

SESUG Paper SD-188-2017**ETL Load performance benchmarking using different load transformations in SAS® Data Integration Studio.****Sai S Potluri, Synectics for Management Decisions.****ABSTRACT**

This paper is primarily intended to provide some helpful insights to the developers in loading large volumes of data into an Oracle Database by carrying out a performance benchmarking between different load transformations available in SAS® Data Integration Studio. The performance benchmarking for bulk loads has been performed on the following Load Transformations using different load styles available with them.

- Oracle Bulk loader
- Table loader

The performance comparison for update operation has been performed between the following transformations

- Merge
- Table Loader

INTRODUCTION

While designing the ETL process, you may need to implement different load strategies based on the amount of data that needs to be processed. Typically, below are the two categories of loads that needs to be processed by the ETL to load the data into the Data warehouse.

1. Incremental data – This is to process the daily delta/incremental data only. The data processed here typically is very less, could be several thousands of rows daily.
2. Historical Loads- The data that is processed here can be very large. You may need to set up Bulk Loading strategy here.

In this paper, we will navigate through the different bulk loading transformations available in SAS® Data Integration Studio and run a performance benchmarking comparisons between them.

PERFORMANCE COMPARISON

To run the performance benchmarking on different Load Transformations in SAS® Data Integration Studio, we have taken a Stage Table that has 13,695,436 rows and executed the loads into an Oracle table using different Load Techniques. The same tables have been used across all different load transformations. The Performance benchmarking is done based on the Run Time, CPU time, I/O operation and Memory usage statistics for each Load Transformation. Each Load transformation/Type is run 10 times for different data volumes (25%, 50%,75% of 13,695,436 rows) and an average CPU time, I/O and Memory usage has been taken. Run time statistics were collected from the SAS DI after the Job completed successfully.

ENVIRONMENT DETAILS:

The performance benchmarking explained in this paper has been done in the following environment. The performance results might vary based on different environments and their configurations.

- SAS Data Integration Studio 4.9
- SAS 9.4 (TS1M2)
- Red Hat Enterprise Linux Server release 6.9

ORACLE BULK LOADER

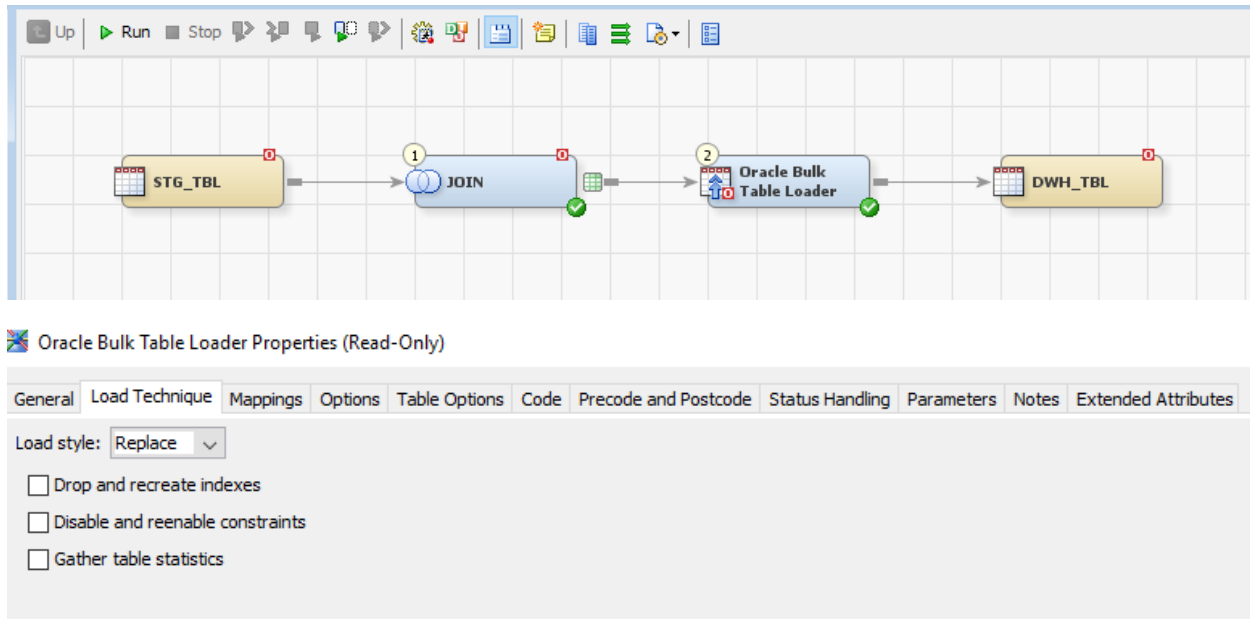
When you have a large volume of data that needs to be processed fast and efficiently to meet the SLA (Service Level Agreement), Bulk Load through the SAS/ACCESS interface is the best option. Bulk Loading is the fastest way to transmit large amounts of data to the DBMS. When the Oracle Bulk Loader transformation is used, the option BULKLOAD=YES will be set. This will make SAS/ACCESS interface to launch the Oracle SQL*Loader (SQLLDR) which will be invoked with direct path. You can further boost the bulk load by leveraging the native oracle options. But, one needs to have a clear understanding of these options and need to be tried in the supervision of a DBA. In the below example, a stage table has 13,695,436 rows. This data has been read and loaded into a SAS Dataset which is of 4.8

