Abstract

Database optimization is one of the major concerns and the most important factor that needs to be considered when designing a very large database system. Database optimization and evaluation are usually concentrated on the query language and the organization of data. The recent advances in technologies of data-warehousing architecture, online analytical processing (OLAP), and data mining demand more rigorous optimization and evaluation strategies that involve the entire data-warehousing environment. While there are standard approaches for optimizing database queries and data structures separately, there appears to be no current and coherent research efforts in assessing optimization techniques in data-warehousing systems. Instead of concentrating on query language and data structure in optimizing a database, our research focused on all the qualitative elements or optimization factors in the entire data-warehousing system. That includes the language, the data, the architecture, as well as the operation. Our previous research [14] developed an approach to evaluate an information system qualitatively using Kivia graphs. In this paper, we will use the Kivia graphs to profile a data-warehousing system so that the results can be used for performance evaluation, comparison, and tuning.

Keywords: Data-warehousing system, Kiviat graph, performance optimization, performance comparison, performance tuning, qualitative evaluation profile.

1. Introduction

Database technologies such as relational databases [6] have been used for transactional mode of processing for a long time. In the past decade, information has become an asset of an organization just like human resources or capital asset. Business intelligence is now vital to the survival of an enterprise in the current competitive economic environment. The transactional nature of databases has proven that it is inadequate to provide the intelligent information required by a business enterprise. Transactional database lacks historical data and summary data or data with higher level of abstraction. Data-warehousing system is the environment that provides business intelligence to an organization with historical and summary level of data. Figure 1 shows a typical data-warehousing system. It is an environment that includes the transactional or operational databases in the back-end, data staging area to prepare historical and summary data for data warehouses and data marts in the middle, and information tools in the front-end where OLAP activities are conducted. As shown in the figure, there is a meta-data management process that describes all the objects in the data-warehousing system so that they can be monitored, managed, and controlled. The reader is referred to [4, 5, 7, 8, 19] for more detailed information about data-warehousing systems.

![Figure 1](image-url)

The traditional approach of system evaluation is to gather performance statistics on a system. System improvements are then carried out to enhance performance. Our research [14] takes the software engineering approach that the construction of a data-warehousing system should start with the qualitative optimization factors during system development. Quantitative data are then collected during system operations. Finally, the qualitative factors and the quantitative data can be reconciled for performance improvement. In this paper, we establish the qualitative profile for a data-warehousing system and we do this not from the limited scopes of language and data structure, but from all the aspects of the entire data-warehousing environment.

Related works in this area include [10, 20] which contributed many quality factors discussed in this paper and [1] which incorporated a performance
perspective in designing a data-warehousing system. But there is no comprehensive view of performance optimization factors that allows easy comparison of data-warehousing systems. In this paper, we propose a qualitative framework for data-warehousing system optimization and performance tuning. We classified the optimization strategies in the data-warehousing system into seven levels and grouped them into four major strategy levels. We then identified twenty-four optimization factors in all the strategy levels for achieving optimal performance in a data-warehousing system. Research results are presented using Kiviat graphs [13]. The Kiviat graphs for two data-warehousing systems of two major groups of users – regular users and ad hoc users are also going to be discussed.

2. Seven Levels of Optimization Strategies

Optimization strategies to efficiently deliver users’ requests are a major concern in data-warehousing environment due to the fact that data warehouse consists of historical, summarized, and huge volume of consolidated data. As shown in Figure 1, the data-warehousing system consists of many components and when OLAP activities are conducted in this environment, the different components may play a role in satisfying a user’s requests depending on the type of information that the user asks. In this paper, we classified performance optimization strategies into four major levels. These four strategy levels are the query language level, the data structure level, the architecture level, and the operation level. The effectiveness and efficiency of answering users’ requests depend on the query language, the data structure, as well as the hardware architecture that constitute the data-warehousing system. The effective and efficient coordination of all the components in a data-warehousing environment depend on the operational processes and procedures.

Within these four major strategy levels, we identified three additional levels that are between these major levels for a total of seven strategy levels. Between query language and data structure levels, we have the language/data level. Between architecture and language/data levels, we have language/architecture level and data/architecture level. The relationships among all the levels in a data-warehousing system are shown in Figure 2.

In order to compare the performance of different data-warehousing systems and to optimize the performance of a data-warehousing system, we need standard points of references for comparison and for performance tuning. We identified twenty-four qualitative elements which we called optimization factors in the data-warehousing system. These optimization factors are the procedures, methodologies, and technologies that are the most important factors to improve the performance of a data-warehousing system. The optimization factors under each of the strategy level are discussed below. The seven strategy levels are designated as L1 to L7 and the optimization factors are identified as F1, F2, and so on under each level.

