IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2320-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 9, Issue 9, September 2020

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.512

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 9, September 2020 ||

Framework for Performance Prediction of Data Warehouse Systems

Dr. Madhu Bhan[1] and Dr. K Rajanikanth[2]

Department of Master of Computer Applications, Ramaiah Institute of Technology, Bangalore, India. [1]

Visvesvaraya Technological University, Belagavi, Karnataka, India[2]

ABSTRACT:The growing size of Data Warehouse systems and the increasing complexity of the query workloads make it necessary for developers to be able to assess the performance of Data Warehouse Systems in the early stages of development when making changes are easy and less expensive. Assessing the performance of a Data Warehouse System in early design stages is crucial for building a successful business application. As Data Warehouse Systems are complex systems with many interacting components, it is essential to have a framework that supports early assessment of performance of such systems. It is in this context that we propose a framework to address the difficult issue of assessing the performance of Data Warehouse systems in the early design stages of its development.

KEYWORDS:Data Warehouse, Performance, Framework, Modelling, Simulation.

I. INTRODUCTION

A Data Warehouse System is more than just a "big database". Data is extracted from heterogeneous relational databases and other legacy systems and is stored in Data Warehouse to be used by organizations to retrieve key information for decision support [1]. In today's decision support environment where time is a very critical factor, the Data Warehouse manager must make sure that end-users are getting reliable answers to their queries within a reasonable time. The performance of a Data Warehouse depends on many complex factors such as the architecture, the quantity of data stored within a Data Warehouse, the query workload, and the performance characteristics of the hardware and software components deployed to implement the business solution. Designing a Data Warehouse application without adequate understanding of its performance characteristics can lead to significant differences in expectations between system users and system designers and administrators. To assess design alternatives or to identify system bottlenecks, quantitative system analysis must be carried out from the early stages of Data Warehouse development when changes are easy and less expensive. Hence there is a need for a framework that represents the tasks of performance assessment of Data Warehouse Systems. In this paper, we describe a framework that allows for assessing performance of Data Warehouse Systems during early design phase of its development. The Framework makes use of Simulation Techniques and the Software Performance Engineering (SPE) approach.

II. RELATED WORK

The simplest Data Warehouse architecture is a two-tier architecture where a client interacts directly with the Data Warehouse Server. The Data Warehouse Server acts as the first tier and the client that hosts the front-end decision support and analysis tools acts as second tier. However, the most common Data Warehouse architecture is a three-tier architecture where the first tier is a Data Warehouse Server, the middle tier is an Online Analytical Processing (OLAP) Server where the materialized views are available and the third tier is the client [2, 3]. The client tier contains query and reporting tools.The increasing demands for fast interactive response time makes query performance one of the central problems of Data Warehousing. A sophisticated design is necessary to ensure high performance of Data Warehouse Systems.Two popularly known approaches to improve performance of such systems are materializing views and indexing.

The approach of materializing views refers to the evaluation of frequent queries beforehand and storing of the results as materialized views for later use by the OLAP Server to provide fast response to those queries [4]. The materialization of all the possible views is constrained by the storage space and maintenance cost and hence one needs to select a sub-set of views to be materialized. This problem of selecting a sub-set of views from a set of possible views is termed as "Materialized View Selection" problem.This problem is known to be NP- hard. Thus the only practical approach to solve this problem is to develop approximate solutions. Many such approximate solutions are available [5, 6].

