Abstract
This white paper explains how Pentaho BI suite - Data Integration Component can be configured and used with EMC Greenplum Data Integration Accelerator (DIA) for big data analytics purposes in terms of structured and unstructured data.

February 2012
# Table of Contents

**Executive summary** ........................................................................................................ 4  
Audience .............................................................................................................................. 4  

**Big Data** ............................................................................................................................ 5  
The data explosion .................................................................................................................. 5  
Scope ..................................................................................................................................... 5  

**Organization of this paper** .............................................................................................. 6  

**Overview of Greenplum Database** .................................................................................. 7  
Overview of EMC Greenplum Data Computing Appliance (DCA) ......................................... 8  

**Overview of the Data Integration Accelerator** ................................................................ 9  
Product Overview .................................................................................................................. 9  
The DIA Configurations ........................................................................................................ 10  
Building Blocks of the DIA .................................................................................................. 10  
DIA/DCA Software and Theory of Operations ..................................................................... 12  
  Greenplum’s Scatter/Gather Streaming Technology ............................................................. 12  
  Parallel Loading .................................................................................................................. 12  
  External Tables ................................................................................................................... 14  
  Greenplum Parallel File Distribution Server (gpfdist) ......................................................... 14  
  How does gpfdist work ....................................................................................................... 14  
  Running gpfdist clients ...................................................................................................... 15  
  Using gpload to invoke gpfdist .......................................................................................... 16  

**Pentaho Data Integration with Greenplum Technology** .................................................... 20  
Using JDBC drivers for Greenplum database connections ................................................... 21  
  Installation of new driver .................................................................................................... 21  
  Pre-configurations Required ............................................................................................... 23  

**Basic Architecture PDI and Greenplum Database** ............................................................ 24  

**Pentaho Data Integration clustering for structured data** .................................................... 24  
  Snapshot of Carte Master-Slaves Registration Process ....................................................... 25  

**Pentaho Data Integration Partitioning** .......................................................................... 26  

**Pentaho Data Integration Partitioning in a Clustered Transformation** ............................ 27  
  Advanced Features: Using the Dynamic Cluster ................................................................. 27  
  Monitoring the Partitioned-Clustered Transformation ....................................................... 27  

**Integrating Pentaho Data Integration with GPHD for unstructured data** ......................... 28  

**Future expansion and interoperability** ............................................................................. 28  

**Conclusion** ....................................................................................................................... 28  

**References** ....................................................................................................................... 29
Executive summary

Greenplum Database is capable of managing, storing and analyzing big data analytics with the help of commodity hardware. This capability is available in an appliance packaging called the Data Computing Appliance (DCA). In additions, Greenplum Data Integration Accelerator (DIA) is a fast, parallel data loading solution, built specifically to integrate with the DCA. The DIA comes pre-configured with Greenplum’s Scatter/Gather Streaming™ loading technology (also referred to as gpload using gpfdist). Greenplum’s Scatter/Gather Streaming™ (SGS) technology, typically referred to as gpfdist, eliminates the bottlenecks associated to data loading, enabling ETL applications to stream data into the Greenplum database and Greenplum Hadoop solutions quickly.

Greenplum database is a popular analytical databases working with open source data integration products like Pentaho Data Integration (PDI), a.k.a. Kettle. Pentaho Kettle is part of Pentaho Business Intelligence suite. The EMC Greenplum Database is capable of managing, storing and analyzing large amount of data. Pentaho Data Integration unifies the ETL, modeling and visualization processes into a single, integrated environment with the use of Greenplum to drive better business decisions and speed up solution development for customers.

Audience

This white paper is intended for EMC field facing employees such as sales, technical consultants, support, as well as customers who will be using Pentaho Data Integration tool to integrate their ETL work with Greenplum DIA. This is neither an installation guide nor an introductory material on Pentaho. It documents the Pentaho connectivity and operation capabilities, and shows the readers how it can be used in conjunction with Greenplum database to retrieve, transform and present data to users. Though the reader is not expected to have any prior Pentaho knowledge, basic understanding of data integration concepts and ETL tools would help the reader understand this document better.
Big Data

The data explosion

All over the world, corporations are dealing with a massive data explosion. It is said that 90 percent of the data today have been created in the last two years alone. Data comes from everywhere: web, business transactions, automated data collection sensors, RFID data, mobile data, social-network data, retail transactions, health care data, electric meter readings, and so on. Our world has been exploding and analyzing large data sets; the amount of corporate data is in the order of Terabytes, Exabytes and Zettabytes of data.

