DATA WAREHOUSE PERFORMANCE EVALUATION

Dr. Madhu Bhan

Department of Master of Computer Applications,
M.S. Ramaiah Institute of Technology, India.

Abstract: *A Data warehouse is a collection of non-volatile, historical, summarized data that is gathered from transactional databases over time. Data warehouse systems are beneficial in supporting decision-making since they are optimized for On-Line Analytical Processing. The ability of an online analytical tool to deliver information when needed, or to provide "just in time" information for efficient decision-making, is a vital sign of success. It is important to evaluate the performance of these On-Line Analytical Processing applications in the early stages of their development. In this paper we simulate a typical On-Line Analytical Processing query sent to a Data warehouse to know its response time. By creating and resolving the Software Execution Model for a Data warehouse system, we are able to retrieve the Performance metrics. A weather case study is used to discuss the findings.*

Keywords: Large Databases, Decision Making Systems, On-Line Analytical Processing, Simulation, Performance.

1. INTRODUCTION

The creation of Data warehouses and the performance of On-Line Analytical Processing (OLAP) on top of them have become essential components of decision-making and forecasting solutions. OLAP has distinct functional and performance requirements than the online transaction processing applications. Transactional databases are based on operational databases whereas OLAP provides a multidimensional view (data cube) of aggregate data to enable quick access to strategic information for further analysis. Data warehouses and OLAP work well together. Data is managed and kept in a Data warehouse. Data warehouse data is transformed into strategic information by OLAP. It includes everything from simple navigation and browsing to computations and more in-depth analyses including time series and intricate modelling. From simple browsing and navigation to computations and more complex analysis, such as time series and complex modelling, it has all. Slicing, Dicing, Pivoting, and Roll-up are some common OLAP procedures that can be carried out on Data warehouses. A section of the data cube is chosen through Slicing and Dicing based on constant values in one or more dimensions. With Roll-up, one or more dimensions are made more generic and the relevant aggregations are carried out in the respective measures, while Pivot, presents the measures in various cross-tabular layouts. The performance of these OLAP procedures is a significant performance challenge. Because OLAP searches must access vast amounts of data and involve numerous aggregation procedures, query response time is the key performance challenge. Sales patterns, product profitability and financial forecasting are some of the applications of OLAP.

A key factor in determining if a piece of software is acceptable is its ability to meet performance requirements. Performance is essential for software systems that carry out customer-service tasks because they must deliver a quick response that customers would accept. Unfortunately, the approaches used for software development tend to overlook performance. For software systems which are complex and big, it is crucial that the performance factors be analyzed starting from the early stages of the software development life cycle. It is a standard practice in the business to evaluate performance at the end of software development, but doing so can result in the use of expensive hardware, time-consuming tuning procedures, or even a sophisticated redesign of the application. Here, in this work a methodology to integrate performance analysis into the software development process, is proposed.

2. LITERATURE SURVEY

The degree of data volatility is the main distinction between a Data warehouse and a conventional transaction database. While data in a Data warehouse is stable and updated at regular intervals (monthly or weekly), information in a transaction database is continually changing. Decision-making regarding future activities is made possible via OLAP. A Data warehouse is built for online analytical processing and decision making, as opposed to relational models, which are used to construct databases for ad-hoc querying and on-line transaction processing. It frequently adopts a snowflake/star schema. Data warehousing and online analytical processing (OLAP), though occasionally used synonymously, pertain to various parts of what are commonly called decision support systems or business intelligence systems. These system components include databases and programs that give analysts the resources they need to support organizational decision-making [1], [2], [3].

There are often many stages of study and planning throughout the life cycle of a traditional system. The system builders in the Data warehouse environment are not frequently allowed this luxury and are instead obliged to quickly put together a Data warehouse with little time for capacity planning and performance analysis.

A study of DBMS tuning elements such as buffer pool, sort heap, prefetching, and number of I/O servers is carried out and its effect on OLAP performance is analyzed. A workload model for OLAP that is based on the TPC-H benchmark was constructed [4], [5], [6] for the study. Simulations was used to determine the performance of processors and micro architectures because of their complexity. There are two methods for simulating processors: execution-driven simulation and trace-driven simulation. Trace-driven simulation replicates the timing behavior of captured or artificially created trace files on a simulated system. This method is well-established and frequently employed. Execution-driven simulation replicates the functional execution of software programs by using them as input.

The Simple Scalar serves as an example of this tactic. The limitation of a fixed instruction set and the requirement to migrate operating systems and drivers to the simulation framework are drawbacks of the execution-driven method [7]. Software performance engineering includes methods that make it easier to evaluate the performance of software systems early in the software development life cycle [8]. Software performance engineering continues to manage and forecast the software's performance throughout the stages of detailed design, coding, and testing. It also keeps track of actual performance in comparison to expectations. Software performance engineering is crucial for software engineering, and software quality in particular. Depending on the state of the system, the software performance engineering process employs a variety of performance assessment tools.

