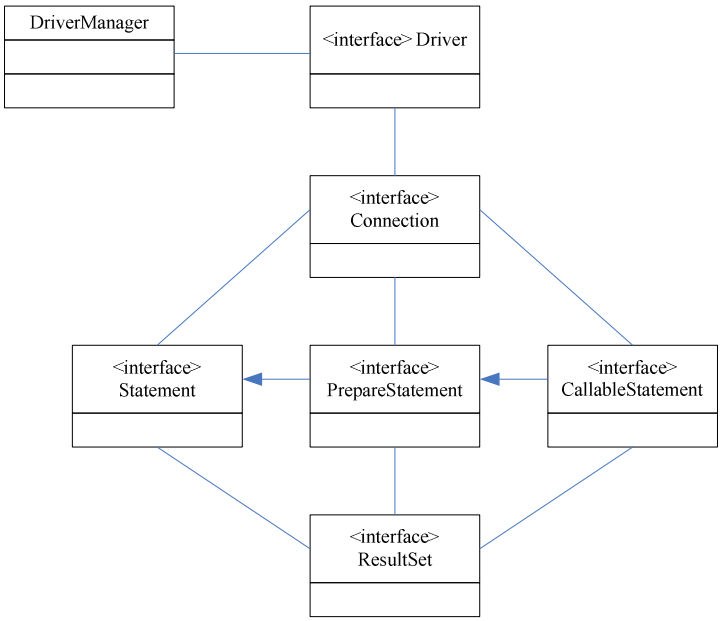


Figure 2 : JDBC basic class diagram

Web application frame mina

It is a high performance network communication module. Apache MINA has high performance of development and high scalability[5]. Asynchronous API is an abstract event-driven, and has several transmission modes as follows: TCP/IP, UDP/IP and serial port, etc. MINA's development is based on NIO of JAVA, and it has high transparency and also well encapsulation.

Database connection pool

Establishing a database connection is a large part in the process of database operations, and it is not a simple sense of the communication process between the server and the user, but several communications between a database server and the database driver. Such as in Mysql connection process, there will be 7-10 times communication, the process of performing is as follows:

(1) Check the user's information
(2) Negotiate the desired code page settings and also set the available code (optional);
(3) To obtain the database version information for confirmation
(4) To determine the ideal size of database protocol packet
(5) The final set of information about painting

Of course, building a “resource pool” for the database connection is the basic idea of database connection pool[6], the first is to set a certain amount of database connection, and obtain from it when there is demand, then put it back into the resource pool after using it. But database connection pool still needs a lot of technology to support, and one of the most important is how to make effective management. It refers to the establishment, destroy and distribution of the database connection.

Concurrent access is frequently used in the database connection pool, and thread safety is the major problems that needed to consider in concurrent access, if the whole connection pool is locked, then it lead to a loss of access efficiency seriously. Currently, the database connection pool like DBCP, CP30 usually adopt simple locking scheme, and of course a more efficient scheme like similar segmented locking can also be adopted.

Load balancing algorithm

Distributed connection management system is the foundation of studying database connection sharing technology, it involves load balancing, and load balancing algorithm has the following kinds:

(1) Round-robin scheduling algorithm
(2) Weighted round-robin scheduling algorithm
(3) Harsh-based scheduling algorithm
(4) Least-connection scheduling algorithm
(5) Least response time scheduling algorithm

In the above algorithms, the first three belongs to static load balancing algorithm, and their requests distribution has no relationship with and server body. The last two algorithms belong to dynamic load balancing algorithm. However, the above algorithms still have some limitations with no clear indicators to measure the load on the server requests. The study adopts load balancing algorithm which is the combination of static and dynamic I type, the overall effect will be more reasonable.

THE DESIGN SCHEME OF DISTRIBUTED CONNECTION MANAGEMENT SYSTEM

Basic function

The basic functions of distributed connection management system are as follows:

(1) To achieve dynamic load balancing, it can adjust the allocation request according to the circumstance of each nodes.
(2) To achieve dynamic node management
(3) To achieve the function of database connection sharing, making the database connection shared by many application service programs.
(4) To achieve distribution function, completing the server request according to the different priority.
(5) To achieve the whole management function.

