CSE 603: Programming Assignment 2

MATRIX CORRELATION COMPUTATION USING NETEZZA UDX

INTRODUCTION:

Implemented a FUNCTION which takes VARIABLES (columns) and OBSERVATIONS (rows) as INPUT and generate all possible CORRELATION VALUES.

Refer to RUN_PROGRAM_SCRIPT.txt for detailed description to RUN this 'netezza_correlation_matrix' Function.

STORED PROCEDURE:

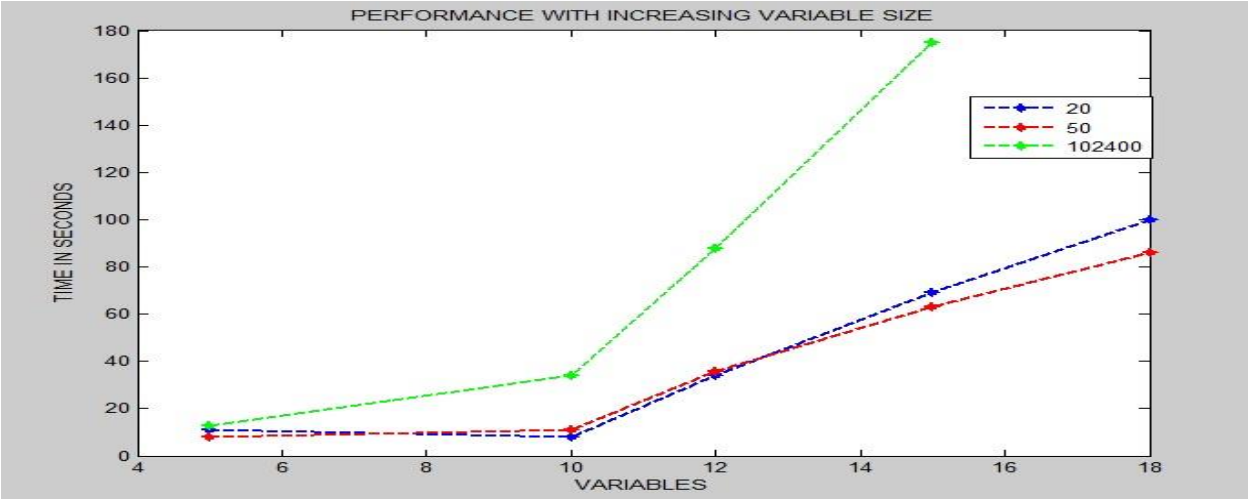
NETEZZA_CORRELATION_MATRIX(r ROWS, c COLUMNS) work flow is shown below,

1. Creates a table and Inserts r Observations and c Variables
2. Calculates Mean for each column in input table placing them into a table
3. Computes Correlation using,
    a. DIFF_ UDTF
    b. CALDEV UDA
    c. CALCORR UDA
4. Calculates TIME TAKEN for UDX to calculate correlation.
5. Computes Correlation using NETEZZA ANALYTICS FUNCTION 'CORR_MATRIX_AGG' and the time taken by it.

Refer to COMPILE_REGISTER_UDX.txt for details regarding functionality of UDXs.

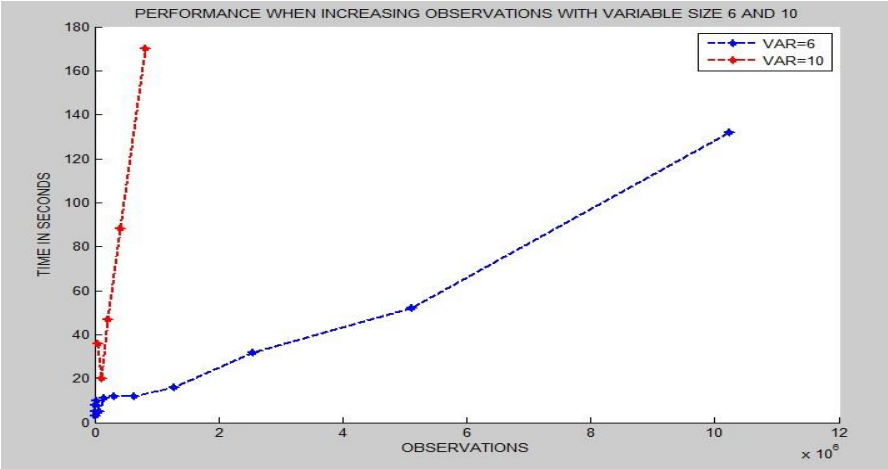
OBSERVATIONS:

1. Impact on performance by increasing Variable Size:



1. Time taken by UDX is significantly increased with increasing variable sizes after 10.
2. You can see that, for Variable size >= 10 there is sharp increase in time taken by the UDX for very less observations 20, 50 and also for 102400 rows/observations.
3. Correlation matrix size we generate depends on the number of columns/variables which explains increase in time with more variables.
4. Note that UDXs take columns as input. Increasing number of columns directly impact the performance of the UDX.

2. Impact on performance by increasing Observations:



1. Time taken by UDX is significantly increased with increasing observation sizes higher than 10^6.
2. Amount of rows less than 10^6 doesn't show consistent behavior with respect to time taken. You can see the graph fluctuating up and down with observations <10^6.
3. Impact on performance by increasing variable size with huge observation sizes:

   You can also observe in this graph, the increase in TIME for 10 variables compared to 6 variables.

You can conclude that increasing Variable Sizes after certain point (var >= 10) reduce the performance of the UDXs and also increasing Observations higher than 10^6 impact the performance of the UDX. If you compare both factors, Variable Sizes show significant impact rather than Observation count.

3. Netezza Analytic Function :- CORR_MATRIX_AGG :

CORR_MATRIX_AGG compared to my UDX implementation gives high performance regardless the increase in Variable sizes and Observation sizes.

4. Limitations:

1. Input Table Generation is a huge performance bottle neck in my implementation. Inserting row by row for huge matrix sizes takes considerable amount of time and also made my testing impossible.

2. As a work around I separated the 'create and insert table' script from NETEZZA_CORRELATION_MATRIX stored procedure and manually inserted rows in bulk to already existing table. Using the following query, INSERT INTO TABLE_NAME SELECT * FROM TABLE_NAME;

3. Using NZ_LOAD would have been a better option for inserting huge data into tables in NETEZZA database.