2.1 Query Language Level (L1)

When a user performs an OLAP analysis, he/she uses a user interface or some query languages such as SQL [15]. Whether it is SQL or a query language behind a user interface, the first optimization strategy is the query language itself. There are a total of three optimization factors that the technologies are mature enough to be employed at this strategy level.

(F1) – Language optimization: It is obvious that the first thing we can improve the performance of a data-warehousing system is to have an optimized compiler and most likely an optimized interpreter for the query language [11].

(F2) – Heuristic optimization: If the language is SQL, move the relational operations (e.g., selection and projection) down the query tree will optimize a query. Another heuristic is to consolidate several queries into a single query to minimize input/output overhead (e.g., disk accesses) [9, 12].

(F3) – Semantic query optimization: The technique of modifying a query into another more efficient query using constraint rules can be used to optimized query processing [2, 18].

2.2 Data Structure Level (L2)

There are a lot of techniques to organize data so that an analyst can efficiently access them. At the data structure level, we identified optimization techniques that can be applied to the operational databases, the data warehouses, or the data marts in the data-warehousing environment. All of these techniques are described in [5, 10, 17, 19].
The most
demanding capability
for OLAP mode of operation
is the multi-dimensional cube.
It organizes data for
efficient processing of multi-dimensional
queries in data-warehousing applications.
It models the data in
terms of what the analyst
think and helps users
understand the data. The
structures include the latest
relational OLAP, multi-dimensional
OLAP, or hybrid
OLAP technologies.
These multi-dimensional
structures are usually
resided in the data marts.

(F2) – De-normalization: This is very common
in relational databases that allow controlled
redundancy or data replication. In data-warehousing
systems, summary tables of pre-aggregated data
can be created to avoid massive computation
at execution time. If the data organization is
the relational OLAP structure, some of the
relational tables can be pre-joined before run time.

(F3) – Clustering: Clustering techniques include
the creation of array of data as an index to more
detailed level of data, data compaction to reduce disk
accesses by eliminating data fragmentation, and data
co-location to speed up data access operation.
Clustering techniques in data-warehousing
environments are discussed in [22].

(F4) – Data searching: Indexing and hashing are
common techniques to eliminate scanning large
volumes of data. Another simple technique is pre-sort
certain data to avoid sorting at run time.

(F5) – Partitioning: The creation of multiple
tables partitioned along some dimension (e.g.,
monthly data partitioned into twelve tables, one for each
month) that will accelerate incremental data backup
and recovery. Load time into indexed tables will also
be reduced.

(F6) – Token-based database technology: The
technique of replacing text with numbers (tokens) to
reduce data volume. Database size will be
substantially reduced if the data records consist of
many repeated fields. Small data volume allows
in-memory manipulation.

2.3 Language/Data Level (L3)

This strategy level concerns the optimization
techniques of the query language that operates
on the data. In other words, the data are organized in such
a way that allows easy access and manipulation of the
data. Three optimization factors are identified at this
strategy level.

(F1) – Views: The creation of view is a well-
established technique to expedite data access.
The physical view belongs to the data structure level but
the virtual view is a better fit in the query language
level. So views are put into this language/data level.
The views are formed usually based on usage pattern.

2.4 Architecture Level (L4)

The architecture level addresses the hardware
and its configuration that make up the
data-warehousing system. Again, the idea is to identify the
technologies that are common and mature to improve
the performance of a data-warehousing system.

(F1) – Hardware infrastructure: This factor
includes all the state-of-the-art technologies that can
be used to upgrade a data-warehousing system such as
CPU, memory, disks, and communication network
e.g., high bandwidth with fiber optics). This
optimization factor also includes the arrangement of
hardware such as two or three tier architecture and
scalable architecture depending on the particular
configuration of an organization to gain optimal
performance [10].

(F2) – Data staging area: The creation of a buffer
area for data extraction, transformation, and loading
for the integration process of a data-warehousing
system will eliminate the bottleneck between the
operational databases and the data deposit area (see
Figure 1). The staging area also allows parallel
integration and parallel loading of data into the data
deposit area.

(F3) – Parallelism: High performance hardware
architecture is definitely required for a high
performance data-warehousing system. This
optimization factor involves the latest parallel and
scalable architectures such as SMPs (Symmetric
Multiprocessors), MPPs (Massively Parallel
Processors), and clusters [10].

