University of Melbourne

Advanced Database Systems

COMP90050

In Memory Database Management Systems

Authors:
Hammad
Saman
Ivan Patricio Valarezo

ID: 601099

May 9, 2013

Abstract

Here we need to put the abstract of the work done.

1 Introduction

The main Introduction.

2 Background and Analysis

2.1 Concepts

We need to put concepts, I've found a good source from (Kemper and Neumann (2011))

2.2 Technology awareness

Let's put here something related with the main concerns that leads to this emerging technology. Also, its important to be aware that the ACID properties of DB Systems should be contrasted. In terms of each one of its meanings.

Atomicity In order to achieve atomicity, the IMDB should be able to handle the effects of unsuccessful transactions. In general terms, the problem has to explicitly with the data existent in the volatile memory, since all successful transaction are in a successful state only after been committed to the logging infrastructure.

Durability It is clear that during a failure, the IMDB Systems has the most obvious disadvantage, (without mention the hardware based solutions like batteries and so on), so the effects of a commit must be restored on a failure, this is the principle of Durability of course. One of the ideas to accomplish this is use *redo Logging* .

3 Data Organization

Relational Databases have been the backbone of business applications for more than 20 years, trying to provide companies with a management information system for a set of applications. During all this time, we have dreamed with the possibility of having all the information at our fingertip, we even have sold this idea(Plattner, 2009).

But due to many reasons, we have not been able to offer this. And our systems have been separated in two main different groups: The online transaction processing (OLTP) and on line analytical processing (OLAP).

3.1 OLTP and OLAP

Under the OLTP classification, could be grouped systems with a high rate of mission-critical transactions, most of the database systems (as we know them) are mainly used for transaction processing. Some examples of OLTP systems are financial, sales, Manufacturing, order entry, banking transaction processing, human resources. All this systems performs relatively well mainly because they work on a small portion of the data. The TCP-C benchmark publish an average processing of 100.000 such sales transactions per second on a powerful system (Kemper & Neumann, 2011).

On the other hand, analytical, business intelligence and financial planning application were moved out to separated systems (for more flexibility and better performance). The OLAP Systems could be aimed toward specific task related to the company data warehousing.

Even though, this solutions are different in context, both are still based on *Relational theory* but with different technical approaches. Moreover, recent development in the field of OLAP and the increased availability of main memory (sufficient enough to hold a completed compressed database) have enabled the processing of complex analytical requests in a fraction of a second and thus ease the development of new business processes and applications. The next step seems obviously to undo the separation between OLTP and OLAP and all requests be handled on a combined data set.

Now that the main memory is abundant, we have gone back to see the possibility of having a all-in-one MMDB. SAP[1] is one of the most enthusiastic companies pushing the use of Main Memory Databases for a mixed OLTP & OLAP environment (Plattner, 2009), the company advertises SAP HANA as a generic MMDB solutions.

The MMDB Structure For these, the Main Memory Database Systems improve two critical points: Indexing and Storage

3.2 Indexing

Although, the performance observed in a MMDB could be outstanding compared to a on disk DB, the index structure is a critical bottleneck (?, ?).

This has been copy pasted , reframe: One approach to achieving high performance in a database management system is to store the database in main memorv rather than on disk. -One can then design new data structures aid algorithms oriented towards making eflicient use of CPU cycles and memory space rather than minimizing disk accesses and disk space efficiently (?, ?)

“Disk vs Main Memory” :

Disk oriented: minimize disk access, and miminize space Main Memory oriented: since there is no disk access -¿ first concern is to reduce overall computation time, and use as less memory as possible.

[1]SAP AG: A German multinational software corporation that makes enterprise software to manage business operations and customer relations.

3.3 The OLTP and OLAP Systems

The transactional processing part of a system OLTP (Online Transaction Processing) defines a Database system oriented to the day to day work. Almost all systems (Financial, Manufacturing, Sales, etc) rely on transactions to perform their goals. On the other hand, systems for decision taking, analytical and planning in general has been isolated from OLTPs toward specialized systems called OLAP (Online Analytical Processing).

The main purposes of this separation could be generalized into *performance* and maybe *technical* issues. Even though both systems keep the essence in terms of relational theory, there are some important differences between them (Plattner, 2009).

In the IMDB Systems

3.4 IMDB Data Organization

3.4.1 Column-Store Database System

Is a good idea to do this for IMDB?. I have some articles in (Krishnamurthi & Berthelson, 2011) and in (Abadi, Boncz, & Harizopoulos, 2009).

From (Plattner, 2009), it seems that although more memory has been always useful, the database systems for OLTP where not well adopted in parallel environments, one of the reasons of this were problems like deadlocks and temporary locking in parallel transactions. So in general terms has not been a good idea. Really? review this since Plattner is one of the pro Column Oriented store.

Also from (Plattner, 2009), initial tests regarding in-memory databases using relational type based on *row storage* has not shown notable performance over RDBMSs. The opportunity for *column based storage* now raises due the abundant availability of main memory.

Column-oriented databases are based on vertical partitioning by columns, this is not a new idea, Column-oriented store where considered around 70s. The core of the functioning is based in the fact that each column is stored separately, having the logical scheme built around unique positioning keys (?, ?). This vertical data organization offers particular advantages to reading fewer attributes, because during requests no unnecessary columns must be read, this is normally the case in row-oriented structures on hard drives.

Compression is another characteristic that Column-oriented ease. The redundancy of column store and the homogeneous domain is an advantage and a convenience for compression techniques. All the data within a column belongs to the same type, and the entropy[2] is relatively low.

Some of the main advantages of Column Storage worth to mention:

- Column Storage performs outstanding on modern CPUS (see (Plattner, 2009))
- The vertical compression along columns is more efficient.

[2]The semantic similarities of data

3.5 IMDB Query Processing

3.6 IMDB Data Recovery

4 Conclusions

The conclusions.

We could talk about: *MonetDB* , *VoltDB*

References

Abadi, D. J., Boncz, P. A., & Harizopoulos, S. (2009, August). Column-oriented database systems. *Proc. VLDB Endow.*, *2*(2), 1664–1665. Available from `http://dl.acm.org.ezp.lib.unimelb.edu.au/citation.cfm?id=1687553.1687625`

Kemper, A., & Neumann, T. (2011). Hyper: A hybrid oltp&olap main memory database system based on virtual memory snapshots. *2011 IEEE 27th International Conference on Data Engineering (ICDE)*, 195. Available from `https://ezp.lib.unimelb.edu.au/login?url=https://search-ebscohost-com.ezp.lib.unimelb.edu.au/login.aspx?direct=true\&db=edb\&AN=80280785\&site=eds-live`

Krishnamurthi, M., & Berthelson, M. (2011). Column-oriented database management systems: What do they deliver for analysts?. *European Journal of Management*, *11*(3), 138. Available from `https://ezp.lib.unimelb.edu.au/login?url=https://search-ebscohost-com.ezp.lib.unimelb.edu.au/login.aspx?direct=true\&db=edb\&AN=78131552\&site=eds-live`

Plattner, H. (2009). A common database approach for oltp and olap using an in-memory column database. In *Proceedings of the 2009 acm sigmod international conference on management of data* (pp. 1–2). New York, NY, USA: ACM. Available from `http://doi.acm.org/10.1145/1559845.1559846`