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 9, September 2020 ||

Indexing is the other popular approach for improving the performance of Data Warehouse Systems. Special indexing techniques are proposed taking into account the specific characteristics (vast data, mostly read access) of Data Warehouse Systems. A widely used technique is Bitmap indexing [7]. Bitmap index uses a "bit structure to indicate the rows containing specific values of the indexed attribute. Alternative schemes such as Multi-level Encoding scheme and Multi-component Encoding scheme have been introduced to reduce storage requirement and speed up performance [8].
SPE is an approach to assess the performance of software systems early in the life cycle [9, 10]. Two models are proposed in this approach. The first one, "Software Execution Model", accepts the performance characteristics of all the resources used in building the system as inputs and builds an "Execution Graph" that can be solved to obtain the performance characteristics of the software system. This model ignores issues like the presence of other service requests, the consequent contention for resources, the need for buffering requests waiting for service, service policies etc. In other words, the performance characteristics are obtained for an isolated execution environment. All such issues ignored by the first model are taken into account by the second performance model, the "System Execution Model". This second model makes use of the results produced by the first model. Additionally, it requires data regarding the workload characteristics and execution environment including all the resources for which contention is possible. The model built at this stage is essentially a variant of layered queuing network which needs to be solved either analytically or through simulation techniques to obtain the final performance results. This entire process can be repeated with different parameters to explore alternative designs. This would mean, in the context of Data Warehouse Systems, alternative view selection algorithms, alternative indexing strategies, and even alternative choices for Hardware / Software components.As this performance assessment is done in the early design phase of the system, such an exploration of alternatives is economically feasible and is thus the preferred approach for building successful Data Warehouse Systems.
Simulation is a de-facto method for the performance assessment of complex software systems [11, 12]. Simulation techniques are used to solve performance models of Data Warehouse systems and thereby collect performance metrics of interest such as average time to respond to requests, the average time for which a request has to wait before being attended to by the server, the average utilization of resources etc. However, for successful use of simulation approaches, it is essential to accurately characterize the workload and performance characteristics of the all the hardware and software components of the Data Warehouse system.For characterizing the workload that may be used in simulation studies of Data Warehouse, there arestandard benchmarks available from the Transactions Processing Council [13].The actual simulation itself can be carried out using available popular tools [14, 15]. Alternatively, one can build a specific simulation tool that is more useful for studying the systems of interest [16].
Software size is an important input parameter for the performance models developed in the SPE approach. These models require the size to be specified in Thousands of Lines of Code (KLOC). As these performance models are to be used for early estimation of the system performance, we need to estimate the size of the software also in the early stages of the software development life-cycle (SDLC). A large number of different techniques have been indeed developed over the years to estimate the size of the software as KLOC in the early phases of software development life cycle [17,18].

III. PROPOSED FRAMEWORK

The proposed framework makes use of the SPE approach of Smith with suitable modifications, Object-Oriented concepts together with the industry-standard Unified Modelling Language(UML), and simulation techniques. The SPE approach, applicable to any software system, includes gathering data required for performance assessment followed by developing and solving the software execution model to get results useful for performance assessment. Simulation techniques allow us to obtain execution histories of complex systems like Data Warehouses which are not amenable to pure analytical approaches. The proposed framework for assessing the performance of Data Warehouse Systems in the early design phase of its development is shown in Figure 1. The framework assumes that the architecture of the Data Warehouse is chosen as three-tier architecture. If required, the choice of the architecture can be included as the first step in the framework. The performance characteristics of all the hardware components involved (server speeds, disk access times, network speeds etc.) are assumed to have been determined. It is also assumed that the queries that will be run against the Data Warehouse are known though their actual arrival rates may vary and need to be estimated. The proposed framework integrates performance analysis and Data Warehouse development process to ensure that the system meets its performance objectives. The Framework consists of the following steps.

1.Set the performance goals: Based on the business requirements, determine acceptable levels of performance. This could mean setting a limit on the average waiting time for a query and / or setting a minimum limit on server utilization etc.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 9, September 2020 ||

