DWH–Performance Tuning For Better Reporting

Sandeep Bhargava
Research Scholar
SGVU, Jaipur, India

Naveen Hemrajani
Associate Professor
SGVU, Jaipur, India

Dinesh Goyal
Associate Professor
SGVU, Jaipur, India

Subhash Gander
IT Professional

ABSTRACT: The concept of data warehouse deals in huge amount of data and lot of analytically queries runs on DWH, which covers base data in terms of thousands of gigabytes, to unveil the hidden pattern of business. So response time of query is exponential proportional (metaphorically) to involved base data. So we can say THUMB RULE as "MORE BASE DATE MORE ACCURATE RESULTS". But it will degrade the performance if not taken care properly. Also we, as human, hate to wait due to natural phenomenon encoded in our DNA. Lot of works has been done by many literates around the globe on DWH performance tuning by proposing many frameworks related with various focus data quality, Metadata management etc...

In this paper, an effort has been made to discuss about the real industry problems and how we improve the performance of data warehouse by minimize the existing CPU cycles wisely using metadata driven approach.

Keywords

DWH- data warehouse, BI- Business Intelligence

1. INTRODUCTION

There is lot of hardware computation power is being involved to make DWH align with requirement of business. The current growth of semi conductor industry and world-class competition made computational power relatively cheaper but still hardware cost has a significant contribution on cost estimation / infra implementation.

Performance of DWH can be considered at below mentioned stages.

- Fetch data from source system
- Data processing through ETL Layer
- Feeding data in to DWH
- Time involved in Fetching data from DWH for reporting

Here our focus will be more on the report response time as discussed above and the proposed solution will help to make that better. So for better reporting many factors are there which can contribute in positive and negative sense depending upon how factors has been considered or implemented?

1.1 Factors Affecting Reporting Layer

Response Time

- Tools Selection
- Physical Design
    - Normalized vs. de-normalized
    - Relational vs. dimensional
    - Hybrid
- Reporting frequency
- Concurrent Users
- Amount of data
- Indexing
- Statistics

All above-mentioned factors can impact reporting response time but in this paper we will be explaining more on how to keep **statistics up to date with help of available metadata for better reporting?[1]**

1.2 What is Statistics?

In DWH statics are milestones directives that help database optimizer / database engine to come up with **"THE COST EFFECTIVE"** execution plan. So what information is there in statistics? In statistics below information can be stored in DWH, which will make sure pre availability of required data to come up with effective execution plan. For example:

- Number of nulls
- Number of Unique values
- Average row correction per values change
- Number of intervals
- Number of rows updated.
- Mode Frequency etc…

With this available precompiled information optimizer / database engine will use this information rather than calculating or estimating the same at run time which will increase the response time unnecessarily.

How statistics can be defined &do we need to define every Time?

Statistics can be defined by developers / DBA / Architects to make sure smooth execution of queries on the box. It can be defined on Column or combination of column level and No, Statistics should be defined only one time and this is a physical attribute of a database table, which can be defined on a column or combination of column but it requires a frequent refresh on the basis of data changing frequency and amount of change this has to be refreshed on a table level[5].

Where this information will be saved & how this will be used?

System will store this information in system tables and depending upon the database software used it can be string field or even a **clob** or **blob** object and when query will be submitted to database then optimizer / database engine will use this pre compiled information from system tables for cost effective execution plan.[7]

1.3 What is Metadata?

Metadata, a very confusing term in word of data warehouse .Most individuals with some level of involvement with data warehouses, from a technical or business perspective know of the term “meta data”. When asked to define “what is meta data” most of these individuals can reiterate the common definition “Data about data”[6]

So

- What is data about data?
- Why this is so important?
- How this piece plays its role?

Suppose in a database software like oracle, sql server etc.. we have 30 databases and each database contains 100 tables and lot many columns. Now every column has its associated attribute but how database will know about that below.

- What is the length of a particular column?
- What all tables are with Database A?

So to answer these questions not only database but every single available tool which has concepts of backend database has its own area dedicated to that application or database only for technical specification of the attributes and in physical terms this area called as sys_dba in oracle, dbc in teradata and so on.