Working with Big Data allows us to spot business trends with high accuracies. It offers us an opportunity to find new insights and mine unexplored trends in our businesses, and enables to answer questions that would have been beyond reach in the past. In so doing, it allows us to be more flexible and agile in meeting our business requirements. Good use of big data will underpin new waves of productivity growth and consumer surplus.

Scope

This document is not intended to be a complete Pentaho Data Integration installation guide or to support training material on Pentaho. Instead it augments the Pentaho installation documentation with particulars that should be noted when deploying PDI (Clustering and Partitioning) to the EMC DIA.
Organization of this paper

This paper mainly covers the following topics:

- Overview of Greenplum Database
- Overview of EMC Greenplum Data Computing Appliance (DCA)
- Overview of the Data Integration Accelerator
- Overview of Pentaho Data Integration (PDI)
- Basic Architecture PDI and Greenplum Database
- Pentaho Data Integration Clustering
- Architecture Greenplum Data Integration Accelerator using Pentaho Carte Clustering
- Pentaho Data Integration Partitioning
- Pentaho Data Integration Partitioning in a Clustered Transformation
- Using JDBC drivers for Greenplum database connections
- Future expansion and interoperability
- Conclusion
- References
Overview of Greenplum Database

Greenplum Database is designed based on a share-nothing MPP (Massively Parallel Processing) architecture which facilitates business intelligence and analytical processing. In this architecture, data is automatically partitioned across multiple 'segment' servers, and each 'segment' owns and manages a distinct portion of the overall data. All communication is via a network interconnect -- there is no disk-level sharing or contention to be concerned with (i.e. it is a 'shared-nothing' architecture).

Highlights of the Greenplum Database:

- Dynamic Query Prioritization: Provides continuous real-time balancing of the resources across queries.
- Self-Healing Fault Tolerance: Provides intelligent fault detection and fast online differential recovery.
- Polymorphic Data Storage-MultiStorage/SSD Support: Includes tunable compression and support for both row-and column-oriented storage.
- Analytics and Language Support: Supports analytical functions for advanced in-database analytics.
- Health Monitoring and Alerting: Provides integrated email and SNMP notification for advanced support capabilities.
Overview of EMC Greenplum Data Computing Appliance (DCA)

Greenplum Data Computing Appliance (Greenplum DCA) is a self-contained data warehouse solution that integrates all of the database software, servers and switches necessary to perform big data analytics. The DCA is a turn-key, easy to install big data analytic solution that provides extreme query and loading performance for analyzing large data sets. The Greenplum DCA is built from four server increments called modules. The DCA starts as a single Greenplum Database module, and can be configured with a maximum of 24 modules, up to a 6 racks. The DCA integrates Greenplum Database, data loading and Hadoop software with compute, storage and network components; delivered racked and ready for immediate data loading and query execution. It takes advantage of large clusters of increasingly powerful, commodity servers, storage and network switches to minimize the customers’ cost of ownership.
Overview of the Data Integration Accelerator

Product Overview

Data Integration Accelerator (DIA) modules are high capacity loading servers. The DIA servers are pre-configured with Greenplum’s gpfdist and gpload software for easy loading and staging of data into GPDB modules. Up to 4 modules of DIA are supported in a Greenplum DCA configuration.

The Data Integration Accelerator (DIA) is specially built to facilitate fast data loading to the DCA. It integrates the Greenplum data loading software called gpfdist with the server, storage and networking gear into a single system. It leverages the high-speed internal 10 Gb/sec communication network to deliver the data quickly to the DCA. Each server in the DIA is preloaded with RedHat Enterprise Linux operating systems with the requested software stack.

The DIA meets the reliability requirements of the most mission-critical enterprises with the data availability consisting of RAID protection at the disk level. It is offered in three models: the DIA 10 (quarter rack), DIA 100 (half rack) and DIA 1000 (full rack) systems. The DIA comes in blocks of 4 servers. Each block is referred to as a module. Customers can order up to 4 modules of DIA for each rack. The DIA architectural diagram is described as follows:
The DIA Configurations

The following table shows some of DIA models. The physical dimensions, weights, power and cooling details can be found in the DIA specification sheet on EMC Powerlink.