GB and is finally loaded into an Oracle table using an Oracle Bulk Table Loader with different load techniques. Below are the different load techniques that were used in the Oracle Bulk Loader for performance comparison.

- Oracle Bulk Loader with Load Style Replace.
- Oracle Bulk Loader with Load Style Truncate.
- Oracle Bulk Loader with load style Truncate and disabling the constraints on the table.
- Oracle Bulk Loader with Append Mode.

Load Technique: Oracle Bulk Loader with Load Style Replace

In this load method, the BL_LOAD_METHOD for the bulk load is set to REPLACE and the data is loaded using the Oracle SQL*Loader with direct path.

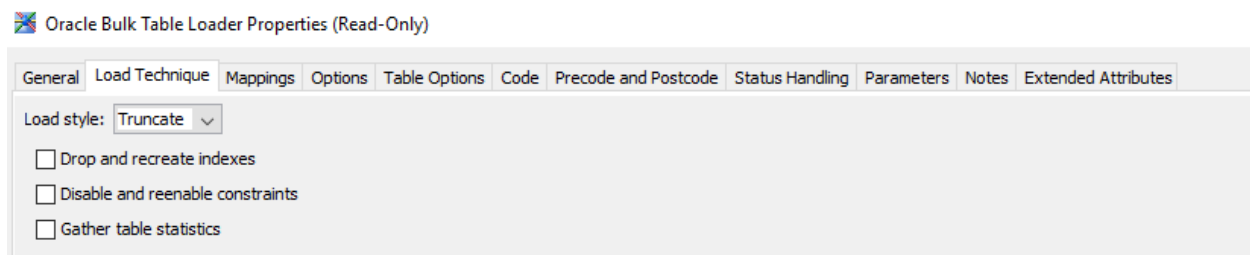


The screenshot shows the Oracle Bulk Table Loader Properties dialog box (Read-Only) with the 'Load Technique' tab selected. The 'Load style' is set to 'Replace'. Below this, there are three unchecked checkboxes: 'Drop and recreate indexes', 'Disable and reenale constraints', and 'Gather table statistics'. The background shows a data flow diagram with components: STG_TBL, JOIN, Oracle Bulk Table Loader, and DWH_TBL.

Display 1. Oracle Bulk Loader with Load Style Replace

Load Technique: Oracle Bulk Loader with Load Style Truncate

In this load method, the BL_LOAD_METHOD for the bulk load is set to Truncate and the data is loaded using the Oracle SQL*Loader with direct path.

