Perform scalable data exchange using InfoSphere DataStage DB2 Connector

Angelia Song
Brian Caufield
Fan Ding
Mi Wan Shum
Sriram Padmanabhan

August 13, 2015

Introduction

InfoSphere Information Server is a data integration platform that enables customers to understand, cleanse, transform, and deliver trusted information to their critical business initiatives. InfoSphere Information Server is designed to provide the infrastructure to create scalable, high-performance information integration environments, helping organizations turn the data they have into the innovation they need.

InfoSphere DataStage is the information integration component of InfoSphere Information Server. InfoSphere DataStage helps businesses integrate data across multiple high-volume data sources and target applications. It also supports the collection, integration, and transformation of large volumes of data with data structures ranging from simple to complex.

DB2 for z/OS is IBM’s relational database management system (DBMS) for the z/OS operating system. DB2 for z/OS is designed for and tightly integrated with the IBM System z® mainframe to leverage the strengths of System z.

DB2 for z/OS delivers innovations in operational efficiency for out-of-the-box savings, business resiliency, warehouse deployment, and enhanced business analytics. DB2 for z/OS is the only cost-effective, simple, and proven database on the market which offers 24x7 availability, reliability and security.
This white paper illustrates how to use InfoSphere DataStage to access DB2 for z/OS database systems. It also provides insights into the performance of different connectivity options and presents valuable best-practice recommendations.

**Overview of InfoSphere DataStage**

InfoSphere DataStage includes design tools that enable developers to create new data integration jobs visually. The term *job* is used in InfoSphere DataStage to describe extract, transform and load (ETL) tasks. Jobs are composed from a rich palette of operators called *stages*. These stages include:

- Source and target access for databases, applications and files
- General processing stages such as filter, sort, join, union, lookup, and aggregations
- Built-in and custom transformations
- Copy, move, FTP, and other data movement stages
- Real-time, XML, SOA and Message queue processing

In addition, InfoSphere DataStage allows developers to apply pre and post-conditions to each of these stages. Multiple data integration jobs can be controlled and linked by a sequencer that provides control logic, processing jobs in a particular order based on user-defined triggers. InfoSphere DataStage also supports rich administration capabilities for deploying, scheduling, and monitoring jobs.

One of the strengths of InfoSphere DataStage is that it enables you to design jobs independently of the underlying structure of the system. If the system changes, is upgraded or improved, or if a job is developed on one platform and implemented on another, the job design itself rarely needs to change. This separation is possible because InfoSphere DataStage gathers information about the shape and size of the system from a configuration file provided at run time. This file defines one or more processing nodes — logical, rather than physical — on which the job will run. Using the definitions in this file, DataStage organizes the necessary resources. When a system changes, the file is changed, not the job.

It is important to note that the number of processing nodes need not necessarily correspond to the number of CPU cores in the system. The file may define one processing node for each physical core in the system, or it may define more or less, depending on the requirements of the job. In this manner, InfoSphere DataStage delivers impressive parallel processing capabilities. The following factors affect the degree of parallelism required by a particular job or application:

- CPU-intensive applications, which typically perform multiple CPU-demanding operations on each record, benefit from the greatest possible parallelism supported by a system.
- Jobs with large memory requirements can take advantage of parallelism if they act on data that has been partitioned, and if the required memory is also divided among partitions.
- Applications that are disk- or I/O-intensive, such as those that extract data from and load data into databases, gain from configurations in which the number of logical nodes equals the number of I/O paths being accessed. For example, if a table is fragmented 16 ways inside a database or if a data set is spread across 16 disk drives, administrators should set up a node pool consisting of 16 processing nodes.
InfoSphere DataStage DB2 connector

InfoSphere Information Server connectivity options enable InfoSphere DataStage jobs to transfer data between InfoSphere Information Server and external data sources, such as:

- Relational databases
- Mainframe databases
- Enterprise resource planning (ERP) or customer relationship management (CRM) databases
- Online analytical processing (OLAP) or performance management databases

An InfoSphere connector is a component that provides data connectivity and metadata integration for external data sources. A connector typically includes a stage specific to the external data source.

