An Experimental Approach and Monitoring Tools for Evaluating a Dynamic Cubing System

Anne Tchounikine, University of Lyon, Lyon, France
Maryvonne Miquel, University of Lyon, Lyon, France
Usman Ahmed, University of Lyon, Lyon, France

ABSTRACT

In this paper, the authors propose an approach and different tools to evaluate the performance and assess the effectiveness of a model in the field of dynamic cubing. Experimental evaluation, on one hand allows observing the behavior and the performance of the solution, while on the other hand it lets one compare the results with those of the other competing solutions. The authors’ proposal includes an experimental workflow based on a set of configuration parameters to characterize the inputs (data sets, queries sets and algorithm input parameters) and a set of metrics to analyze and qualify the output (performance and behavior metrics) of the solution. They have identified a number of useful tools necessary to develop an experimental evaluation strategy. These monitoring tools allow elaborating the execution scenarios, collecting output metrics and storing and analyzing them online in real-time as well as later in off-line mode. Using a use-case model, the authors show that the framework and the proposed environment help carrying out a rigorous experimental evaluation of a dynamic cubing solution.

KEYWORDS

Behavior and Performance Analysis, Dynamic Cubing, Experimental Evaluation

INTRODUCTION

This article addresses the issue of experimental evaluation we faced during a research project within the field of real-time data warehousing. It presents an extended version of (Tchounikine 2015). We will try here to highlight how the use of a rigorous approach can promote experimental evaluations and complement academic evaluation performed with formal demonstration and qualification of models and algorithms. The concerned research project was conducted in the field of new agile BI applications, operating in fast evolving environments. In this earlier work, a solution is discussed in order to provide better data freshness and reduce analysis latency (Ahmed, Tchounikine, Miquel & Servigne, 2010). This solution allows on-the-fly insertions of facts and members by means of frequent atomic insertions, leading to fast aggregate updates. We provide the definition of a Dynamic Cube based on an un-ordered multidimensional and multi-level data space, enabling its evolution. A tree structure incrementally stores detailed data and aggregates for the densest regions of the data space thanks to a split strategy that promotes refinement of aggregates at increasing lower level. This proposal was implemented in a prototype that consists in a suite of tools starting from facts loading to OLAP navigation in the dynamic cube. This prototype was firstly aimed at functional testing and helped us
to demonstrate the feasibility of the solution. In a second step, we used it to carry on experimental evaluation of this solution.

Experimental evaluations can be of two types. A first type of evaluations consists in observing the behavior and the performance of the solutions under some varying although tightly controlled circumstances: this helps in understanding its working, allows experimenting some formal postulates and provides feedback to improve the solution or tune some running parameters. A second type of evaluations consists in carrying out comparative studies and compares obtained results to those of competing solutions. Here again, it is necessary to tightly control the execution circumstances in order to ensure fair comparison.

We carried out these two types of evaluation using the prototype in context of our work on Dynamic Cubing. For this purpose, we followed an experimental approach based on a workflow that outlines the input parameters determining the execution context and experiments, the output metrics that are relevant for the evaluation of results, and a set of tools allowing the design of scenarios, tuning of inputs, collection and interpretation the outputs. We believe this experience can contribute to show how experimental evaluations can gain in being designed and adopted from the very beginning and maintained throughout a project.

This paper is organized as follows: The second Section summarizes our earlier contribution in the field of Dynamic Cubing and motivates our work on experimental evaluation. Next section lists and defines the input parameters used to customize the settings of an experimental evaluation. Section “Evaluation metrics” defines the metrics used to evaluate and observe the solution. Next section presents the tools used to perform experimentation and gives samples of experiment tracks and shows how the input parameters are set to reproduce use cases of test scenario, and how metrics are used to analyze them. We end with related work and conclude.

**MOTIVATIONS FOR EXPERIMENTAL EVALUATION OF THE DYTREE**

**Dynamic Cubing Model**

Data warehouses are increasingly desired in time critical applications such as supervision applications, fraud/threat detection, energy management, operational intelligence etc. In these applications, the facts are expected to be inserted on-the-fly, by means of frequent atomic insertions leading to the requirement of fast aggregate updates in order to maintain the hypercube in real-time. Moreover, if new dimension members are frequently inserted at run time, then it is clear that the re-arrangement for ordering data is not feasible, leading to a need for fast and dynamic cubing. The special nature of a dynamic cube makes the usual strategies and data structures inefficient. In this article, we will not expand on the topic of dynamic or real-time cubing. We will just give a brief overview of our earlier work in the followings. The detailed proposal is available in Ahmed, Tchounikine and Miquel (2014).

Let us take a very simplified use case of a data warehouse for Building Temperature Control, with two dimensions, time (day< month) and sensor location (probe < floor) (Figure 1). The measure is the temperature with average as aggregation function.

Our earlier contribution in the field of dynamic cubing includes the following propositions: a multidimensional data model is defined that operates in a data space $S$ defined by un-ordered hierarchical dimensions (Figure 2). Each axis represents a dimension of the model and coordinates are made from all its members. Points in $S$ are detailed or aggregated facts. We note $l^i_j$ the $j^{th}$ level of a hierarchical dimension $D_i$ and domain $\{l^i_j\}$ its set of members. Here is how the data space is dynamically built.

Initially, the multidimensional data space $S$ and its dimension axes are empty. When the first temperature is recorded ((sep-1, a3) in Figure 2), then its coordinates (day, probe) are plotted in first position on the axes and the point appears in the multidimensional space. Thereafter gradually as the facts come in, their coordinates are added to the axes and the points are placed in the multidimensional
Related Content

An Exposition of CaRBS Based Data Mining: Investigating Intra Organization Strategic Consensus
[www.igi-global.com/chapter/exposition-carbs-based-data-mining/44293?camid=4v1a](www.igi-global.com/chapter/exposition-carbs-based-data-mining/44293?camid=4v1a)

Query Interaction Based Approach for Horizontal Data Partitioning
[www.igi-global.com/article/query-interaction-based-approach-for-horizontal-data-partitioning/125650?camid=4v1a](www.igi-global.com/article/query-interaction-based-approach-for-horizontal-data-partitioning/125650?camid=4v1a)
OLAP Analysis Operators for Multi-State Data Warehouses
Franck Ravat, Jiefu Song and Olivier Teste (2016). *International Journal of Data Warehousing and Mining* (pp. 20-53).
[www.igi-global.com/article/olap-analysis-operators-for-multi-state-data-warehouses/171118?camid=4v1a](www.igi-global.com/article/olap-analysis-operators-for-multi-state-data-warehouses/171118?camid=4v1a)

Bagging Probit Models for Unbalanced Classification
[www.igi-global.com/chapter/bagging-probit-models-unbalanced-classification/40411?camid=4v1a](www.igi-global.com/chapter/bagging-probit-models-unbalanced-classification/40411?camid=4v1a)