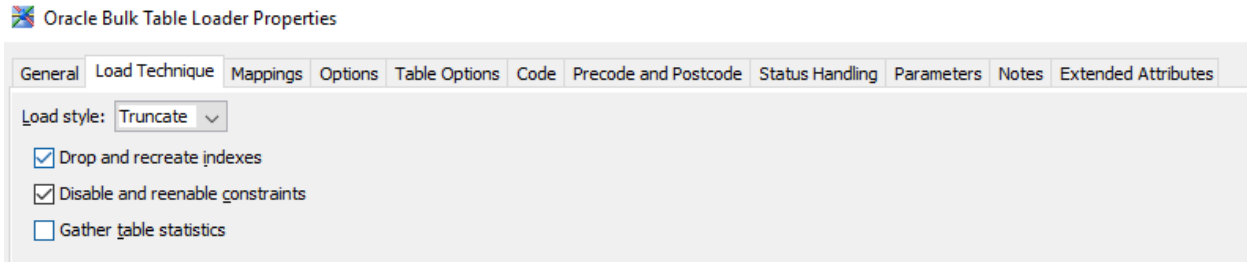


The screenshot shows the Oracle Bulk Table Loader Properties dialog box (Read-Only) with the 'Load Technique' tab selected. The 'Load style' is set to 'Truncate'. Below this, there are three unchecked checkboxes: 'Drop and recreate indexes', 'Disable and reenale constraints', and 'Gather table statistics'.

Display 2. Oracle Bulk Loader with Load Style Truncate

Load Technique: Oracle Bulk Loader with load style Truncate by disabling the constraints on the table

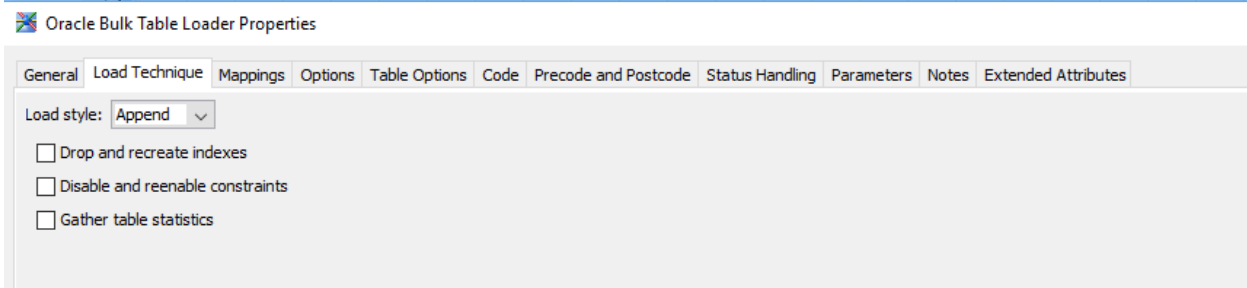
In this method, the bulk load is performed by disabling the Constraints, Indexes before the load and enabling them after the load is complete. In this case too, BL_LOAD_METHOD is set to Truncate and the data is loaded using the Oracle SQL*Loader with direct path.



Display 3. Oracle Bulk Loader with Load Style Truncate with Disable and Enable Constraints

Load Technique: Oracle Bulk Loader with Append Mode

In this Load Style, BL_LOAD_METHOD is set to Append and the data is loaded using the Oracle SQL*Loader with direct path.



Display 4. Oracle Bulk Loader with Load Style Append

TABLE LOADER

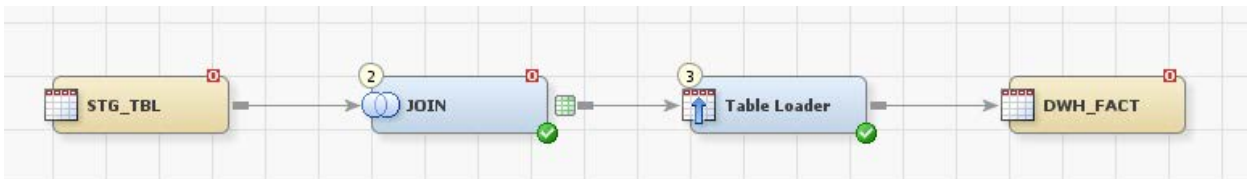
Table Loader is another transformation in SAS DI studio that can be used to perform inserts and updates on the data. It uses PROC APPEND to insert data in the target table. In this section, we are only interested in the bulk loads using Table Loader in two different load styles. and a

