International Journal of Computer Science
and Engineering (IJCSE)
ISSN: 2278-9960
Vol. 2, Issue 4, Sep 2013, 191-194
© IASET

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

JAVA STORED PROCEDURE AND USER DEFINE FUNCTION ON DB2 FOR Z/OS

SAGAR GUPTA[1] & SRINIVAS GOPALA KRISHNA[2]

[1]Systems Engineer, Infosys Limited, Bangalore, Karnataka, India

[2]Principal Technology Specialist, Infosys Limited, Bangalore, Karnataka, India

ABSTRACT

Stored procedures are an efficient way to write applications as they reduce Network Traffic, make better use of server memory, and ease of management of code. And Java allows us to design Stored Procedures as well as UDFs which provides additional functionality in terms of data handling and manipulation. This paper discusses on ways to develop Java SP and UDFs using Mainframe DB2 as Database Server. Also, discusses about the customization of the connectivity to leverage strengths and facilities of Mainframe.

This document draws from our experiences while implementing SP and UDF using Java on DB2.

KEYWORDS: JDBC, WLM, RAS, COBOL, JAVA, UDF, DB2

INTRODUCTION

In current IT scenarios database integration is a common feature for application designing. Database connectivity involves various factors like data-transmission ratio, bandwidth usage along with basic factors of data layer designing in optimized manner.

Executing Ad-hoc queries directly from distributed environment on database using multiple thread approach will result in excessive bandwidth usage as well as also gives performance impact on overall application throughput. Programming routines such as stored procedures or user defined functions are majorly used for data layer programming to achieve low data transmission ratio and high optimization, it also give feature for designing robust distributed application.

Stored Procedures

Stored procedures are user-written programs which can be stored on databases to perform tasks which are not supported by T-SQL queries. Stored procedures are written in a host language which a database can understand. Stored procedure can be written in two ways, one is by using PL/SQL language which supports to process algorithms designing using modular programming or by using high level programming languages which can be used to process data in a way which can be understood by the database.

In case of DB2 stored procedures can be written in C, C++, COBOL, REXX or JAVA which are called as external stored procedures. Java can be used for designing stored procedures and UDFs (User Defined Functions) which give more functionality to programmer in terms of data handling and manipulation while writing T-SQL queries which are not supported by DB2 natively.

MATERIALS AND METHODS

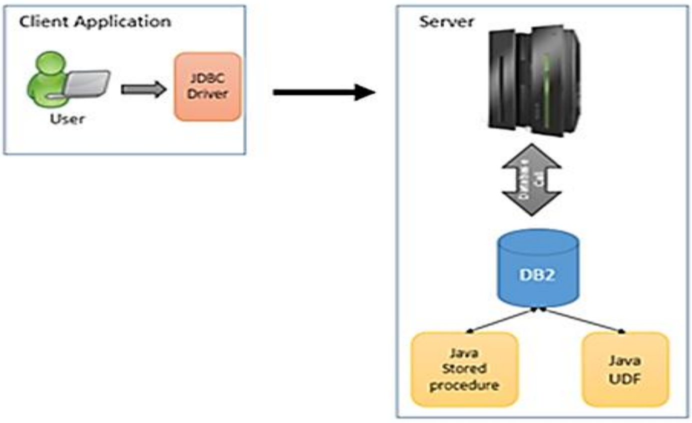

Figure 1: Client-Server Communication

Client Application: Client invoking JAVA stored procedure or JAVA UDF will use the corresponding JDBC/ODBC driver based on the platform. The JDBC/ODBC drivers are responsible for converting the database routine invocation into low level database calls to the server, and converting the data retrieved from the server into corresponding data types supported by that platform.

Database Server: Server will receive the calls made my client application and calls the relevant instance of DB2 running on the server to complete the request. DB2 will internally follows below steps for data retrieval.

- **Syntax and Semantic Checking:** The query will be first checked for correct syntax based on standards set for various DML/DDL calls and if the syntax checking fails an error will be returned to the invoking application. Once the syntax checking completes semantic checking will get started which verifies the reference to the database objects made by the call and the host data type comparison with database data types, if any error occurs during this processing the client application will get notified with corresponding error message.
- **Executes the DML/DDL Statement:** Once the syntax and semantic check succeeds, the call will be made to stored procedure or UDF based on the request and the inputs will be passed to the routines for processing. Once the execution completes based on the SQLCODE either the data or the error code will be returned to the client application.

