Configure parameters for IBM Data Server Driver for JDBC and SQLJ against HADR for DB2 pureScale

Sujan Ghosh (sghosh.raj@in.ibm.com)
Senior Software Engineer
IBM

Vinayak Joshi (vijoshi5@in.ibm.com)
Senior Software Engineer
IBM

04 September 2015

IBM High Availability Disaster Recovery (HADR) for DB2® pureScale® (on DB2 for Linux, UNIX, and Windows) offers functionality high availability and scalability through IBM Data Server Driver for JDBC and SQLJ. Mission-critical applications that lack end-to-end optimal configurations don't failover/failback for transactions. This tutorial explains configuration parameters for IBM Data Server Driver for JDBC and SQLJ against HADR for DB2 pureScale, such as covering different scenarios/criteria in deciding values of configuration parameters suited as per application/environment needs.

Introduction

DB2 pureScale (for Linux, UNIX and Windows) is a shared disk architecture clustering technology, where multiple DB2 members process database requests against a single shared copy of the database. HADR for DB2 pureScale provides the functionality of failover (continuous availability) and disaster recovery with a copy of data in a standby member, respectively. We will go over the fundamentals of configurations/setup/environment most optimally from a client application standpoint.

Note: HADR for DB2 pureScale is supported from DB2 10.5 onward. The copying of data between the primary and standby database happens via transaction of log streams. The standby database replays the logs to sync the data.

The following image depicts HADR for DB2 pureScale connectivity between client applications and a data sharing group with HADR configuration. The clients could be stand-alone applications or middleware, such as WebSphere® Application Server, that utilize connectivity offered by the IBM Data Server Driver for JDBC and SQLJ to connect to a DB2 data sharing group. The primary database and standby cluster with three members each can be at different locations. HADR uses
TCP/IP communication for shipping transaction logs, and the preferred replay member in the standby cluster receives and replays the logs.

Figure 1. HADR for DB2 pureScale connectivity

Workload balancing with pureScale

Applications running in a cluster environment like DB2 for pureScale or Sysplex (DB2 for z/OS® data server) use the functionality of workload balancing (WLB) to distribute the incoming request for a new transaction or a new incoming connection from the application to each DB2 member of cluster at transaction level and connection level. The distribution of workload to each DB2 member is determined by the target data server cluster. The distribution will happen as per capacity of each DB2 member of the cluster mentioned in the server list returned by the cluster. The server list is the base for workload balancing and ACR operations by the IBM Data Server Driver for JDBC and SQLJ. The server list contains the information for the DB2 member's availability and capacity to handle the workload.

The IBM Data Server Driver for JDBC and SQLJ only supports transaction-level WLB, whereas transaction-level balancing encompasses connection-level balancing automatically.

WLB can be used with the pureScale cluster using IBM Data Server Driver for JDBC and SQLJ property enableSysplexWLB=true.

Alternate group support: Driver perspective

Failover using IBM Data Server Driver for JDBC and SQLJ along with ACR/WLB happens at the group level, where the primary and the alternate data servers form a cluster/group. A cluster or group consists of multiple members. A failover can also occur across a group or within a group. HADR support for DB2 pureScale with an alternate group starts when all members of the group go down and the transaction will failover to the alternate group. When one member of a group goes down and the others are available, transaction processing will continue on the remaining members of the same group.
In HADR for pureScale, alternate group support in IBM Data Server Driver for JDBC and SQLJ can be achieved by using `alternateGroupServerName`, `alternateGroupPortNumber`, and `alternateGroupDatabaseName` at the connection URL or data-source level.

These properties can be set at the connection URL, data source-level, properties file, custom properties of JNDI in WebSphere Application Server, etc.

At the connection URL:

- String url = "jdbc:db2://jaxxx.svl.ibm.com:50000/
databaseName:user=user;password=passwd;
alternateGroupDatabaseName=databaseName;alternateGroupPortNumber=50000;
alternateGroupServerName=jaxxx1.svl.ibm.com;enableAlternateGroupSeamlessACR=true;";

At the data-source level:

- com.ibm.db2.jcc.DB2SimpleDataSource ds__ = new DB2SimpleDataSource();
  ds__.setAlternateGroupDatabaseName("databaseName");
  ds__.setAlternateGroupPortNumber("50000");
  ds__.setAlternateGroupServerName("jaxxx1.svl.ibm.com");
  ds__.setEnableAlternateGroupSeamlessACR(true);

Add custom properties using WebSphere Application Server:

- The following steps and figures illustrate how to configure custom properties using IBM Data Server Driver for JDBC and SQLJ data-source properties, which are not present in the admin console by default.

