Association for Information Systems
AIS Electronic Library (AISeL)

MWAIS 2012 Proceedings | Midwest (MWAIS)

5-2012

V+H: Architecture to Manage DSS and OLTP Workloads

Maria Del Pilar Angeles
Universidad Nacional Autónoma de México, pilarang@unam.mx

Victor Gonzalez-Castro
Universidad Nacional Autónoma de México, victor.gonzalez.castro@exalumno.unam.mx

V+H: Architecture to Manage DSS and OLTP Workloads

María del Pilar Angeles
Universidad Nacional Autónoma de México
pilarang@unam.mx

Victor González-Castro
Universidad Nacional Autónoma de México
victor.gonzalez.castro@exalumno.unam.mx

ABSTRACT

In the last few years research has been done in order to define the best approach that DBMSs must follow to manage different workloads. Some approaches have followed the "One size fits all" trying to incorporate all features in a row-oriented DBMS (also called horizontal) to manage both OLTP and DSS workloads. Additionally, there have been specialized DBMS following a columnar approach (also called vertical) that focuses on the growing demand to efficiently manage DSS workloads. The present paper is aimed to propose a combination of both vertical and horizontal DBMS to best manage OLTP and DSS workloads. We have used mature, commercially available products from a single vendor and developed a custom middleware Decision Query Module that identifies the best option for most efficient execution of a query. This V+H architecture also offers the functionality of a mirrored DB without paying twice of the storage.

Keywords

Vertical DBMS, Columnar DBMS, OLTP, DSS, Architectural Configuration.

INTRODUCTION

Providers of DBMS are incorporating many features to address to the growing demands of corporate information systems. The most implemented approaches are the columnar approach, also called vertical and the relational approach also called row-oriented or horizontal DBMS. Vendors and open source DBMSs have followed the "one size fits all" approach, by implementing almost any kind of Information Systems through relational DBMS. Therefore, current horizontal DBMSs are trying to solve any kind of workload, but mainly OLTP and DSS. For instance, Oracle (Akadia, 2012) introduced the materialized views for improving DSS queries resolution. Sybase (Garbus and Gupta, 2006) has introduced configuration parameters within the Adaptive Server Enterprise (ASE) indicating the query optimizer to behave differently on OLTP, DSS or mixed workloads.

From the research perspective, there has been further investigation (Gonzalez-Castro, V., MacKinnon, M. L. and Marwick, D. 2006; Gonzalez-Castro, V., MacKinnon, M. L. and Angeles M.P. 2009a; Gonzalez-Castro, V., MacKinnon, M. L. and Angeles M.P. 2009b; MonetDB 2012; Stonebreaker M. Abadi D., Batkin A., Chen X., Cherniack M., Ferreira M, Lau E., Lin A, Madden S., O'Neil E., O'Neil P., Rasin A, Tran N. and Zdonik S. 2005) that has demonstrated that columnar is more appropriate for data warehousing environments and DSS workloads achieving unbeatable performance numbers on analytical queries. This efficiency derives from the fact that analytical queries only retrieve few columns of the table and the inherent reduction on system I/O, but if the query retrieves all columns of the table involving many rows this columnar approach can decreases its performance, (Stonebreaker M., Bear Ch., Çetintemel U., Cherniack M., Ge T., Hachem N., Harizopoulos, S., Lifter J., Rogers J., and Zdonik S., 2007). In order to remediate the present situation we are proposing to combine vertical and horizontal DBMSs within architecture to address both kinds of workloads. The following section presents existing approaches for managing mixed workloads from the industry and academic perspective. Section 3 explains our approach termed V+H architecture. Section 4 is focused in our research in progress regarding the implementation and a set of experiments we have carried out. The last section corresponds to conclusion and future work.

BACKGROUND: EXISTING APPROACHES TO MANAGE MIXED WORKLOADS

Most of the organizations have both OLTP and DSS workloads. However, there is not a single approach able to efficiently manage and execute queries within a specific DBMS where the subjacent model is the best option for better performance.

In order to manage mix workloads different approaches have been followed.

Disparate OLTP and DSS systems

The first approach has been the implementation of separate systems to perform OLTP and Decision Support for reporting purposes. These systems are also called reporting servers which might have the same structure and keeping apart reporting from OLTP. Further approaches are based on the creation of data warehouses to manage the DSS or reporting workloads. The main difference from the previous approach is that Data Warehouses usually have a different data model design (star or snowflake) from the transactional system, (Devlin B. A., Murphy P.T., 1988).

The disadvantages of this approach are that different data models are used and data is populated on asynchronous fashion, generally the Data Warehouse is feed from the transactional system. Furthermore, queries are predetermined to be executed against one system or other.

