Performance evaluation between Oracle JDBC drivers

Young-Ho Shin*, Sung-Geun Han**, Nam-Kyu Park***, Joo-Yeong Kim****, Gyu-Seok Kim*****
Korea Institute of Science and Technology Information, Korea
E-mail : shinyh@kisti.re.kr*, sghan@kisti.re.kr**, namkyu.park@kisti.re.kr***,
jyghim@kisti.re.kr****, gskim@kisti.re.kr*****

1. Introduction

These days, Relational database has been widely used for various purposes such as OLTP, DataWareHouse etc. Also, Java is the most popular and highly used programming language which has many advantages of platform-independent, multi-threading, and so on. A JDBC driver is a software component enabling a Java application to interact with a database. In this paper, We have conducted performance evaluation of Oracle JDBC drivers that are widely used from relational database in research and industry. From now on, The Performance evaluation result can be used as fundamental data on service qualities and compatibilities.

2. JDBC driver types

JDBC(Java Database Connectivity) defines how a Java program can communicate with a database, and it provides standard API(Application Programming Interface) for developers. Using JDBC, a Java developer is able to create database applications that are independent of specific DBMS[1]. Through this JDBC service of a consistent API, We could connect, search, modify and manage data of database. JDBC driver has four types of connection and access method, and the characteristics of respective types are as follows.[2]

2.1 Type 1 : JDBC-ODBC Bridge plus ODBC Driver (JDBC-ODBC), that calls native code of the locally available ODBC driver.

2.2 Type 2 : A native API partly Java technology-enabled driver, that calls database vendor native library on a client side. This code then talks to database over network. (corresponds to Oracle OCI(Oracle Call Interface))

2.3 Type 3 : Pure Java Driver for Database Middleware, that pure-java driver that talks with the server-side middleware that then talks to database.

2.4 Type 4 : Direct-to-Database Pure Java Driver, that pure-java driver that uses database native protocol. (corresponds to Oracle thin), Most database have this driver. In fact, This type is the most commonly used method in JDBC, easy and simple.

3. Experimental Environments, Background and Methods

3.1. Experimental Environments

We ran the performance test on the system specification which described below :

- PC(2.6GHz, 2CPU(8core), Memory 8GB) : Windows7, Apache-JMeter-2.11
- Web Server ((2.6GHz, 1CPU, Memory 1GB) : RHEL AS6, Apache-2.2
- WAS Server (2.6GHz, 1CPU, Memory 1GB) : RHEL AS6, Tomcat5.5.12, Tomcat6.0.39, Tomcat 7.0.52
- DB Server (3.0GHZ, 2CPU(8Core), Memory 32GB) : RHEL AS6, Oracle 11gR2

The system architecture is a common three-tier structure. Currently, the most commonly used version of Oracle database is Oracle10g, Oracle11g and Oracle12. Thus, as the driver for these DBMS versions ojdbc14.jar, ojdbc5.jar, ojdbc6.jar, and ojdbc7.jar was installed in the WAS server. Additionally, Tomcat engine for Oracle JDBC drivers was installed in the WAS server. The main features of these drivers are as follows:

- ojdbc14.jar – classes for user with JDK 1.4 and 1.5
- ojdbc5.jar – classes for use with JDK 1.5. It contains the JDBC driver classes, except classes for NLS support in Oracle Object and Collection types.
- ojdbc6.jar – classes for use with JDK 1.6. It contains the JDBC driver classes except classes for NLS support in Oracle Object and Collection types.
- ojdbc7.jar – For use with JDK 7; It contains the JDBC driver classes except classes for NLS support in Oracle Object and Collection types.

3.2. Experimental Background and Methods

Oracle database 10i and 11g JDBC drivers intentionally trade off large memory use for improved performance, but in Oracle database 12c the driver's memory management has been designed to minimize memory use[3]. Oracle JDBC drivers consistently have been developed to maximize the performance for use. The background of this performance test have resulted from characteristics of Oracle JDBC drivers. The Web program for the test using DBCP(Database Connection Pool) and JDBC by each Oracle JDBC driver was prepared respectively. The Web program is showing results that correspond to the search condition and the Web application, the search results show 30 rows per page. For each driver, two types (OCI of type2 and thin of type4) of performance testing proceeds, and the results are presented. As the tool for testing, Apache JMeter was used. The Apache JMeter is a popular open source software for load testing with many useful modeling features such as thread group, HTTP request, and Summary Report elements[4]. For the test, A thread group element contains the following properties and values:

- Number of threads (users) : 100
- Ramp-up period (in seconds) : 0
- Loop Count (the number of times to execute the test) : 10

It means that a hundred of users access to the web page ten times simultaneously. There are many samplers available we need HTTP request sampler. The summary report shows values about the measurement Apache JMeter has done while calling the web page. The throughput is the number of requests per unit of time that are sent to the server during test. Through the summary report, we can provide performance information.

4. Experimental results and Evaluation

As shown below figure, they show no regularities. In the case of DBCP, in accordance with the version of Oracle JDBC driver is very little difference. But, in the case of JDBC, ojdbc14.jar and ojdbc5.jar are compared with ojdbc6.jar and ojdbc7.jar showed a significant performance difference.

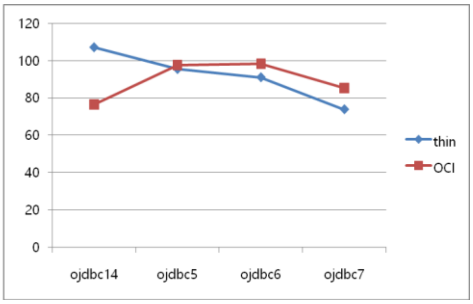

Figure 1. Throughput in DBCP

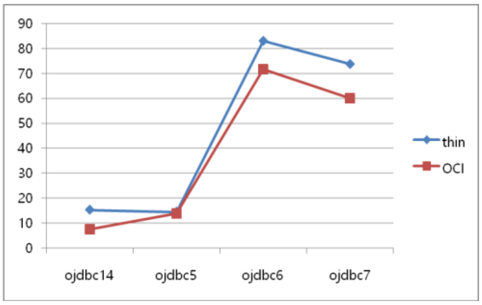

Figure 2. Throughput in JDBC

In the case of DBCP, OCI scheme has slightly better performance. However, in the case of JDBC, thin scheme is more excellent way. Performance of DBCP method was better than Performance of JDBC. In the case of latest drivers, The performance between DBCP and JDBC method did not show any significant difference.

5. Conclusion

We have run the performance evaluation test of various connection and access methods of Oracle JDBC drivers. Actually there are some limitations of this test that has only a select query, But it is certain that this performance test is meaningful enough in an OLTP(On-Line Transaction Processing) database. By this experiment, It shows that there are some differences between JDBC drivers and access types. The performance evaluation results are expected to use as a reference and basis in a development task.

6. References

[1] Julia Gutjahr, Andreas Loew, “Scalability and Performance: JDBC Best Practices and Pitfalls”, Sun Microsystems GmbH, Sun Java Center, Ampèrestraße 6, D-63225 Langen, Germany, Feb 12, 2002

[2] http://www.oracle.com/technetwork/database /features/jdbc/

[3] Douglas Surber, “Oracle JDBC Memory Management”, Oracle Corporation, World Headquarters 500 Oracle Parkway Redwood Shores, CA 94065 U.S.A , August 2009

[4] http://jmeter.apache.org/index.html