3. METHODOLOGY

To assess the performance of Data warehouse/OLAP Systems, we have proposed an algorithm for its Software Execution Model. The algorithm has the following steps

3.1. Consider a Data warehouse schema.

The most widely used schema models for data warehousing are the Fact constellation schema, Snowflake schema, and a Star schema. Project needs, available tools, and project team preferences should be used to decide which schema model should be used for a Data warehouse. Perhaps the simplest Data warehouse schema is the Star schema. It offers a clear and straightforward link between the schema design and the business entities that end users are analyzing. For typical OLAP queries, it offers performance that is highly optimized.

3.2. Calculate the size of data.

The physical size of the database is considered for measuring the performance of a system. In general, the larger the data source, the longer it takes to retrieve the query results.

3.3. Explore the expected query and its corresponding operations.

Performance depends on the query, data, indexes, and the hardware that it operates. You can get an idea of how many rows are going to be scanned and what indexes are going to be used.

3.4. Assess the hardware configuration of the deployment model

Performance can be significantly impacted by deployment settings. Workloads that create huge service demands on system resources are known to dramatically harm the performance of multi-tier systems.

3.5. Find resource usage for each query operation

To determine what resources are being used by the query. For example, the CPU and I/O usage is just one piece of information that can be used to determine your query's resource usage.

4. CASE STUDY

For handling sizable amounts of manual, digital, sensor data, performing statistical analysis, and the extraction of useful trends and patterns, the use of Data warehouse technologies in the field of meteorology and climatology might be valuable. Temperature, wind speed, precipitation, cloud cover, and other weather-related statistics may be included in a weather star schema that stores weather data, along with location, date/time, and other details. Automatic measurement stations regularly collect significant amounts of numerical and multimedia data. This data includes measurements of various meteorological factors taken every 10 minutes, every hour, or more frequently. Applications for meteorology and climatology continuously analyze the data. It frequently goes through aggregations using unique formulas. Relational database management systems have steadily been implemented over the past few years in various meteorological organizations to replace proprietary file-based application storage models. Weather forecasting, where quick access to real data is crucial, and climatology, where flexible access to high quality information on historical weather is crucial, are the two principal applications of meteorological data [9].

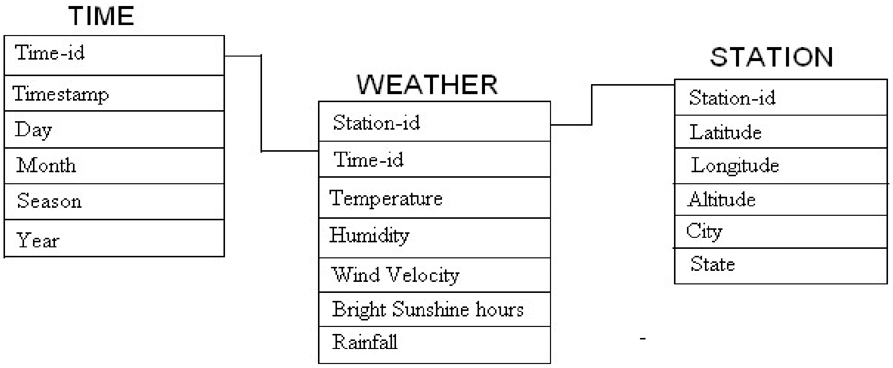

Figure 1. Star schema model of a Weather Data warehouse

In order to assess the response times to weather related queries we use our proposed algorithm to determine approximate response times in an isolated environment where there is no contention of resources. The steps of this algorithm are illustrated with queries typical of a Weather Data warehouse.

4.1. Consider a Data warehouse schema

Figure 1 shows a Star schema model of a Weather Data warehouse with two dimensions and five measures.

4.2. Calculate the size of data

In order to calculate the size of data that needs to be processed, we make the following assumptions.
Let there be 1k tuples in Station table.
Size of a row in station table = 2+8+8+8+2+2=30 bytes=240 bits
Size of **Station table** = 240000bits
On an average there are 5 weather readings per day for each station.
Size of a row in time dimension = 8+8+4+2+2+4) = 28 bytes = 224 bits
The No. of rows in Time table (assuming 10 years of data) = 5* 365 *10 =18250
Size of **Time table** = 4088000 bits
Size of the row in fact table = 8+2+8+8+8+4+8 = 46 bytes = 368bits
Number of rows in fact table = (number of recordings per day)* (365) * (number of stations) * (number of years of historical data i.e., 10)
Number of rows in fact table = 5*365*1000*10=18250000
Size of **Weather table**=18250000*368 =6716000000 bits

4.3. Explore the expected query and its corresponding operations

Let our example query be: Select Station-id, Avg(Temprature), Avg(Humidity) from Station, Time, Weather where Station-city = {‘Madras’, ’Srinagar’, Simla’} and Month= ‘June’ and Station.Station-id = Weather.Station-id and Time.Time-id = Weather.Time-id; Figure 2 shows the order of operations involved for query execution of Example.

