A New Method in Improving Database Connection Pool Model

Guo-liang Feng, and Lian-he Yang

***Abstract*—**Through analysis to present database connection pool technology, the bottleneck of its performance infection is found. In this paper, the present database connection pool is optimized through two fashions which are automatically adjusting the configuration parameters of the connection pool and optimizing the management strategy of the connection pool. And through statistical principle the specific adjustment strategy of he two optimization fashions is presented. Finally, a group of experiments prove that the accessing response speed to database of the optimized database connection pool has improved noticeably when ever-increasing user number.

***Keywords*—**Database, Java, connection pool, XML, life cycle, B/S.

I. Introduction

IN the system of B/S structure, each dynamic page and application system must frequently access to the database, connecting database needs users validation and after using the database must be ensured that it can be correctly closed to prevent memory leakage. Therefore connecting database is a time-consuming operation. In a specific period of time, there will be a lot of database operation requests in an application website, and then system performance will be become very badly. A good solution to solve the bottleneck of connection database is to adopt the connection pool technology which can efficiently realize the management of database connection. Thus the performance optimization to the database connection pool is crucial to improve the accessing database performance.

II. Principle and Model of Database Connection Pool

A. Connection Pool Principle

Connection pool is an aggregate of connection objects which actually refers to a "buffer storage pool" with a certain amount of connection objects. The bosom of connection pool provides a kind of management mechanism which can control the number, frequency and time of connection objects in the bosom of connection pool, and provide interface of access to and release connection to application program. Basic idea of connection pool technology is to pre-establish some connection objects and place them in memory for use, when needing to establish database connection in program only to take one to use from memory without having a new, after using only need to replace them in memory, and the establishment and disconnection of linking are all managed by the connection pool itself. The workflow of connection pool is shown in Fig. 1.

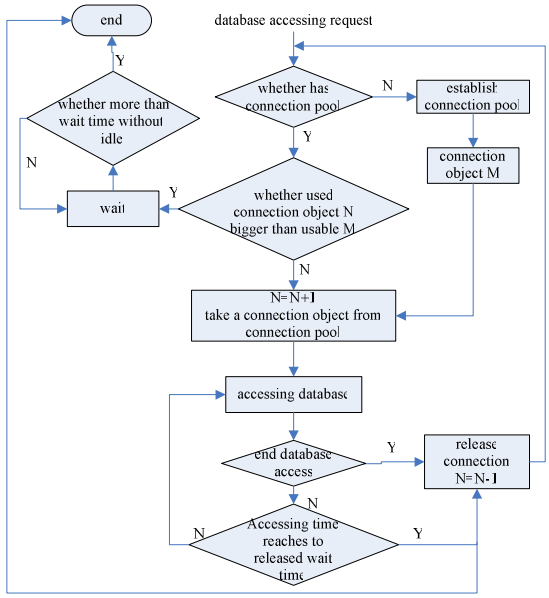

Fig. 1 Workflow of connection pool

B. Connection Pool Model

The typical model of database connection pool is shown in Fig. 2. The processing flow of database connection pool is mainly composed by the following three parts: initialization of the configuration parameters in connection pool; obtaining connection from connection pool and accessing to the database; releasing obtained connection and returning it to connection pool. The configuration parameters of centralized connection pool include: (1) Min-Connections established by connection pool, namely the idle connections maintaining dynamic connection pool; (2) Max-Connections in connection pool; (3) connections maintaining dynamic connection pool, and in which Max-Connections = Min-Connections + Connections; (4) Wait-Connection-Times without idle connection; (5) Connection-Use-Count; (6) Wait-Release-Time.

Guo-liang Feng is with School of Computer Technology and Automation of Tianjin Polytechnic University, Tianjin, China (phone:+8613522169377; e-mail: fengglying@gmail.com).

Lian-he Yang is with School of Computer and Automation of Tianjin Polytechnic University, Tianjin, China.

