



Analysis of Recovery Techniques in Data Base Management System

Sarita Sharma^{1*}, Priyanka Agiwal¹, Rakesh Gaherwal², Shival Mewada³ and Pradeep Sharma¹

¹Department of Computer Science, Govt. Holkar Science College, DAVV, Indore, India

²Idyllic Institute of Management, Rau, Indore, India

³Department of Computer Science, MGCGV, Chitrakoot, India
sharmasarita0703@gmail.com

Available online at: www.isca.in

Received 29th July 2015, revised 22nd January 2016, accepted 10th March 2016

Abstract

As system continue to grow in size and complexity, they pose increasingly greater safety and risk management challenges. The danger of data being corrupted due to reckless application writes is increased. When a DBMS crashes, all or a portion of the data may become unusable. Appropriate procedures must be followed to restore, validate and return the system to normal. Recovery algorithms are techniques to ensure database consistency and transaction atomicity and durability despite failure. In this paper, we have studied when a user face the problems of failure, he may confuse which technique should be used. So we compare the recovery techniques Log Based Scheme and Shadow Paging on the basis of their performance. A review of recovery techniques give the idea of modifying log based recovery technique to be used in multi-user environment more precisely if some problem of writing log each time might be solved. Shadow paging is an alternative approach in place of log based scheme Shadow paging may be considered to use for recovering if its drawbacks of garbage collection and data fragmentation may resolve.

Keywords: DBMS, Transaction, Recovery, Concurrent Control Method, ACID, Checkpoint.

Introduction

The database is a collection related data in an organized manner. This is the best way of storing the data. A distributed system can be visualized as a set of sites, each site consisting of a number of independent transactions¹. Database management systems are used to enable developers to create a database, fill it with information and create ways to query and change that information without having to worry about the technical aspects of data storage and retrieval². When we perform transaction on the database then sometimes we face failure. A transaction is an event which occurs on the database³. It is a set of changes that must all be made together. It is a program unit whose execution may or may not change the contents of a database⁴. Transaction has following states⁵ as in figure-1.

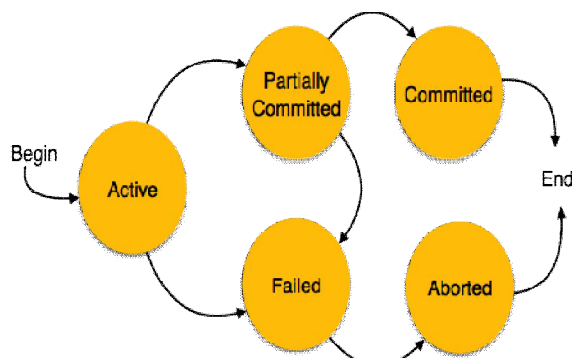


Figure-1
States of a Transaction

Data recovery is the process of salvaging and handling the data through the data from damaged, failed, corrupted or inaccessible secondary storage media when it cannot be accessed normally. Often the data are being salvaged from storage media such as internal or external hard disk drives, solid state drives (SSD), USB flash drive, storage tapes, CDs, DVDs, RAID and other electronics. Recovery may be required due to physical damage to the storage device or logical damage to the file system that prevents it from being mounted by the host operating system (OS)⁶. Recovery is needed to achieve the basic properties (ACID properties) of a transaction.

The storage structure is divided into two categories Volatile and Non-Volatile Storage. Volatile Storage is that which do not survive system crashes. Cache, main memory etc. are some examples of volatile storage. And the Non-Volatile storage which survives system crashes. Hard disks, magnetic tapes, flash drives etc. are some examples of non-volatile storage. On the basis of these storage structures we use recovery techniques to resolve the problems of data failure. In this paper we have discussed the basic recovery techniques Log based and Shadow Paging and try to compare them on the basis of their nature.

The rest of this paper is organized as follows: In section II, We have explained what Recovery is? In section III, we have defined problem of statement through transaction failure. In section IV, we have explained the related work. In section V, we have described the performance of recovery techniques. In section VI suggestion and recommendation are given and finally

in section VII we have concluded the recovery techniques through conclusion and future study.

Recovery

Data recovery is the process of restoring data that has been vanished, unintentionally deleted, corrupted or made inaccessible for any reason. In enterprise information technology (IT), data recovery consistently indicates the restoration of data to a desktop, laptop, server, or external storage system from a backup.

