WARLOCK : A Data Allocation Tool for Parallel Warehouses

Thomas Stöhr Erhard Rahm

University of Leipzig, Germany
http://dbs.uni-leipzig.de

1 Introduction

We present the *WARLOCK* tool to automatically determine a parallel data ***war***ehouse's al***loc***ation to dis***k***. This GUI-equipped tool is implemented in Java and utilizes an internal cost model and heuristics to determine a disk allocation minimizing both I/O work and query response times. *WARLOCK* recommends a ranked list of fragmentation candidates, a detailed query performance analysis and a tailored physical allocation scheme for relational star schemas and bitmap indexes. It supports multi-dimensional fragmentations and can deal with data skew for parallel data warehouses based on a Shared Everything or Shared Disk architecture.

2 Data allocation approach

We consider relational star schemas with denormalized, hierarchically organized dimension tables and one or more fact tables. Each dimension level is represented by a particular dimension attribute. Fact tables contain a set of measure attributes for aggregation and refer to dimension attributes by foreign keys. The considered workload consists of a variety of multi-dimensional join and aggregation (star) queries on the fact tables that refer to dimension attributes. We support standard bitmaps and encoded bitmaps that work as bitmap join indexes [2] to avoid costly fact table scans.

A suitable horizontal fragmentation of the fact tables is essential for parallel query processing and to limit the number of fragments to be processed for a given query. We follow a multi-dimensional, hierarchical range fragmentation strategy called MDHF that we recently proposed and evaluated in [5]. One-dimensional fragmentations are a special case of this approach. A fragmentation is defined by selecting a set of fragmentation attributes from the dimensional attributes, at most one per dimension. All fact table rows corresponding to a single value combination of the fragmentation attributes are assigned to one fragment. This approach is able to confine star query work to a subset of the fragments if at least one fragmentation dimension is accessed [5]. Bitmap fragmentation exactly follows the fact table fragmentation to keep the relationship of indicator bits and fact table rows.

Proceedings of the 27th VLDB Conference, Roma, Italy, 2001

We support a logical round-robin allocation scheme where fact table and bitmap fragments are stored on disk according to a logical order of the fragmentation dimensions. Under notable data skew we apply a *greedy* size-based allocation scheme to keep disk occupancy balanced. The scheme stores fragments, ordered by decreasing size, onto the least occupied disk at a time.

3 Tool overview

With *WARLOCK*, we want to automate the complex data fragmentation and allocation task as much as possible. The goal is to find a disk allocation that optimizes I/O overhead for bitmap and fact table access as well as query response times by utilizing parallel processing for a representative set of queries. *WARLOCK* utilizes several heuristics and a respective cost model. Fig. 1 presents the tool architecture.

3.1 Input layer

The user (DBA) specifies parameters relevant to the internal cost model. As a first step, a *star schema* with its attributes, hierarchy cardinalities, row sizes and fact table volumes has to be defined. Data skew may be incorporated at the bottom level of each dimension by specifying a zipf-like data distribution. Furthermore, a few *database* and *disk parameters* are to be set (page size, number of disks and their capacity, average rotational, seek and data transfer

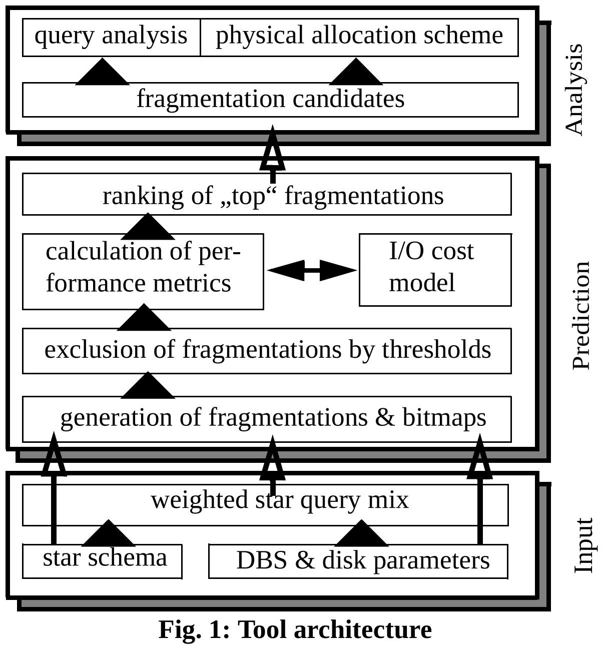

query analysis
physical allocation scheme
fragmentation candidates
Analysis
ranking of „top“ fragmentations
calculation of performance metrics
I/O cost model
exclusion of fragmentations by thresholds
generation of fragmentations & bitmaps
Prediction
weighted star query mix
star schema
DBS & disk parameters
Input

Fig. 1: Tool architecture

times, prefetching granule). With respect to the performance-sensitive prefetch size [3,5], WARLOCK offers the choice to set a fixed value or to determine itself optimal values for fact tables and bitmaps, which strongly differ with respect to fragment sizes.

Similar to APB-1 [1], several *weighted query classes* can be specified according to the subset of dimensions they access and their relative share of the workload.

3.2 Prediction layer

At this layer, I/O access costs and query response times are predicted for different fragmentation candidates to determine the most suitable choices. We limit the overall evaluation space to „point“ fragmentations (attribute range size = 1), which keeps enough potential to achieve a sufficient number of fragments [5]. Additional thresholds are applied to exclude fragmentations that, for instance, cause fragment sizes to drop below the prefetching granule etc.