PAX

The approach, Partition Attributes Across PAX, (Ailamaki A., DeWitt D. J., Hill M. D. and Skounakis M., 2001) is aimed to the improvement of the cache performance by data value clustering at attribute level within contiguous data pages.

Traditionally, data are stored following the N-ary Storage Model. The NSM stores row relations in contiguous data pages; each row has a record header RH, which contains a nulls bitmap, offsets in case of variable length data values and further specific information from the implementation. Each new row is inserted in the first free contiguous data space starting from the beginning of the data page. However, NSM shows poor cache performance, during predicate evaluation, because the query processor retrieves the value of the attribute involved in the query from each record in the relation along with other attribute values stored next to it, wasting useful cache space to store data no referenced.

The PAX motivation is to preserve the attribute data values of each row within the same page as is used in the NSM and take advantage of the data cache.

PAX partitions data rows vertically within each data page, storing together the same attribute data values within mini pages.

In the case of applying a predicate on a fraction of a record, as PAX stores together all the values for the same column, on a cache miss all the data values corresponding to the involved attribute are loaded to cache, and to reconstruct a record requires performing a mini-join among the minipages contained in the same page.

Oracle Exadata

Oracle describes the Exadata Hybrid columnar compression in (Greenwald R., Stackowiak R., Maqsood A. and Bhuller M. 2011), where tables are organized in sets of thousands of records called Compression Units CUs. Within CUs, data are organized per columns and then compressed.

The columnar organization allows similar data values stay together, improving the data compression. This strategy is very useful in case of massive data loads and heavy queries, improving performance of update activity.

When data that is required to satisfy a query predicate does not need to be decompressed, only data returned to the user is decompressed.

As each compression unit contains all the columns of a set of records, entire rows are retrieved with a single I/O, satisfying queries involving all columns without sacrificing response time.

Fractured Mirrors

The Fractured Mirrors approach (Ailamaki, A. Ramamurthy R., DeWitt D. and Su Q, 2002) proposes a new mirroring scheme using both NSM and DSM models. This scheme combines the best aspects of both models, along with the added benefit of mirroring to better serve an ad-hoc query workload. While DSM seems to be ideal for selections with low projectivity and low selectivity, the NSM model is tuned for workloads that are highly selective and uses most of the attributes. Therefore, fractured mirrors involved mirroring by software storing in two disks NSM and DSM models each. A set of experiments were conducted to prove the effectiveness of both models in a mirror fashion. However, their evaluation of the system had primarily focused on the TPC-H query workload. They never tested OLTP applications nor approached that given a query workload how to decide which of the storage models would be best suited. In our current approach we are using already efficient schemes to handle variable length records and NULL values from commercial data management providers.

THE V+H ARCHITECTURAL CONFIGURATION

The present research focuses on the proposal of the V+H architecture for the improvement of mixed workloads derived from OLAP and OLTP applications. This proposal is concerned to the development of a middleware called Decision Query Module (DQM). This module will be built upon two commercial database manager systems supporting relational and columnar database models from a single vendor. The DQM would be able to decide from which database (vertical or horizontal) execute the query on the bases of the query plans in order to solve them more efficiently. Our approach is more easy and cost effective to implement in comparison to the fractured mirrors approach which considers developing a DBMS from scratch. As mentioned before, the Decision Query Module is aimed to manage mixed workloads while the work of (Ailamaki, A. et al., 2002) was more concern about mirroring data for recovery purposes.

The Decision Query Module automatically sends the query to the more convenient repository rather than sending the query manually by the DBA with the corresponding disadvantage of waste of time and making wrong decisions on where to send each query. Also once the decision has been made by the DBA, the query always go to the vertical or the horizontal database without considering that the conditions within the DBMS may change at any time (new indices created, fresh or out of date statistics, data fragmentation, considerable changes on the size of the data both large increase or decrease, etc.) However, these changing conditions are considered at the generation of the query plan, which is taken into consideration by the DQM.

Previous research has demonstrated that vertical model is the best option for Data warehouse environments (Gonzalez-Castro, V., et al., 2009a; Gonzalez-Castro, V., et al., 2009b), and relational model is best suited for transactional applications. However, there would be cases where is not that easy to assume such behavior. For instance, within OLAP applications might be some queries that considering the query plan the cost is lower if it is executed on a relational repository.

