White Paper

EMC INFRASTRUCTURE FOR THE SMART GRID ANALYTICS PLATFORM IN PARTNERSHIP WITH SILVER SPRING NETWORKS

EMC Greenplum Data Computing Appliance, Silver Spring Networks, Attunity CDC Stream, Silver Spring Networks UtilityIQ

- Scalable deep archive for smart grid data
- Analytics platform for smart grid information
- Configuration and implementation reference architecture

EMC Solutions Group

Abstract

This white paper describes EMC's smart grid analytics platform solution. The solution integrates the Silver Spring Networks UIQ system with the EMC Greenplum Data Computing appliance to help utility companies manage and mine their smart grid data for business value and operational efficiency.

June 2012

Table of contents

Executive summary

Business case

Electric utilities are increasingly deploying smart grids—electronic meters that replace the electro-mechanical meters of yesterday - to enable the collection of much more detailed information about the power usage of its customers and the state of its grid. This kind of detailed information allows utilities to develop advanced analytic models to address challenges such as voltage compliance, energy theft, Power Factor issues, high bill prediction, and many others.

Along with the benefits however, come the challenges of managing this increase in data streaming into utility company data centers, by some estimates, up to 3,000x more data than is produced by today's traditional power network. Existing operational databases cannot keep up with this influx of "Big Data" and typically do not have enough storage to be able to perform long-term analysis. In addition, even if they could store that much information, they cannot perform the advanced analytics against the huge datasets in a timely manner and without impacting production operations.

Customers need an analytics platform solution that is capable of handling large-scale smart grid data and that provides a platform for performing advance analytics on that data.

Solution overview

EMC and Silver Spring Networks have teamed to define and test a solution and reference architecture for a smart grid analytics platform. This infrastructure integrates with the existing Silver Spring Networks UtilityIQ (UIQ) database to provide a scalable smart grid analytics platform for ingest and long-term storage of meter data. It also provides a platform for advanced analytics on the meter data so that utility companies can derive additional value from their smart grid investment.

The smart grid analytics platform solution is an infrastructure capable of pulling smart grid data from an SSN UIQ application built on Oracle and loading it into a Greenplum Data Computing Appliance (DCA). This infrastructure enables petabytes of data to be queried and analyzed in near real time using massively parallel compute strategies. The net result is that analytics, which currently take hours or days, can now be computed within minutes of the real-time events occurring. The solution is designed to work seamlessly with the existing UIQ and cause no impact or additional work for database administrators (DBAs).

The Attunity Stream Change Data Capture (CDC) software processes Oracle transaction to record new and updated meter information and efficiently transfers that information into the Greenplum database. The system avoids the use of database triggers and is capable of scaling to millions of meters and makes the meter data available for analysis mere minutes after it comes in from the meter.

Key results

The key benefits provided by this solution are the ability to initialize and then continually capture large amounts of smart grid data in the Greenplum database. Then, the Greenplum database becomes a platform to help utility companies derive value from this information in areas such as voltage compliance, energy theft, system loading, to name a few.

The solution design features these benefits:

- **Automated schema migration**

  The solution performs an automated schema migration from Oracle tables and data types into Greenplum tables and data types. This includes creating primary keys as well as the proper distribution clause for the Greenplum tables.

- **Automated initial data load**

  The solution automatically performs an initial data migration from the Oracle database into the Greenplum database. No intermediate files are created and no additional disk space is consumed because the data is streamed from source to target through the network.

- **Change data capture**

  The solution continually captures changes in the source database and propagates those changes (inserts, updates, and deletes) into the Greenplum database. New smart grid data in the Oracle database is available for analysis in the Greenplum database within minutes.

- **Analysis**

  EMC Greenplum's standard and open interfaces enable developers to access data using SQL and advanced in-database analytics, resulting in lightning fast correlation of previously disparate data sources.

Introduction

Purpose

The purpose of this white paper is to describe a smart grid analytics platform solution that integrates with the Silver Spring Networks UIQ system. This solution provides long-term storage of the meter data as well as a platform for analysis of that data to derive business value and reduce operational expenses for utility companies.

Scope

The scope of this white paper is to:

- Provide an overview of the key technology components in the solution
- Present the solution architecture
- Document solution design considerations
- Provide implementation details of the software tools used in the solution
- Describe performance test scenarios, procedures, and results from a lab testing environment

Audience

This white paper is intended for a technical audience, including:

- Customers who have had a technical briefing and are interested in moving to the pilot stage
- The systems integrator (SI) who can use this as a blueprint for implementing a smart grid analytics platform solution
- EMC consultants who want to understand the capabilities of this solution

Terminology

Table 1 defines the terminology used in this paper.

Table 1. Terminology

Term	Definition
Analytics	The study of operational data using statistical analysis with a goal of identifying and using patterns to optimize business performance.
Business intelligence (BI)	The effective use of information assets to improve the profitability, productivity, or efficiency of a business. IT professionals use this term to refer to the business applications and tools that enable such information usage. The source of information is frequently the analytics platform.
Data load	The term used to refer to the input of data into an analytics platform. We use the term data load as an action that can also be applied to the speed at which data can be fed into the analytics platform. The data load rate is often quoted in bandwidth, for example, GB/s.
Change Data Capture (CDC)	The process of capturing changes made at the data source and applying them to the target database.

Term	Definition
Data warehousing (DW)	The process of organizing and managing information assets of an enterprise. IT professionals often refer to the physically stored data content in some databases managed by database management software as the analytics platform. They refer to applications that manipulate the data stored in such databases as DW applications.
Massively Parallel Processing (MPP)	Many servers working in parallel on database tasks.
Shared-Nothing Architecture	A distributed computing architecture made up of a collection of independent, self-sufficient servers. This is in contrast to a traditional central computer that hosts all information and processing in a single server.

Key technology overview

Overview

This section provides an overview of the key components of the solution:

- EMC Greenplum Data Computing Appliance
- EMC Outsourcer Oracle to Greenplum integration software
- Attunity Stream CDC
- Solution-specific integration scripts

EMC Greenplum Data Computing Appliance

The EMC Greenplum Data Computing Appliance (DCA) is a purpose-built appliance that delivers a fast-loading, highly scalable, and parallel computing platform for next-generation data warehousing and analytics. The appliance architecturally integrates database, computing, storage, and network resources into an enterprise-class, easy-to-implement system. The DCA offers the power of MPP architecture, delivers the fastest data loading capacity in the industry, and provides the best price-to-performance ratio without the complexity and constraints of proprietary hardware.

For this solution, we configure the DCA with two standard Greenplum Database (GPDB) modules (which is also known as a half-rack configuration). Depending on the storage, ingest, and query requirements for any analytics platform, a customer may require fewer or more modules to match specific requirements.

Figure 1 illustrates the modular approach of the DCA. In this figure, "GPDB" refers to Greenplum Data Base, "HD" identifies a Hadoop module, and "DIA" identifies a Data Integration Accelerator.

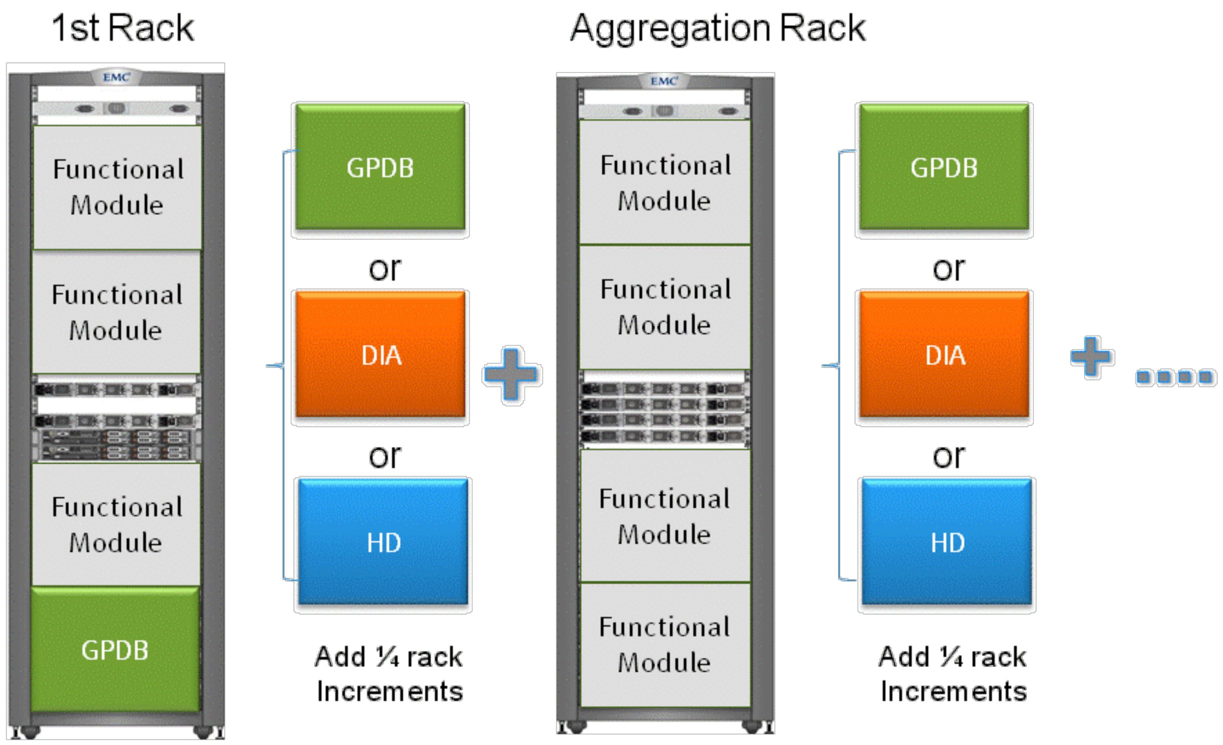

Figure 1. DCA modules

For more information about the DCA, refer to the white paper: *EMC Greenplum Data Computing Appliance: Performance and Capacity for Data Warehousing and Business Intelligence—A Detailed Review.*