Is It Necessary?
No, having statistics is not necessary but **depending on the filter and join condition in queries it will help to avoid full table scan and over consumption of CPU cycles.[2]**
But if statistics are there on the database tables then it is essential to have it up to date otherwise it will provide false information to optimizer / database engine which will turn as a feed in to bad execution plan. e.g. if statistics is not up to date ? Statics says there are 10, 00,000 records in a table. But actually it has increased by 10 millions so this type of inconsistency will lead in to cost and CPU consuming execution plans.

2. TRADITIONAL METHODOLOGY FOR STATISTICS REFRESH.

At many sites, to address this refresh process, they have a weekly / fort nightly / monthly job to run on box which will simply go and refresh all available statistics on the box regardless of the functional data load frequency.[3]

On this traditional solution we did some analysis and below are some more description for the same.

2.1 What was done in investigation?

During analysis we started to plot the various graphs and trends to identify the nature of wastage. And we figured out that absence of intelligence in weekly stats collection process is causing this wastage in CPU cycles.

Evident Traits

1) Every time stats were getting refreshed on database level
2) No separation for static data
3) No special consideration on heavily floating facts
4) Requires a large chuck of CPU once in a week, reduces other process’s slices
5) No Aging information available
6) No auditing about was, what is and what should be?

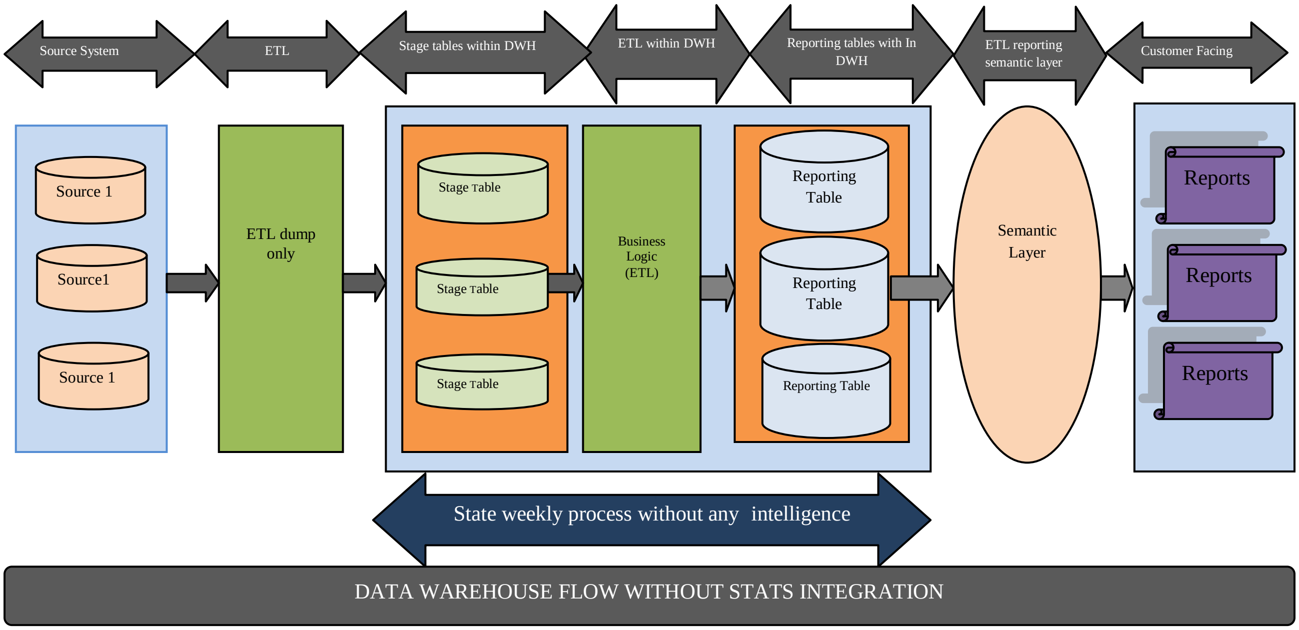

Fig. 1 Available Traditional Methodology

What alerted us for solution?