InfoSphere supplies a native connector to access DB2 for Linux, UNIX, and Windows, and DB2 for z/OS databases. This connector is called the DB2 Connector. When you use InfoSphere DataStage to access DB2 database systems, the DB2 connector provides a single way to work with DB2 data. With the DB2 connector, you can perform transactional and non-transactional operations on DB2 data:

- Read, write, and look up data in DB2 database tables.
- Import metadata of table and column definitions, lists of DB2 databases, tables, views, and database aliases into a DB2 connector job.
- Manage rejected data.
- Work with national language data. The DB2 connector provides national language support (NLS).

Prior to the introduction of the DB2 Connector, there were a variety of older stages that had been developed to access DB2 data. The DB2 Connector effectively replaces all these older techniques.

The DB2 connector is the latest technology. It offers more functionality and improved performance, and is recommended for new job designs. If you have existing jobs that use the older stages and would like to update to use the connectors, the InfoSphere Connector Migration Tool can be used to facilitate the migration.

Bulk load in to DB2 for z/OS

InfoSphere DB2 connector provides the capability to bulk load data into DB2 for z/OS. The following graph shows a DataStage job that reads data from a text file and loads into a DB2 for z/OS table using the DB2 connector.

Figure 1. Bulk loading data in to a DB2 table
To enable the DB2 connector for bulk loading data into DB2 for z/OS, several parameters must be set in the DB2 connector. Visit the IBM Knowledge Center IBM InfoSphere Information Server 9.1.2 documentation for the necessary configuration steps.

To take advantage of the parallel bulk load feature in the DB2 connector, the Partition type property under Input > Partitioning tab must be set to DB2 connector.

Figure 2. The DB2 Connector

In addition to that, under the Properties tab, the Write mode must be set to Bulk load, and a target table name needs to be specified for Table name.

Figure 3. Setting the write mode and table name in DB2 Connector

Next, the Bulk load to DB2 on z/OS must be set to Yes, and the hostname or IP address of the DB2 for z/OS system must be specified for the Transfer to property.

If you choose to bulk load to DB2 for z/OS, you must specify a load method. You can select one of the following valid values:

- MVS™ dataset(s)
- Batch pipe(s)
- UNIX System Services (USS) pipe(s)

In this example, the Load method is set to USS pipe(s), hence a valid USS pipe directory must be specified. If other load methods are selected, you must specify the fields required for that load method.
**Figure 4. Selecting the load method**

<table>
<thead>
<tr>
<th>Yes or No</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Bulk load to DB2 on z/OS</td>
</tr>
<tr>
<td>Yes</td>
<td>USS pipe(s)</td>
</tr>
<tr>
<td>Yes</td>
<td>ftp</td>
</tr>
<tr>
<td>Yes</td>
<td>Remove Hostname1</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>/u/USER1</td>
<td>USS pipe directory</td>
</tr>
</tbody>
</table>

**Figure 5. Bulk load into a partitioned DB2 for z/OS table using the USS pipe load method**

The above figure illustrates the data movement during bulk loading into a partitioned DB2 for z/OS table using the USS pipe load method.

On the InfoSphere DataStage system, the DB2 connector runs in parallel with one connector process per DataStage partition. In addition, one UNIX pipe and one FTP connection are established for each DataStage partition for data transport.

Once the USS pipes are created on the DB2 for z/OS system to buffer data received by the FTP server (step 1A), the DB2 connector calls the LOAD utility via the DSNUTILS stored procedure (step 1B), and one LOAD utility is invoked for each table partition.