EMC Outsourcer Migration Assistance Software

EMC has developed utility software that assists in the integration of the Greenplum database with existing Oracle and SQL Server databases. Integration in this case can include schema translation from Oracle or SQL Server into Greenplum database, initial load, ongoing refresh, and change capture. This solution uses EMC Outsourcer to efficiently translate and create existing Oracle schemas and data types into matching Greenplum schemas and data types. Outsourcer also performs the initial data load to seed the analytics platform with a full copy of the meter data in the SSN UIQ Oracle database.

The EMC Outsourcer software simplifies the initial setup of the solution by:

- Providing automated schema translation between Oracle and Greenplum
- Automating the bulk data load of data from the source Oracle database into the Greenplum analytics platform

Additional features include:

- Dynamic data type mapping from Oracle to Greenplum
- Network-based streaming of initial load requiring no disk space for intermediate files
- Easily restart-able so that any target table can be refreshed from the source database table

Attunity Integration Suite: Attunity Stream

The Attunity Integration Suite (AIS) is a comprehensive integration platform for on-demand access and integration of enterprise data sources and legacy applications. This solution uses the Attunity Stream component of AIS to perform CDC from the Oracle database to the Greenplum analytics platform. As meter data comes into the Oracle database, Attunity Stream tracks the database changes via the Oracle log files and records these transactions in flat files, which are subsequently loaded into the Greenplum database using the Greenplum gpfdist high speed parallel loading functionality.

With Attunity Stream, you do not need to add triggers to the source database to capture changes. Instead, it uses log mining to track changes with little to no impact on the performance of the Oracle database.

Attunity Stream includes an easy-to-use graphical user interface that makes it simple to connect to the source database and define the CDC jobs to capture data. CDC jobs identify the tables to track in the source database and include options for filtering the columns and limiting the rows that are included in the capture.

Attunity Stream CDC seamlessly resumes where it left off in the event of planned or unplanned outages of the Attunity server or source database.

Using Attunity Stream CDC, organizations can:

- Easily capture changes in a source database for efficient transfer into a target database
- Avoid performance impact on the source database by leveraging existing log mining agents to capture changes to designated tables
- Avoid additional DBA maintenance requirements by not requiring triggers to be placed on tables being monitored in the source database
- Scale to track many source databases and multiple schemas within a database to consolidate change streams into one central analytics platform

Solution-specific Integration Scripts

This solution consists of multiple components that must be integrated to work together to meet the goals of the solution. We developed several integration scripts to carry out functions related to automating and coordinating these components to provide a seamless and easily managed system.

In addition, to provide high-speed ingest of captured meter data, we developed Greenplum stored procedures to automatically import the meter transaction data and apply those transactions to the target tables.

The design and implementation sections below provide details about these integration scripts.

Solution architecture

Overview

This section illustrates the architectural layout of the smart grid analytics platform environment. It also provides details of the hardware and software resources that were used in the reference environment that underpins the testing described in the *Performance testing* section.

Solution architecture

Figure 2 provides a high-level illustration of the architectural layout of the smart grid system. The outlined area indicates the smart grid analytics platform solution that is described in this document. This document describes the design, integration, and operation of this solution.

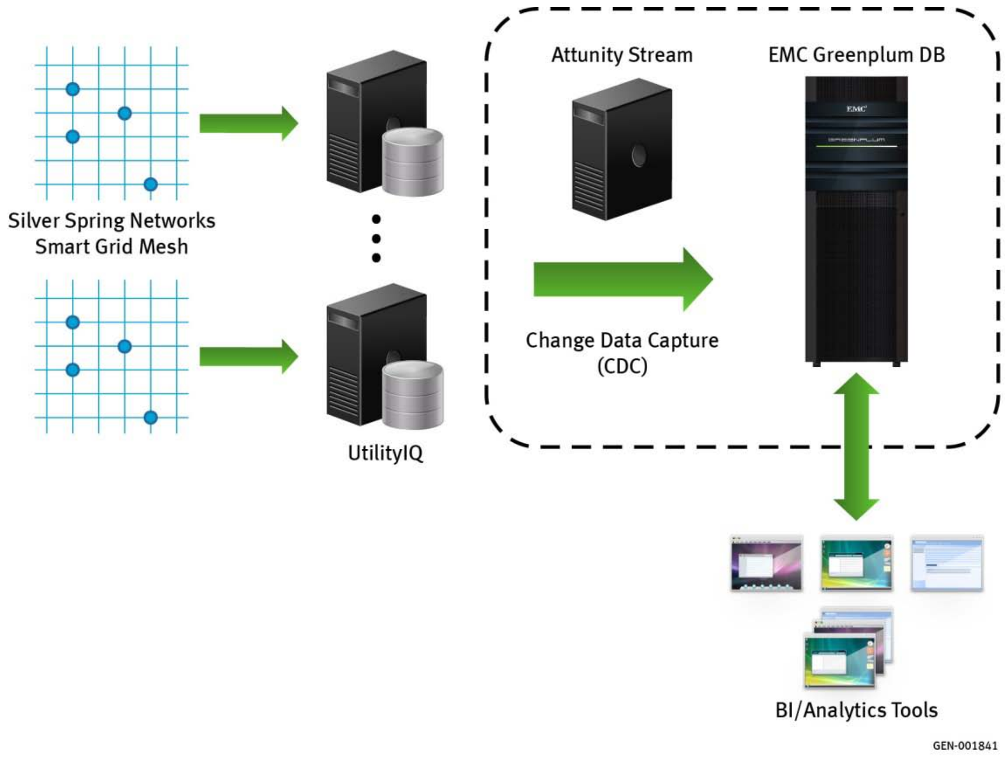

Figure 2. Solution architecture

A general overview of the operation of the smart grid environment is as follows.

Silver Spring Networks (SSN) smart grids, which are deployed in homes and businesses served by the utility company, record power usage at set intervals during the day. These meters communicate with each other to form a mesh network of meters and eventually return meter data to the SSN UIQ Oracle database. The SSN UIQ database validates and organizes the incoming meter data into a defined database schema.

Depending on the scale of the meter deployment, there may be multiple instances of the SSN UIQ Oracle database collecting meter data.

The smart grid Data Warehousing solution adds the Greenplum database for long-term storage of meter data and provides a platform for analytics to be run on that data. The Change Data Capture process using Attunity Stream provides the mechanism to capture meter data being loaded into the UIQ, and replicate that data into the analytics platform.

Environment profile

Table 2 details the environment profile for the SSN Solution benchmark environment.

Table 2. Solution profile

Profile characteristic	Details
Network	• The connection between all servers in the environment is nominally a 1 GbE network.
Source database	• One (1) or more oracle databases comprise the source database. • Overall, approximately 95% of the changes to the source database are inserts (primarily of new meter data), and 5% are updates. A very small number of the changes are deletes. • Meter data must be available for analysis in the analytics platform no more than four hours after it arrives in the UIQ. • Peak updates to the Oracle database happen every 4 hours.
Analytics platform	• Greenplum DCA consists of two standard modules. • The Greenplum DCA must be sized according to ingest and query load for a specific meter deployment.

Hardware resources

Table 3 details the hardware used in the SSN Solution benchmark environment.

Table 3. Hardware components

Equipment	Quantity	Configuration
EMC Greenplum DCA	1	Configured with two standard Greenplum Database modules.
Oracle Server	1	4 CPUs, 64 GB RAM, 14 TB SAN storage
Attunity Stream Server	1	2 CPUs, 4 GB RAM, 144 GB SAN storage
Attunity Studio Client	1	1 CPU, 4 GB RAM, 20 GB SAN storage

Software resources

Table 4 details the software used to validate the solution.