<table>
<thead>
<tr>
<th></th>
<th>DIA 10 Quarter Rack</th>
<th>DIA 100 Half Rack</th>
<th>DIA 1000 Full Rack</th>
</tr>
</thead>
<tbody>
<tr>
<td># servers</td>
<td>4</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td># CPU cores</td>
<td>48</td>
<td>96</td>
<td>192</td>
</tr>
<tr>
<td>Total memory</td>
<td>192 GB</td>
<td>384 GB</td>
<td>768 GB</td>
</tr>
<tr>
<td>Total SATA disks</td>
<td>48</td>
<td>96</td>
<td>192</td>
</tr>
<tr>
<td>Usable capacity</td>
<td>70TB</td>
<td>140TB</td>
<td>280TB</td>
</tr>
</tbody>
</table>

As with most DCA appliances, the DIA can be orderable in building blocks (modules) of 4 servers each. For the DIA, the customer must already have at least one module of DCA, and the DIA can reside in the same rack above the DCA, if there is room in the rack.

Building Blocks of the DIA

DIAs are purchased together with DCAs. Currently, customers can order their DCA/DIA in functional blocks of up to 6 racks. In the near future, this number will be increased to up to 12 racks.
The physical DCA and DIA racks and blocks are assigned with numbers (see table below):

<table>
<thead>
<tr>
<th>Rack 1</th>
<th>Rack 2</th>
<th>Rack 3</th>
<th>Rack 4</th>
<th>Rack 5</th>
<th>Rack 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block 4</td>
<td>Block 4</td>
<td>Block 4</td>
<td>Block 4</td>
<td>Block 4</td>
<td>Block 4</td>
</tr>
<tr>
<td>Block 3</td>
<td>Block 3</td>
<td>Block 3</td>
<td>Block 3</td>
<td>Block 3</td>
<td>Block 3</td>
</tr>
<tr>
<td>Block 2</td>
<td>Block 2</td>
<td>Block 2</td>
<td>Block 2</td>
<td>Block 2</td>
<td>Block 2</td>
</tr>
<tr>
<td>Block 1</td>
<td>Block 1</td>
<td>Block 1</td>
<td>Block 1</td>
<td>Block 1</td>
<td>Block 1</td>
</tr>
</tbody>
</table>

Table 1: Physical numbering of blocks in DCA/DIA racks.

The racks are numbered, starting from 1 and incrementing by 1, up to the maximum number of supported racks. Within each rack, we can house up to 4 blocks, each block containing 4 DCA/DIA servers. Each block can contain balanced DCA segment servers, DCA capacity segment servers, Hadoop nodes, or DIA servers.
DIA/DCA Software and Theory of Operations

The overall theory of operation for the DIA/DCA is to place the Data Integration Software (i.e., ETL tools) stack on the DIA platform so that users can maximize and fully leverage the intrinsic capabilities of the Greenplum Data Computing Appliance. This approach allows us to continue using the traditional direct database connection approach of ETL tools, but also allows the tools to directly access the Greenplum Scatter/Gather Streaming (SGS) technology as required.

Greenplum's Scatter/Gather Streaming Technology

Parallel Loading

Greenplum's Scatter/Gather Streaming™ (SGS) technology, typically referred to as gpfdist, eliminates the bottlenecks associated to data loading, enabling ETL applications to stream data into the Greenplum database very fast. This technology is intended for loading big data sets that are normally used in large-scale analytics and data warehousing.

This technology manages the flow of data into all nodes of the database. It does not require additional software or systems and takes advantage of the same parallel dataflow engine nodes in Greenplum database.

Figure 1 shows how Greenplum utilizes a parallel everywhere approach to loading.
Greenplum’s SGS technology ensures parallelism by scattering data from source systems across 100s or 1000s of parallel streams that simultaneously flow to all nodes of the Greenplum Database. Performance scales with the number of Greenplum Database nodes, and the technology supports both large batch and continuous near-real-time loading patterns with negligible impact on concurrent database operations.