DISCUSSIONS

Java Stored Procedure

Java stored procedure on DB2 Z/OS can be designed in two flavors: compiled Java stored procedures and interpreted Java stored procedures, till DB2 V7 both compiled and interpreted Java stored procedures are accepted but from DB2 V8 onwards only interpreted Java stored procedures can be used. Interpreted Java stored procedures can be developed either by using SQLJ or by using JDBC. Prerequisites software for Java stored procedure:

- DB2 for z/OS at Version 7.
- OS/390 Version 2 Release 8.
- IBM Developer Kit for OS/390, Java 2 Technology Edition, SDK 1.3.1 or 1.4.1 level (5655-A46).

After setting up the Java environment in DB2, Java stored procedure can be created using CREATE PROCEDURE statement with required parameters like EXTERNAL NAME, WLM ENVIRONMENT, LANGUAGE (by default JAVA), PARAMETER STYLE (by default JAVA), STAY RESIDENT YES or STAY RESIDENT No etc. While creating JAVA stored procedures points to be remembered for giving external name:

- **When packages and jars are not used:** <Class-Name>.<Method-Name>
- **When packages are used:<**package-name>.<Class-name>.<method-name>
- **When jars are used:**
- **When jars not defined to DB2:** <package-name>.<Class-name>.<method-name> and .jar file path should be present in CLASSPATH variable.
- **When jars are defined to DB2:** .jar file should be registered to DB2 using IBM supplied stored procedure SQLJ.INSTALL_JAR and entry of .jar file should be removed from CLASSPATH, after that the external name will be: <RACF-ID>.<name of registered jar>:<package-name>.<class-name>.<method-name>.
- If the DB2 connection is created then only Type-2 JDBC driver should be used whose path should be mentioned in DB2_HOME environment variable in WLM job. To use Type-2 driver SQL connection can should be instantiated using: jdbc:default:connection connection string .
- If a result-set needs to be returned then an array of ResultSet class needs to be created.

While creating definition program for the stored procedure points to be remembered are:

- All the input parameters should be declared as native data-types like for VARCHAR and CHAR, String should be used.
- All the output parameters should be declared as arrays of corresponding compatible data-types as because DB2 passes INOUT and OUT parameters as single-entry arrays like for VARCHAR it should be String[].

Once the stored procedure JAVA class is create, the compiled class needs to be put under one of the directories mentioned in CLASSPATH variable of WLM job. After finishing compilation the WLM region job needs to be refreshed in DB2 so that the new class definition can be loaded in DB2.

JAVA UDF (User Defined Function)

User defined function can be created in DB2 using CREATE FUNCTION statement and rest other parameters remains almost same as mentioned in stored procedure case, one special parameter which can be mentioned in case of UDF is FENCED or NOT FENCED. In case of FENCED the UDF runs in special DB2 process outside the DB2 engine address space while NOT FENCED UDFs will run in same address space giving better performance, but if an abnormal termination happens with NOT FENCED UDFs (registered as NO SQL), DB2 engine will fails. But for FENCED UDFs if an abnormal termination happens DB2 will terminate the thread in which the UDF was running.

BENEFITS

- JAVA complied code can be transferred from one machine to another without dealing with complexities of machine level code conversion as JAVA runs on separate address space (JVM) which is common across all platforms.
- Data conversion from client-end to server-end and vice versa can be avoided.

- JAVA provides advanced features for handling data types like String, Integer etc. natively, these features can be used to build more robust routines in DB2.
- Template programming is supported in JAVA using which the independency from input and output data types.
- JAVA supports package level design approach which gives more modularity to application designing.
- JAVA supports unit testing frameworks where DB2 routine can be checked with various test cases.
- Platform independency for DB2 routine invocation from client machine as JDBC connectivity can be used standard connectivity medium.
- JAVA routines are easier to design and debug.

CONCLUSIONS

The application using Java Stored Procedures and User Defined Functions give better performance and throughput than applications executing queries directly from the distributed environment.

REFERENCES

1. Redbook: Java Stand-alone Applications on z/OS Volume II.
2. Redbook: SMP/E for z/OS V3R5.0 User's Guide z/OS V1R10.0-V1R11.0 SA22-7773-14
3. Redbook: DB2 Version 9.1 for z/OS Application Programming Guide and Reference for Java - SC18-9842-11