University of Melbourne

Advanced Database Systems

COMP90050

In Memory Database Management Systems

Authors:
Ivan Patricio Valarezo

ID: 601099

April 2, 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* .

2.3 IMDB Data Organization

2.3.1 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.

Also from (Plattner, 2009), initial tests regarding in-memory databases using relational type based on *row storage* has not shown notable performance over RDBMSs.

2.4 IMDB Query Processing

2.5 IMDB Data Recovery

3 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`