Utilizing Data Marts for Performance Analysis

David Wiese

Lehrstuhl für Datenbanken und Informationssysteme
Friedrich-Schiller-Universität Jena
Ernst-Abbe-Platz 2
07743 Jena
wiese@informatik.uni-jena.de

Abstract

Improving performance analysis and moving towards automation of large information management platforms requires a more intuitive and flexible source of decision making. This paper addresses the idea of employing data marts and leveraging the OLAP paradigm for the efficient navigation through large volumes of performance measurement data hierarchically. The administrators or system managers navigate through adequately (re)structured measurement data trying to detect performance bottlenecks, identify causes for performance problems or assess the impact of configuration changes on the system and its representative metrics. Of particular importance is finding the root cause(s) of an imminent problem, threatening availability and performance of an information system. This is supposed to be accomplished within moderate amount of time and little processing complexity in contrast to traditional static reporting.

1 Introduction and Motivation

In order to maximize productivity and efficiency, minimize the total cost of ownership and to ensure service level agreements of an information management platform, Performance Management, that is the collection, storage and analysis of performance-relevant data, has become of fundamental importance and can be regarded as a mixture of science and art. Performance monitoring and analysis tools like IBM DB2 Performance Expert [IBM05] aim high at assisting database and system administrators in coping with these tedious tasks. However, using and reporting against such tools often reveals an overwhelming wealth of data, making it almost impossible for analysts and even experienced database administrators to separate the wheat from the chaff. Consequentially, it seems more than warrantable to yield a more intuitive source of information on which to base performance-analytic decision making. This paper deals with ideas to harness widely adopted multidimensional concepts in order to represent hierarchical layers of performance measurement data and to allow explorative, interactive and intuitive OLAP problem analyses on large (real-time and historic) data volumes in a hierarchical top-down manner.

The rest of this paper is organized as follows. In Section 2 we briefly motivate common performance data collection and storage. We then present the basic principles of OLAP and the multidimensional data model in Section 3, and discuss the application of multidimensional analysis for performance analysis. Subsequently, we conclude with a summary and the obligatory outlook in Section 4.

2 Performance Data Collection and Storage

In order to achieve at least sufficient performance, pursuing a customized, proactive and reactive performance management strategy becomes vitally important. Instead of detecting problems when they

occur, or, worse, have already been affecting system performance adversely, the proactive effort targets at avoiding problems in the first place by trend recognition, prediction and strategic as well as preventative planning. However, not all adverse effects on the system can be avoided beforehand. Hence, a sophisticated reactive methodology for localizing the problem, pinpointing when and where it occurs, who is affected and what resources are involved, determining the root-cause(s) and finally fixing the situation is needed as well.

As data must be collected, before it can be analyzed and embedded in the decision making process effectively, both approaches are in the need of a profound amount of system measurement variables (metrics) that represent performance and system resource allocations (system state) at fixed points in time and time periods. Metrics originate from various sources [IBM03] (for IBM DB2 these are, among others, the System Monitor, Explain Facility, DB and DBM configuration parameters, registry variables, schema definitions, log files, operating system data, etc.) with different locations and formats. But wherever the measurement data comes from, it needs to be stored to allow sophisticated (time series) analyses. Storing performance data can be done in manifold manners (flat files, RDBMS, spreadsheets, etc.) as evaluated in [SMP98]. Ideally, there should be a single, global monitor that supervises all layers in the software stack and extracts useful performance indicators for the application, network, DBMS, operating system and hardware (CPU, memory, disk) into a central, consistent and integrated short- (nearly real time) and long-term diagnostic performance metric repository. The truth is, that in practice every layer has its own set of monitoring and analysis tools which hardly communicate and do not deliver an acceptable basis for analytical decision making.