Transaction Failure

Transaction failure is the state of a transaction when it fails to execute or reaches a point from where it cannot go any further. There are several reasons for a transaction failure as follows: i. Logical errors in which transaction cannot complete due to some code error or any internal error condition. ii. System error in which transaction cannot complete due to some system error like deadlock or resource dearth. iii. System crash in which transaction cannot complete due to the external problems like intervention in power supply. iv. Disk failure which occurs due to formation of bad sectors, impassable to the disk, disk head crash or any other failure.

Related Work

Mr. Mourad Benchikh has described that “Recovery algorithms are techniques to ensure transaction atomicity and durability despite failures”. The recovery subsystem, using recovery algorithm, ensures atomicity by undoing the actions of transactions that do not commit and durability by making sure that all actions of committed transactions survive even if failures occur⁷.

Two main approaches in recovery process Log-based recovery using WAL protocol and Shadow-paging. Joost Verhofstad has presented the “seven techniques to recover the data in the database”.

Salvation program: Run after a crash to attempt to restore the system to a valid state. No recovery data used. Used when all other techniques fail or were not used. Good for cases where buffers were lost in a crash and one wants to reconstruct what was lost.

Incremental dumping: Modified files copied to archive after job completed or at intervals.

Audit trail: Sequences of actions on files are recorded. Optimal for "backing out" of transactions. (Ideal if trail is written out before changes).

Differential files: Separate file is maintained to keep track of changes, periodically merged with the main file.

Backup/current version: Present files from the current version of the database. Files containing previous values form a consistent backup version.

Multiple copies: Multiple active copies of each file are maintained during normal operation of the database. In cases of failure, comparison between the versions can be used to find a consistent version.

Careful replacement: Nothing is updated in place, with the original only being after operation is complete⁸.

Performance of Recovery Techniques

Log Based Recovery: Logs are the sequel of records which are used to manage records of the actions during a transaction. Logs are written before the actual alteration and stored on a stable storage media. Log Based Recovery technique works in three different manners: i. Deferred Update, ii. Immediate Update, iii. Checkpoint

Deferred Update Method: In this method a database is not physically updated on disk until after a transaction reaches its commit point. After it, the updates are stored persistently in the log and then written to the database. Before the commit point the transaction updates are managed in the local transaction workspace like buffers. If transaction fails before reaching commit point, it will not have changed the database; hence there is no need to UNDO. So it is necessary to REDO the effect of the operations of a committed transaction from the log, because then effect may not yet have been recorded. Hence it is also termed as “NO UNDO/REDO” Algorithm. Let there be a transaction T1 as follows: read(X), write (10,Y), write(20,Z), Commit.

Table-1
Using Deferred Update

Time	Action	Log
t1	START	-
t2	read(X)	-
t3	write(10,Y)	Y = 10
t4	write(20,Z)	Z= 20
t5	COMMIT	COMMIT

Table-2
After Update

DISK	Before			After		
	-	-	Y=6	-	-	Y=10
X=5	Z=2	-	X=5	Z=20	-	

Immediate Update Method: In this method database may be updated by some operations of a transaction before the transaction compasses its commit point. These operations are consistently recorded in the log on disk effectively writing before adapted. If a transaction aborts after keeping record some changes to the database, but before commit point, the effect of its operations on the database must be undone. This method needs both undo and redo in recovery. Hence it is also termed as “UNDO/REDO” Algorithm to the database. Transaction T1 in the above example works as below:

Table-3
Using Immediate Update

Time	Action	LOG
t1	START	-
t2	read(X)	-
t3	write(10,Y)	Was Y == 6, now 10
t4	write(20,Z)	Was Z == 2, now 20
t5	COMMIT	COMMIT

Table-4
After Updates

DISK	Before			During			After		
	-	-	Y=6	-	-	Y=10	-	-	Y=10
	X=5	Z=2	-	X=5	Z=2	-	X=5	Z=20	-

Checkpoint: When more than one transaction is being executed in parallel, the logs are interleaved. At the time of recovery, it would become hard for the recovery system to backtrack all logs, and then start recovering. During execution system manages the log, using one of the two techniques Deferred Update or Immediate Update. In addition, System repeatedly performs checkpoints. Checkpoint can be occur automatically and manually constantly whenever the database is shut down or when a redo log switch occurs. We use the following SQL command vigorously to use checkpoint: i. alter system switch log-file, ii. alter system checkpoint.