Nowadays, organizations invest large amounts of money on disaster recovery plans, fault tolerant applications thorough replication or mirroring data. Information is so important for business that decision makers do not mind to pay twice storage, software and hardware infrastructure and IT specialists. The V+H architecture is not requiring such investment of money, not even twice storage. The OLAP and OLTP data would be stored within a relational database and a columnar database, is well known that columnar database takes less disk storage than relational database (Gonzalez-Castro, V., et al., 2009a; Gonzalez-Castro, V., et al., 2009b). Therefore, same information would be stored twice without paying twice of the storage. Considering the DQM, each query would be executed in the best repository, which is determined as follows:

- User or application write a "select" statement and send it to be resolved by V+H.
- The DQM sends the query to be evaluated on both DBMSs.
- Each DBMS get the Query Execution Plan (QEP) to determine the cost of the select statement. There are different methods to do the costing of the plan. We have chosen to do the costing based on the number of physical I/Os. Each DBMS returns its own QEP as output to the DQM.
- The DQM parses both QEPs and normalizes both costs in terms of physical I/Os as each DBMS gives its weight on different units.
- Once the normalized number of I/Os is computed, then it is send to the "decisor" module which decides to which repository the select statement will be send, according to the one with the minor number of I/Os (lower cost).
- Now the query is send to the chosen repository for execution.
- Finally the DBMS solves the query and returns the result set.

In the case of updates, inserts or deletes the DQM would be able to send the transaction to both databases and each DBMS will use its own native mechanism to do the insert, update or delete. The V+H architecture is shown in Figure 1.

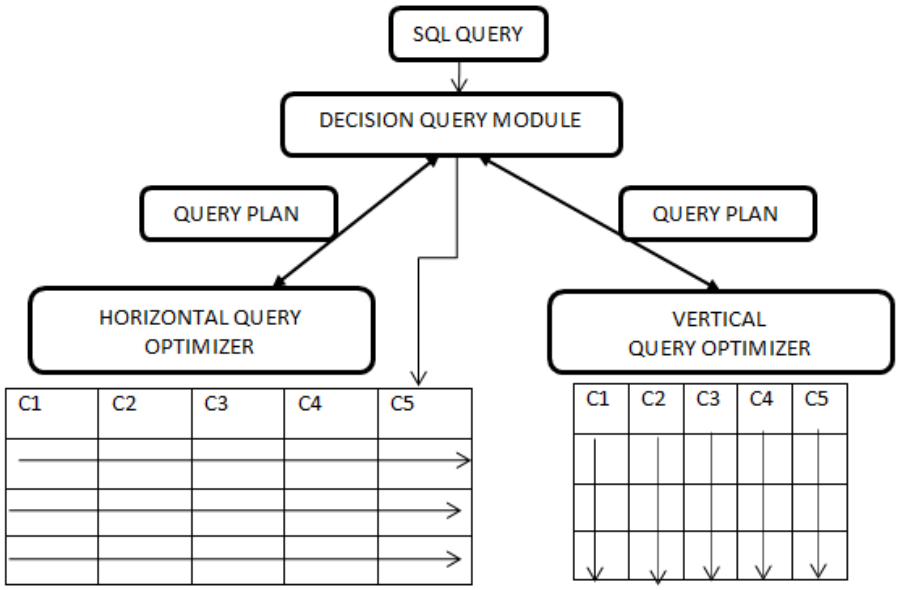

Figure 1 V+H Architecture to manage mixed workloads

THE V+H ARCHITECTURE IMPLEMENTATION AND EXPERIMENTS FIGURES/CAPTIONS

The V+H architecture has been implemented through OLAP and OLTP queries from TPC-H and the TPC-C benchmarks, a columnar database manager, a relational database manager from a single vendor. The DQM would be able to send update, delete and insert transactions to both databases.

A set of experimental test plan has been developed and still under execution. Such experiments are being executed with a Scale Factor of 1GB (database). This section presents the results we have until now. In the case of analytical applications, Table 1 shows queries 1 to 19 which are defined for the TPC-H benchmark which is oriented to DSS workloads. In the case of operational applications Table 2 shows queries 31 to 37 which are defined for the TPC-C benchmark, to represent OLTP workloads. As expected those queries from TPC-H, were evaluated to be executed on the vertical repository represented as column (e) and those coming from TPC-C, were evaluated to be executed on the horizontal repository. Columns (f) and (g) are the actual times for each query. In this case the queries were executed on both repositories in order to validate that the decision made by the DQM is right. Of course after validate that the decisions are right, the DQM will send the queries to just one of the repositories to avoid being solving both queries all time and generate unnecessary work load on both DBMSs; instead of that the query will be send to the horizontal or vertical repository whichever is more convenient.