2.5 Language/Architecture Level (L5)

The overall architecture governs the entire data-
warehousing system. We describe the various data
drilling capabilities by an OLAP analyst under this
language/architecture level. This can arguably fit into
the language/data level. However, the fact that the
drill-through capability allows the accessing of data
from the data mart to the data in the staging area or
data in the operational databases, the architecture of
the data-warehousing system should be constructed to
allow such access path. Therefore, we put the data
drilling capabilities under this strategy level.

(F1) – Data drilling capabilities: This
optimization factor includes the two major capabilities
drill across and drill through. Drill across allows
access paths between multi-dimensional cubes. Drill
through allows access paths from multi-dimensional
structures to the staging area or to the operational
databases.

2.6 Data/Architecture Level (L6)

The organization of the data under the data-
warehousing architecture is the concern of this
optimization strategy level.

(F1) – Data marts: Not all data-warehousing
system consists of data marts. The creation of data
marts extracts subset of the data warehouse or
departmental data. This allows customized views of
data to tailor to user requirements. For example, the
construction of a particular multi-dimensional cube is
the usual structure of a data mart.

(F2) – Exploration data warehouse: If the
architecture of a data-warehousing system is flexible
effective to allow the creation of an exploration data
warehouse or data mart, then it will increase the
performance of the data-warehousing system by
offloading data to a temporary storage place to do
special analysis. This exploration data warehouse is
only for short-term purpose but its operation will not
affect the main operations of the entire data-
warehousing system.

2.7 Operation Level (L7)

The level that governs the entire data-
warehousing environment is the administration for the
operational processes and procedures of a data-
warehousing system. This is an important level to
ensure all the components of the architecture are well
coordinated to maximize the data-warehousing system
availability.

(F1) – Software architecture: The first
optimization factor we considered at the operation
level is the underlying software in the data-
warehousing system. Using third party software that is
fast by industry benchmark will improve that overall
performance of the system.

(F2) – Activity and data monitoring: Lots of
activities and data can be monitored to enhance a data-
warehousing system. Examples of monitoring
activities are: usage statistics can help pre-
computation or pre-aggregation decision at the data
structure level; workload analysis to plan and schedule
OLAP activities and do capacity planning; data usage
tracker to identify dormant data; and data monitoring
to ensure the quality of data.

(F3) – Data loading: When data is integrated in
the staging area and load into the data repository,
pipelined and partitioned parallelism can speed up the
loading process. Data loading (e.g., in different time
zones) can ensure the availability of data in a wide
geographical area.

(F4) – Scheduling: An overall intelligent
scheduling for data loading, backups, maintenance,
refreshing data, monitoring data, etc. can improve the
performance of a data-warehousing system by
maximizing the availability of the environment to the
analysts.

(F5) – Dormant data archiving: This is the
process of eliminating data that have zero or a very
low probability of access. It will reduce the size of the
data and thus speed up any OLAP activities.

(F6) – Workload management: This optimization
factor related to scheduling but this factor concerns the
management of different requests from the analysts. It
also involves the blocking of query if the query
consumes too much system resources or hang the
entire system.

3. Qualitative Evaluation Profiles

There are totally twenty-four optimization
factors in seven strategy levels that can be considered
to fine tune a data-warehousing system to maximize
its performance. If a data-warehousing system has all
these twenty-four optimization factors, it will be a
completely optimized system that in theory should
have the best performance. The best way to profile
these qualitative elements is to show the profile in a
Kiviat graph [13, 16]. The major strength of Kiviat
graph is the visualization of patterns that is easy to
compare one system with another system.

A Kiviat graph consists of arbitrary number of
axes that are the variables of interest to profile a
system. Each axis is scaled from say, one to ten to
indicate the relative performance measure of each
variable. Usually the maximum scales in all the axes
are the same so that the shape of the Kiviat graph for
an optimal system is a circle. To profile a data-
warehousing system in a Kiviat graph, we used the
optimization strategy levels as the axes and the
optimization factors as the scales on the axes.
In order to compare different systems using Kiviat graphs, we need an optimized graph as a standard for comparison. In [14], we showed the construction of an optimized graph by combining the seven levels of optimization into four strategy levels (SL1 to SL4) and distributing the twenty-four optimization factors evenly on the axes. The resulting Kiviat graph is shown in Figure 3. It shows the Kiviat graph of a totally optimized data-warehousing system. The figure profiles an optimal data-warehousing system qualitatively. The circle represents that each strategy level has the maximum optimization factors of six. The diamond inside the circle shows the shape for performance comparison purpose and we called it the shape of the Kiviat graph.