Figure 2 shows how the final gathering and storage of data to disk takes place on all nodes simultaneously, with data automatically partitioned across nodes and optionally compressed. This technology is exposed via a flexible and programmable external table (explained below) interface and a traditional command-line loading interface.
**External Tables**

External tables enable you to access data in external sources as if it were in a table in the database. In Greenplum there are two types of external data sources, external tables and Web tables. They have different access methods, external tables contain static data that can be scanned multiple times. The data does not change during queries. Web tables provide access to dynamic data sources as if those sources were regular database tables. Web tables cannot be scanned multiple times. The data can change during the course of a query.

**Greenplum Parallel File Distribution Server (gpfdist)**

gpfdist is Greenplum’s parallel file distribution server utility software. It is used with read-only external tables for fast, parallel data loading of text, CSV, XML files into a Greenplum data warehouse. The benefit of using gpfdist is that users can take advantages of maximum parallelism while reading from or writing to external tables, thereby offering the best performance as well as easier administration of external tables.

gpfdist can be considered as a networking protocol, much like the http protocol. Running gpfdist is similar to running a HTTP server. It exposes the target file via TCP/IP to a local file directory containing the files. The files are usually delimited files or CSV files, although it can also read tar and gziped files. In the case of tar and gzip files, the PATH much contains the location of the tar and gzip utilities.

For data uploading into a Greenplum database or data warehouse, you generate the flat files from an operational database or transactional database, using export, COPY, dump, or user-written software, depending on your business requirements. This process can be automated to run periodically.

gpfdist is pre-installed on each DIA server by default.

**How does gpfdist work**

gpfdist runs in a client-server model. The DIA server acts as the server, while the master server of the DCA is the client. You start the gpfdist process on a DIA server, by indicating the directory where you drop/copy your source files. Optionally, you may also designate the TCP port number to be used.
A simple startup of the gpfdist server is the following command syntax:

```
gpfdist -d <file_files_directory> -p <port_number> -l <log_file> &
```

For example:

```
# gpfdist -d /etl-data -p 8887 -l gpfdist_8887.log &
[1] 28519
# Serving HTTP on port 8887, directory /home/gpadmin/etl-log
```

In the above example, gpfdist server is set up to run on one of the DIA host, anticipating data loading from flat files stored in a file directory /etl-data. Port 8887 is opened and listening for data requests and a log file is created in /home/gpadmin called etl-log.

For each DIA server, you can run multiple instances of gpfdist, under different TCP ports and directories.

**Running gpfdist clients**

To initiate the data extraction process, we use the DCA Master Server as the client. We connect to gpfdist through the external tables. These tables can be created using the psql command.

For example:

```
# psql -d gpdb
gpdb=#
gpdb=# create external table ext_load_performance (like performance_table)
gpdb=# location ('gpfdist://etl3:8887/performance_test.dat')
gpdb=# format 'text' (delimiter '|')
gpdb=# segment reject limit 20 rows;
CREATE EXTERNAL TABLE
```

In the above example, we create an external table. This table has all the attributes of a table called ‘performance_table’ (using the like performance_table clause), and makes uses of flat files stored in the host ‘etl3’, using port 8887. The flat file name is expected to be ‘performance_test.dat’.

Combining this statement with the statement started in the gpfdist server, we expect the file to be in the directory ‘/etl-data’ (I thought it is “etl-data”). Both client and server will be communicating using TCP port 8887 on the DIA side.

Of course, the external table itself will not initiate any process, or do any data loading. It has simply defined a connection between the client and the server.
To start the data loading process, you issue the following command, for example:

```
insert into performance_table select * from ext_load_performance;
```

When this command is run, each segment server will connect to gpfdist simultaneously. The gpfdist server will divide the flat file in chunks, and distribute the work amongst the segment servers. Taking advantage of the ‘share nothing’ architecture, Greenplum database is able to make use of the parallelism given to the data loading operations, achieving linearly scalable data loading performance as one adds DIA/DCA components.

**Using gpload to invoke gpfdist**

DIA leverages the parallel bulk loading capabilities of the Greenplum data loading utility called “gpload”. “gpload” is a data loading utility that acts as an interface to Greenplum Database’s external table parallel loading feature. The Greenplum EXTERNAL TABLE feature allows us to define network data sources as tables that we can query to speed up the data loading process. Using a load specification defined in a YAML formatted control file, “gpload” executes a load by invoking the Greenplum parallel file server(gpfdist) – Greenplum’s parallel file distribution program, creating an external table definition based on the source data defined, and executing an INSERT, UPDATE or MERGE operation to load the source data into the target table in the database.