**Figure 2. Selecting the DB2 connect database**

In the navigation window on the left under Resources, click **JDBC > Data sources**, then select the **JNDI resource / data source** to which the web application wants to connect and execute.
Configure parameters for IBM Data Server Driver for JDBC and SQLJ against HADR for DB2 pureScale

Figure 3. Selecting custom properties for chosen database

Click the selected JNDI/data source on the right side under Additional properties and click custom properties.

Figure 4. Providing name value inputs for new property

Add a property by clicking on the New tab and providing the name value inputs for the new property, then save.
The IBM Data Server Driver for JDBC and SQLJ makes it easy to do a seamless group failover in HADR for pureScale environment by using enableAlternateGroupSeamlessACR. The enableAlternateGroupSeamlessACR indicates whether the seamless failover behavior across the groups is enabled. Seamless failover behavior across groups is available only if the alternate group is specified.

**Configuration of HADR for DB2 pureScale using IBM Data Server Driver for JDBC and SQLJ**

Failover and disaster recovery in HADR for DB2 pureScale can be achieved using IBM Data Server Driver for JDBC and SQLJ as follows:

1. Classic HADR automatic client reroute (ACR) setup using WLB
2. Alternate group support

WLB with HADR for DB2 pureScale can be achieved using the IBM Data Server Driver for JDBC and SQLJ property enableSysplexWLB=true. If WLB is enabled, ACR will also be enabled implicitly.

**Classic HADR ACR for DB2 pureScale with WLB**

Automatic client reroute with HADR for DB2 pureScale also provides the same functionality of failover (continuous availability) and disaster recovery. Automatic client reroute, which can also be referred to as classic configuration allows the connections to failover to an alternate data server if there is an outage to the primary data server. The failed transactions need to be handled and resubmitted after failover at the application, when the classic configuration enabled in client applications non-seamless transactions failover throws error code -4498 or -30108 to the application.

The IBM Data Server Driver for JDBC and SQLJ periodically receives the server list from a data sharing group. Each member of the group maintains a copy of the server list, which is available to IBM Data Server Driver for JDBC and SQLJ. When attempting to acquire a new connection, or begin a new transaction on an existing connection, the IBM Data Server Driver for JDBC and SQLJ determines the best member using WLB algorithm to route the request to and from the server list. When there is any outage of any member, the IBM Data Server Driver for JDBC and SQLJ will try each member of the primary cluster received from the server list. If there is no member available in the primary cluster, IBM Data Server Driver for JDBC and SQLJ will try failover to an alternate server. If it's still not successful connecting after trying all the alternate servers, IBM Data Server Driver for JDBC and SQLJ will loop back to the primary server and repeat for every 10-minute increment.

In the classic configuration, the client can choose to configure an alternate server, so failover to the alternate server can take place even if no connection to the primary (which returns an alternate server list) has succeeded. The SQL command `UPDATE ALTERNATE SERVER FOR DATABASE` is used to locate the alternate server in the primary and secondary cluster. That is, the primary cluster's alternate server would be the secondary cluster, and the secondary's cluster alternate server would be the primary cluster. This is known as Classic HADR ACR setup for DB2 pureScale.
HADR setup for DB2 pureScale enables ACR by default. HADR for DB2 pureScale setup consists of the primary and secondary cluster, when a client gets a connection to the primary cluster, the alternate server list is loaded into client cache. The server/alternate server list would contain primary cluster's members and secondary cluster's information.

**Configuration of classic HADR ACR setup for DB2 pureScale**

Using IBM Data Server Driver for JDBC and SQLJ Classic HADR ACR for DB2 pureScale can be configured as:

1. Using `clientRerouteAlternateServerName` and `clientRerouteAlternatePortNumber` for alternate cluster along with details of server name and port number of the primary cluster
2. Using optional `maxRetriesForClientReroute` to traverse each member of server list the number of times
3. Using optional `retryIntervalForClientReroute` to wait for the amount of time between server list passes

The properties `clientRerouteAlternateServerName` and `clientRerouteAlternatePortNumber` for alternate cluster at application level will be ignored by IBM Data Server Driver for JDBC and SQLJ if the very first connection to the primary cluster is successful. If unsuccessful, alternate cluster information mentioned in `clientRerouteAlternateServerName` will be used for the first connection.