Vilferdo Pareto, an Itilian economist, observed 80% of the land in Italy was owned by 20% of the population; he developed the principle by observing that 20% of the pea pods in his garden contained 80% of the peas.

Later on same study was generalized by Joseph M. Juran, business management consultant, and named after Vilferdo Pareto as Pareto Principle "Pareto Principle states that 80% effects come from 20% causes." If we allow the same extension in BI life cycle then we will see that more then 20 % CPU cycles were consumed by approx 80% system management tasks and if we were running low on CPU cycles then saving every bit will be an add on to pocket, which indirectly converts in saving. So basically "what we save is also a kind of earning", quoted by Warren Edward Buffett, an American investor.

On the similar lines when we start analyzing systems then we found that out of 100 % CPU cycles assigned to system management, approx 80 % were getting consumed in stats management.

Upon further investigation we found that out of assigned 80%, approximately more then 50% was kind of *wastage because of unnecessary repetition of process even though it is not required.[4]

.

3. PROPOSED SOLUTION

3.1 Flow Chart

Step 1: As this process is so tightly integrated with ETL at table level so to initiate with we will pass Database name and Table name to the process.

Step 2: In this Step process will check the existence of the database object in database to make sure this table does exist in database as a table. If table is there in database then it will go on step 3 otherwise will break the process and come out.

Step 3: After getting confirmation on existence with help of step 2, process will make sure that table has data. If table is populated then it will go on next level otherwise will break the process and come out.

Step 4: After getting confirmation on data population with help of step 3, process will make sure that stats are there on the table. If stats are there then it will proceed to next step otherwise will break the process and come out.

Step 5: Now as process knows with help of previous steps about stats so now question is to know weather data load is incremental or will be truncate and load. Now because this cannot be identified using metadata information so end user as an input to the process will pass this information. If table is truncate and load then process will refresh the whole stats on the table but if not then simply proceeds to next step.

Step 6: In this step process will make sure that data of base table has been modified by how much percentage. Ideally if data change is more the 9% then go with a refresh otherwise leave it alone.

Step 7: In this step process will check the age of existing stats and if it is more then 30 days then will refresh the stats otherwise simply come out of the loop.