3.1 Performance Comparison and Performance Tunning

To quantitatively compare the performance of two data-warehousing systems, certain metrics and benchmarks have to be set and run. The results will then be analyzed and compared. To qualitatively compare two data-warehousing systems, we can check the optimization factors and profile these factors in Kiviat graphs for the respective systems. If the area bounded by the maximum values in each axis of the Kiviat graph (i.e., the shape of the Kiviat graph) is less than the one shown in Figure 3, then there is room for improvement. The list of optimization factors checked will indicate what other factors to look for to fine-tune the performance. But that does not mean if a system with less optimization factor and then it is not an optimized system. It all depends on the usage of the data-warehousing system as discussed in the following.

3.2 Kiviat Graph for Farmers

As discussed in [10], there are two different groups of users of data-warehousing systems – the farmers and the explorers. The farmers are regular users. They have precise requirements and they tend to access higher level of summarized data. These users know what they are looking for so they can employ all the optimization factors to their data-warehousing system to optimize their operations. Figure 4 shows a typical Kiviat graph of a data-warehousing system that is optimized for regular users. In other words, if we want to build an optimized data-warehousing system for farmers, the Kiviat graph shows what factors we can design for the system.

Since the regular users can established all the optimization factors, the shape of Figure 4 is similar to Figure 3. The only difference is that the factor of “exploration data warehouse” has been taken out from SL3. It is because regular users do not usually need a flexible architecture to create an exploration data warehouse. So for an optimal performance of a data-warehousing system used by regular users, the shape of the Kiviat graph is similar to the shape of the totally optimized Kiviat graph.

3.3 Kiviat Graph for Explorers

The other major group of data-warehousing system users is the explorers or the ad hoc users. Explorers do not use the system regularly. They do not have requirements and they tend to access detailed level of data. They look at relationships of data rather than occurrences of data. Figure 5 shows a typical Kiviat graph of a data-warehousing system that is optimized for ad hoc users.

In Figure 5, SL1 has two optimization factors: language optimization and heuristic optimization. Even though the ad hoc users do not employ all the optimization factors at SL1, the data-warehousing system at least provides an optimized language compiler or interpreter and the users can do some heuristic optimization to speed up query processing.
SL2 has only one optimization factor and that is the de-normalization factor. It is because for explorers, they do not have all the requirements to build a multi-dimensional structure. They do their analysis out of the data from the operational data sources that are usually in relational architecture. So the data-warehousing system at least has de-normalized data in relational tables to expedite query processing. SL3 has three factors. They are hardware infrastructure, parallelism, and exploration data warehouse. For ad hoc users, a data-warehousing system can be built with the best hardware infrastructure and the latest parallel and scalable architecture. It is obvious that the creation of an exploration data warehouse is useful for explorers. Finally, SL4 consists of only half the optimization factors at this strategy level. The three factors are software architecture, scheduling, and workload management. The reason for including software architecture is the same as hardware infrastructure. Even for ad hoc users, we can do some scheduling and workload management to manipulate their requests.

Even though the area bounded by the shape of this Kiviat graph is very small compared with Figure 4, it is an optimized system for ad hoc users.

4. Conclusions and Future Research

To totally optimize a data-warehousing system, not only we have to look at the query language and the data structure as in transactional databases, we also need to account for all the elements in a data-warehousing system. In this paper, we classified the total optimization strategies into seven levels and grouped them into four major strategy levels. We then identified a total of twenty-four optimization factors in these strategy levels. The strategy levels and the optimization factors are presented using Kiviat graphs for performance comparison and for performance tuning. The qualitative performance profiles of most data-warehousing systems lie somewhere between the profiles of the two extremes as shown in figures 3 and 5.

The twenty-four optimization factors we identified are qualitative in nature. To actually compare the performance of data-warehousing systems, quantitative data need to be gathered. Future research is needed to apply and generate new performance metrics on the data-housing environment. It will be interesting to see the Kiviat graph with the performance quantitative data versus the Kiviat graph with the qualitative optimization factors. In theory, both Kiviat graphs should have a direct relationship with each other. That is, the bigger shape of the Kiviat graph with the qualitative optimization factors should correspond to a better (bigger shape) of the Kiviat graph with the performance quantitative data.

With the advances in data-warehousing architecture, OLAP systems, and data mining technologies, future research is also needed to identify additional or modify the existing optimization factors mentioned in this paper. Inter-relationships among optimization factors also need to be examined.

5. Acknowledgements

This research was supported by a grant from the Center for Research, College of Science and Health, William Paterson University of New Jersey.

6. References