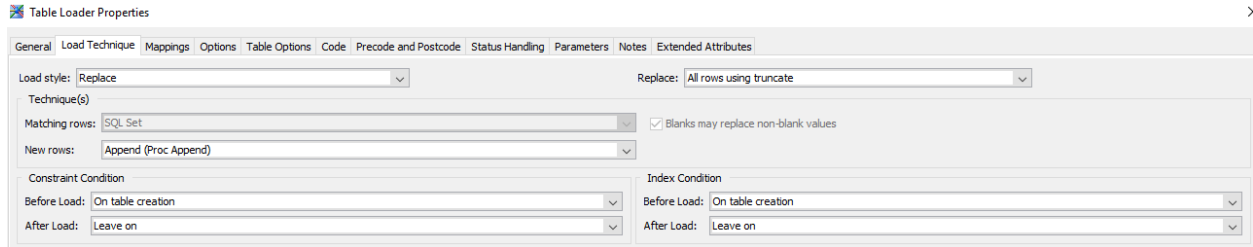
1. Table Loader with Replace Load Style.
2. Table Loader with Append to existing Load Style.

The performance comparison of the above two load styles is done with Oracle Bulk Loader.

Load Technique: Table Loader with Replace Load Style

In this load style, the target table is truncated and the load is performed using SAS PROC APPEND. In this method, the target table is completely rebuilt from the source tables. Any historical data in the target table will be lost. Use this load type, when there is no need to store the historical data or to bulk load an empty target table.

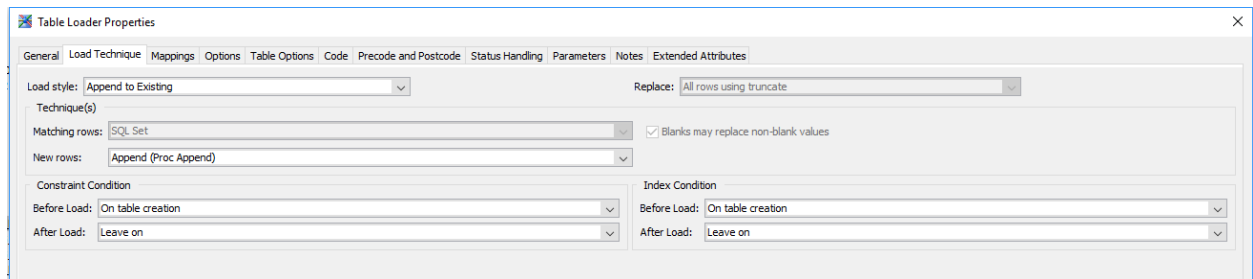
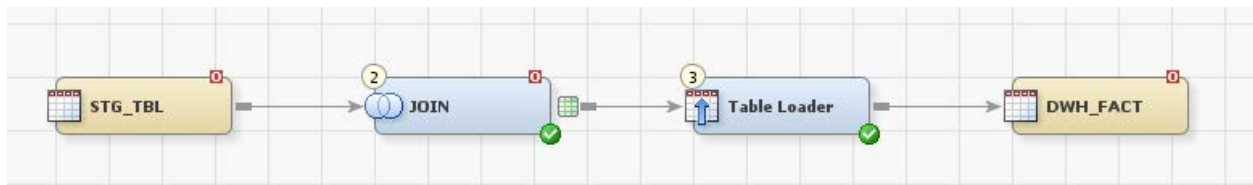




Display 5. Table Loader with Replace Load Style

Load Technique: Table Loader with Append to Existing Load Style

In this load style, the target table is loaded using SAS PROC APPEND. This load style appends data to the target table.



Display 6. Table Loader with Append load Style

Below are the short abbreviations used for different load styles in the line graphs

OBL-Append: Oracle Bulk Loader with Append Load Style

OBL-REPLACE: Oracle Bulk Loader with Replace Load Style

OBL-TRUNCATE: Oracle Bulk Loader with Truncate Load Style

OBL-TRUNCATE-DEC: Oracle Bulk Loader with Truncate Load Style, Disable and Enable Constraints