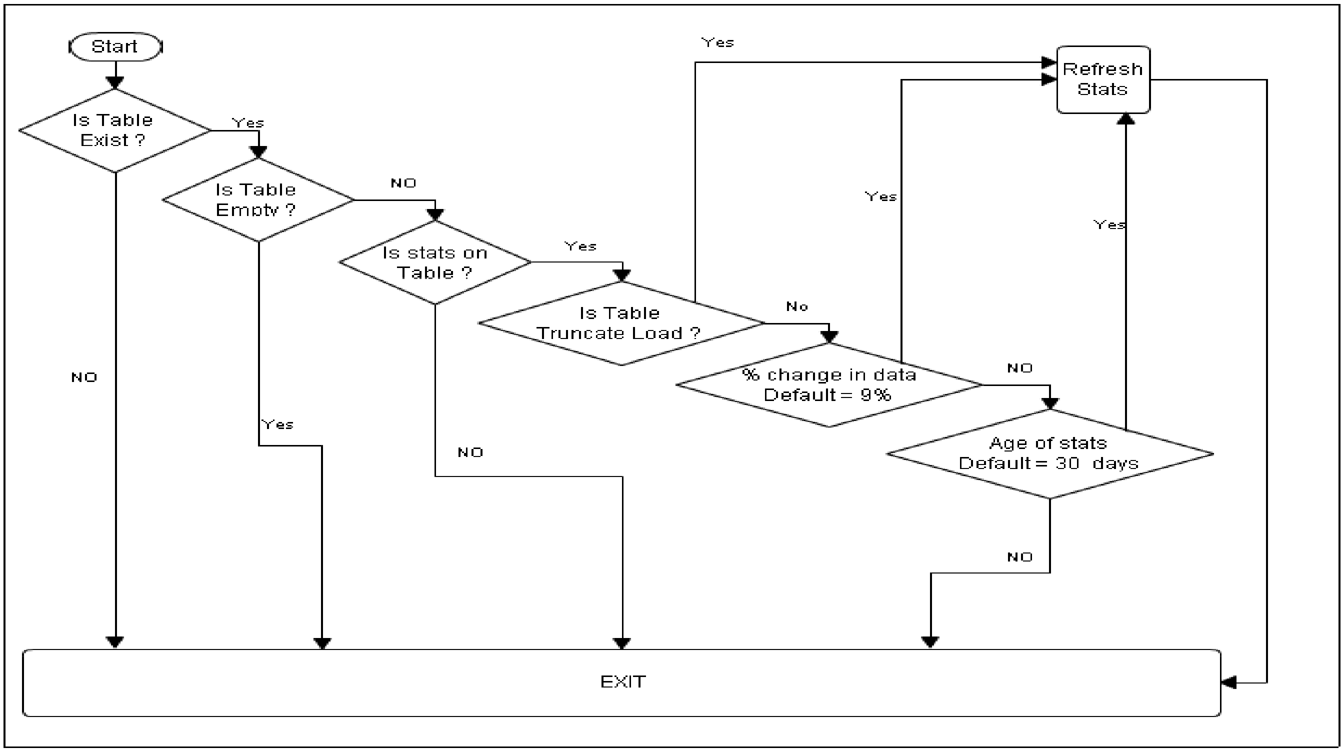

Fig. 2. Flow Chart of Proposed Solution

3.2 Process Flow Model

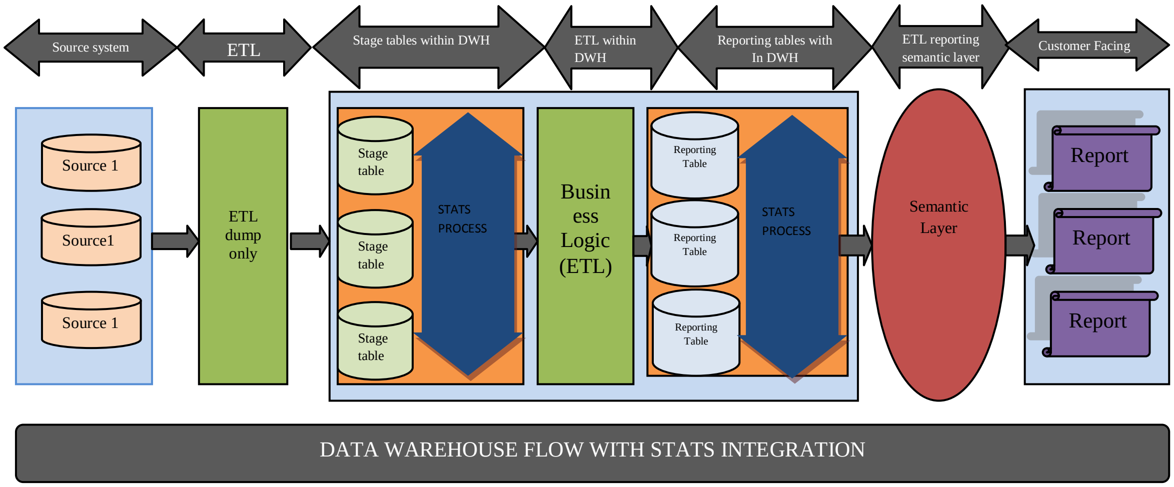

Fig. 3 Proposed Model Using STATS

4. RESULTS

The above mentioned graph is showing CPU utilization of four different users for a application. These user were responsible to operates on stage load, reporting table loads, reporting needs etc. for a sample data of a firm M/s Adsect Technologies, Jaipur for two consecutive months in year 2011. This process was implemented for user 3 alone. And the trend of CPU utilization for user 3 (green) is supporting the fact process is working. Because in three and half spikes it has more requirement for CPU as compare to last three and half spikes.

The above graph proves that the technique introduced in this paper has reduced the CPU utilization drastically and the same may be implemented practically for further better usage of technology.

4.1. Cost Benefit Analysis

Let us assume the cost of 1 CPU Cycle is 1 unit then in first week cost occurred is approximately 4900 units which continues till first four week, this result is obtained without using stats, while next four week the cost is reduced to almost less than half when STATS approach was applied & new cost is just 2400 units.