Table 4. Software components

Software	Version
EMC Greenplum Database	4.0.3.0
Attunity Stream CDC	5.3.2.5 with patch p052_53243
EMC Outsourcer	1.9
Oracle	10*g* R2

Designing a smart grid analytics platform solution

The overall system design

The design of the solution is driven by the high-level requirements, which can be broken down into these main functional areas:

- Schema migration and initial data load
- Ongoing Change Data Capture
- Analytics enablement
- Data protection and disaster recovery

The solution development process consists of assembling and configuring the tools and processes, and then testing the resulting design to ensure that it meets the stated requirements.

The requirements for this solution represent a generalized smart grid analytics platform at the upper end of the scale of anticipated deployments by Silver Spring Networks. Any specific deployment will have different throughput, storage, and integration requirements and needs to be tailored to meet those requirements. This solution serves as a reference architecture to use as a baseline when designing and implementing an actual smart grid analytics platform.

Schema migration

In this solution, the analytics platform contains tables that are essentially clones of a subset of the source tables in the database that initially collects the meter data and metadata. Because the Greenplum database has different data types than the Oracle database, a process of mapping from Oracle data types to the Greenplum data types (Schema Migration) must be performed when creating the target tables in the Greenplum analytics platform. To efficiently create target tables with the proper data types, make sure the primary keys in the source tables are also the primary keys in the target table.

Challenges

Logically, the process of mapping data from the source schema to the target schema consists of looking at the DDL of the source table and writing a table definition in Greenplum syntax that matches the source. For each data type in the source table, you must use a corresponding data type in the target table definition.

Most of the source columns have data types that are relatively easy to map to Greenplum data types. Fields of type VARCHAR would typically map to TEXT or CHARACTER VARYING and numeric fields such as INT or NUMBER would map to similar types in Greenplum.

Constructs must also be added to the table definition to define the primary keys to match those in the source table.

Additionally, you need to define a distribution key for tables in Greenplum to ensure that the table is balanced across all the segments.

With more than a few tables, and for tables with many columns, this process becomes tedious and error prone. An automated process for constructing target tables with the proper mapping and design is required.

There are also cases where the data type mapping is not clear. In the SSN source database, several tables contain columns that are defined with the RAW data type. Normally, a RAW data type would map to a BYTEA data type in Greenplum. In some situations however, it may be necessary to map to different data types and perform a transformation of the data so that it will load into that type.

Initial data load

The second major functional requirement is the initial data load into the target database. The CDC process only tracks changes (inserts, updates, and deletes) from the point in time at which it is started and therefore does not "see" data that already exists in the source database. An initial data load must be carried out to seed the target database with the existing source data. From that point, the CDC process keeps the target database updated.

Requirements:

- Copy the data from the designated source tables to the target database.
- Convert the designated columns from RAW (binary) data to the target type.

Derived Requirements:

- Given the large number of tables and amount of data in the source database, the initial load process should be automated and repeatable in the event that the target database needs to be reloaded.
- Extracting data from the source database into flat files and then importing those flat files into the target database doubles the storage requirements of the solution. The initial load process should minimize the need to stage a copy of the data in the file system.

Challenges

Several source tables contain fields defined with the RAW data type but actually hold character data. For proper analytics, SSN needs to have these fields converted to specific data types in the analytics platform.

The most common way of performing an initial data copy from one database to another involves exporting the data from the source database into flat files in a file system, and then importing the data from the flat files into the target database. One problem with this approach is that the exported data requires large amounts of file space to be available to hold the files. You must then move this data over the network to the target database, potentially requiring even more space, and then load the data into the target tables. Managing these files can quickly get out of hand.

In addition, during system development, testing, and deployment, you may need to perform the initial load process several times. Therefore, the solution needs to provide an initial load process that is easily repeatable and does not require large amounts of disk space that ends up not being used after the solution is deployed.

Schema mapping and initial data load solution design

For schema mapping and initial data load, EMC has developed a Java software tool called Outsourcer, which automates the process of migrating a schema and data from a source database such as Oracle or SQL Server, into Greenplum. Outsourcer connects to the target database, examines the structure of the source tables, and then connect to the target Greenplum database and create the corresponding tables there. Outsourcer determines the proper Greenplum data type to match the source data type and also re-creates the primary keys in the target table, based on the primary keys in the source table. Outsourcer creates the table's distribution key based on the primary keys in the source table as well.

After creating the schemas, Outsourcer will perform the initial data load and copy the data from the source to the target.

Figure 3 shows the logical architecture of how Outsourcer performs data loading. Outsourcer uses a special feature of the Greenplum database called External Web Tables to stream the data from the source to the target over the network, avoiding the need to load the data into intermediate files in the file system.

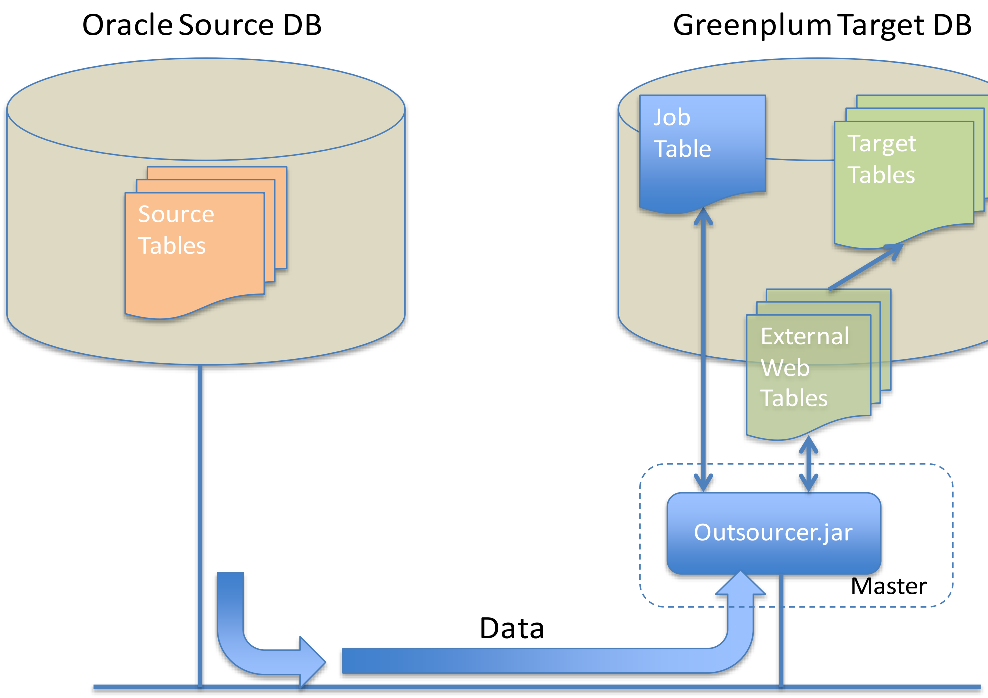

Figure 3. Logical architecture of how Outsourcer performs data loading

Outsourcer is controlled by, and uses, several tables that must be created in the Greenplum database under a schema named “os”. One of the control tables is a job table that defines which source database tables should be mapped, and the target database in which those tables should be created. The relevant fields are described in Table 5.

Table 5. Job table properties

Field Name	Description		
Id	Record ID to identify the jobs in the table.		
Source	A composite field comprised of values that describe the source table.	**Field**	**Description**
		Type	The database type of the source—‘oracle’ in this case.
		server_name	The IP or connection name of the source database.
		Port	The port to use to connect to the source database.
		database_name	The name of the database to connect to.
		schema_name	The name of the schema that holds the table to migrate
		table_name	The name of the table to migrate
		user_name	The username to use to login to the source database.
		Pass	The password to use to login.
Target	A composite field comprised of values that describe the target of the migration.	**Field**	**Description**
		schema_name	The name of the schema that holds the target table.
		table_name	The name of the target table.

Example entries in the os.job table are shown in Table 6.

Table 6. Example entries in the os.job table

Id	Source	Target
1	(oracle,10.10.1.1,1521,db1,schema1,tablea,orauser,pass)	(ssn,tablea)
2	(oracle,10.10.1.1,1521,db1,schema1,tableb,orauser,pass)	(ssn,tableb)
3	…	…

In this example, record 1 tells Outsourcer to connect to an Oracle database at the given IP address using the “orauser” user and “pass” password and copy the schema of “schema1.tablea” to “ssn.tablea” in the target Greenplum database. Connection information for the target database is not needed in the table because Outsourcer already has this information in order to be able to read the job table in the first place.

Using this job table, you can define schema migration "jobs" to tell Outsourcer which schemas to copy from the source to the target database.

Outsourcer does not handle certain source data types. For example, Outsource does not map RAW, BLOB, and CLOB fields to the target database because Outsourcer cannot necessarily make assumptions about the target type. The SSN source tables have a number of RAW fields that need to be mapped to the target tables. For these tables, a corresponding view, that casts these RAW fields into character types are created in the source database and Outsourcer migration jobs are configured to point to the view instead of the underlying table.