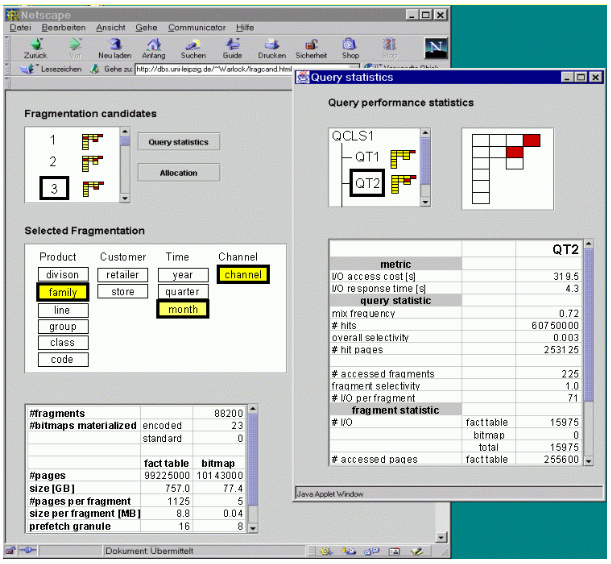
WARLOCK uses a twofold metric to determine the goodness of a fragmentation considering both *I/O access cost* or *overhead* (throughput) and *I/O response time which* are estimated by means of a analytical model we proposed in [3]. Often the throughput and response time goals are contradicting making it necessary to find good compromise solutions. Fragmentations declustering query hits broadly over many fragments and database pages often enable a high degree of parallelism and small response times, but may lead to a high number of disk I/O thus limiting throughput. On the other hand, fragmentations clustering query hits within a few fragments may limit I/O volume but only allow for small degrees of parallelism thus causing high response times. To consider these trade-offs, WARLOCK uses a simple heuristic preferring fragmentations reducing overall I/O requirements, which is also advantageous with respect to multi-user query processing. For all fragmentations, it first determines the overall I/O access cost for the considered query mix. Subsequently, the leading X% fragmentations are ranked with respect to the overall I/O response time they achieve. The resulting top fragmentations are then presented to the user for analysis and final selection.

WARLOCK determines a bitmap scheme per fragmentation that encompasses standard bitmaps on low-cardinal attributes and hierarchically encoded bitmaps on high-cardinal attributes.

3.3 Analysis and output layer

WARLOCK outputs the ranked list of fragmentation candidates and for each fragmentation a set of bitmap indexes and a disk allocation. Moreover, the user can request a detailed statistic per fragmentation for query classes (Fig. 2). It comprises a database statistic (#pages, #fragments, fragment sizes), I/O access statistic (#accessed fragments and pages, #I/Os), I/O response times and a prefetch granule suggestion. The leading fragmentations are visualized for further analysis and comparison.

The physical allocation of a fragmentation specifies the distribution of fact table and bitmap fragments down to single fragments as well as the resulting disk occupancy and access distribution. Furthermore, a disk access profile


Netscape
Datei Bearbeiten Ansicht Gehe Communicator Hilfe
Zurück Vor Neu laden Anfang Suchen Guide Drucken Sicherheit Shop Stop
Lesezeichen Gehe zu: http://dbs.uni-leipzig.de/~Warlock/fragcand.html
Query statistics
Allocation
Dokument: Übermittelt
Java Applet Window

Fragmentation candidates

1
2
3

Selected Fragmentation

Product	Customer	Time	Channel
divison	retailer	year	channel
family	store	quarter	
line		month	
group			
class			
code			

#fragments		88200
#bitmaps materialized	encoded	23
	standard	0
	fact table	bitmap
#pages	99225000	10143000
size [GB]	757.0	77.4
#pages per fragment	1125	5
size per fragment [MB]	8.8	0.04
prefetch granule	16	8

Query statistics

Query performance statistics

QCLS1
QT1
QT2

		QT2
metric		
I/O access cost [s]		319.5
I/O response time [s]		4.3
query statistic		
mix frequency		0.72
# hits		60750000
overall selectivity		0.003
# hit pages		253125
# accessed fragments		225
fragment selectivity		1.0
# I/O per fragment		71
fragment statistic		
# I/O	fact table	15975
	bitmap	0
	total	15975
# accessed pages	fact table	255600

Fig. 2: Fragmentation / query analysis (screenshots)

per query class is visualized.

WARLOCK provides several options to facilitate *interactive fine tuning*. Disk parameters, query load specifics and bitmap configurations can be interactively adapted to examine the performance variations they imply. For instance, the user may decide to exclude some of the suggested bitmap indices to limit space requirements [4].

4 Tool demonstration

During the demonstration we will use WARLOCK for various schemas and workloads, including APB-1-based configurations. We demonstrate the input and analysis process and present a detailed query performance statistic for the resulting fragmentations candidates as well as a calculated and visualized allocation scheme. Attendants may modify the parameter configurations and let WARLOCK compare the results and, furthermore, may enter their own data warehouse schema and query mix that is relevant to their work and examine the results.

References

[1] *APB-1 OLAP Benchmark, Release II*. OLAP Council, Nov. 1998. http://www.olapcouncil.org/research/bmarkly.htm

[2] P. O'Neil, G. Graefe: *Multi-Table Joins Through Bitmapped Join Indices*. ACM SIGMOD Record 24 (3), 1995

[3] T. Stöhr: *Analytical I/O analysis of data allocations for parallel data warehouses (in German)*. Proc. German database conference BTW, Oldenburg, Germany, Springer, March 2001. http://dol.uni-leipzig.de

[4] T. Stöhr: *A Data Allocation Tool for Parallel Data Warehouses*. Technical Report, University of Leipzig, Germany, June 2001

[5] T. Stöhr, H. Märtens, E. Rahm: *Multi-Dimensional Database Allocation for Parallel Data Warehouses*. Proc. 26th VLDB Conference, Cairo, Egypt, Sep. 2000.