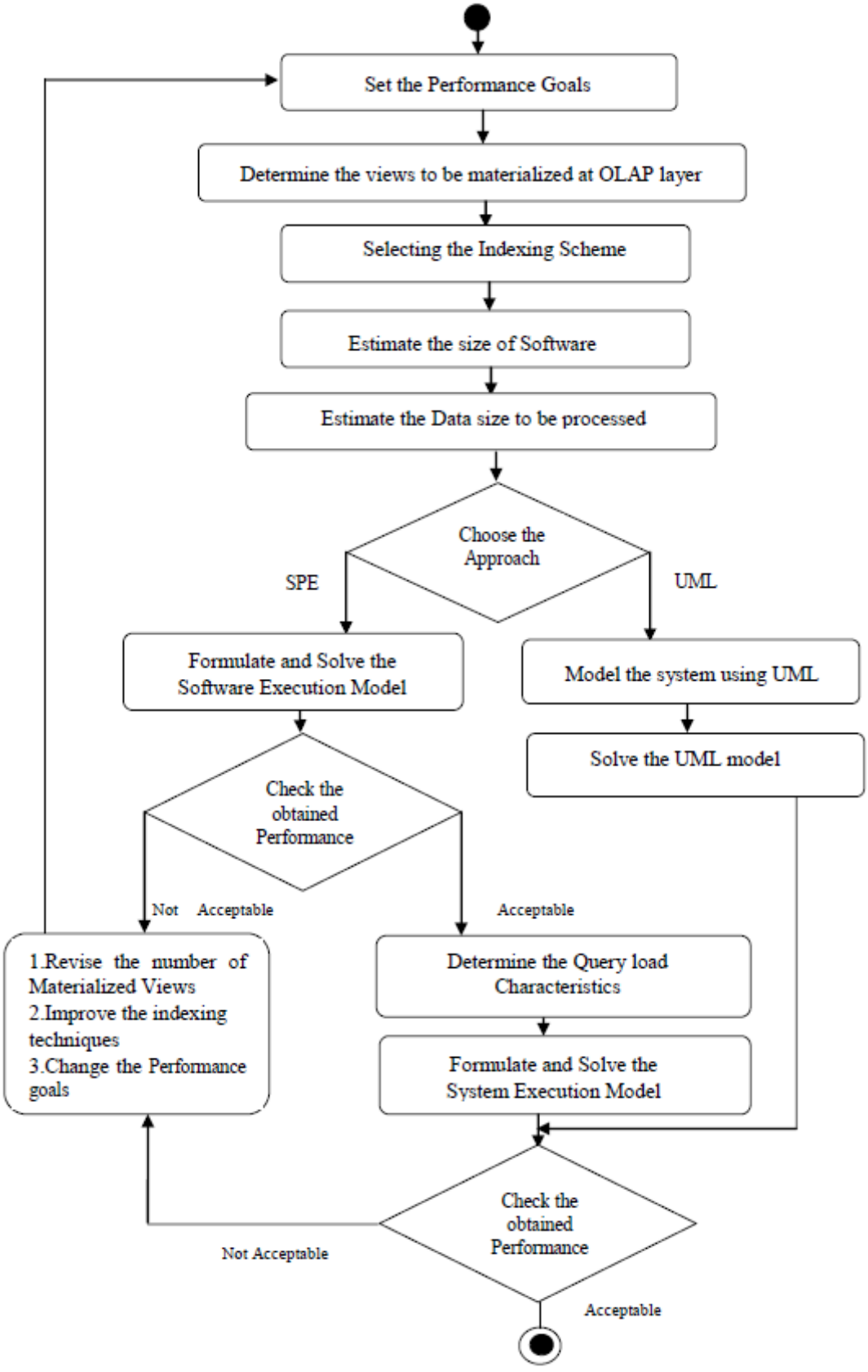

Fig. 1Framework for Performance Prediction of Data Warehouse Systems

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 9, September 2020 ||

2.Determine the views to be materialized at OLAP layer: The availability of views at the OLAP layer will allow the queries that depend only on these views to be processed by the OLAP Server itself. There will be no need to forward the query to the Data Warehouse Server. As the Data Warehouse server needs to refer to the usually much larger Tables, it will take more time to process the queries. On the other hand, materializing more views may impose severe demands on storage space. Thus, one needs to determine the reasonable number of views to be materialized.

3.Select the Indexing Scheme: Indexing the physical tables (at the Data Warehouse level) and materialized views at OLAP Server level has a dramatic impact on the time required to process a query. However, indexing does consume significant storage and leads to increased costs of maintenance. Thus one must select an indexing scheme that is most appropriate for the given business context.

4.Estimate the Data size to be processed: The amount of data (tables and views) that needs to be processed to satisfy each query from the set of possible queries needs to be estimated next. This will depend on the query itself and the sizes of the tables that need to be referred as well as the sizes of any views materialized which are useful in answering the query. Further, determining the actual sizes of views may be quite expensive and thus one may need to estimate them using some form of sampling technique.

5.Estimate the Size of the Software: Size of the Software to be executed is an important parameter for the process of assessing the performance of the system. Thus it is necessary to estimate the size of the software that is to be executed on the Data Warehouse Server as well as OLAP Server.