Change Data Capture

The process of tracking changes in the source database and copying them to the target analytics platform is called Change Data Capture, and is the main purpose of the solution.

Requirements:

- The solution must not impact Oracle source database performance. No triggers can be added to source tables to track changes.
- While most data is appends to the source tables, the solution must be able to process updates.
- The solution also must be able to process re-do log files. Data must be available in the target system within 4 hours of arriving in the source system. Ideally, this should be only minutes.
- CDC process must be able to be switched off and then resumed, and be able to catch up from up to 5 days of changes in the source.

Challenges

Change Data Capture is a much different process than bulk data load. Bulk data load consists only of inserts into the target database and the process can run in parallel across all target tables. On the other hand, with CDC, data must be inserted, updated, and deleted in the target database in the same order as was performed in the source. The sequential nature of CDC makes it harder to use Greenplum features such as parallel loading with gpfdist, which is more suited to bulk loading. Loading data changes through traditional SQL statements via the Greenplum Master however, does not provide the throughput to handle large transaction volumes.

One of the overriding requirements for the solution is to cause no performance impact on the UIQ database. This means that the solution must make use of Oracle transaction log mining instead of relying on database triggers.

CDC tools exist to extract relevant insert, update, and delete transactions from the re-do logs of the source database but they fall short of being able to efficiently apply those transactions to the target database.

The solution must be able to track transactions in the source database with no impact on production, and also make use of high-speed data loading while still keeping the transactions sequential.

Change Data Capture solution design

This solution uses Attunity Stream to perform the Change Data Capture process from the Oracle source database into the Greenplum target. Attunity Stream tracks changes in the source database tables. Depending on the type of the target database, it applies those changes in different ways.

For a Greenplum target database, Attunity Stream, by default, connects through ODBC to the Master and execute insert, update, and delete statements using that connection. However, given the volume of transactions, this approach will not meet the throughput requirements.

Attunity Stream provides the option to export the transactions as flat files instead of loading them through ODBC. Using this option, you can use Greenplum's gpfdist program to load large volumes of data into the database very quickly. There is a wrinkle in this option however that requires some additional "glue" to work around.

The transactions are captured from the source database and the transaction information is recorded in flat files. The transaction information contains information such as the timestamp of the transaction, the operation (insert, update, delete), and other "meta-data" about the transaction, in addition to the data itself.

For example, consider a source table with a structure and data such as that shown in Table 7.

Table 7. Example of the source table

RecordID	Column A	Column B
1	John	Smith
2	Mary	Jones

If the second record is deleted and a new record is added, it may look like Table 8.

Table 8. The changed table

RecordID	Column A	Column B
1	John	Smith
3	Bob	Walker

The transaction file generated from those changes may look like:

```
Timestamp, Transaction Type, RecordID, Column A, Column B
10:20:03,Delete,2,,
10:20:25,Insert,3,Bob,Walker
```

A flat file like this cannot simply be bulk loaded into the target table. The records in this file are not rows to be added to the table, they are instructions to carry out on the target table.

In order to properly apply the source database transactions and still take advantage of high-speed bulk loading, we developed a stored procedure and added it to the target database. This stored procedure creates an external table that maps to the transaction file and then walks through the transactions in the external table, applying them to the target table.

Figure 4 gives an overview of the CDC architecture.

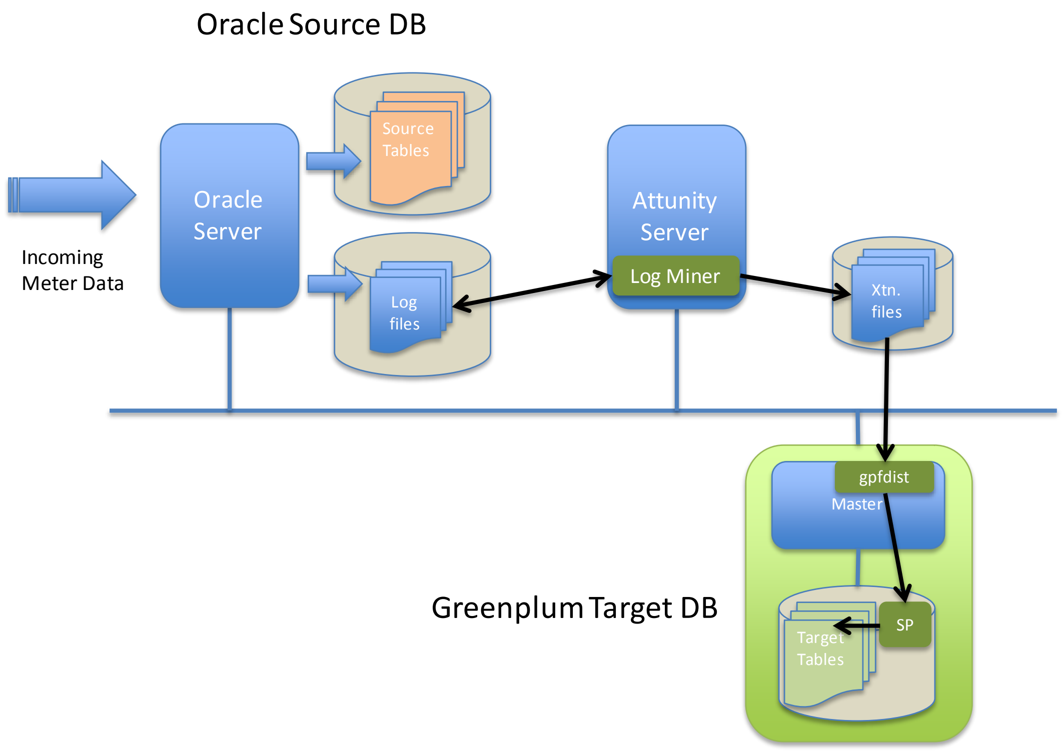

Figure 4. Overview of the CDC architecture

Smart grid data continually flows into the source Oracle database. Attunity Stream, monitors the Oracle re-do logs and captures changes to the set of tables that are designated to be copied into the analytics platform. Attunity Stream writes these changes as transaction files into the local file system. This file system is visible to the Greenplum master via an NFS mount. A script on the Attunity server consolidates the files and calls a stored procedure in the Greenplum database that creates an external table that maps to the transaction files and processes the transaction records to the target table(s).

Glue code

When Attunity Stream is configured to write the captured transactions to flat files, a separate process must be developed to move those transactions into the target database. For this solution, a Perl script was written that is executed periodically (from a cron entry) to collect the transaction files (multiple files are generated for each table) and pass them to a stored procedure in the Greenplum database. The high-level operation of the script is as follows:

For each table configured in CDC:

- Gather the .csv transaction files into one file
- Call the custom Stored Procedure to process the file
- Process the next tables' transaction files

The stored procedure

The custom stored procedure used to process the CDC transactions is called by the Glue Code for each transaction file to be processed. One transaction file represents the changes to that source table since the last cycle.

Because the transaction file may have insert, update, and delete operations, the stored procedure must be able to process each of these types of operations. Figure 5 illustrates the process of applying the transactions to the target tables using the stored procedure.

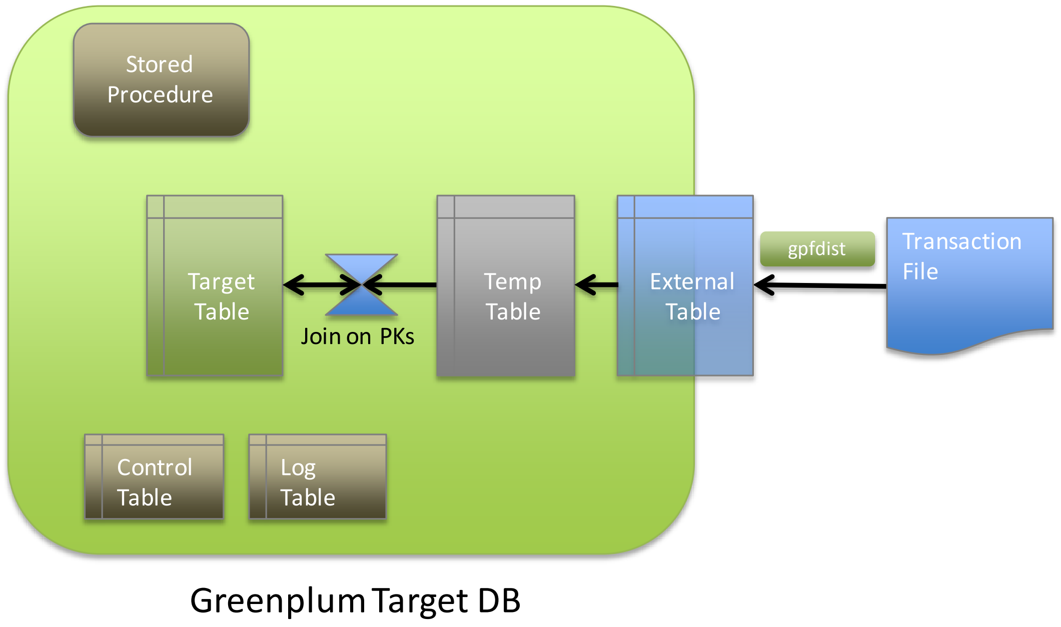

Figure 5. Process of applying the transaction to the target tables with the stored procedure

At a high-level, the stored procedure creates an external table that maps to the transaction file. It then executes a delete statement on the target table by joining the target table and the external transaction table on the primary keys where the transaction operation is either update or delete. It then executes an Insert statement on the target table by joining the target table and the transaction table on the primary keys and inserting the data from rows in the transaction table that are either Update or Insert.