3 Harnessing OLAP for Performance Analysis

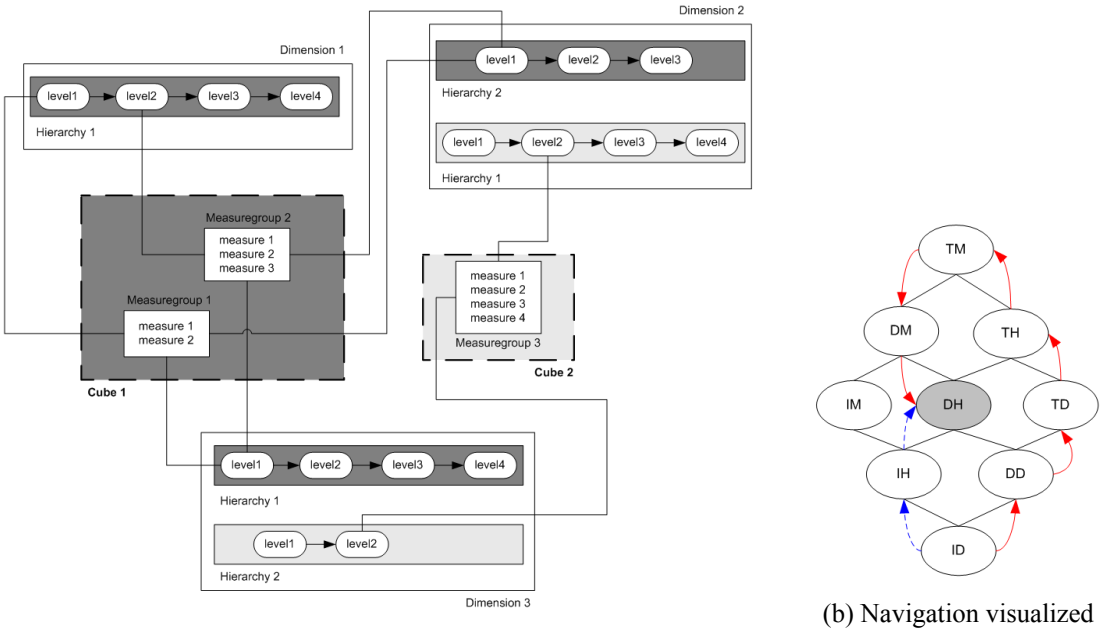
Present performance reports delivered either from monitoring tools or from the RDBMS itself rather overwhelm or confuse end-users like administrators and analysts with a poorly conceived information overload than supporting them in decision making. Furthermore, the rudimentary SQL analysis capabilities turned out to be unsuitable for more sophisticated reporting requirements. Manually filtering out the feasible facts from relational tables becomes exhausting, time-consuming and error-prone, even for experienced users.

OLAP can be seen as a set of technologies and tools that assist in the quick analysis of (business) data [Cod+93]. For the analysis using OLAP, the manifold relationships among (business) data are mapped to multidimensional data structures. OLAP technologies have gained ground within the last decade in business-oriented decision support environments and the business community. Therefore, it seems legitimate to evaluate the merits and drawbacks of utilizing OLAP techniques for representation and analyses of performance-relevant measures.

3.1 The Multidimensional Data Model

The basis of the multidimensional data model (see Figure 1(a) and detailed explanation in [Wie05]) is rooted in the difference between qualifying and quantifying data that are reflected by two key concepts: dimensions and measures. Dimensions in multidimensional models serve for the unambiguous, orthogonal structuring of the data space and describe different ways of looking at the information (e.g. time, database objects and workload). The intersection of dimensions acts as an index and identifies the data points the analysts intend to analyze, the so-called measures (e.g. index pool and data page hit ratios, physical writes/reads, number of commits/rollbacks as well as dynamic/static SQL statements). In contrast to the descriptive, textual and qualifying dimension attributes, these are mostly numerical, additive and quantifying information. Measuregroups group sets of measures semantically by subject area in order to reduce and manage complexity. The structure of the data is similar to that of an array. Like the dimensions of an array, dimension levels provide the indexes for identifying individual cube cells, as well as the situation in which the measures were taken and where they are meaningful. As it

is often sufficient to draw decisions from a high-level point of view, not everybody is interested in detailed (raw) data. In order to increase manageability, dimensions are broken down into hierarchy levels (e.g. table - tablespace - bufferpool - database - instance; secs - mins - hours - days) with differing granularities. Such an abstraction process is an instinctively known human activity. Utilizing these hierarchical relationships, analysts can drill-down/roll-up along their data to view different levels of abstraction and only deal with the level of information appropriate to their current assignment.