4.4. Assess the hardware configuration of the deployment model

Figure 3 depicts the web-based Data warehouse/OLAP system deployment model [10]. The Users use the Web browser to submit requests for analyses. After receiving the user's request, the Web server sends it to the OLAP Server. If the response is on the OLAP server, it is forwarded to the Web server, which then sends it to the user. If the solution is

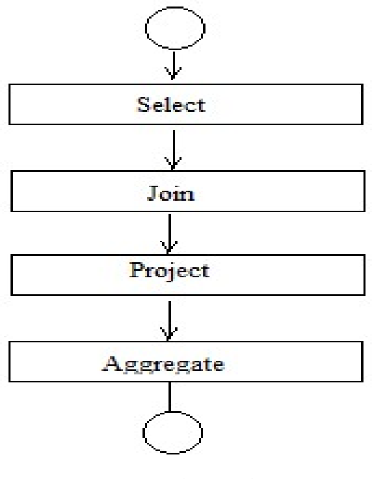

Figure 2. Query Operations

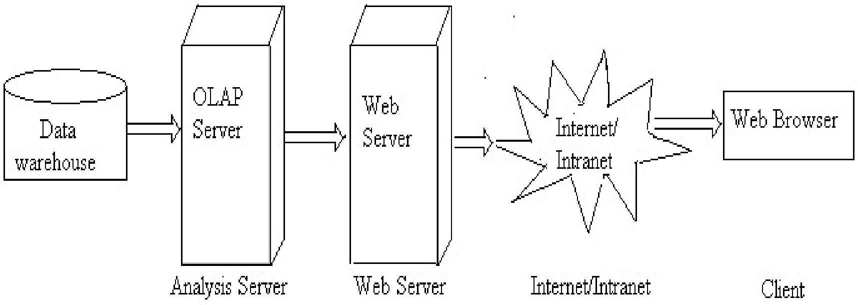

Figure 3. OLAP structure based on Web

not offered by OLAP Server, it calls the data from the Data warehouse, completes the analysis, and then returns the result to the Web server, which then sends the result to the client. Considering a typical deployment of a Data Warehouse we assume the following configurations of the hardware/software resources:

1. Data Warehouse /OLAP CPU time for data processing equal to 0.1 microseconds per page.
2. Internet (NET) speed equal to 2048 kb/second
3. Disk page size equal to 128 kb;
4. Time to access one Database/Proxy Disk page equal to 0.1 milliseconds;
5. Performance goal: The usual query is returned in under 5 seconds.

4.5. Find Resource Usage for each query operation.

Resource usage for all the query operations is listed in Table I, Table II and Table III. The hardware resources required and time taken for select, join, project and aggregate operations is calculated [11], [12].

Table 1. Select Operation

Resource	Operation	Time
Disk	Access of Station Table (240000bits)	No. of pages = *1.875* = 2 Access Time = 0.0002 sec
	Access of Time table(4088000bits)	No. of pages = 31.93=32 Access time = 0.0032 sec
	Access of Weather Table (6716000000 bits)	No. of pages=52468.75sec=52469 sec Access Time = 5.25sec

Table 2. Join Operation

Resource	Operation	Time
CPU	1st join: 18250000 Comparison operations to join Weather and Station Table.	= 1.82sec
	2nd Join: 54750(18250 * 3) Comparison Operations to join the result of 1st join and Time Table	= 0.0055sec

Table 3. Project and Aggregate

Resource	Operation	Time
CPU	Average of Temperature and Humidity	=0.01

The time consumed by each hardware resource for various components of the Query is given in Table 4. These values are obtained by summing the various execution time given in Table 1, Table 2 and Table 3.

Table 4. Hardware/Software Execution Times

Hardware Resources Query Component	DW Disk	DW CPU
Select	5.2534 sec	
Join		1.8255 sec
Aggregation		0.01 sec
Total	7.09(7.0889) sec	

5. RESULTS AND DISCUSSION

The query's overall execution time of 7.09 seconds is significantly longer than the goal of 5 seconds. Due to the massive amounts of data that must be retrieved. Hardware resources like Data warehouse disc access time are recognized as bottleneck resources.