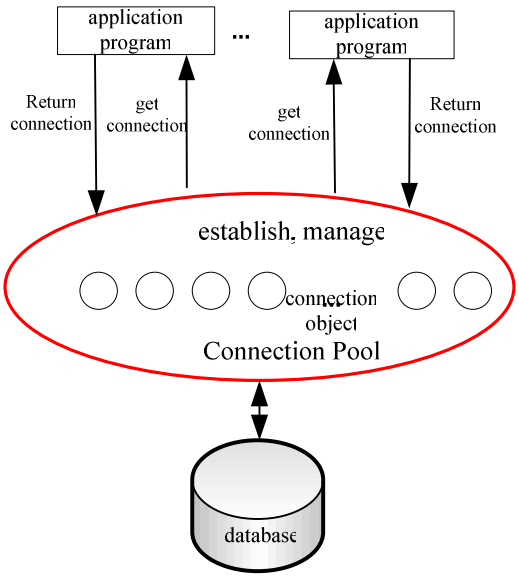

Fig. 2 Connection pool model

The obtaining and release of connection are all realized through the management program of the connection pool. Therefore, in order to optimize the performance of the database connection pool, the management of the connection pool in Fig. 1 is the focus of optimization.

III. Optimized Scheme of Performance in Database Connection Pool

According to the above-mentioned principle and model introduction of the database connection pool, now using two improvements to realize the performance improvements of the database connection pool.

A. Adaptive Adjustments of Configuration Parameters in Connection pool

Firstly establishing a XML table to store the initialization parameters of the connection pool, and complete the work of registration drive and initialization the connection pool. Its file format is:

```
<PoolPara>
 <Pool>
  <PoolType>
    sql sever 2000
  </PoolType>
  <Driver>
    com.rawstrin.jdbc.sqlserver.SQLServerDrive
  </Driver>
  <Url>
  <![CDATA[jdbc:rawstrin:sqlserver://Localhost:1433
 user=foxpc &password=coffee & useUnicode=true &
 characterEncoding=gb2312]]>
  </Url>
  <User>
    ...
  </User>
  <Password>
    ...
  </Password>
  <MaxConNum>
    30
  </MaxConNum>
  <MinConNum>
    8
  </MinConNum>
  <MaxWaitTime>
    1000
  </MaxWaitTime>
 </Pool>
<PoolPara>
```

Establishing a log file to record the initial and ending time of each link in connection pool.

```
<ConnectInfo>
 <Connection>
  <ID>
    1
  </ID>
  <StartTime>
    20070601162723
  </StartTime>
  <EndTime>
    20070601164500
  </EndTime>
 </Connection>
</ConnectInfo>
```

The material realization thinking is: when the concurrent application program requesting access to the database and in Wait-Connection-Time, if this link has not been answered, it will add 1 to the invalid connection number in this recording cycle, and when this number is greater than the pre-set maximum value the sign needing to adjust the parameters will be true, otherwise it will be the initial sign number which is false (this sign is false when initializing). After the final application program releasing connection in application, the parameter adjustment sign will be inspected: if the parameter adjustment sign is true, the parameter adjustments will be done; otherwise, maintaining the original database connections and working according to the original configuration parameters.

If the parameter adjustment sign is true after the final link circulated, then the corresponding parameter adjustment will be done. And the specific process of adjustment is: according to the failure connection number in this time of the connection begin and end time in log file to adjust. Now assuming that the failure connection number in this time is Lost-Num (i), the total lifecycle of this connection pool is Time-Run, and in the Time-Run cycle there are Lost-Total-Num time segments which have failure connection. Synchronously in the time segment of Time-Run there are Lost-Total-Num time segments which have not failure connection and assuming the Min-Connections in this time segment is Natural-Num (j). Then the formula of parameter adjustment is:

$$\text{Max-Connections}=\frac{\sum_{i=1}^{\text{Lost-Total-Num}}\text{Lost}-\text{Num}(i)}{\text{Lost}-\text{Total}-\text{Num}} \quad (1)$$

$$\text{Min-Connections}=\frac{\sum_{j=1}^{\text{Natural-Total-Num}}\text{Natual}-\text{Num}(j)}{\text{Natural}-\text{Total}-\text{Num}} \quad (2)$$

The above formula is a weighted statistical formula and it can primely find a best value of the Max-Connections and Min-Connections to achieve the best operation configuration parameters of the whole system.

B. Management Strategy Optimization of Connection Pool

Traditional connection pool management flow is shown in Fig. 3.