(a) Multidimensional concepts

(b) Navigation visualized with a lattice

Figure 1: Multidimensional model as basis for navigation

3.2 Multidimensional Navigation

Typical multidimensional navigation is a top-down approach. With the intention to obtain a "big-picture" view, the user normally starts analysis from an upper level of abstraction (entry point) by looking at a specific set of summarized problem- or situation-relevant key performance metrics (totals, averages, counts, etc.) that might not seem quite right. Issuing a combination of sequential and mutually dependent queries, from there he crawls along paths through the multidimensional data space in order to reveal inter-connections and dependencies between metrics that should be examined next. When requiring insight into more fine granular data, the analyst can incrementally increase the level of detail with a drill-down operation. Rolling-up does the opposite, it moves upwards in hierarchical relationships, thereby decreasing level of detail. Besides, setting focus on specific regions of data that appear promising can be performed by slicing, dicing and pivoting on the current data cube.

A typical business analyst will not query against relational tables[1] directly. Instead, he specifies his requests with a graphical navigation interface. In contrast to common static reporting, it is not necessary anymore to generate an entirely new report in order to see more details or to change the perspective on the data. With an appropriate graphical tool[2], it can be as easy as a mouse click, and the requested points of interest can be displayed. Access to valuable information can be gained quickly and intuitively without having to learn a new query or programming language or dialect.

---

1 Presuming a relational OLAP implementation via star or snowflake schemas.

2 See www.tdwi.org/marketplace for a comprehensive list of vendors and tools.

Aside from utilizing sophisticated tools, multidimensional data can also be analyzed using SQL directly, applying the ANSI-SQL99 OLAP extensions [Mog05], like the extended GROUP BY functionality (grouping sets, GROUPING function, ROLLUP and CUBE operators, etc.) as well as OLAP functions (aggregating, partitioning, windowing, ranking, etc.) to form the typical queries on star schemata, affecting a large number of tables and mainly applying equi-join predicates between fact and dimension tables, as well as local selective predicates on the dimension tables.

3.3 Navigating Multidimensional Performance Metric Cubes

OLAP user access is typically intuitive, multidimensional, dynamic, navigational, explorative, and session-oriented with task- and user-specific patterns and the goal of identifying new or unexpected relationships between variables as quick as possible. However, several fundamental challenges and problems in analyzing the multidimensionally structured data prevail. These are creating, finding, correlating and navigating cubes in order to resolve or foresee possible bottlenecks within acceptable time.

One of the biggest difficulties for users is to find the appropriate set of cubes for conducting analyses on. This can be a highly non-trivial and possibly iterative trial-and-error process that should be accomplished by experienced DBAs only. At the beginning, the user has to get some idea which measures (cubes) are best suited for addressing the problem and providing an entry point into following analyses. In [Wie05] some aspects of an autonomic cube advising and selection component in the area of performance tuning are introduced.

Once the appropriate cube candidates have been selected, there are many ways for finding the actual cause of the problem. Utilizing dimension hierarchies for navigation provides the analyst with the flexibility to explore various paths of interest that may all lead to the root cause(s). Some paths seem more naturally intuitive to the user, but require more time for analysis. In order to reduce system down times or time for decisive analyses, the paramount goal is to find the cause(s) as quickly as possible. From a set of representative query sequences, ideally, it seems optimal to select the one with the lowest accumulated execution time. Apparently, this can be difficult and depends on the skill and experience of the user. Interestingly, the authors of [HHY99] propose a layer on top of multidimensional data that provides for automated problem isolation of performance problems. They claim that automated drill down is sufficiently general so that it offers the possibility of improved productivity in a wide range of computing environments. However, analysts still must explore different dimensions manually in order to determine which provides the best characterization of the problem being isolated.

