Astera
DATA INTEGRATION MADE EASY

Centerprise
Data Integrator

Centerprise Best Practices

Working With the High Volume Data Warehouse

Overview

Data warehouses and data marts provide the business intelligence needed for timely and accurate business decisions. High quality extract, transform, and load (ETL) functionality is vital to the success of any data warehousing project. The ability to pull data from a variety of application silos, cleanse, transform, and aggregate data in a timely fashion establishes your data warehouse as a reliable tool upon which your organization can depend for decision-making insight.

Centerprise is the ideal solution for transferring and transforming records from a transactional database to the data warehouse. It provides all the functionality needed for today's demanding data storage and analysis requirements, including sophisticated ETL features that ensure data quality, superior performance, usability, scalability, change data capture for fast throughput, and wide connectivity to popular databases and file formats.

Data warehousing comes with its own unique set of challenges, mostly revolving around the sheer volume of data as well as key maintenance. This best practices paper provides best practices to be kept in mind during the entirety of the development process in order to make certain data warehousing projects will be successful.

Data Quality

High performance data warehousing is all about achieving speed and scale while effectively managing increasing data complexity. Data warehouses typically deal with large volumes of data and any sort of scan or write is prohibitively expensive, making it crucial to ensure that it is done in the best and most efficient way possible. Consequently, data quality has become a key piece of the data warehouse solution and is often a barrier to successful warehouse and analytical implementations.

An important question that must be asked before making an initial effort to migrate from the online transaction processing (OLTP) system to the online analytical processing (OLAP) system is what kind of shape the data being moved is in. Since the whole point of most OLAP systems is to facilitate business decisions, it is best to identify data problems as soon as possible and it is for this reason, we recommend making liberal use of Centerprise's built-in data quality components.

Data Quality Rules Action

Centerprise's data quality rules action is designed to flag records that do not conform to a set of pre-established business logic rules. Records passing through these rules will be evaluated according to each rule and be marked as either a warning or, if applicable, an error.

The data quality rules action is typically used for two purposes: data profiling and validation. It can be found in the "Data Quality" section in the tool box. You'll want to place the action right after a data set; typically, between a source and target. You can also add more than one rule per action. While this will not affect performance, it will increase the readability of your diagram.

Data Profiling

Data profiling can be useful aid during the development phase as well as a first-class dataflow producing reports detailing the health of a data source.

A good idea is to create a dataflow with just the source and a data-quality rule action on it. Create a list of rules in the Data Quality Rules action based on a series of assumptions about the data. For example, I might need to load a fact table with product dimensions of category, name, and variation based on a source field expected in a format of ProductCat_Name_SubType. I first run my source data-set through my data quality rule of IsMatch(@"\w+_\w+_\w+") . Then, I attach the Data Profile transformation at the end. This will give a lot statistical information about this dataset, including how many records have failed my data check. Let's say that half of my records have errors. At this point, I may need to change my flow to handle this type of data instead of simply rejecting it.

Validation

No matter how well you prepare for exceptions, bad or unexpected data will come and it is a good idea validate key data fields before records are written to your target tables. Validation works exactly the same way as data profiling. The only difference in this case is instead of using the "Quick Profile" command or writing to a report file like Excel, you place the actual target you are writing into after the Data Quality Rule action. Everything else remains the same. The benefit is that records that do not meet business requirements do not poison your data warehouse.

Logging

Once you've incorporated data quality rules into your data flows, you'll most likely need to keep tabs on the records that are automatically filtered out due to a failed data quality rule. After all, records that are omitted are often still causing your reporting in the OLAP system to be erroneous. There are two ways to handle logging in Centerprise: using the built-in record level log or writing records to an alternate destination.

Record-Level log

Using the record level log is the easier option. It is ready to go right out of the box. There is not much configuration needed. Just connect to your target output and set the location where you wish the log to end up. Upon your run, you will see (by default) all records that were discarded along the way along with the reason why. However, the preferred approach here at Astera is to create your own custom log.

Custom Log

A custom log is nothing more than an alternate target in your dataflow. It can be file or database table, but we recommend a table as multiple runs will be more manageable. To attach a custom log, simply drag and drop a "Destination" from the toolbox onto the end of your flow. To get only the erroneous records, you'll have to split your data flow path in two.

Divide and Conquer

You'll want to "divide and conquer" your data by utilizing a route transformation to separate your data stream into records that can go straightaway from records that have data quality deficiencies that require further processing. Within this group of records, you can break data down even further into more specific buckets so that you can make some assumptions about data in these temporary containers. This can greatly simplify expressions in Centerprise. Bad data falling into one of the deficient buckets can be corrected and later re-joined to the normal path using a "Union" transformation.

Translating Into Star Schema

While data quality rules may be options, translating a relational schema into a star schema is something that is done in every OLTP to OLAP data transfer. The biggest hurdle you will face is keeping relationships intact while migrating between the two systems.

Dealing With Foreign Keys

Loading fact tables typically involves writing a couple of facts about the transactional record along with a large amount of foreign keys to related dimension tables. These keys will invariably be unavailable from the source data set and a lookup will be required. The lookup, in this case, can be the biggest pitfall in this entire process. Too many lookups can either make the diagram hard to read and maintain, or worse, slow the flow down to an unacceptable performance level.

Reduce the Number of Lookups

Consider using joins instead of lookups. Before placing a lookup on the diagram, ask yourself the following question: how unique are the values going into my lookup? If the values are pretty unique and you have a large data set transferring, you may want to consider using a join transformation. Using a join transformation achieves the exact same thing as a lookup, but in many cases appears cleaner and in some cases may actually be a boon for performance compared to lookups that do not have caching enabled.