The schema, the name of the target table, and the name of the corresponding transaction file are passed to the stored procedure. It is assumed that gpfdist is already running and pointing to the directory that contains the transaction files. The stored procedure performs the following steps:

1. Get the primary keys for the table from the control table that describes the target tables (configured in a solution setup step).
2. Create the join phrase of primary keys for the delete statement that will be built in a subsequent step.

3. Create an external table that maps to the table being loaded. The columns in the external table consist of the columns in the corresponding target table as well as the “transaction description” columns that exist in the transaction file.
4. Create a temporary table that matches the external table definition and bulk load (gpfdist) the transaction data into it.
5. Delete the records in the target table that have delete or update transactions in the temporary (transaction) table. This uses the join statement that was built above.
6. Execute an SQL statement that inserts records into the target table that have Update or Insert transactions in the temporary (transaction) table. This is a nested select statement that uses a look up table to call conversion functions for columns with certain data types. Ordering and ranking is performed on the records so that only the last update transaction of a particular item is performed.
7. Drop the external table and update a logging table with the number of errors (bad records in the external table) found.

This design allows CDC transaction data to be high-speed loaded into the Greenplum database using gpfdist, but still applied to the target table in the proper sequence with respect to inserts, updates, and deletes.

Analytics Enablement

The Greenplum database platform integrates with a number of Analytics and Visualization tools. Detailed information regarding specific tools can be found on EMC Powerlink. Specific information on analysis of smart grid data is outlined in a separate paper.

Data Protection and Disaster Recovery

Data protection for the Greenplum database is provided through integration with EMC Data Domain and documented in the Configuring EMC Greenplum DCA and Data Domain for Backup white paper located on Powerlink. Data Domain backups can be replicated offsite to provide for effective Disaster Recovery of the analytics platform.

Performance testing

Introduction

The two key performance measurements for this solution are the initial data load time and the throughput for ongoing CDC operations.

Notes

- Benchmark results are highly dependent upon workload, specific application requirements, and system design considerations and implementation. Relative system performance will vary as a result of these and other factors. Therefore, you should not use this workload as a substitute for a specific customer application benchmark when contemplating critical capacity planning and/or product evaluation decisions.
- All performance data contained in this report was obtained in a rigorously controlled environment. Results obtained in other operating environments may vary significantly.
- EMC Corporation does not warrant or represent that a user can or will achieve similar performance results expressed in transactions per minute.

Test objectives

The objectives of the test are to:

- Validate the design of the schema translation and initial data load between Oracle and Greenplum
- Determine the duration of the initial data load operation
- Validate the design of the CDC process
- Determine the throughput of the CDC process
- Provide some guidance for system sizing for a given artificial workload

Note that the validation lab environment was not built for "ultimate" performance. We use the real-world configurations which are more likely to match the systems that are in use in a typical data center.

Test scenarios

Table 9 describes the test scenarios.

Table 9. Test scenarios

Test scenario	Description
One	Perform a schema translation and Initial data load from the Oracle source database to the Greenplum target database. Validate that target tables were created properly and data was loaded without errors. Determine the throughput of the data load operation.
Two	Simulate incoming smart grid data in the Oracle database and validate the CDC process which moves that data to the Greenplum target database. Validate that insert, update, and delete transactions are properly replicated to the target tables. Measure the throughput to ensure it meets specified requirements.

Note that, as this is a one-time operation, the design favored ease of configuration and operation over loading performance.

Test procedures

To validate and measure the schema migration and initial data load, we initialized the Oracle source database with actual smart grid data. The Oracle server has 10 tables that are replicated to the Greenplum database.

The time it takes to replicate the schema and perform the initial copy is highly dependent on the number and size of the source tables. For this validation test, we replicated a selection of small, medium, and large tables and loaded them from Oracle into Greenplum. Table 10 shows the load times and throughput for each representative table size.

Table 10. Load times and throughput for each representative table size

Size	Rows	Bytes	Load time (hh:mm:ss)	Throughput (rows/sec)	Throughput (bytes/sec)
Small	5 M	443 MB	0:01:23	60 K Rows/s	5.3 MB/s
Medium	996 M	270 GB	4:00:55	65 K Rows/s	18 MB/s
Large	7.0 B	2.138 TB	27:30:36	70 K Rows/s	21 MB/s

Note that while the overall loading rate is relatively low compared to the maximum Greenplum ingest values (using Greenplum high-speed parallel loading with gpfdist for example), this process is a set-it-and-forget-it configuration. There are no intermediate steps of extracting the data into flat files, setting up ETL connections into Greenplum, creating external tables, or inserting from external tables into target tables.

These timings also include the export step from the source database, which is usually not factored into benchmark import tests. During these tests, we observed that the bottleneck on the ingest throughput is the process of extracting the data from the Oracle source and moving it across the network. The ingest into the Greenplum database was not the limiting factor.

Figure 6 shows the CPU activity on the Oracle source database during the load.

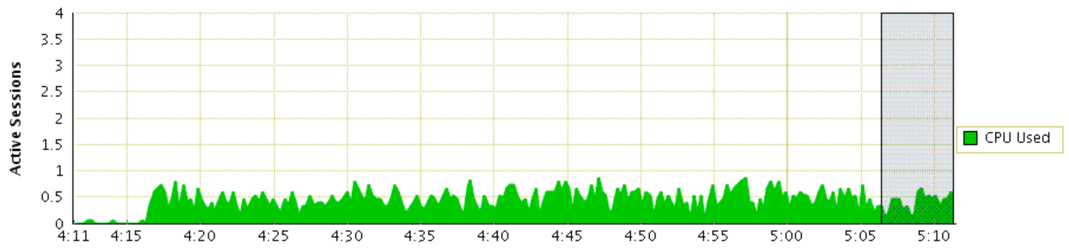

Figure 6. Load on Oracle server during initial data load process

Figure 7 shows the user I/O activity on the Oracle source database during the load.

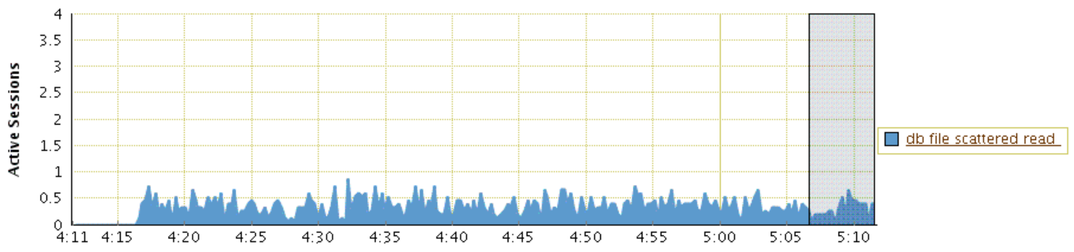

Figure 7. The user I/O activity on the Oracle source database

Figure 8 shows the disk I/O activity on the Oracle source database during the load.

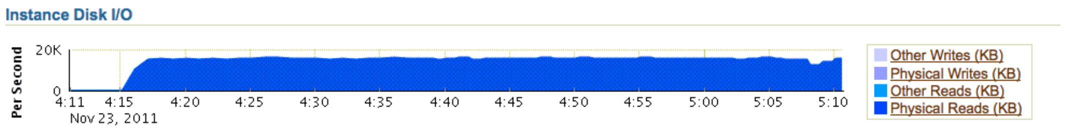

Figure 8. The disk I/O activity on the Oracle source database

To validate and measure the CDC process, we added simulated smart grid records to the Oracle source database.

In the smart grid environment, approximately 95 percent of the source database operations are inserts of new meter data being added to the database. Peak database operations occur every 4 hours (six times per 24-hour period), and last for approximately 1 hour. During this peak time, approximately 200 rows are loaded every second. In addition, two other tables are loaded with approximately 100 rows per second, and 50 rows per second respectively.