The sequential nature and diversity of exploration possibilities (possible paths) in multidimensional analysis can be clarified visually using the lattice approach (see [SMP98] for details). There is one vertex for each combination of dimension levels, representing an SQL query on the base tables with a certain cardinality (number of rows returned), and an edge among vertices that can be calculated by a single drill-down/roll-up operation. The art in analyses lies in considering only those queries of importance. Figure 1(b) aims at showing various "movement" strategies and illustrating the possible navigation process of an analyst. Each node is labeled with a combinatory instance of dimension levels. The chosen hierarchies for both dimensions are: (T)able - (D)atabase - (I)nstance and (M)inute - (H)our - (D)ay. The lowest node represents the highest level of abstraction with the least cardinality in contrast to the uppermost one which describes information at highest detail. The arrows (dotted and drawn through) ought to represent two of numerous sequential paths (with operations restricted and simplified to drilling-down and rolling-up) an analyst could take to best localize the (database-related) problem. Obviously, the dotted path seems more effective regarding time and effort to arrive at the conclusion. Regrettably, the shortest path is not always the most evident to a human analyst.

In addition, analysis paths may also lead to a dead-end. Often, users have to go back in history, backtrace their routes through the data and find alternative paths to draw conclusions. Unfortunately, as

problems and cubes might correlate to each other, besides intra-cube exploration, inter-cube comparisons need to be considered as well. Therefore, finding a clue by navigating one initial cube might serve as a starting point for the next (set of) cube(s). Regrettably, setting focus on new cubes might, at worst, enforce their creation and loading with data, thus causing additional unwanted overhead on the system.

4 Summary and Outlook

Applying OLAP techniques supports sophisticated problem detection and analysis instead of naively guessing problem causes and parameters that take effect. Aligning complex data by dimensions that influence the key business factors as well as representing and capturing natural hierarchical relationships in data gives users the ability to comprehend the conceptual scheme and recognize dependencies and implications more intuitively. In spite of all promising sounding preferences, practice must show suitability of the multidimensional paradigm. The biggest challenge lies in finding the trade-off between more intuitive analyses and the overhead in creating, filling, selecting and navigating these structures.

Hence, in [Wie05] we show how a simple, lightweight and extensible data mart creation framework (XDMF) has been developed to easily rearrange and aggregate performance data and construct subject-oriented multidimensional data marts as well as valid and complex SQL-based data transformation and movement mappings in order to support subsequent analyses. Data marts can then be created on-demand at run-time in order to organize data (into symptoms or categories) and to allow subsequent analyses over time, tracking back problems or even correlating imminent incidences to past events.

Literature

[BaGu01] Bauer, A.; Günzel, H.: *Data Warehouse Systeme Architektur, Entwicklung, Anwendung*. dpunkt. Heidelberg. 2000.

[Cod+93] Codd, E.F. et. al.: *Providing OLAP to User-Analysts: An IT mandate*. Arbor Software. 1993.

[HHY99] Hart, D.G.; Hellerstein, J.L.; Yue, P.C.: *Automated drill down: An approach to automated problem isolation for performance management.* In: Proceedings of the Computer Measurement Group. 1999.

[IBM03] Alur, N.; Falos, A.; Lau, A.; Lindquist, S.; Varghese, M.: *DB2 UDB/ WebSphere Performance Tuning Guide*. Redbook. IBM Corp. 2003.

[IBM05] Chen, W-J; Ma A.; Markovic, A.; Midha, R.; Miskimen, M.; Siders, K.; Taylor, K; Weinerth, M.: *DB2 Performance Expert for Multiplatforms V2*. Redbook. IBM Corp. 2005.

[Mog05] Mogin, P.: *OLAP Queries and SQL1999*. Issues in Database and Information Systems. Victoria University of Wellington. 2005.

[SMP98] Sriram, C.; Martin, P.; Powley, W.: *A Data Warehouse for Performance Management of Distributed Systems*. Dept. of Computing and Information Science, Queen's University at Kingston. 1998.

[Wie05] Wiese, D.: *Framework for Data Mart Design and Implementation in DB2 Performance Expert*. Diploma thesis. University of Jena and IBM Böblingen. 2005.