Load Lookups Up Front

If using lookups, consider using the "load all records at start." This will incur an initial overhead action of selecting all records from the lookup table, but is often more desirable than continuously chatting with the OLAP to retrieve a single value. Do not use this option if the amount of records being transferred is very small and the lookup table is very large. In this case, the cost of initialization might not be worth it.

Options for Related Tables

Besides key resolution, another key challenge when doing this kind of transfer is dealing with late arriving dimensions. Handling this situation correctly requires writing to the dimension table before inserting into the fact table. To do this, there are several options available to you.

Pass Through Needed Table

The first option is to pass the record directly through the dimension table. The idea here is to utilize the identity column (or sequence object) properties to generate a new primary key for the newly-inserted record and use that value to populate the fact table.

Note that in order to use this method, there are a couple of properties that must be set. First, you must divide the records that insert a record into the dimension table from those that do not. This is typically done with a router after a lookup on the dimension table produces no value. Second, you must have the database writing option for the dimension table set to "single record insert." This will return nothing in the case of a bulk insert.

The downside of this approach is that it is considerably slower than using a bulk insert. In many cases dealing with records in millions, it is prohibitively slow and is not recommended. The biggest benefit of this approach is that since the identity column is being used to get the dimension key, referential integrity on the data warehouse is virtually guaranteed.

Dynamic Lookup

The second option for late arriving dimensions is to use the "dynamic" caching option for a database lookup. In essence, this lookup will retrieve a required dimension key and for the late arriving dimension that is not there yet, it will create one for you. You can then load the fact table along with the dimension table at the same time.

We recommend using the "Load all Values" at the start for this scenario as this will keep you from having to recalculate the next id for each lookup. A limitation of this approach is that it only works with integer keys.

Lastly, make sure to check the "Use Constraint Base Write." This will ensure that all of the dimension inserts will happen before the fact table inserts. Keep in mind that this will all happen in one go.

Load tables in two different flows

The last option is the easiest. Just process the source data twice; once for dimensions and again for facts. This has the benefit of being incredibly simple and making dataflows very straightforward.

Dimension Key Maintenance

Another common task besides inserting into fact and dimension tables is to update dimension tables. Due to the nature of dimension tables, you cannot update just a single record. You must also update plus insert in the case of an SCD type 2 operation. Be sure to make use of Centerprise's slowly changing dimensions (SCD) component for these scenarios. It will archive the existing record and create a new record suitable for inserting. You can learn more about Centerprise SCD at https://www.youtube.com/watch?v=QTuxKCGKlvM).

Performance Considerations

Finally, due to the volumes of data typically involved when moving data from an OLTP system to an OLAP one, you'll have to be continually cognizant of potential performance bottlenecks and what is optimal. Here are some best practices for what to be particularly aware of.

Same Network

The most obvious potential source of poor performance is the network itself. Because of how Centerprise sits as middleware between the source and the target, be aware of the costs of moving large amounts of data over a network. This may eat up large amounts of bandwidth and possibly be feasible to do only during off-peak hours. Also, it is advisable to have Centerprise as well as the source and target all on the same network. Transferring over a WAN or VPN is not recommended. In fact, if network bandwidth is a major concern, you may wish to look at deploying Centerprise on the same machine as either the source or target.

Keep an Eye on Queries

Another common issue with these projects is the amount of data being retrieved from the source. If, for example, there is a filter being used on the diagram to filter out a large part of the source data set, the amount of data being pulled and traveling across the network does not change. In this case, a full table retrieval is being performed. It is therefore recommended that you filter out unneeded data via the database query itself. You can do this using the WHERE clause section of the database table source. Use the $() syntax to make use of parameters that can be controlled from an outer workflow. For example, WHERE startDate > '$(myVariable.Start)' and endDate < '$(myVariable.End)'.

Make Use of Caching

Caching is extremely important in Centerprise. However, there is nothing that will slow down a data transfer more than an incorrect caching setting.

Art Form

But finding the "correct" setting is often more of an art form than a definite directive. It often depends on the data itself being the driver for when to use certain cache options. For example, if the source data being used to look up items is not very distinct, using the "load all records up front" option is a waste. If the source data is fairly distinct or there are enough records, we recommend using the "load up front" option.

Persisted

Due to the nature of star schemas, you'll find yourself using the same dimension lookups over and over again. In this case a best practice is to utilize the "persisted" cache option. This will store the results of the lookup on disk in a more permanent location. Upon subsequent runs of other data flows using the same lookup, it will use the local cache instead of heading back to the database. If you use this option, you will typically want a lead data flow that will refresh the cache before all other data flows in a project are set to use it.

Use the “In Database Join” Option

When using Centerprise join actions to join multiple servers, consider leveraging the “Join in Database” option. You can find this in the options screen of the Join transformation. This will prompt Centerprise to build a complex SQL query that will perform the join on the database server rather than fetching both datasets and performing the join in Centerprise. If this join cuts down on a significant amount of records, this could be a huge win for the efficiency of your flow.

ELT

There are some exciting new features coming to Centerprise that will further optimize the time needed to complete complex data warehouse jobs. Among these is the new “Pushdown” option. This option will allow you to move many of the transformations and expressions to the database server itself, thus bypassing the loading of entire streams of data into the Centerprise server. This has the potential of reducing load times by orders of magnitude. Be on lookout for this feature in Q4 of 2015!