Thus, for performance testing, 350 rows of data per second are added across roughly three main tables. This rate is for a base configuration of approximately 685 K meters. To test the scalability of the solution up to approximately 8.2 M meters, we duplicated the base schema 12 times and loaded the simulated smart meter data to each of these schemas.

For purposes of testing, we used the peak insert rate to measure the performance of the system. Updates and deletes, were functionally validated but, as they represent a small fraction of the overall database transactions, were not factored into the throughput testing.

CDC throughput testing consisted of adding 1.2 million rows to the source table at different multiplication factors from 1 to 12. The data was added over the course of 1 hour, which represents a rate of approximately 350 rows per second for the base configuration.

Scale	Inserted rows/sec	Inserted rows/hr	MB/sec
685 K meters	350	1.2 M	201
8.2 M meters	4200	14.4 M	2419

As the records are added to the source database, the CDC process tracks the changes by monitoring the Oracle log files. The CDC process records the transactions into CSV flat files on the CDC server. As the CSV files are generated, a separate, EMC-developed, process bulk loads the CSV files into the Greenplum database to add them to the target table.

During testing, we found that the limiting factor in the throughput of the solution was the speed at which Attunity generated CSV files. In other words, after an hour of peak load on the source database, how much longer after that hour of loading did it take the CDC process to finish generating the corresponding CSV files for that load. Ideally, the last CSV file would be generated shortly after the end of the load period. For light loads, this was the case but as loads increased, this lag time also increased. This is the lag-time indicated in the tables of testing results below.

The process to bulk load those CSV files into the Greenplum target database and apply the changes to the target tables ran in parallel with the CSV generation process and easily kept up with the CSV files being generated.

The Attunity CDC program allows for different ways to configure the CDC jobs. You can configure a separate CDC job for each schema—resulting in 12 CDC jobs for the full scale test, or a single CDC job to track the source tables in all 12 schemas.

Figure 9 shows a separate CDC job configured for each schema (10 tables tracked in each schema).

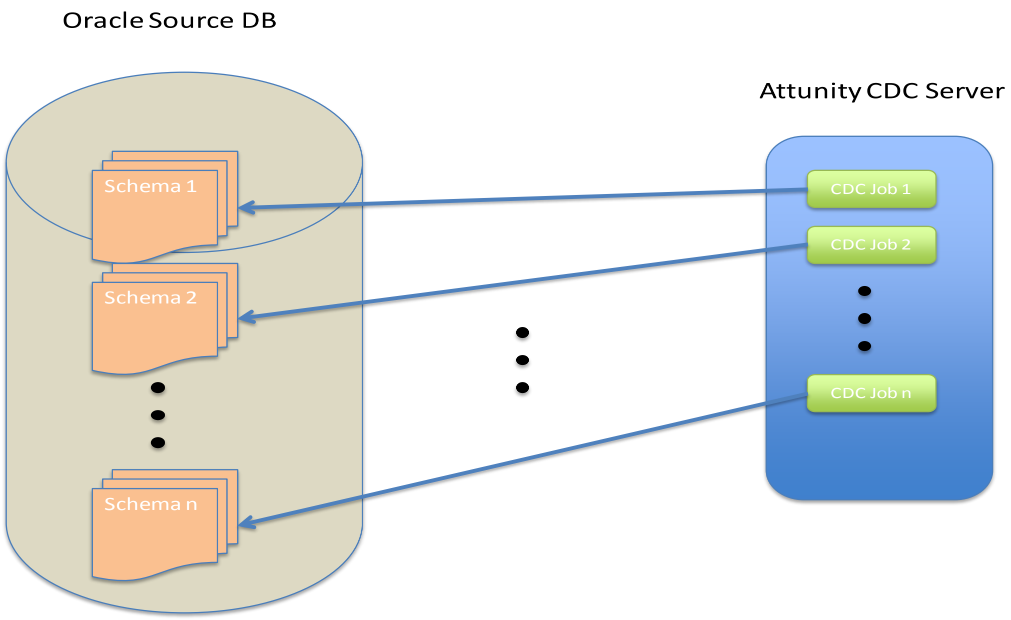

Figure 9. The configuration where a separate CDC job is configured for each schema

This method might be considered the "normal" way of configuring CDC jobs and seems to work well for small numbers of CDC jobs. But, as the number of schemas, and therefore CDC jobs, was increased, the impact of these separate Log Miner jobs on the Oracle source database became significant.

Figure 10 shows the impact on the Oracle source database server of adding 12 CDC jobs one by one. Note that no data loading was occurring during this time. This represents CDC jobs simply starting and looking for changes in the source database.

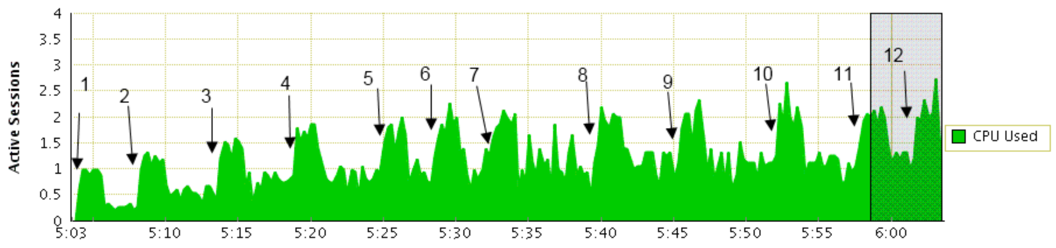

Figure 10. The impact on the Oracle source database server

Table 11 shows the timings and results of loading different amounts of meter data into the source database and tracking the lag time of CSV file generation. As shown in Table 11, loads up to 5.4 Million meters resulted in no CSV generation lag. That is, the CSV file generation process kept pace with the loading process such that they both finished at the same time. Beyond this number of simulated meters however, timeouts and failures in the CDC jobs were encountered. The load on the Oracle database was such that the log mining process would time out and fail to capture the change data.

Table 11. Timings and results

Meters Simulated	Rows loaded	Load Duration	CSV Generation Time	CSV Latency (*)
685K	1,200,000	1:03	1:03	0:00
1.3 M	2,400,000	1:03	1:03	0:00
2.7 M	4,800,000	1:03	1:03	0:00
4.1 M	7,200,000	1:03	1:03	0:00
5.4 M	9,600,000	1:03	1:10	0:07
6.8 M (High load on Source Database. CDC failures encountered)	12,000,000	1:03	1:18	0:15
8.2 M	N/A	N/A	N/A	N/A

* CSV Latency is the difference in time between Load Duration and CSV Generation Time.