The management architecture of distributed connection

Distributed connection management system, as the database middleware, is located in the middle of application service and the database, so the application services need to communicate through distributed connection management system and database, and the database is very transparent to application services. And through the concentration encapsulation of database access and database connection pool management, it can manage the database connection resource effectively. Connection pooling node and the control node are the basic component of distributed connection management system. Figure 3 shows the basic structure of the distributed connection management system:

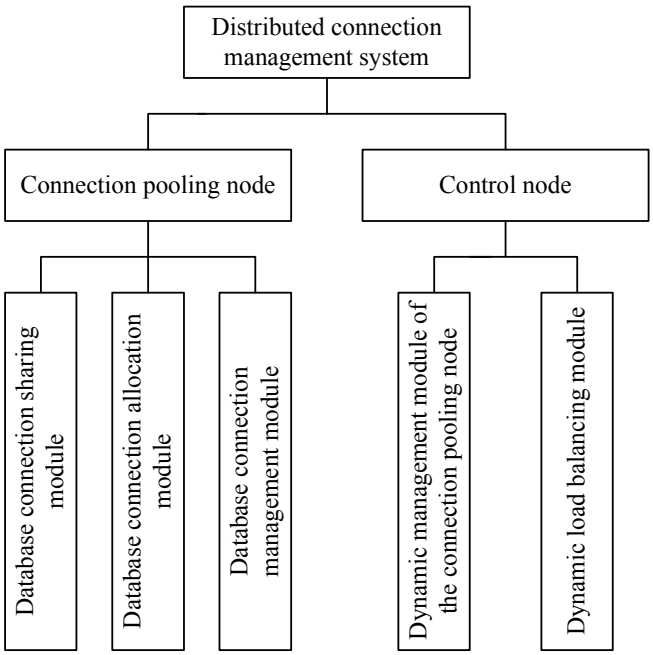


Figure 3 : The structure of the distributed connection management system

As can be seen from the above, the distributed connection management system is very effective for achieving dynamic load balancing, and it can be real-time adjusted according to the actual load. The connection pooling node list is one of the key factors, and TABLE 1 shows the connection pooling node list

TABLE 1 : The connection pool node list

Connection pooling node	Ip	Weight	State
node 1	192.168.10.1	3	available
node 2	192.168.10.2	2	available
node 3	192.168.10.3	1	available

As can be seen from the table above, the connection pooling nodes all have a specific item on the list of connection pool, the specific information including is IP, weights and state. However, the dynamic management of connection pool nodes has the following three parts:

(1) When the load of the connection pool node is at full capacity, start a new node to provide service
(2) And when the load of connection node is low, it will stop using some nodes to improve the utilization rate of resources on each node.
(3) When some nodes lose function, it will stop sending requests and re-enable all nodes after the function recovery.

Data sharing is mainly that the direct database of different application service can be shared, and it is not provide service for a single application service. The research achieves database sharing mainly by building a connection pool, which can well improve the ability of concurrent access[7].

Data distribution refers to that when there is a lot of application request, the point of junction will allocate the connection object according to the actual situation, thus to access the database. It mainly uses the priority allocation strategy, and even different requests of the same service also have different priorities.

The specific implementation content

To build the overall structure of the connection pool node, the overall structure of the connection pool node compete the interaction mainly through threads, realizing the whole connection pool process. The main contents are as follows:

(1) Primary thread
(2) Worker thread
(3) Monitor thread
(4) Connection management thread

The realization of the request processing model, this study adopts an asynchronous request processing model to carry out communication, and its core content has the following points:

(1) Event class
(2) Handle class
(3) Selector class
(4) Event Handler class
(5) Task class

The realization of data encapsulation and parsing has the following two functions: the first is to parse the data package send by the application program; the second is to subpackage the response packet of connection pool. That is the request packet and sending packet.

The realization of the connection object pool, its core class includes: request class, sql, operation and property.

To achieve multi priority request queue, by a basic entry AbstractQueue, and then two methods including enqueue and dequeue.

The realization of the control node, it generally has the following three functions: one is to maintain connection pool node list; and the second is to complete the load adjustment; the third is to control the management service node.

The test and analysis of the system

Generally three computers are needed to be set as A, B, C three application service programs, firstly, the priority of allocation side road test is conducted, and assuming that the priority of service program is 1, the priority of application program B is 2, the priority of application program C is 3, make three service programs continuously send queries quest to the database with the cycle for 5s. Figure 4 shows the statistical results of the proportion of application service request processing. As can be seen from the above, priority allocation method can process the request task of each application service program effectively.

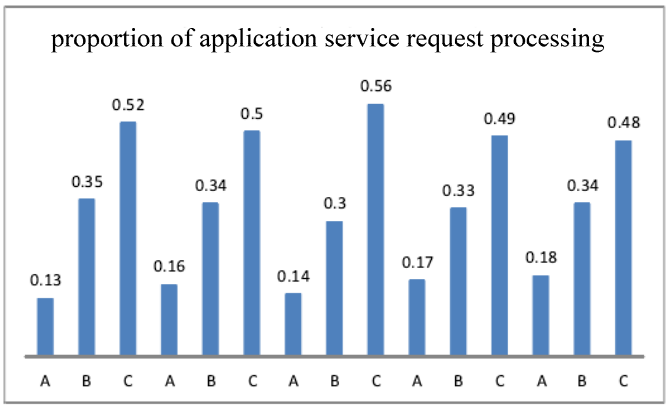


Figure 4 : The client request processing proportional graphs

Then the dynamic adjustment test of the connection pool is conducted, using the formula (1), (2), (3) to calculate, the test can be carried on after setting the parameters.