**Recommendation:** Use the Classic HADR ACR setup with WLB for DB2 pureScale when customers want WLB with failover to the alternate cluster. If the failover is not successful within the cluster, failover will occur across the clusters with workload balancing.

**Note:**

1. When the target server is not configured with HADR for DB2 pureScale setup and the client wants alternate group support, it's recommended to use client-side alternate group properties `alternateGroupServerName`, `alternateGroupPortNumber`, and `alternateGroupDatabaseName`.
2. When the target server is configured with HADR for DB2 pureScale setup, clients can ignore properties for alternate group support. Server configuration HADR for DB2 pureScale will take care of the failover to the alternate group. For Classic HADR ACR, it's always recommended to use client-side properties `clientRerouteAlternateServerName` and `clientRerouteAlternatePortNumber`.

The sample code in Listing 1 shows how an application may be programmed for Classic HADR ACR setup for DB2 pureScale during a transaction. The properties are configured in the property file without modification of application. These properties can be set at the connection URL, data source-level, properties file, custom properties of JNDI in WebSphere Application Server, etc.

The following code listing shows regular JDBC code that applications typically use for the classic ACR with HADR for DB2 pureScale for setting properties on property file, loading the file, getting a connection, and starting a transaction. Set the property file in this way to create a file with any name like `jcc.properties` and add the following properties with key value pairs:
enableSysplexWLB=true
clientRerouteAlternateServerName=jafvt2xxx.svl.ibm.com
clientRerouteAlternatePortNumber=50000
maxRetriesForClientReroute=2
retryIntervalForClientReroute=1

jafvt1xxx.svl.ibm.com is the primary cluster with two members and jafvt2xxx.svl.ibm.com is the secondary cluster with two members.

*/ // Load the driver class
Class.forName("com.ibm.db2.jcc.DB2Driver");
// Create DB2SimpleDataSource object
com.ibm.db2.jcc.DB2SimpleDataSource ds__ = new
DB2SimpleDataSource();
// Create statement, PreparedStatement, property utility and InputStream objects
java.sql.Statement stmt = null;
java.sql.PreparedStatement pstmt = null;
java.util.Properties prop = null;
java.io.InputStream input = null;
// Connection URL with primary cluster
String url = "jdbc:db2://jafvt1xxx.svl.ibm.com:50000/
SAMPLE:user=user;password=passwd;";
// Load the properties file
try {
    input = new FileInputStream("jcc.properties");
} catch (IOException ex) {
    ex.printStackTrace();
}
// Get a connection using the property utility
java.sql.Connection con = java.sql.DriverManager.getConnection(url, prop);
// execute transaction SQLs
stmt = con.createStatement();
stmt.executeUpdate("sql statement1");
stmt.close();
pstmt = con.prepareStatement("sql statement2");
pstmt.setString(1, "..");
pstmt.executeUpdate();
pstmt.close();
}

Configuration of HADR for DB2 pureScale without WLB

Assuming the setup of HADR for DB2 pureScale consists of two clusters (primary and secondary), each cluster with two members, when the client requests a new connection for JDBC, SQLJ chooses one of the DB2 members to route the request to and from the server list. When there is an outage of any of the members, the IBM Data Server Driver for JDBC and SQLJ will try each member of the primary cluster received from the server list. If there is no member available in the primary cluster, IBM Data Server Driver for JDBC and SQLJ will try failover to alternate group members. If it is still not successful getting a connection after trying all the alternate groups, IBM Data Server Driver for JDBC and SQLJ will loop back to the primary group and repeats for the 10 minute duration.

Using IBM Data Server Driver for JDBC and SQLJ HADR for DB2 pureScale can be configured as:

1. Using alternateGroupName\name, alternateGroupPortNumber and
   alternateGroupDatabaseName for alternate group cluster specifications
2. Using enableAlternateGroupSeamlessACR to enable seamless transaction across the alternate clusters
3. Using optional maxRetriesForClientReroute to traverse each member of the server list the
given number of times
4. Using optional retryIntervalForClientReroute to wait for the amount of time between server
   lists passes

HADR for DB2 pureScale with alternate group support is recommended when IBM Data Server Driver for JDBC and SQLJ wants the failover within cluster. If unsuccessful, the failover happens outside the cluster group.
The code below shows how an application may be programmed for alternate group support using IBM Data Server Driver for JDBC and SQLJ against HADR for DB2 pureScale (with two clusters). The properties can be set using the property file without modifying the application code. The property file can be loaded using the load function of utility class java.util.Properties.