The gpload program processes the control file document in order and uses indentation (spaces) to determine the document hierarchy and the relationships of the sections to one another. The use of white space is significant. White space should not be used simply for formatting purposes, and tabs should not be used at all.
The basic structure of a load control file is:

```---
VERSION: 1.0.0.1
DATABASE: db_name
USER: db_username
HOST: master_hostname
PORT: master_port
GPLOAD:
INPUT:
- SOURCE:
  LOCAL_HOSTNAME:
  - hostname_or_ip
  PORT: http_port
  | PORT_RANGE: [start_port_range, end_port_range]
FILE:
- /path/to/input_file
- COLUMNS:
  - field_name: data_type
  - FORMAT: text | csv
  - DELIMITER: 'delimiter_character'
  - ESCAPE: 'escape_character' | 'OFF'
  - NULL_AS: 'null_string'
  - FORCE_NOT_NULL: true | false
  - QUOTE: 'csv_quote_character'
  - HEADER: true | false
  - ENCODING: database_encoding
  - ERROR_LIMIT: integer
  - ERROR_TABLE: schema.table_name
OUTPUT:
- TABLE: schema.table_name
- MODE: insert | update | merge
- MATCH_COLUMNS:
  - target_column_name
- UPDATE_COLUMNS:
  - target_column_name
  - UPDATE_CONDITION: 'boolean_condition'
- MAPPING:
  target_column_name: source_column_name | 'expression'```
PRELOAD:
- TRUNCATE: true | false
- REUSE_TABLES: true | false

SQL:
- BEFORE: "sql_command"
- AFTER: "sql_command"

Above example shows syntax for GPLOAD using YAML file. This file is divided into sections for easy reference; those horizontal lines are not to be placed in a YAML file. For example, you can run a load job as defined in my_load.yml using gpload:

    gpload -f my_load.yml

It is recommended that we confirm that gpload is running successfully, to reduce any chance of future errors.
Example of the load control file - my_load.yml:

```yaml
---
VERSION: 1.0.0.1
DATABASE: ops
USER: gpadmin
HOST: mdw-1
PORT: 5432
GPLOAD:
  INPUT:
    - SOURCE:
      LOCAL_HOSTNAME:
        - etl1-1
        - etl1-2
        - etl1-3
        - etl1-4
    PORT: 8081
    FILE:
      - /var/load/data/*
        - COLUMNS:
          - name: text
          - amount: float4
          - category: text
          - desc: text
          - date: date
          - FORMAT: text
          - DELIMITER: '
          - ERROR_LIMIT: 25
          - ERROR_TABLE: payables.err_expenses
    OUTPUT:
      - TABLE: payables.expenses
      - MODE: INSERT
    SQL:
      - BEFORE: "INSERT INTO audit VALUES('start', current_timestamp)"
      - AFTER: "INSERT INTO audit VALUES('end', current_timestamp)"
```

Note: YAML file is not a free formatted file, field names and most of the content need to be in a certain format.
Pentaho Data Integration with Greenplum Technology

Pentaho Data Integration (PDI) delivers comprehensive Extraction, Transformation and Loading (ETL) capabilities using a meta-data driven approach. It is commonly used in building data warehouses, designing business intelligence applications, migrating data and integrating data models. It consists of different components:

- **Spoon** – Main GUI, graphical Jobs/Transformation Designer
- **Carte** – HTTP server for remote execution of Jobs/Transformations, Carte is both a master and client software, it runs at PDI master and PDI slave nodes
- **Pan** – Command line execution of Transformations
- **Kitchen** – Command line execution of Jobs
- **Encr** – Command line tool for encrypting strings for storage

Pentaho is capable of handling huge data sets (i.e. in terms of Terabytes and Petabytes) with Greenplum Database (for structured data) and Greenplum Hadoop (for unstructured data) taking full advantage of the massively parallel processing environment provided by the Greenplum product family.
Using JDBC drivers for Greenplum database connections

Pentaho Kettle ships with many different JDBC drivers that reside in a single java archive (.jar) file that are present in the libext/JDBC directory. By default, Pentaho PDI is shipped with a postgresql jdbc jar file, which is used to connect to Greenplum.

Java 1.6 (JDK 6) is required for the installation.

Installation of new driver

To add a new driver, simply drop/copy the .jar file containing the driver into the libext/JDBC directory. For example,

• For Data Integration Server: `<Pentaho_installed_directory>/server/data-integration-server/tomcat/lib/

• For Data Integration client: `<Pentaho_installed_directory>/design-tools/data-integration/libext/JDBC/

For BI Server: `<Pentaho_installed_directory>/server/biserver-ee/tomcat/lib/

• For Enterprise Console: `<Pentaho_installed_directory>/server/enterprise-console/jdbc/

If you installed a new JDBC driver for Greenplum to the BI Server or DI Server, you have to restart all affected servers to load the newly installed database driver. In addition, if you want to establish a Greenplum data source in the Pentaho Enterprise Console, you must install that JDBC driver in both Enterprise Console and the BI Server to make it effective.

In brief, to update the driver, the user would need to update the jar file in `/data-integration/libext/JDBC/`.

Assume that a Greenplum Database (GPDB) is installed and ready to use, users can define the Greenplum database connections in the Database Connection dialog. Users can give a connection name, choose Greenplum as the Connection Type, choose “Native (JDBC)” as Access, and give the Host Name, Database Name, Port Number, User Name and Password in the Setting section.

Special attention may be required to setup the host files and configuration files in Greenplum database as well as the hosts in which Pentaho is installed. For
instance, in Greenplum database, the user may need to configure pg_hba.conf with the IP address of the Pentaho host. In addition, users may need to add the hostnames and the corresponding IP address in both systems (i.e. Pentaho PDI server and the Greenplum Database) in order to ensure both machines can communicate with each other.

Please refer to the white paper “WORKING WITH PENTAHO DATA INTEGRATION USING GREENPLUM - The interoperability between Pentaho Data Integration (Kettle) and Greenplum Database” for details of JDBC configurations.
Pre-configurations Required

There are a few pre-requisites before configuring the spoon and carte for parallelism:

1) Pentaho PDI is installed in all the Master and Slave servers
2) Pentaho Spoon is configured and started correctly in Master server
3) Pentaho Carte is configured and be able to started correctly as a background process in master and slave servers correspondingly.
4) Carte.sh is both a master and client software, it runs at PDI master and PDI slave nodes.
5) Master is required to distribute loads to all slaves, monitors their activity.
6) GPLOAD is pre-installed and tested in each of the master and slaves servers.

You can test GPLOAD by correctly define the yaml file and run gpload. For example, you will see something like this if the test ran through:

```bash
[gadmin@etl4 gpload_test]$ gpload -f a.yaml
2011-11-12 11:43:39|INFO|started gpfdist -p 8000 -P 9000 -f
"/home/gadmin/gpload_test/lineitem.dat" -t 30
2011-11-12 11:43:45|INFO|running time: 6.25 seconds
2011-11-12 11:43:45|INFO|rows Inserted = 274191
2011-11-12 11:43:45|INFO|rows Updated = 0
2011-11-12 11:43:45|INFO|data formatting errors = 0
2011-11-12 11:43:45|INFO|gpload succeeded
```
Basic Architecture PDI and Greenplum Database

The following diagram shows the basic interoperability between Pentaho Data Integration with the Greenplum Database:

Pentaho Data Integration clustering for structured data

With structured data, in order to take the full advantage of parallelism on Data Integration Accelerator (DIA), Carte is heavily used in the design to demonstrate both the PDI clustering and partitioning features among different segment servers in EMC Greenplum DIA.

Clustering is an important concept in Pentaho when you have to handle some complex transformations and a huge load into your target database. In order to handle this requirement, PDI offers an advanced feature that implement the clustering concepts to break the workload into trunks and let each of the clustered server to handle one portion of the load. The clustered servers will follow a Master-Slaves architectural model. The master node will decide how to distribute the workload into different portions for each slave worker node to process. Basically, the global workload is divided into different smaller workloads and execute in all the slave worker nodes at the same time in parallel (or near parallel). The tool that created this master-slaves architecture is called “Carte”. In this architecture, there is one master node, which is acting like a conductor, assigning the sub-tasks (the original task is broken into different sub-tasks) to the slaves and merging the results that come back from the slave nodes when the sub-tasks are completed. For each slaves, it is a node that will execute the job/transformation, process the tasks and then send back the results to the
Master node for reconciliation. In a word, the relationship of Master Carte Node and Slave Carte Nodes is one-to-many (1..n, where n >= 1) relationship. Usually, the more PDI slaves you implement in this Master-Slave architecture, the better parallelism/performance you can achieve.

For example, you can create a job (with transformation) like this:
Pentaho Data Integration Partitioning

Partitioning splits up the database tables or data sets during processing. In Pentaho Data Integration, Kettle has the capability to direct rows of data to a certain step copy based on a partitioning rule. There are a few partitioning rules that you can apply:

- Mirror to all partitions
- Mod partitions on the last 2 digits before the extension of a filename “hash-partitioned”, “Remainder of division” (modulo).

The objective of using partitioning is to increase the parallelism in a transformation. The implementation of partitioning in Kettle is straight-forward: for each partition that is defined, multiple step copies are started for steps with a partitioning method defined. That means that if you define “N” partitions, you have “N” copies to do the work. Database partitions can also be defined in PDI in the database connection.
Pentaho Data Integration Partitioning in a Clustered Transformation

By spreading the partitions over a number of slaves in a clustered transformation, we can address the problem when the number of step copies is growing rapidly. You can define a partitioning schema with a static list of partitions with those partitions divided over the number of slave transformations at run-time. However, if you define too many partitions across too many slave servers, it will cause a lot of network traffic as well as CPU and memory consumption. Therefore, the clustered and partitioned transformations should be carefully designed to achieve the desirable performance.

Advanced Features: Using the Dynamic Cluster

There is an advanced feature called “dynamic cluster” enables you to take advantage of a dynamic pool of computer resources. A dynamic cluster is a cluster schema where the slave servers are only known at runtime. Creating a dynamic cluster schema is like creating a normal cluster schema by checking the “Dynamic cluster” option in the “Clustering Schema dialog”.

Monitoring the Partitioned-Clustered Transformation

You can monitor the Partitioned-Clustered Transformation via the browser. For example, you can monitor different cluster through either the browser or spoon.
Integrating Pentaho Data Integration with GPHD for unstructured data

The demand of handling unstructured data is increasing and effectively implemented analytics can give customers a competitive advantage. Hadoop has rapidly emerged as the preferred solution for Big Data analytics across unstructured data. Greenplum HD is a certified and supported version of the Apache Hadoop stack with the java-based file system HDFS (Hadoop Distributed File System). By combining the newly release Pentaho PDI features, such as MapReduce Input/Output, Hadoop Copy Files, Hadoop File Input/Output, with the build-in Greenplum bulk loader in DIA, customers can easily manipulate unstructured data with their familiar ETL environment by moving data easily in and out Greenplum Hadoop and Greenplum Database.

Future expansion and interoperability

Both Greenplum and Pentaho are increasing their capability to adopt the Big Data trends as there is a significant growth in data sizes in the industry. Therefore, both companies are expanding their interoperability to adopt the upcoming demands. This will benefit the customers by providing more choices with enhanced performance options.

Conclusion

In this white paper, the process of how to create and apply JDBC driver to connect Pentaho Data Integration with Greenplum Database is discussed using JDBC driver in particular. It only covers the preliminary interoperability between both Pentaho PDI with a clustering and partitioning features and Greenplum database for basic data integration and business intelligence projects.

It also discussed briefly the anticipated interoperability and integrations of both technologies to accommodate the Big Data Trend in the coming future, such as, the Greenplum native bulk loader, Pentaho Integration with Greenplum Hadoop HD solutions and Greenplum DIA integration with Pentaho Data Integration tools. We will discuss those future expansions in upcoming white papers.
References


2) Getting Started with Pentaho Data Integration guide from www.pentaho.com


4) EMC Greenplum Data Computing Appliance Getting Started Guide Version 1.1.0.0 (P/N: 300-012-917 Rev: A02)

5) EMC Greenplum Data Computing Appliance: High Performance for Data Warehousing and Business Intelligence – An Architecture Overview white paper

6) EMC GREENPLUM DATA INTEGRATION ACCELERATOR white paper

7) EMC GREENPLUM DATA COMPUTING APPLIANCE white paper

8) Greenplum Database 4.1 Load tools for Unix

9) Other EMC Greenplum Database, DCA, and DIA Documentation

The most recent documentation is available on the EMC Powerlink web site: http://powerlink.emc.com