(a)	(b)	(c)	(d)	(e)	(f)	(g)
Query	TPCH-queries	Horizontal Cost	Vertical Cost	Sent to	Horizontal Time (sec.)	Vertical Time (sec.)
Q1	Pricing Summary Report Query	26,945,878	1,536	Vertical	20.74	7.41
Q2	Minimum cost supplier query	746,538	768	Vertical	13.02	7.20
Q3	Shipping Priority Query	23,092,924	15,104	Vertical	217.25	4.31
Q4	Order priority checking query	9,428,696	49,888	Vertical	435.70	1.51
Q5	Local supplier volume query	13,578,510	98,688	Vertical	46.35	4.41
Q6	Forecasting revenue change query	23,416,700	98,688	Vertical	47.45	0.97
Q7	Volume shipping query	20,297,094	51,288	Vertical	4.20	1.37

Q8	National Market share query	7,784,406	221,288	Vertical	53.46	2.23
Q9	Product Type Profit Measure query	11,524,833	49,496	Vertical	3.63	13.03
Q10	Returned item reporting query	13,760,452	237,056	Vertical	166.69	4.91
Q11	Important stock identification query	3,272,746	246,536	Vertical	142.33	1.18
Q12	Shipping modes and order priority	25,787,208	246,536	Vertical	6.67	1.26
Q13	Customer Distribution query	12,318,092	320,464	Vertical	178.59	5.48
Q14	Promotion Effect query	24,593,422	482,320	Vertical	8.49	0.77
Q15	Top Supplier Query	330,071,155	482,320	Vertical	3.62	1.40
Q16	Small-Quantity-Order Revenue query	2,552,320	482,320	Vertical	3.94	3.94
Q17	Discounted Revenue query	894,430	489,528	Vertical	2.27	1.15
Q18	Potential Part Promotion query	820,902	667,496	Vertical	3.01	1.82
Q19	Suppliers who kept orders waiting query	20,830,532	667,888	Vertical	124.89	6.75

Table 1. Experiment results according to TPCH benchmark

(a)	**(b)**	**(c)**	**(d)**	**(e)**	**(f)**	**(g)**
Query	TPC-C queries	Horizontal Cost	Vertical Cost	Sent to	Horizontal time (sec.)	Vertical time (sec.)
Q31	Find address of the warehouse	108	2968	Horizontal	0.45	3.88
Q32	Find address of the district	108	3920	Horizontal	0.10	1.10
Q33	Find number of customers with an specific last name	196	4872	Horizontal	0.12	0.90
Q34	Find name, address, phone number balance, credit, discounts with an specific last name ordered by first name	220	7816	Horizontal	0.09	1.01
Q35	Find the name, balance of an specific customer and the items of the his/her most recent order	162	7816	Horizontal	0.05	0.69
Q36	Find the most recent order	328	8224	Horizontal	0.05	0.94
Q37	Find the items of an order	224	11168	Horizontal	0.12	1.02

Table 2. Experiment results according to TPCC benchmark

Considering the results of the experiments, the transactional queries (31 to 37), as they access few data, they require few I/O operations. Therefore, the DQM evaluates the QEP with small number. However, the same set of transactional queries evaluated for the vertical repository returns a large number of I/O operations. There are a number of reasons for this behavior. The first reason is that the vertical DBMS manages larger page size than the horizontal DBMS; this is because on OLAP environments large amounts of data are processed and a large page size helps to return such quantity of information. The

second reason is due to the vertical databases need to rebuild the records for full row return, requiring a large number of I/O operations in comparison with horizontal DBMS. Row-oriented DBMSs store all record values contiguously while the columnar-oriented DBMSs store contiguously as many values as possible for the same attribute on a single page. Furthermore, the opposite is also true, the horizontal DBMS needs to read more information (reflected here as a large number of I/Os), because it reads all values of each record even when few columns are presented to users.

The experiment results have also shown that analytical queries (1 to 19) executed within the vertical DBMS read fewer pages as each page has many values for the same attribute, and because as a larger page size is utilized the number of I/Os is consequently reduced.

The experiment results are consistent with the expected outcomes, due to the nature of the queries. The only exception is query 9, which was evaluated to be cheaper to execute on the vertical repository, but the actual execution times are showing that it ran faster on the horizontal repository, it will be necessary to do more research on this query behavior but until now our DQM is making 96% right decisions on where to execute the query. It could be fine as the Query Execution Plan cost is made based on statistical information therefore QEP are good estimates of which will be the execution behavior, but is not 100% precise.

CONCLUSION

The present research is aimed to improve the execution time within mixed workload environments with OLAP and OLTP applications.