TBL-APPEND: Table Loader with Append Load Style

TBL-REPLACE: Table Loader with Replace Load Style

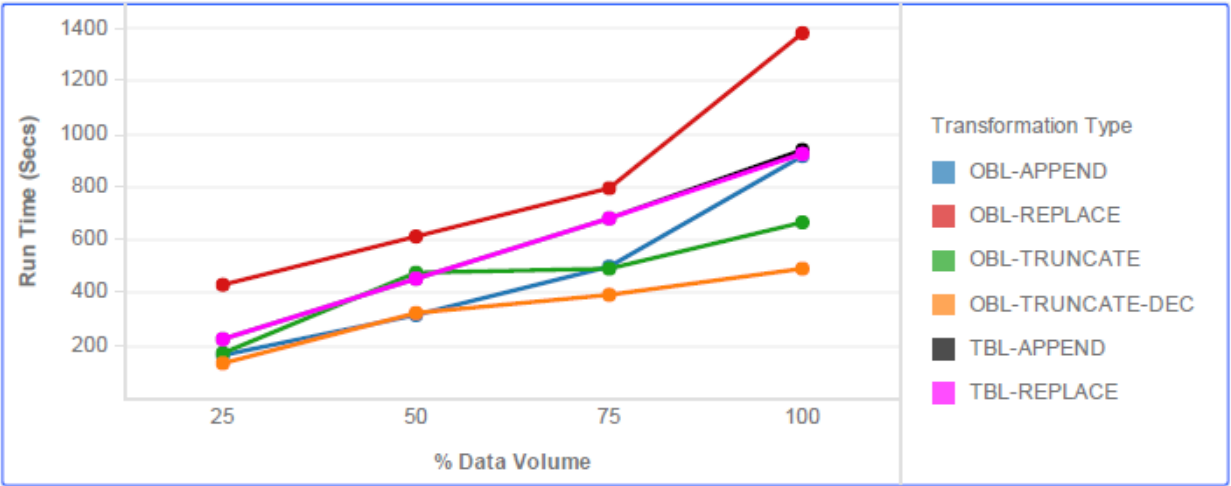


Figure 1. Run Time comparison for different load Styles

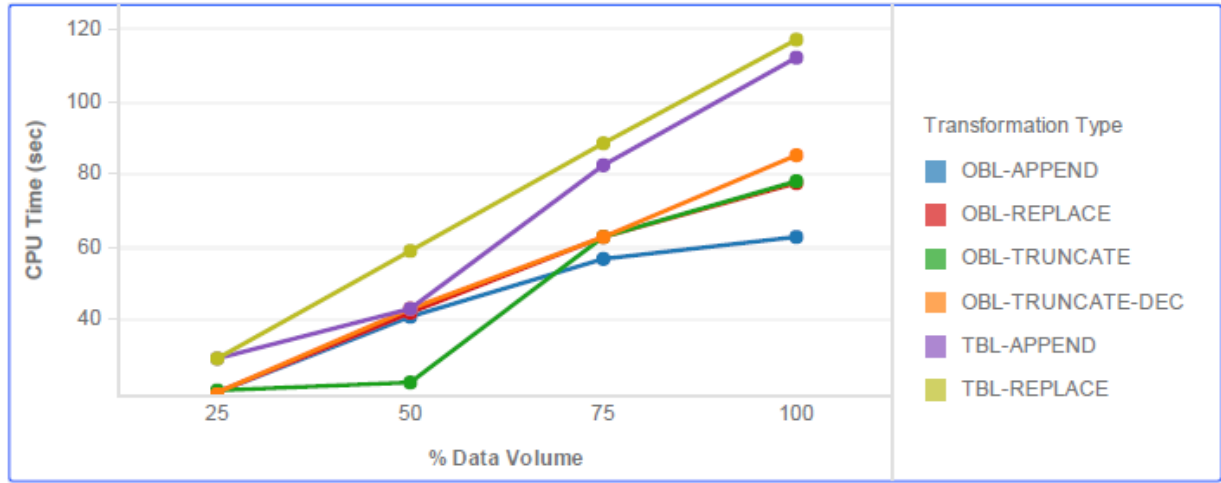


Figure 2. CPU Time comparison for different load Styles

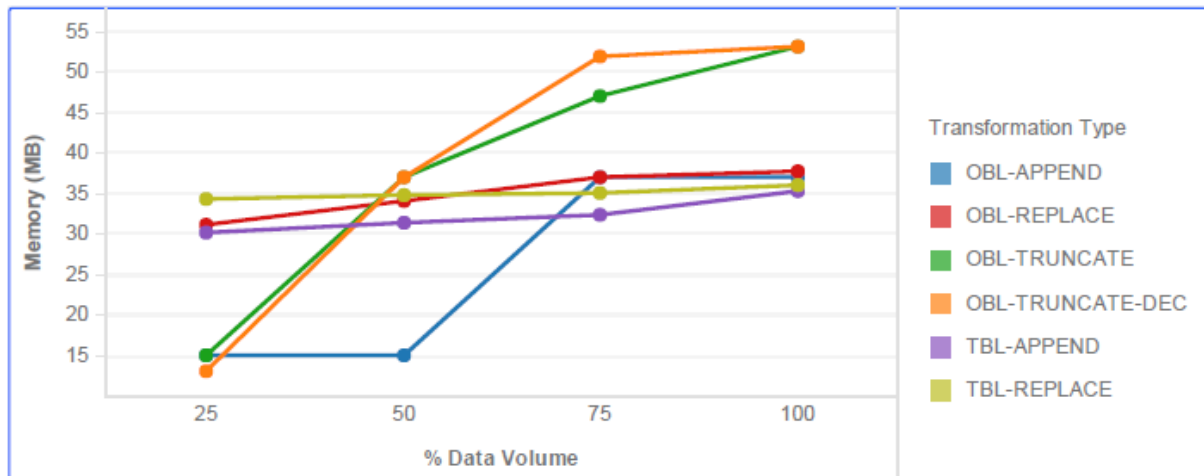


Figure 3. Memory comparison for different load Styles

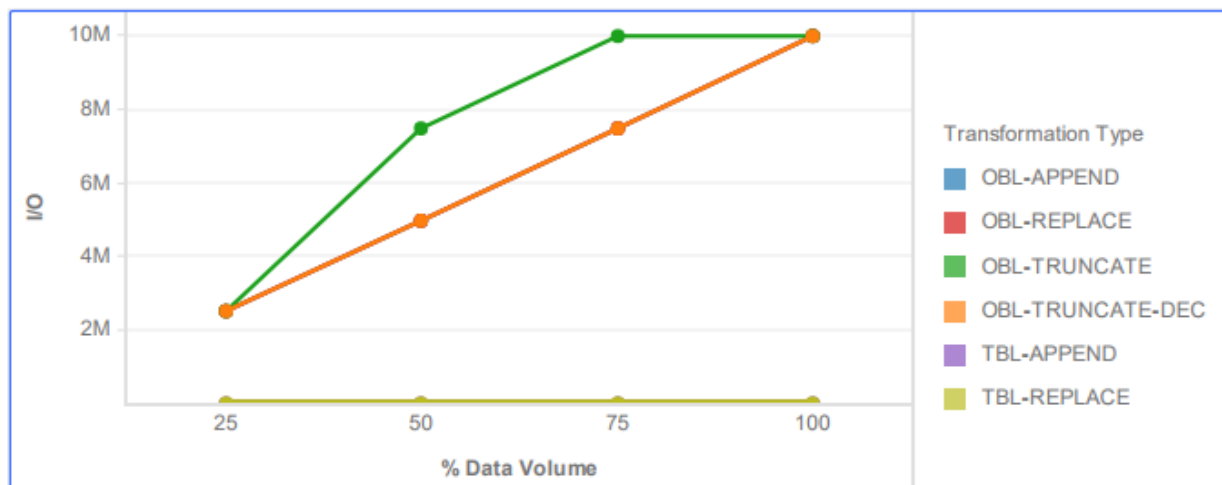


Figure 4. I/O comparison for different load Styles

Run Time: OBL-TRUNCATE-DEC has the first best performance and OBL-TRUNCATE offers the second-best performance with the Run Time.

CPU Time: All the Oracle Bulk Loader load styles consumed relatively similar CPU time, but OBL-APPEND is the best out of them.

Memory Utilization: OBL-TRUNCATE-DEC utilized most of the memory and OBL-Append took utilized less memory of all the different load transformations. Table Loader append and replace load styles have similar memory usage and the usage is linear for the load volumes.

I/O Operation: All the Oracle Bulk Loader load styles have similar I/O performance, but OBL-REPLACE has the best performance out of them. Table Loader has very low performance compared to Oracle Bulk loader.

PERFORMANCE COMPARISON FOR UPDATES

As part of the daily ETL processing, there will be inserts and updates to the data. SAS® Data Integration Studio provides the following transformations to perform updates on the data.

