Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 48 (2015) 256 – 262

Procedia Computer Science

International Conference on Intelligent Computing, Communication & Convergence

(ICCC-2015)

Conference Organized by Interscience Institute of Management and Technology,

Bhubaneswar, Odisha, India

Architecture Based Materialized View Evolution: A Review

Anjana Gosain[a], Sangeeta Sabharwal[b], Rolly Gupta[c*]

[a]*Professor, USICT,Guru Gobind Singh Indraprastha University, Delhi, India*
[b]*Professor, NSIT, Delhi University, Delhi, India*
[c]*Research Scholar, NSIT, Delhi University, Delhi, India*

Abstract

Data Warehouse evolution is a critical problem in present scenario due to perpetual transactions and change in their structure arising out of continual evolving users' requirements. Handling properly all type of changes is a crucial process as it forms the core component of the modern DSS. Therefore DW has to be updated periodically according to different type of evolution of information sources. The problem of evolving an appropriate set of views is subjected to as the materialized view evolution problem. Many different materialized view evolution methods have been proposed in the literature to address this issue. This paper provides a survey of materialized view evolution methods. The paper aims at studying the materialized view evolution in relational databases and data warehouses as well as in a distributed setting. It defines an evolutionary approach for highlighting the materialized view evolution problem by identifying the three main dimensions that are the basis in the classification of materialized view evolution methods namely; (i) Framework, (ii) Architecture and (iii) Model/Design Model. This study reviews architecture based materialized view evolution methods, by identifying respective potentials and limits.

Peer-review under responsibility of scientific committee of International Conference on Computer, Communication and Convergence (ICCC 2015)

Keywords: Architecture ; View Maintenanc;, Materialized view evolution

Peer-review under responsibility of scientific committee of International Conference on Computer, Communication and Convergence (ICCC 2015)
doi:10.1016/j.procs.2015.04.179

* Corresponding author. Tel.: 0-98-99-148-597
*E-mail address:*rollygupta02@gmail.com

1. Introduction

Materialized views act as a data cache that gather information from distributed databases and support faster and reliable availability of already computed intermediate result sets (i.e. responses to queries). Evolution in data warehouse may be generated by change in schema, changes in software and the change in data warehouse requirements. Materialized view evolution approach focuses on choosing materialized views in the design process of data warehouses or maintaining a materialized view in response to data changes or to data sources changes and sometimes to monitor the DW quality under schema evolution. Whenever the underlying base relation is modified the corresponding materialized view also evolves in reaction to those changes so that it can present quality data at the view level.

The materialized view evolution issue has been investigated in several contexts: query optimization, warehouse design, data placement in a distributed setting, web databases, etc. Many diverse solutions to the materialized view evolution problem have been proposed and analyzed through surveys [Dhote et. al. 2009, Halevy 2001, Labrinidis et. al 2009]. However, none of the above mentioned surveys provides a classification of materialized view evolution approaches in order to identify their advantages and disadvantages. Our survey fills this gap.

The goal of the materialized view evolution is to simplify the design, implementation, maintenance and management of data warehousing approaches. Therefore, we classified the materialized view evolution into following dimensions -- Framework, Architecture and Model/Design Model. Based on the methods involved in evolution of materialized views in a data warehousing dimensions, they can be categorized further. So, the taxonomy used for further classification is – View Evolution, Basic View Maintenance, Incremental VM, Self Maintainable Maintenance, Not self Maintainable Maintenance, View selection, View Synchronization, View Adaptation. We present a comparative study of the various research works explored in context of architecture based dimensions and methods. The rest of the paper is organized as follows: Section 2 presents a comparative study of the various research works explored. Section 3 presents the reviews and result. Finally section 4 contains the conclusion and discusses open issues.

2. Comparative Study

We have analyzed architecture based materialized view evolution methods on several parameters and presented their comparative results in the table below:

TABLE1: COMPARISON OF ARCHITECTURE BASED MVE METHODS

S. No	Authors	Techniques /Category Adapted	Issues Addressed/ Changes Handled	Architecture support/ perspective	Method's Activities/ Goals	Addressed attributes	Applicable framework stage	Advantages	Disadvantages	Types of Queries/ Operation	Tool Support/ Implementation
1.	Sumi Helal, et al.	IVM	manual and automatic hoarding	3-Tier Architecture	flexible synchronization	accessibility, availability, and consistency	VM	Ubiquitous data access	fierce competition addressing	RM	Coda-based
2.	Janet L. Wiener, et.al.	VM	autonomous sources +	WHIP prototype	Distributed	scalability Modularity,	VM	Modular and scalable	Issues of crash	RM	C++ and C