We present a V+H architecture with a Decision Query Module that can decide to which data repository (horizontal or vertical) send the execution of queries according to their query plan. The outcomes from the experiment results show that the DQM module is making the right decisions so far. Organizations can benefit of implementing the proposed V+H architecture as they could reduce their queries processing time by execute them towards the most efficient repository. The results presented in this paper show that OLTP queries are best resolved by the horizontal approach and those OLAP queries are best resolved on the vertical repository, but organizations have mixed workloads and if they continue implementing any kind of information system on just one repository, horizontal or vertical, some of their queries would perform badly, referenced as a "normal" expected behavior lately.

Even on organizations that could have both repositories, the decision of where to execute the queries depends on the application or the database administrator, with no consideration of the specific workloads that are occurring at each instant on the database environment. With the implementation of our V+H architecture, in the case of a DBMS failure, the queries can be executed over the remaining DBMS. This might present performance degradation rather a system completely down. Furthermore, performance degradation is already experienced within organizations which are using just one DBMS.

As future work, the next experiments will include other queries and not as specific as the ones from TPCs benchmarks to measure the DQM capacity to mix queries on both repositories.

The V+H architecture allows investment savings; response time improvement; it also reduces query execution cost by considering two data models within a mirroring configuration with less than twice disk storage requirements.

The V+H architecture is focused on organizations that are planning or have implemented a fault tolerant at database level, and are able to spend onto a strong infrastructure in order to avoid loss of information.

The present work presents a Data Query Module that allows the reduction of time overhead of mixed workloads.

The DQM is easy and low cost implemented because is based on already developed strong database manager systems rather than developing a new database manager.

ACKNOWLEDGMENTS

This work was supported by a grant from Dirección General de Asuntos del Personal Académico, UNAM (IACOD project IC100411).

REFERENCES

1. Gonzalez-Castro, V., MacKinnon, M. L. and Marwick, D. (2006) A Performance Evaluation of Vertical and Horizontal Data Models in Data Warehousing, in Olivares Ceja, Jesus. Guzman Arenas, Adolfo. (Eds.) Research in Computing

Science Volume 22. Special issue: Data Mining and Information Systems. Instituto Politécnico Nacional, Centro de Investigación en Computación, México. pp. 67-78 ISSN: 1870-4069.

2. Gonzalez-Castro, V., MacKinnon M. L. and Angeles M. P. (2009a) An Alternative Data Warehouse Reference Architectural Configuration, Proceedings of the Twenty sixth British National Conference Of Databases, Birmingham UK, Springer-Verlag Berlin, Heidelberg ISBN: 978-3-642-02842-7.
3. Gonzalez-Castro V., L.M. MacKinnon, and Angeles M.P, (2009b) The Use of the Binary-Relational Model in Industry, A Practical Approach , Proceedings of the Twenty sixth British National Conference Of Databases, Birmingham UK, Springer-Verlag Berlin, Heidelberg ISBN: 978-3-642-02842-7.
4. Akadia website, (2012), http://www.akadia.com/services/ora_materialized_views.html
5. Garbus J.R. , Gupta A., (2006) Administrator's Guide to Sybase ASE 15, ISBN-13: 978-1-55622-360-0, Word ware Publishing, Inc.
6. MonetDB website, (2012), http://monetdb/cwi.nl
7. Stonebreaker M. Abadi D., Batkin A., Chen X., Cherniack M., Ferreira M, Lau E., Lin A, Madden S., O'Neil E., O'Neil P., Rasin A, Tran N. and Zdonik S. (2005) C-Store: A Column Oriented DBMS, Proceedings of the 31st Very Large Database Conference, Trondheim, Norway 553-564.
8. Stonebreaker M., Bear Ch., Cetintemel U., Cherniack M., Ge T., Hachem N., Harizopoulos, S., Lifter J., Rogers J., and Zdonik S. (2007) One Size Fits All? Part 2: Benchmarking Studies. Proceedings of the Third Conference on Innovative Data Systems Research, 173-184.
9. Devlin B. A., Murphy P.T. (1988) An Architecture for a business and information system Proceedings of the IBM Systems Journal, 27,1,60-81.
10. Ailamaki A., DeWitt D. J., Hill M. D. and Skounakis M., (2001) Weaving Relations for Cache Performance, Proceedings of the 27th International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc. San Francisco, CA,USA ISBN:1-55860-804-4
11. Greenwald R., Stackowiak R., Maqsood A. and Bhuller M. (2011) Achieving Extreme Performance with Oracle Exadata, Oracle Press, McGraw-Hill Companies.
12. Ailamaki, A. Ramamurthy R., DeWitt D. and Su Q, (2002) A case for fractured Mirrors, Proceedings of the 28th International Conference of Very Large Databases, Wisconsin-Madison.