When data starts arriving from upstream stages in the DataStage job, the DB2 connector immediately sends it to the UNIX pipe, feeding the FTP client. The data is then sent to the DB2 for z/OS system via FTP directly into the USS pipe, and loaded into the table partition. Data loadings are executed independently and in parallel for all table partitions, making the DB2 connector a highly efficient approach for bulk load into DB2 for z/OS.
Read from DB2 for z/OS

The DB2 connector supports reading in parallel according to how the data is partitioned in DB2 for z/OS — namely, how the data is natively partitioned. The following image shows a DataStage job that reads data from a DB2 for z/OS table using the DB2 connector and outputs data to a COPY stage.

**Figure 6. DB2 Connector reading data from a DB2 table to a COPY stage**

![Diagram of DataStage job reading data from DB2 to COPY stage]

The connector determines the number of partitions in the table and dynamically configures the number of DataStage processing nodes to match the number of partitions. For DB2 for z/OS tables this is the number of table ("range") partitions in the table. The connector associates each node with one partition. For each node, the connector reads the records that belong to the partition that is associated with that node.

The connector accomplishes this by modifying the `SELECT` statement defined in the stage. The connector adds a `WHERE` clause that uses the column(s) in the partitioning key to select rows belonging to the partition associated with the current node.

The example below demonstrates how the DB2 connector reads data in a natively partitioned fashion. Assume a four-way-partitioned DB2 for z/OS table defined as:

```sql
CREATE TABLE TABLE1
(
  COL1 INTEGER,
  COL2 BIGINT
)
PARTITION BY (COL2)
(
  PART 1 VALUES (10000000000),
  PART 2 VALUES (20000000000),
  PART 3 VALUES (30000000000),
  PART 4 VALUES (MAXVALUE)
)
```

If the DB2 connector is configured in the following way.
Perform scalable data exchange using InfoSphere DataStage
DB2 Connector

Figure 7. The DB2 Connector configuration

<table>
<thead>
<tr>
<th>Usage</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate SQL</td>
<td>No</td>
</tr>
<tr>
<td>Enable quoted identifiers</td>
<td>No</td>
</tr>
<tr>
<td>Prefix for expression columns</td>
<td>EXPR</td>
</tr>
</tbody>
</table>

SQL

<table>
<thead>
<tr>
<th>SQL</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Select statement</td>
<td>SELECT * FROM TABLE1 WHERE COL1 &gt; 10</td>
</tr>
<tr>
<td></td>
<td>Read select statement from file</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Enable partitioned reads</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Partitioned reads method</td>
<td>DB2 connector</td>
</tr>
<tr>
<td>Table name</td>
<td>Table1</td>
<td></td>
</tr>
<tr>
<td>Column name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generate partitioning SQL</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

The connector will run four distinct SELECT statements in parallel:

Node 1: SELECT * FROM TABLE1 WHERE (COL2 <= '10000000000') AND (COL1 > 10)

Node 2: SELECT * FROM TABLE1 WHERE (COL2 > '10000000000') AND (COL2 <= '20000000000') AND (COL1 > 10)

Node 3: SELECT * FROM TABLE1 WHERE (COL2 > '20000000000') AND (COL2 <= '30000000000') AND (COL1 > 10)

Node 4: SELECT * FROM TABLE1 WHERE (COL2 > '30000000000') AND (COL1 > 10)

To enable the native parallel read in DB2 connector, the Enable partitioned reads property must be set to Yes, and Partitioned reads method property to DB2 connector. The name of the table whose partitioning will be used must also be specified in the Table name sub-property.

In addition, the Generate partitioning SQL sub-property defaults to Yes and must be set to Yes when reading from DB2 for z/OS.

Discussion of performance results

Performance test system configuration

To evaluate the performance of different connectivity options in DB2 connector, the following InfoSphere DataStage server and DB2 for z/OS system are setup.