6.Choose the SPE-based Assessment Approach Or UML-based Assessment Approach: There are two choices available now to estimate the performance of the system. One can use either the SPE-based approach or UML-based approach. The UML-based approach is more suitable for systems based on Object-Oriented concepts. Further, this approach assesses the performance of the system at relatively higher level of abstraction. It is to be noted that, in principle, it is possible to even use both the approaches. However, resource constraints in practice lead to the use of only one of these approaches. If SPE-based approach is to be used, go to Step 7. If UML-based approach is to be used go to Step 12.

7.Formulate and Solve the Software Execution Model: Using the data gathered already, formulate the Software Execution Model following the SPE approach. The Software Execution Model provides the performance characteristics ignoring issues like resource contention. Solve the model using analytic or simulation approaches and obtain the performance characteristics.

8.Check the obtained performance: Check if the performance is acceptable. If the performance is not acceptable, one must revise the number of materialized views and / or improve the indexing techniques; or even change the performance goals. Now, repeat Steps 1 to 8. If the performance is acceptable, proceed to Step 9.

9.Determine the Query load characteristics: Estimate the arrival process of the queries, the population size of the queries, and the distinct queries / query classes to be used as representative of the workload.

10.Formulate and Solve the System Execution Model: Using the data gathered already and setting other parameters like Buffer Size, formulate the System Execution Model following the SPE approach. The System Execution Model provides the performance characteristics taking into account issues like finite buffer size, resource contention etc. Solve the model using analytic or simulation approaches and obtain the performance characteristics.

11.Goto step 14.

12.Model the System using UML: The system is assumed to have been developed using Object-Oriented concepts. Thus its static structure and dynamic behaviour can be modelled using UML.

13. Solve the UML model: Solve the UML model using analytic or simulation approaches and obtain the performance characteristics.

14. Check the obtained performance: Check if the performance obtained using SPE approach or UML-based approach is acceptable. If the performance is not acceptable, one must explore alternatives including revising the number of materialized views and / or improving the indexing techniques; or even changing the hardware components. In some cases, one may have to even revisit the performance goals. Now, repeat Steps 1 to 14. If the performance is acceptable, exit the process.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 9, September 2020 ||

IV. FRAMEWORK REQUIREMENTS

The Framework demands five generic requirements. For Data Warehouse systems, knowing the size of data to be processed plays an important role to estimate the performance of the system and hence has been included in the framework. User queries have to be executed against Data Warehouse Tables or against Views residing in OLAP Server. By knowing the views that can be materialized, we can estimate the size of data that needs to be processed for each user query. We need to solve the Materialized View Selection problem which allows us to estimate the size of the data that need to be processed when the query is executed against OLAP Server.

The software size plays a major role in the performance assessment using SPE approach. The Class Point approach provides a size measure of software by suitably combining well known OO measures [19]. The Class Method Points approach estimates the size by considering five distinct scope inputs [20]. A more accurate approach for effort and size estimation of OLAP systems is required.

Because of the characteristics of Object Oriented Modelling the companies can continue to evolve their warehouse and query tools as their business changes, instead of continuously having to restructure and rewrite their existing systems. An Object Oriented Framework for modelling Data Warehouse systems using UML models is required.

If software performance models would be generated automatically, the designers would not find it difficult to employ them in software development cycle, thus bridging the gap between software development and software performance domains. A customized simulation tool to obtain the performance metrics from a Data Warehouse architecture is required. A tool that focuses on evaluation of software performance by considering two models, namely software execution model and system execution model. So that it can quantify impact of Database design factors on OLAP performance.

Alternative Design strategies can be used to meet performance objectives. Indexing is a physical database design strategy. Most of the bitmap index implementations in commercial Data Base Management Systems are relatively simple, such as the basic bitmap index or the bit-sliced index [21,22]. New bitmap indexing techniques in the physical database design can improve the query response time.

V. CONCLUSION

We have proposed a Framework to support assessment of performance of a Data Warehouse System in the early design phase of development. The proposed Framework makes use of the Software Performance Engineering (SPE) approach of Smith with suitable modifications, Object-Oriented concepts together with UML, and simulation techniques. The proposed framework integrates performance analysis process and Data Warehouse development process to make sure that the system meets its performance goals.