4.2. Solution Piloting

To verify the results of this approach pilot testing was done on Teradata database so terminology is inclined toward that database software but this framework can be generalized across any platform.

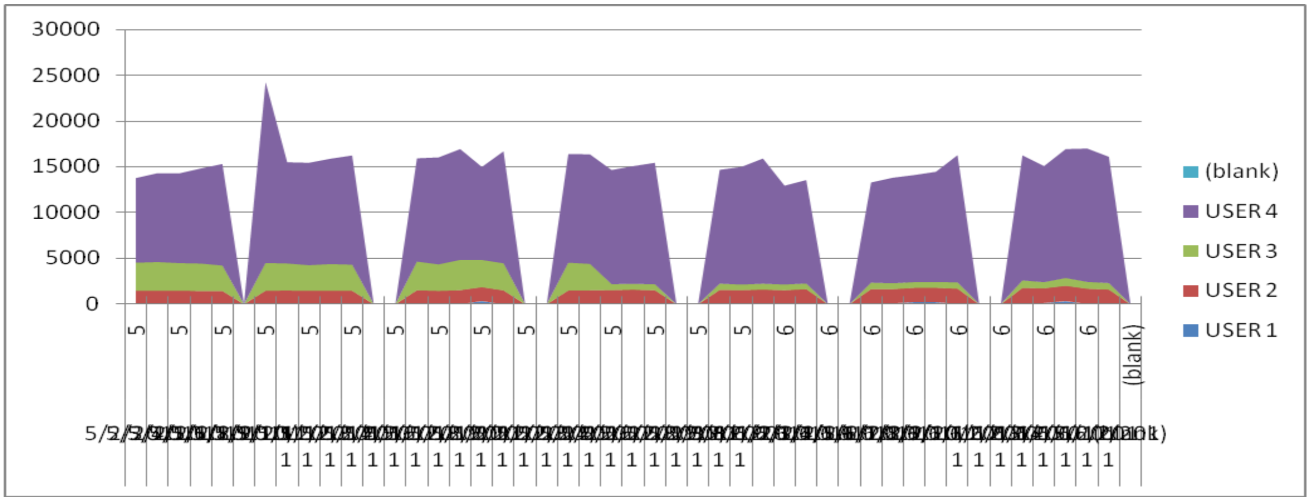

Fig. 4 Typical CPU consumption chart for an application in a DWH

5. CONCLUSION

Today Data Ware Housing is the back bone of most of MNC's and large scale organization. But most of them have not looked into database administration cost especially over hardware. The data entry & updation cost.

The above work is an aim for optimizing the cost of data warehousing by tuning the performance over OLAP. Instead of updating all attributes and rows of a table for even a single value of a entity we have proposed to a methodology to reduce the effort on the machine side. In this effort is made by reducing the number of CPU Cycles to be used for editing the value by using STATS.

5.1 Intended Audience

This paper involves very advanced concepts of data ware housing. Also while writing this paper Tier 3 and Tier 4 population of IT industry was kept in mind as per below list.

- Senior Developers
- Project Managers
- Technical Leads
- DBA
- Solution Architects

6. ACKNOWLEDGMENTS

I would like to take this opportunity to thank & express my special gratitude to Mr Subhash Gander and Ms.Ruchi Dave (Program Coordinator) and also all my friends and classmates for their support and help me.

Last but not the least I wish to express my gratitude to god almighty for his abundant blessings without which this work would not have been successful.

7. REFERENCES

[1] Data Warehouse Performance Management Techniques, Andrew Hold sworth, Oracle Services, Advanced Technologies, Data Warehousing Practice. 2/9/96

[2] Discover Teradata Meta Data Services

[3] Performance Tuning Mechanisms for Data Warehouse: Query International Journal of Computer Applications (0975 – 8887) Volume 2 – No.2, May 2010

[4] Teradata® RDBMS performance optimization, NCR Corporation

[5] Oracle® Database Performance Tuning Guide10g Release(10.2)Part Number B14211-03

[6] Metadata implementation with Ab Initio EME, Article by Mike Green on 13 May 2009

[7] www.oracular.com/white...pdfs/DataWarehousingwithOracle.pdf