1. Table Loader

2. SQL Update
3. Merge
4. SCD Type 1 Loader

SCD Type 1 loader and SQL Update are not the ideal transformations to perform updates on large volume of data, as they generate an individual update statement for each field as shown below. Similar update statement will be generated for all the columns that need to be updated and for all the rows in the table. This operation is very expensive and takes a very long time.

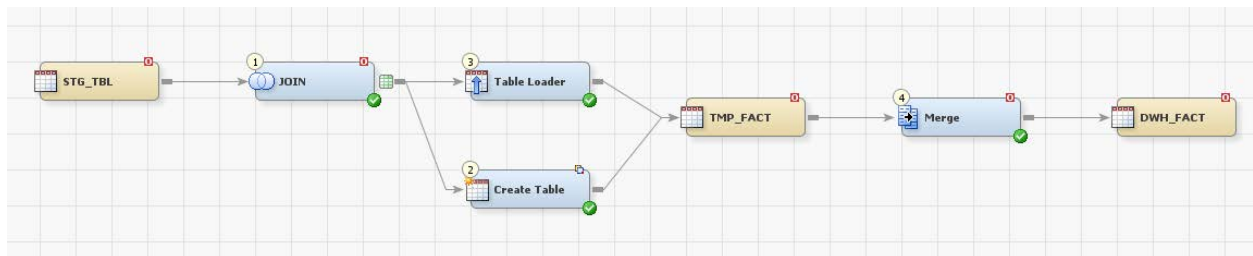
```
update DWH_TBL as m
```

```
  set STTS_CODE = (select STTS_CODE from STG_TBL as t
                  where m.STTS_DIM_ID = t.STTS_DIM_ID)
```

Table Loader and Merge are the other two transformations that can be used to perform the updates. Here in this section, the performance comparison is done between the Table Loader and Merge transformations. To compare the performance for the updates between these transformations, the same stage table with 13,695,436 was taken and we have updated the value for one field in the target table DWH_FACT for different volumes (10,000, 20,000, 40,000 row updates), so the number of updates to be run on the target table would also be the same.

Transformation used to update data: Merge

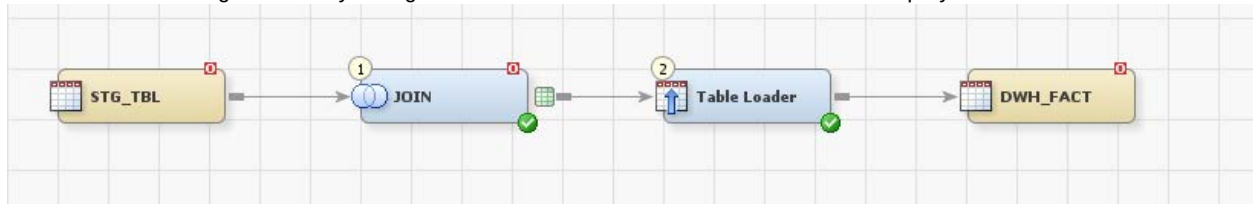
In the below Job, SQL Merge transformation has been used to perform the updates. Merge can only merge the data from one table in to another and requires that all source and target tables reside on the same DBMS server. Perform the merge by specifying the matching keys and the columns and values in the update tab of the merge transformation. For that reason, we have landed the data in to a temporary table called "TEMP_FACT" and merged the data in to the target table "DWH_FACT" using MERGE Transformation.



Display 7. Updates performed using Merge transformation

Transformation used to update data: Table Loader

Same source and target tables were taken and the same update operation has been performed by table loader transformation using the Modify Using Index on the table as shown in the below display 8.



Display 8. Updates performed using Table Loader transformation

Below is the performance comparison (Run Time, CPU Time, I/O) between Merge and Table Loader for different data volume of updates. Merge has the best performance compared to Table Loader when performing the updates. If there are bulk updates to be performed daily, which can take a very long time, consider truncate and re-load approach in those cases. One downside with that approach is that you may lose any historical data in the target table.