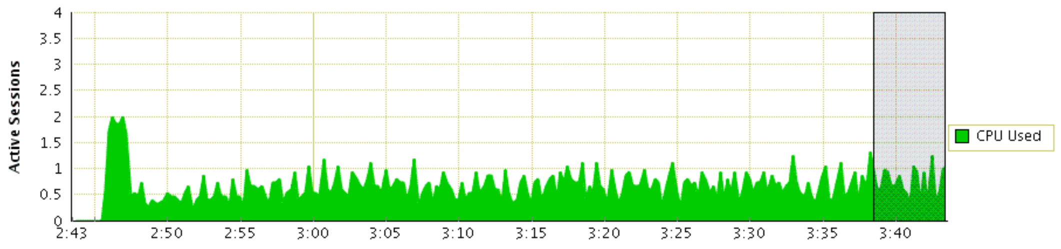

Figure 11. Loading 685 K meters

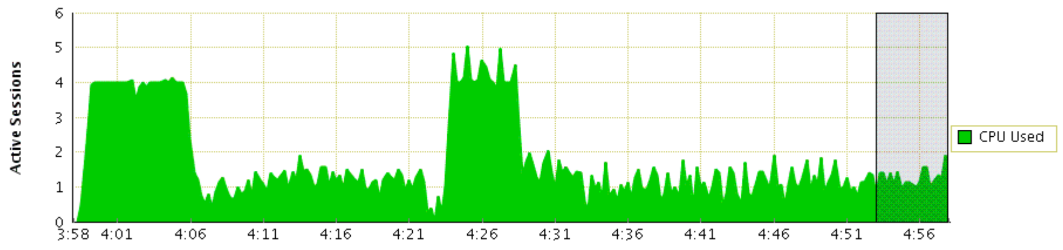

Figure 12. Loading 1.3 M meters

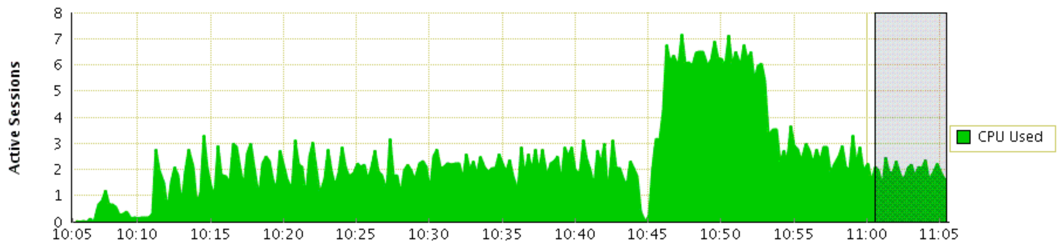

Figure 13. Loading 2.7 M meters

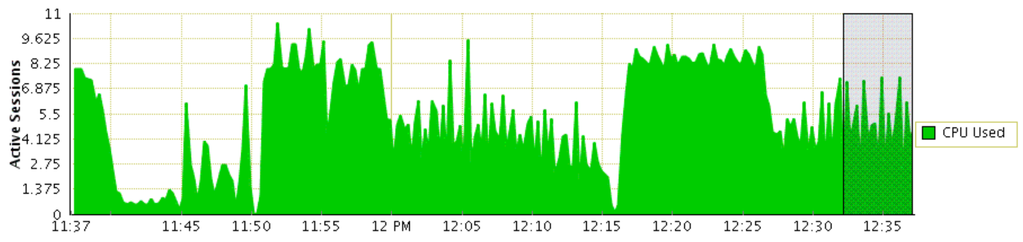

Figure 14. Loading 4.1 M meters

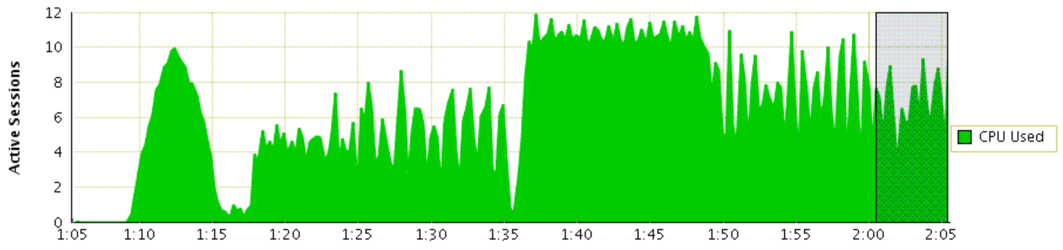

Figure 15. Loading 5.4 M meters

An alternate CDC configuration is to have one CDC job (or at least fewer jobs) track the changes in all the source tables in all the schemas, as shown in Figure 16.

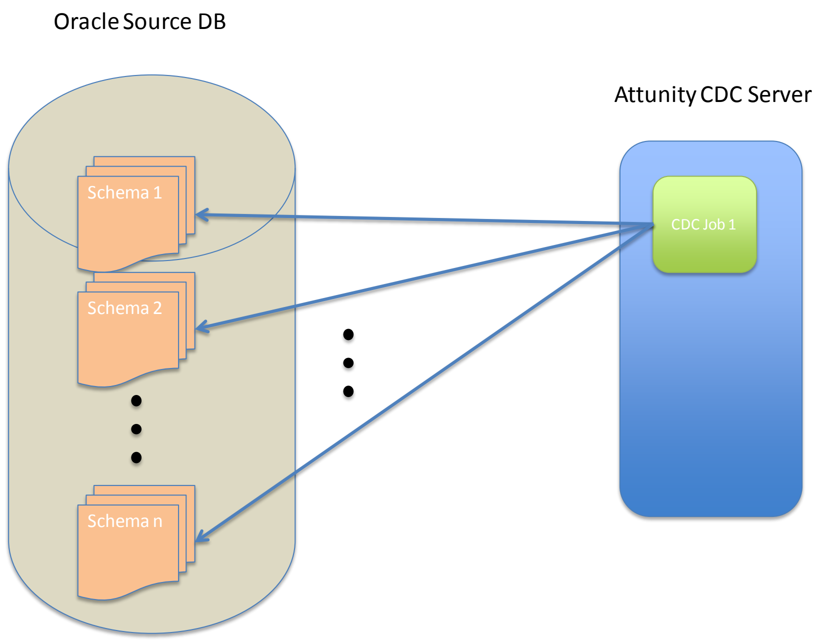

Figure 16. An alternate CDC configuration

This method resulted in much lower loads on the Oracle source database, but also caused lag time in CSV file generation because the lone CDC job had to process the log files for all the schemas.

Table 12 shows the testing results for increasing the number of segments being loaded and tracked by one CDC job.

Table 12. Testing results for increasing segments

Simulated meters	Rows loaded	Load duration	CSV generation time	CSV latency
1.3 M	2,400,000	1:00	1:00	0:00
2.7 M	4,800,000	1:00	1:04	0:04
4.1 M	7,200,000	1:00	1:32	0:32
5.4 M	9,600,000	1:00	2:03	1:03
6.8 M	12,000,000	1:00	2:35	1:35
8.2 M	14,400,00	1:00	3:08	2:08

Figure 17, Figure 18, and Figure 19 show the load on the Oracle source database for a selected numbers of meters loaded.

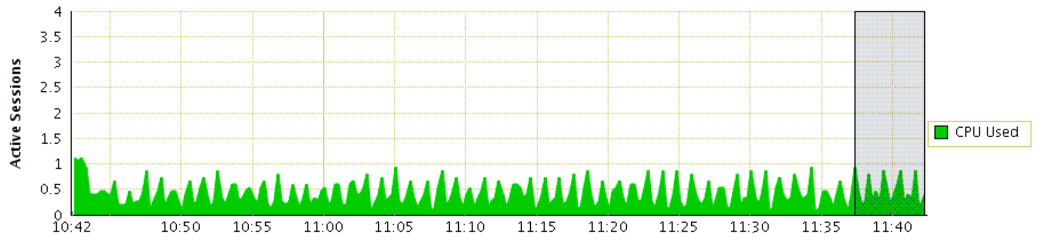

Figure 17. Loading 1.3 M meters

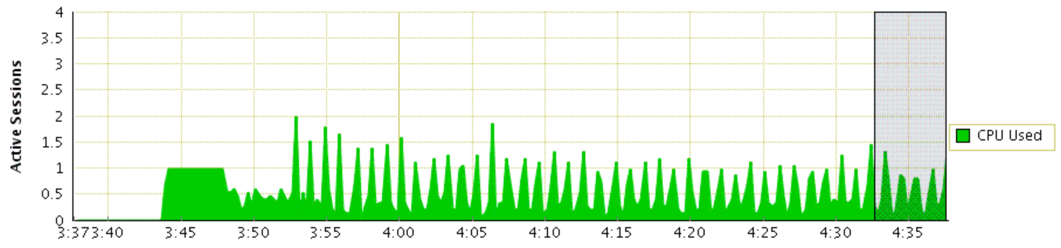

Figure 18. Loading 5.4 M meters

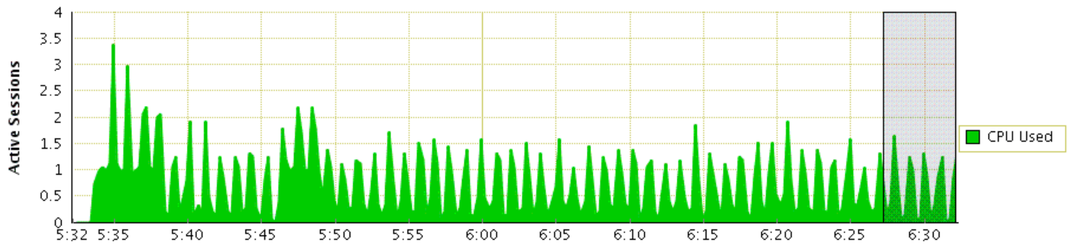

Figure 19. Loading 8.2 M meters

At the maximum load of over right million meters), the CSV lag time is just over 2 hours. This means that after loading for 1 hour, it takes an additional 2 hours for the CDC process to finish generating the CSV files containing the data loaded. This is within the window of 4 hours between meter loading sessions but it does have implications for situations where the CDC process is paused or otherwise unavailable for some amount of time. If the CDC process is paused – due to maintenance or some other reasons, changes in the source database will accumulate. When the CDC process is resumed, it will take time to generate the additional CSV files and "catch up" with the latest meter data being loaded. In the case of prolonged interruptions, it may be more efficient to simply perform and initial load again and resume CDC from that point.

To gauge the effect of dividing the CDC process among more jobs, additional configurations were tested. Table 13 shows results for two CDC jobs, each tracking two schemas.