A three-tier architecture that places the majority of the query results in the OLAP Server allows us to optimize it. The amount of information that is needed in order to answer every question from the collection of potential questions, are stored as views at the OLAP Server. The amount of data that needs to be processed to satisfy each query from the set of possible queries, will depend on the query itself and the sizes of the tables that need to be referred. We have created a software execution model where data needs to be fetched from the Database. It is a model-based execution environment, since the prediction of performance is done during feasibility study of Software Development Life Cycle (SDLC). The execution time has been obtained while ignoring problems of resource contention and multiple requests. We have further solved the model using an analytical approach to obtain the performance characteristics of the system.

6. CONCLUSION AND FUTURE ENHANCEMENT

In this paper, we present a paradigm for evaluating the performance of Data Warehouse Systems during the initial design stage of their creation. Software designers would be able to investigate multiple designs and choose the one that offers the best overall performance with the help of early performance evaluation. In order for developers of big and complex systems like Data warehouse systems, to be able to assess and understand the performance effects of various design decisions at early stages of development when changes are simple and less expensive, it is important to support early performance evaluation. Designing algorithms for the System Execution Model that include factors like the existence of other service requests, the competition for resources, service policies, etc., is part of the scope for future work.

REFERENCES

[1] *G. Garani, A. Chernov, I. Savvas and M. Butakova, "A Data Warehouse Approach for Business Intelligence", Proceedings of IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, **(2019)**, pp. 70-75. DOI: 10.1109/WETICE.2019.00022.*

[2] *D. Wang, Q. Li, C. Xu, P. Wang and Z. Wang, "Research of Data Warehouse for Science and Technology Management System", Proceedings of 2021 International Conference on Service Science (ICSS), **(2021)**, pp. 65-69. DOI:10.1109/ICSS53362.2021.00018.*

[3] *Deepak Asrani and Renu Jain, "Designing a Framework to Standardize Data Warehouse Development Process for Effective Data Warehousing Practices", International Journal of Database Management Systems, vol. 8, no. 4, **(2016)**, pp. 15-32. DOI: 10.5121/ijdms.2016.8402.*

[4] *Karwan Jameel, Abdulmajeed Adil Yazdeen, Abass Kh Ibrahim, Maiwan Bahiat Abdulrazzaq and Mayyadah Ramiz Mahmood, "Analyses the Performance of Data Warehouse Architecture Types", Journal of soft computing and data mining vol. 3 no. 1, **(2022)**, pp.45-57. DOI: 10.30880/jscdm.*

[5] *Prayag Tiwari, Sachin Kumar, Avinash Chandra Mishra,Vivek Kumar and Bodena Terfa, "Improved Performance of Data Warehouse", Proceedings of International Conference on Inventive Communication and Computational Technologies, **(2017)**, pp. 94-104. DOI: 10.1109/ICICCT.2017.7975167.*

[6] *G C Rorimpandey, F I Sangkop, V P Rantung, J P Zwart, O E S Liando and A Mewengkang , "Data Model Performance in Data Warehousing", in proc. IOP Conference Series: Materials Science and Engineering, vol 306, no. 1, **(2018)**. DOI: 10.1088/1757-899X/306/1/012044.*

[7] *Issamjebreen, Mohammed Awad, "Preliminary Study of Software Performance Models", International Journal of Advanced Computer Science and Applications, vol. 7, no. 2, **(2016)**, pp. 239-242. DOI: 10.14569/IJACSA.2016.070233.*

[8] *Gunnar Kudrjavets, Jeff Thomas, Nachiappan Nagappa, "The Evolving Landscape of Software Performance Engineering" , Proceedings of International Conference on Evaluation and Assessment in Software Engineering, **(2022)** , pp. 260–261. DOI:10.1145/3530019.3534977.*

[9] *Zhan Jie Wang and A. B. M. Mazharul Mujib, "The Weather Forecast Using Data Mining Research Based on Cloud Computing", Journal of Physics Conference Series ,vol. 910, no. 1, **(2017)**. DOI: 10.1088/1742-6596/910/1/012020.*

*[10] Roza Dastres, Mohsen Soori, "Advances in Web-Based Decision Support Systems", International Journal of Engineering Research, vol. 19, no. 1,**(2021)**, pp. 1-15. Available at: https://www.researchgate.net/publication/351065107_Advances_in_Web-Based_Decision_Support_Systems.*

*[11] Yesdaulet Izenov, Asoke Datta, Florin Rusu, Jun Hyung Shin, "COMPASS: Online Sketch-based Query Optimization for In-Memory Databases", Proceedings of International Conference on Management of Data, **(2021)**, pp. 804–816. DOI: https://doi.org/10.1145/3448016.3452840.*

[12] *Viktor Leis, Bernhard Radke, Andrey Gubichev, Atanas Mirchev, Peter Boncz, Alfons Kemper and Thomas Neumann, "Query optimization through the looking glass, and what we found running the Join Order Benchmark", International Journal of Very Large Databases, vol. 27, **(2018)**, pp. 643-668. DOI: 10.1007/s00778-017-0480-7.*