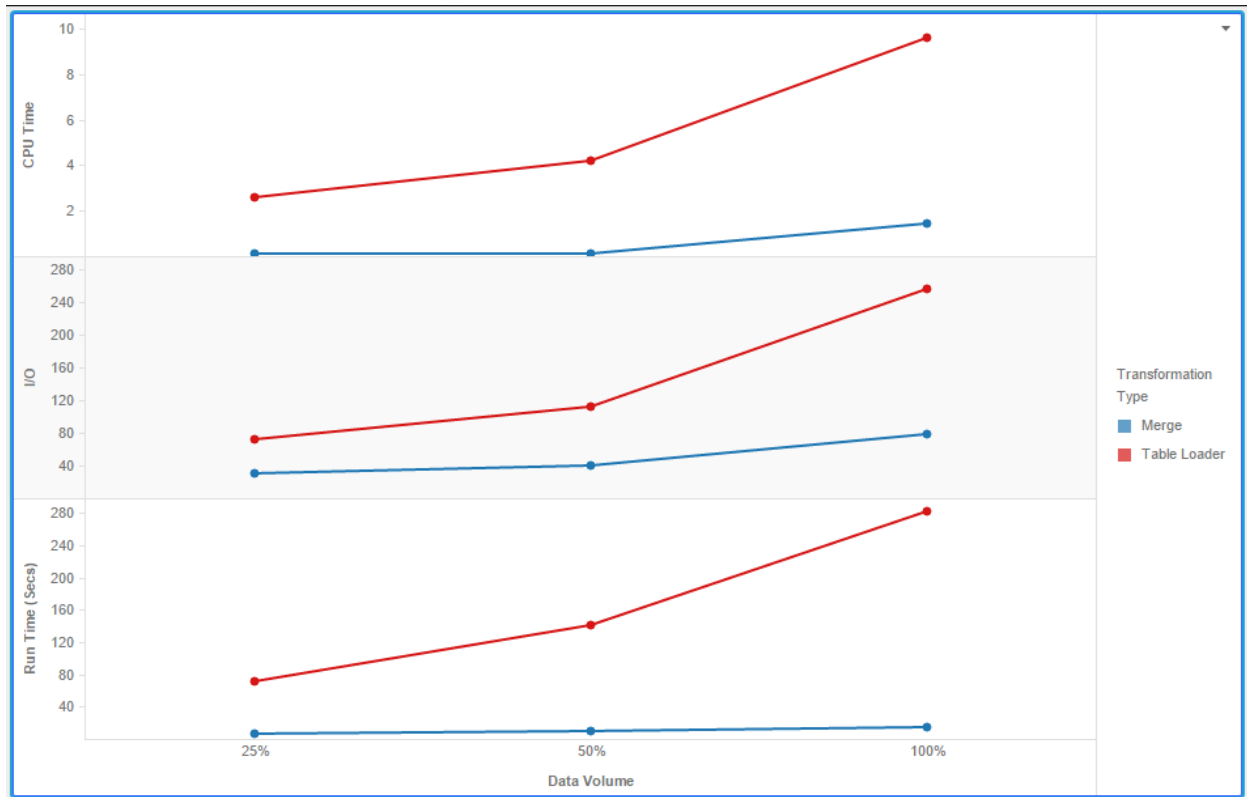


Figure 5. Performance comparison between Merge and Table loader for updates

Ideally, if there is a need for bulk updates, consider replacing the whole table rather performing bulk updates if there is no need to store the historical data in the target table or the historical data is present in the source tables.

CONCLUSION

When designing the load strategies, it is important to do a performance benchmarking between different load techniques to identify the best load strategy that suits your data loads. It is wiser to spend some time in this area during the design phase. This will be helpful in projects where ETL must handle load large volume of data loads very frequently. This may also help in evaluating the current Infrastructure and make any necessary changes to it, if needed. Based on the above bulks load benchmarking Oracle Bulk Loader with Truncate- Disable and Enable constraints offers the best performance with bulk load and Merge offers the best performance with updates. The performance results might vary based on different environments, software version, tool versions used and their configurations.

REFERENCES

Qingfeng Liang, 2011. "Best Solutions for Tuning Performance of ETL Jobs in SAS® Data Integration Studio" *Proceedings of the SAS Global Forum 2011 Conference, Las Vegas, Nevada*. Available at <https://support.sas.com/resources/papers/proceedings10/116-2010.pdf>

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author's at

Sai S Potluri, ETL Architect

Synectics for Management decisions Inc

E-mail: sai.s.potluri@gmail.com