REFERENCES

[1] Muhammad Obeidat, Max North, Lloyd Burgess, Richard Parker1 and Sarah North, DRIP – "Data Rich, Information Poor: A Concise Synopsis of Data Mining", Universal Journal of Management 2015, pp-29-35.
[2] Jiawei Han and Micheline Kamber, "Data Mining", Morgan Kuffman, 2nd Edition, 2006.
[3] Daniel J.Power and Shashidhar Karparathi, "Building Web-based Decision Support Systems", Studies in Informatics and Control, Vol. 11 No.4, 2012, pp 291-312.
[4] Harinarayan, V., Rajaraman A. and Ullman J.D, "Implementing Datacubes Efficiently", International Conference on Management of Data, Canada, 1996, pp 205-216.
[5] B.Ashadevi, R.Balasubramanian, "Cost Effective Approach for Materialized Views Selection in Data Warehousing Environment", International Journal of Computer Science and Network Security, Vol. 8 No.10, 2008, pp 236-242.
[6] Ravindra N. Jogekar, Ashish Mohod," Design and Implementation of Algorithms for Materialized View Selection and Maintenance in Data Warehousing Environment", International Journal of Emerging Technology and Advanced Engineering,Vol.3 Issue 9, 2013, pp 464-470.
[7] O'Neil, P., "Model 204 architecture and performance", 2nd International Workshop in High Performance Transaction Systems. Lecture Notes in Computer Science, Vol. 359. Springer,1987, pp 40–59.
[8] Kurt Stockinger, Kesheng Wu, and Arie Shoshani, "Analyses of Multi-Level and Multi-Component Compressed Bitmap Indexes", ACM Transactions on Database Systems, Vol. 35, No. 1, Article 2, February 2010, pp 1-52.
[9] C.U.Smith, L.G.Williams, "Performance Solutions: A practical guide to creating responsive, Scalable Software", Addison Wesley, 2001.
[10] C U Smith and L G Williams, "Performance Engineering Evaluation of Object Oriented Systems with SPE-ED", Computer Performance Evaluation Modelling Techniques and Tools, LNCS Vol. 1245, 1997, pp 135-154.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 9, September 2020 ||

[11] M. Marzolla, "Simulation-Based Performance Modeling of UML Software Architectures", Ph.D. Thesis: TD-2004-1, Dipartimento di Informatica, Universita Ca' Foscari di Venezia, Italy, 2004.
[12] Simonetta Balsamo Moreno Marzolla, "A Simulation-based approach to Software Performance Modeling", 9th European Software Engineering Conference held jointly with 11th ACM Sigsoft International Symposium on Foundations of Software Engineering, 2003, pp 363-366.
[13] www.tpc.org.
[14] Tony Vignaux, Klaus Muller and Bob Helmbold, "SimPy Manual,version 2.3b1" , 2012, simpy.readthedocs.org
[15] Tayfur Altiok and Benjamin Melamed, "Simulation Modeling and Analyss with Arena", Elsevier, 2007.
[16] K.S Trivedi, SHARPE, "Symbolic Hierarchical Automated Reliability and Performance Evaluator", International Conference on Dependable systems and Networks, 2002, pp 544.
[17] Linda M. Laird, M. Carol Brenna, "Software Measurement and Estimation: A Practical Approach", John Wiley and Sons, 2006.
[18] Chunhua Ju, Minghua Han,"Effectiveness of OLAP based Sales Analysis in Retail Enterprises", in the Proc. of ISECS International Colloquium on Computing, Communication, Control and Management, Vol. 3, 2008, pp 240-244.
[19] Wei Zhou and Q.Liu , "Extended Class Point Approach of size estimation for OO product", in the Proc. of International Conference on Computer Engineering and Technology, ICCET, Vol 4, 2010, pp 117-122.
[20] William Roetzheim, "Estimating Effort Using Use-Case and UML Class-Method Points", in the Proc. of UML & Design World 2005, Austin, Texas, June 2005.
[21] Wu, M.-C., & Buchmann, A.P., "Encoded Bitmap Indexing for Data Warehouses" in the Proc. of 14th International Conference on Data Engineering (ICDE), Orlando, Florida, USA. IEEE Computer Society Press, 1998, pp 201-230.
[22] Wu, K., Otoo, E.,and ShoshaniI, "A Compressed bitmap indices for efficient query processing", Tech. Report LBNL-47807, Lawrence Berkeley National Laboratory, Berkeley, 2001.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor:
7.512

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर
NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details