Table 13. Results for two CDC jobs

Schemas	Simulated Meters	Rows loaded	Load Duration	CSV Generation Time	CSV Latency
2	1.3 M	2,400,000	1:00	1:00	0:00
4	2.7 M	4,800,000	1:00	1:00	0:00
6	4.1 M	7,200,000	1:00	1:00	0:00
8	5.4 M	9,600,000	1:00	1:15	0:15
10	6.8 M	12,000,000	1:00	1:35	0:35
12	8.2 M	14,400,00	1:00	1:50	0:50

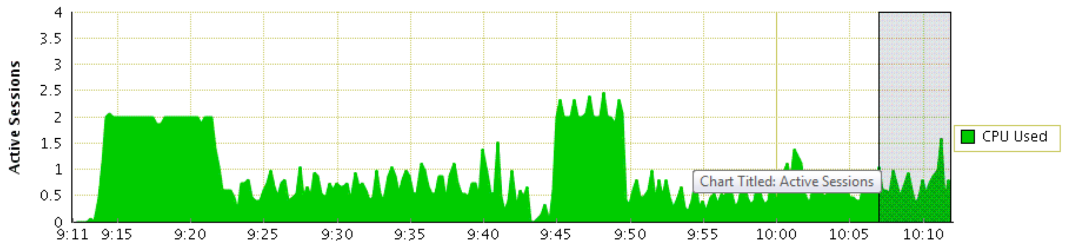

Figure 20. Loading 1.3 M meters

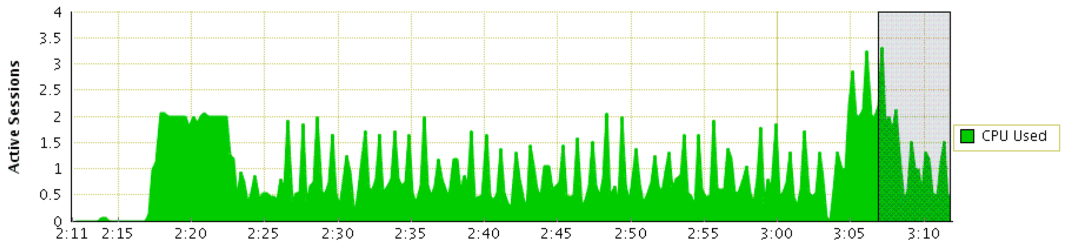

Figure 21. Loading 5.4 M meters

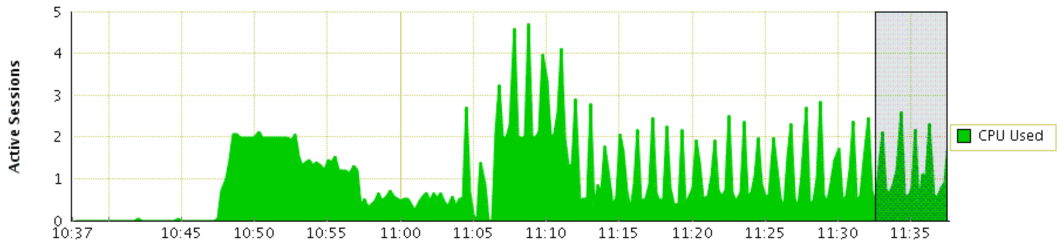

Figure 22. Loading 8.2 M meters

Note that the maximum lag time is reduced to 50 minutes in this configuration, but there is a corresponding slight increase in source database CPU usage.

In another configuration, we configured three CDC jobs, each tracking a maximum of four schemas. The results are shown in Table 14.

Table 14. Results for 3 CDC jobs.

Schemas	Simulated meters	Rows loaded	Load duration	CSV generation time	CSV latency
6	4.1 M	7,200,000	1:00	1:00	0:00
9	6.1 M	10,800,000	1:00	1:05	0:05
12	8.2 M	14,400,00	1:00	1:25	0:25

Note that to create a balanced set of meters across three jobs, the eight and 10 schema tests were replaced by a nine schema test.

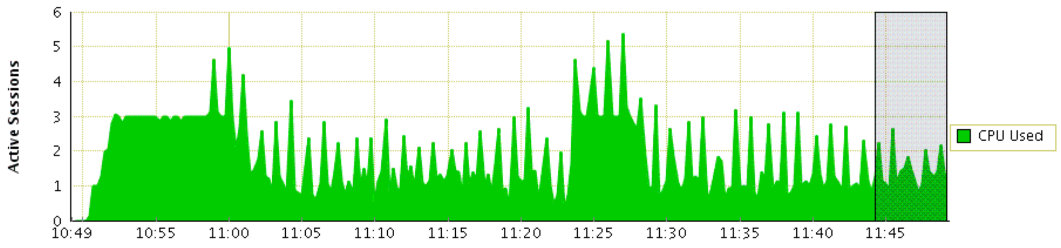

Figure 23. Loading 6.1 M meters

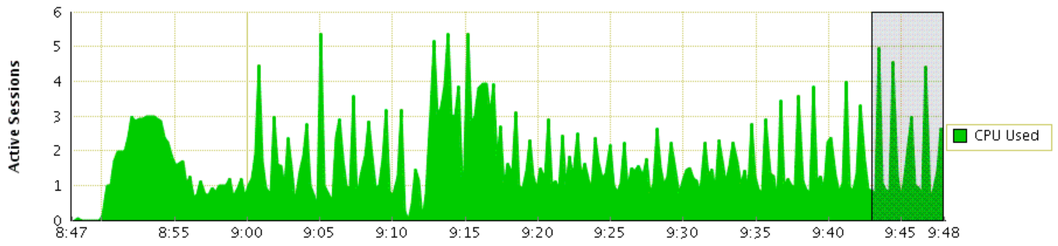

Figure 24. Loading 8.2 M meters

Figure 25 summarizes the results of the tests using three different job configurations.

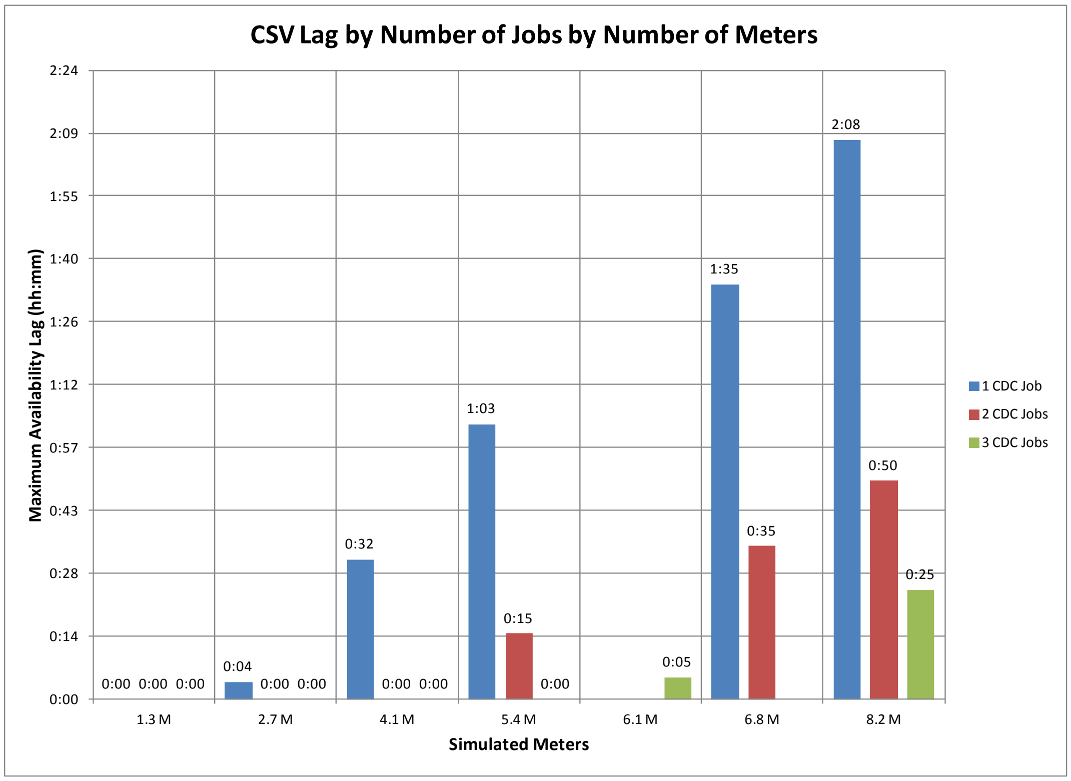

Figure 25. Results of the tests using three different job configurations

Conclusion

Summary

EMC and Silver Spring Networks have teamed to develop an adaptable solution that provides a robust analytics platform for smart grid data that enables utility companies to derive value from their smart grid deployments.

Findings

The solution provides:

- Automated schema translation and creation from Oracle to Greenplum
- Automated initial data load from source to target without requiring intermediate file space and manual manipulation of flat files
- Automated Change Data Capture that tracks changes in the Oracle source database and applies those changes to the target Greenplum database
- A platform for Analytics and Business Intelligence using in-database analytic functions as well as third-party tools

References

EMC white paper

For additional information, see the white paper listed below.

- *EMC Greenplum Data Computing Appliance: Performance and Capacity for Data Warehousing and Business Intelligence—A Detailed Review*

EMC product documentation

For additional information, see the product documents listed below.

- *Greenplum Database 4.1 Administrator Guide*
- *Greenplum DCA Getting Started Guide*
- *Data Sheet: EMC Greenplum Data Computing Appliance*