Table 1. Hardware configuration for the InfoSphere DataStage server

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>40 Intel Xeon processor @ 2.4GHz</td>
</tr>
<tr>
<td>Memory</td>
<td>256 GB</td>
</tr>
<tr>
<td>Network</td>
<td>10Gbit Ethernet</td>
</tr>
<tr>
<td>Storage</td>
<td>DS5300 SAN Storage</td>
</tr>
<tr>
<td>Operating system</td>
<td>REHL 5.8 64bit</td>
</tr>
</tbody>
</table>

Table 2. Hardware configuration for DB2 for z/OS system

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>14 IBM system Z10 CPs (Central Processors)</td>
</tr>
</tbody>
</table>
The following table lists the server software used in the performance study.

**Table 3. Server software for InfoSphere DataStage and DB2 for z/OS**

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>InfoSphere DataStage Server</td>
<td></td>
</tr>
<tr>
<td>InfoSphere Information Server V9.1.2</td>
<td></td>
</tr>
<tr>
<td>Database system</td>
<td>DB2 for z/OS V10</td>
</tr>
</tbody>
</table>

**Bulk load into DB2 for z/OS**

**Figure 8. Throughput of data bulk load into partitioned DB2 for z/OS tables using different load methods**

The image above shows the throughput of a bulk load into a partitioned DB2 for z/OS table using different load methods. To evaluate the bulk load scalability, 1-, 2-, 4-, 8-, and 16-way partitioned tables are tested. For each table, DataStage loads data in \( n \) ways parallel, where \( n \) matches the number of partitions of the target table. Tests are repeated using the three load methods for each partitioned table.

For bulk load using the DB2 connector, our results show that the throughput increases with the number of table partitions, though the degree of improvement depends on the specific load method. Furthermore, the USS pipe and batch pipe loading methods perform comparably, both outperforming the MVS dataset load method.

The two pipe-based loading methods demonstrate high scalability (close to linear) with the number of table partitions. On our DB2 for z/OS system, during bulk load into a 16-way partitioned table, the 14 CPs were over-committed, compromising the load throughput and causing noticeable...
deviation from the theoretical linear scalability. We expect that enabling additional CPs on the system would increase the throughput in these cases.

USS pipes are a standard feature of the USS environment. Configuration and usage of USS pipes is straightforward. In addition, USS pipes demonstrated comparable performance to batch pipes, and significantly better performance over the MVS dataset method. For bulk load into DB2 for z/OS using the DB2 connector, USS pipes are recommended.

**Reading from DB2 for z/OS**

**Figure 9. Throughput of data read from partitioned DB2 for z/OS tables using different methods**

The image above shows the throughput of data read from partitioned DB2 for z/OS tables using different methods. To evaluate the native parallel read scalability, 1-, 2-, 4-, 8-, and 16-way partitioned tables are tested. For each table, DataStage reads data in \( n \) ways parallel, where \( n \) matches the number of the target table partitions. Tests are repeated using the DB2 connector and DB2 for z/OS stages. The legacy DB2 for z/OS stage is another option for reading data in parallel using the native table partitioning. To illustrate the efficiency of the native parallel read, tests are also conducted using the DB2 connector in the sequential mode, where data is read from each table partition serially, instead of in parallel.

Our results show that the native parallel read, using either the DB2 connector or DB2 for z/OS stage, substantially outperforms the sequential read. In addition, for the same number of table partitions, the DB2 connector yields a higher throughput than the DB2 for z/OS stage. In addition, the DB2 connector consumes less computing power than the DB2 for z/OS stage on the DataStage server, which is attributed to the difference in their underlying implementations.

Finally, the throughput of native parallel read using DB2 connector demonstrates high scalability (close to linear) with the number of table partitions.

Performance, lower resource overhead and scalability makes the DB2 connector a preferred option for data retrieval from DB2 for z/OS.
Conclusion

As demonstrated by the performance study results in this article, the InfoSphere DataStage DB2 connector enables a high-performance, scalable approach for data exchange between DB2 for z/OS and InfoSphere DataStage. This connectivity option enables organizations to leverage these industry-leading database system and data integration platform offerings.
Related topics

- Learn more about the InfoSphere Platform.
- Visit Data Integration to learn more.

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