The sample JDBC code uses the property file for alternate group support on HADR for DB2 pureScale. The following code shows sample code with two clusters jafvt1xxx and jafvt2xxx setting properties for alternate group support on property file, load the file, get a connection and start a transaction. In this code snippet, jafvt1xxx is a primary cluster and jafvt2xxx is the alternate cluster. Set the property file in this way by creating a file with any name like jcc.properties. Add the following properties with key value pairs:

```java
   enableAlternateGroupSeamlessACR=true
   alternateGroupServerName=jafvt2xxx.svl.ibm.com
   alternateGroupPortNumber=50000
   maxRetriesForClientReroute=2
   retryIntervalForClientReroute=1

   // Load the driver class
   Class.forName("com.ibm.db2.jcc.DB2Driver");
   // Create DB2SimpleDataSource object
   com.ibm.db2.jcc.DB2SimpleDataSource ds__ = new DB2SimpleDataSource();
   // Create statement, PreparedStatement, property utility and InputStream objects
   java.sql.Statement stmt = null;
   java.sql.PreparedStatement pstmt = null;
   java.util.Properties prop = null;
   java.io.InputStream input = null;

   // Connection URL with primary cluster
   String url = "jdbc:db2://jafvt1xxx.svl.ibm.com:50000/SAMPLE:user=user;password=passwd;";
   // Load the properties file
   try {
     input = new FileInputStream("jcc.properties");
     prop.load(input);
   } catch (IOException ex) { ex.printStackTrace(); }
   // Get a connection
   java.sql.Connection con = java.sql.DriverManager.getConnection(url, prop);
   // execute transaction SQLs
   stmt.executeUpdate("sql statement1");
   pstmt = con.prepareStatement("sql statement2");
   pstmt.setString(1, ".");
   pstmt.executeUpdate();
   pstmt.close();
   con.close();
```

**Conclusion**

The recommendation is to use the classic HADR ACR setup with WLB enabled for DB2 pureScale when the customer wants WLB and failover to alternate clusters. This would be applicable where the customers want the failover to happen within the cluster with workload balancing. If the failover is not successful within the cluster, failover will happen across clusters with WLB. If the driver is at the last alternate group, it should cycle back to the head of the alternate group list.

With IBM Data Server Driver for JDBC and SQLJ, it is easy to develop applications that derive the full benefit of HADR for DB2 pureScale data-sharing technologies from IBM. WLB needs to turn on only properties in the driver, thereby extending the scalability and failure resilience of the applications. With this, we’ve covered the technical details, driver properties, and programming recommendations related to WLB with the classic HADR ACR support.
Resources

- For more information about HADR for DB2 pureScale, see [Set up and manage disaster recovery for pureScale using HADR](#).
- For more information about setup of HADR for DB2 pureScale with alternate group support, see [Operation of alternate group support for connections to DB2 for Linux, UNIX, and Windows](#).
- The [Information Management area on developerWorks](#) provides resources for architects, developers, and engineers.
- Stay current with [developer technical events and webcasts](#) focused on a variety of IBM products and IT industry topics.
- Follow [developerWorks on Twitter](#).
- Watch [developerWorks demos](#) ranging from product installation and setup demos for beginners, to advanced functionality for experienced developers.
- Get involved in the [developerWorks Community](#). Connect with other developerWorks users while you explore developer-driven blogs, forums, groups, and wikis.
About the authors

Sujan Ghosh

Sujan Ghosh has been a software engineer at the IBM lab in India for more than four years as a member of the DB2 Java common connectivity and pureQuery client optimization team. Most of his projects have been based on DB2 and Java technology. He is an IBM Certified Database Administrator for DB2 for Linux, UNIX, and Windows, and an IBM Certified Solution Developer for XML.

Vinayak Joshi

Vinayak Joshi is a senior software engineer at IBM with more than 10 years of experience with various IBM technologies. For the past three years, he has been part of the IBM Data Server Driver for JDBC and SQLJ development team, focusing on the workload balancing and automatic client reroute support in the driver.

© Copyright IBM Corporation 2015
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)