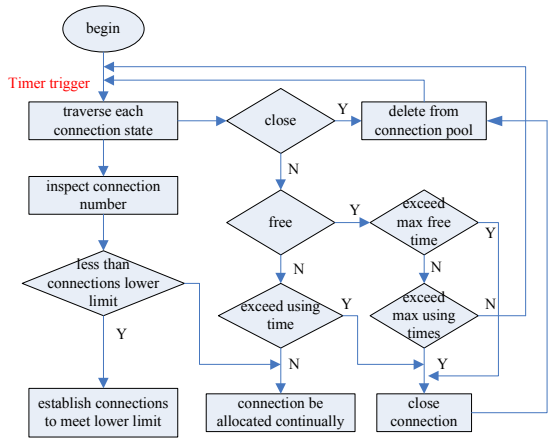

Fig. 3 Traditional connection pool management flow

From Fig. 3 it can be seen that the above management flow triggers for one time to inspect in a Timer interval. When there are multiple connections in the connections pool, such frequent inspects certainly will consume excessive system resources, thus it may be considered that inspecting for one time after a reasonable interval to reduce unnecessary system spending. Therefore, how to choose suitable interval becomes key of optimizing the management strategy of the connection pool. Hereon the interval will be chosen through the following fashions. The specific realizing process is:

(1) Figure out the duration of each connection through established log file, assuming that the duration of connection i is Persistence-Time (i);

(2) The connecting times of connection i in this connection pool lifecycle is Used-Connect-Num (i);

Assuming that there are M connections in connection pool, then the calculating formula of reasonable interval is:

$$\text{Time}-\text{Alternation}=\sum_{i=1}^{M}\text{Persistence}-\text{Time}(i)\times\frac{\text{Used}-\text{Connect\_Num}}{M} \quad (3)$$

The interval of traversing inspection in connection pool management process is carried through in the time of parameters adjustment, its initial value can be a reasonable default value.

IV. Performance Test and Result Analysis

A. Testing Circumstance

Database server configuration: CPU: PIV 2.4G; HD: 120G; EMS memory: 1G; operating system: windows2000 Advanced Server; program language: Java.

B. Testing Result and Analysis

In JSP pages, separately transferring the interface object of present database connection pool and the optimized connection pool realized in this paper to access to the same database information and dynamically generate Web pages. Using Web application performance testing tool OpenSTA to test overall response delay of the system and the testing result is shown in Table I.

TABLE I
Delay Time Comparative Table of Different Connection Pools

	500 online users	800 online users	1200 online users
present database connection pool	10281 ms	38761 ms	137623 ms
optimized database connection pool	4019 ms	14970 ms	64611 ms
time ratio	2.56	2.59	2.13

From Table I it can be seen: when continually increasing user number, the response time of the system which applied optimized connection pool to the database obviously faster than the system using present connection pool.

V. Conclusion

In this paper a kind of database connection pool scheme which can improve the processing capacity of subsequent connection is designed. This scheme generates log file which records connection information through readining the initialization value of the configuration file in the specific using process of the connection pool, and analyzes the log file to obtain better configuration parameters and write them into pool configuration file. Using such fashion can read more reasonable configuration parameters at each startup of the connection pool to avoid artificial setting of the connection pool parameters. Through comparison with present connection pool, it can improve the capacity of subsequent response to database access, then the database connection resources in application system can be more effectively managed.

References

[1] Siva Visveswaran, Dive into connection pooling with J2EE. http://java.sun.com/developer/technicalArticles/J2EE/pooling/, October, 2000
[2] Oscar Sànchezi Vilar. Automatic Web Publishing [D]. Degree of Master of Science in Networks and Distributed Systems in University of Dublin, 2003. P34-35
[3] http://java.sun.com/j2se/1.4/docs/guide/jdbc/index.html
[4] http://otn.oracle.com/products/content.html

[5] Sun Microsystems. The JDBC API Specification [EB/OL]. http://java.sun.com/j2se/1.3/does/guide/jdbc/index.html, 2000-03-03.
[6] Chen.SLu, F.Y, Web-based simulations of power systems [J]. IEEE Computer Applications in Power, 2002, 15(1), P35–40.
[7] Stanford Compiler Group. SUIF Compiler System Version1.0 [Z]. US: Standford University, 1994
[8] Sun Microsystems, Java2 Platform Standard Ed. 5.0[M]. .NewYork: Sun Microsystems, 2004, P55-180.
[9] Carla Sadtler, Web Sphere Application Server-Express V5.0.2 Administrator Handbook [M]. New York: IBM Corp International Technical Support Organization, 2003, P82-96.

Guo-liang Feng was born in Shandong Province, P.R China, in 1980. He received Bachelor degree in Computer application technology in ShanDong University of Technology in China. Currently, he is studying toward master's degree in Computer application technology in Tianjin Polytechnic University in China. His interest is in database.

Lian-he Yang is a professor of School of Computer Technology and Automation of Tianjin Polytechnic University. His research is Computer simulation and database (phone: +86-13522169377,022-24584581; e-mail: fengglying@gmail.com).