$$Server\_Load_i = E_1 * R_i + E_2 * C_i + E_3 * M_i + E_4 * P_i + E_5 * W_i + E_6 * O_i \quad \textbf{(1)}$$

$$Server\_Load_{avg} = \sum_i^n Server\_load_i / n \quad \textbf{(2)}$$

$$weight\_new_i = weight\_cur_i + A * \sqrt[3]{Server\_Load_{avg} - Server\_Load_i} \quad \textbf{(3)}$$

Figure 5 shows the change of the maximum connection in the 120s, as can be seen from the diagram, with the increasing number of database access request, the maximum connection has been adjusted upward. And with the reduction of system pressure, the maximum connection has been reduced constantly. So the connection pool dynamic adjustment strategy is very effective.

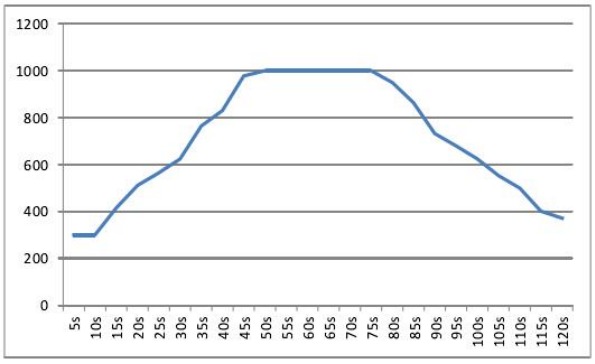


Figure 5 : The changes of the maximum connection

Next a dynamic load balancing test is conducted, and the test results are shown in Figure 6, when a node has a high load, its load will gradually be reduced. However, when a node has a lower load, its load will be gradually increased.

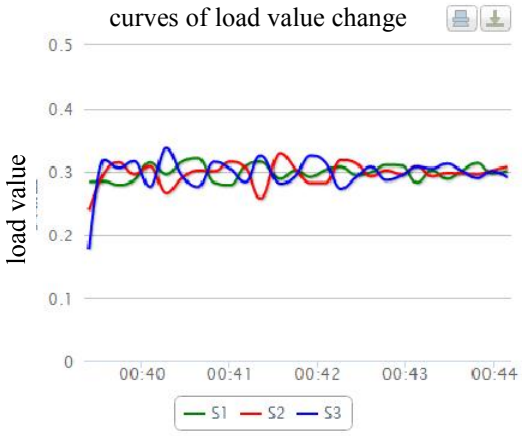


Figure 6 : The curves of load value change

CONCLUSION

The main contents of the database connection sharing technology research is mainly to solve the distribution imbalance problems when multiple applications using the database connection at the same time. The solution is to establish a more advanced connection sharing scheme using JDBC, which is based on distributed connection management system. The main contents of this scheme include strengthen dynamic load balancing and dynamic management on the node, and the allocation algorithm of the database connection and the design of connection pool need to be realized, completing a set of dynamic database connection sharing technology management scheme finally. From the overall function, it can meet the requirements. But there are still a lot of deficiencies in database connection sharing between the application programs, later

the query result of the connection pool shall be optimize, and the analysis and processing capacity of data shall be improved, and the data cache need to be designed perfect, so as to improve the overall capacity of the system.

REFERENCES

[1] Li Zhihui; Large website technology architecture: the core theory and case analysis[M] Xi'an: Electronic Industry Press, **1013**, 100-155.
[2] R.Cattell, G.Hamilton; Jdbc database access with java: A tutorial annotated reference[M], Addison-Wesley Longman Publishing Co., Inc., **(1997)**.
[3] Wang Xiuyi; Database connection- pool and its implementation based on JDBC[J], Computer Systems & Applications, **4**, 36-39 **(2005)**.
[4] Li Deshui, Yin Guofu; Study on the architecture of JDBC [J], Computer Development & Applications, **19(10)**, 55-57,60 **(2006)**.
[5] M.Apache; Apache MINA project[J], 2012-07-07.http://mina.apache,org, **(2009)**.
[6] Liu Haijuan, Su Zhao, Bao Jianguang; Research on data access middleware[J], Radio Engineering, **39(6)**, 2-14 **(2009)**.
[7] Xu Qingsong, Chen Tianhuang; Study on QoS-based database connection middleware prototype [J], Computer and Digital Engineering, **37(1)**, 57-59 **(2009)**.