When a system with concurrent transactions crashes and recovers, it behaves in the following manner –

Shadow Paging: Shadow paging is a method used to attain atomic and durable transactions and provides the competence to manipulate pages in a database. In this method the database is partitioned into fixed-length blocks known as pages. For each database page, a page table with n entries is maintained. Each page contains a pointer to a page on disk. To recover from failure there are two page tables “current page table” and “shadow page table”. At the starting of a transaction both tables

are identical. During the transaction, shadow page table is never altered. When there is write operation during a transaction, the current page table may be changed. Current page table is used by all input and output operations to locate database pages on disk. Hence this method does not need UNDO/REDO algorithm. It recovers the data faster. But in this method data gets fragmented. If the transaction completes then all pages which contains old version of altered data need to be garbage collected. Because of these reasons this method is hard to extend to allow transaction to run simultaneously.

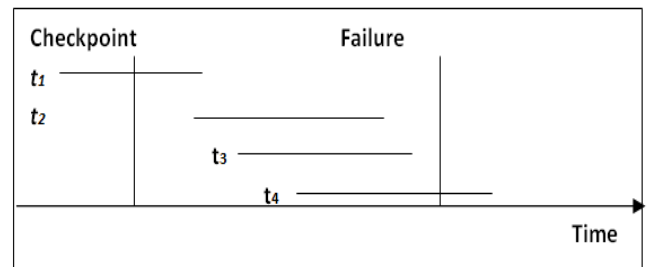


Figure-2
Nature of System in case of Concurrent Transaction

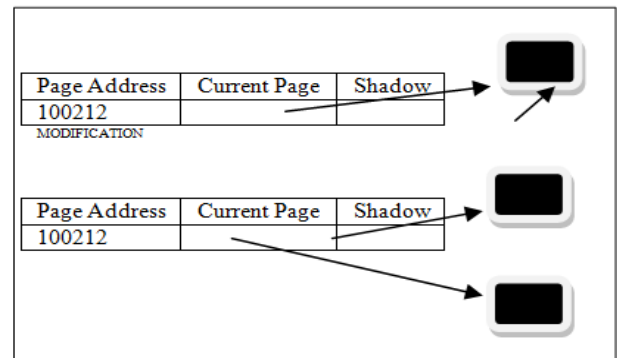


Figure-2
Recovery through shadow paging

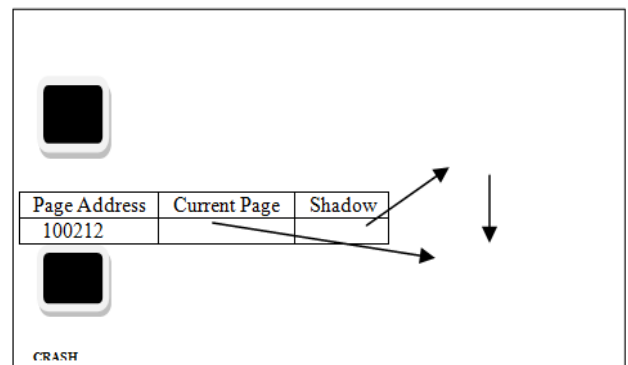


Figure-3
Recovery through shadow paging after crash

Table-5
Comparison of Recovery Techniques

Basis of Comparison	Log Based Recovery	Shadow Paging
Overhead of writing Logs	Require	No Require
Recovery	Not Trivial	Trivial
Data Fragmentation	No fragmentation	Get fragmentation
Need of Garbage Collection	No	Yes
Extension of Algorithm	Easier for concurrent transaction	Hard for concurrent transaction
Time in case of System failure	Increases to recover	Decreases to recover
Use of operations	Undo(T), Redo(T)	NoUndo(T), NoRedo(T)

Table-6
Statistical Representation of Recovery Technique

Factors	Log Based Recovery Technique(T1)	Shadow Paging(T2)
Use of operations	2	0
Overhead of writing Logs	3	1
Commit Overhead	1	5
Time	10	5
Garbage Collection	1	10
Data Fragmentation	2	10
Extending Algorithm	2	8

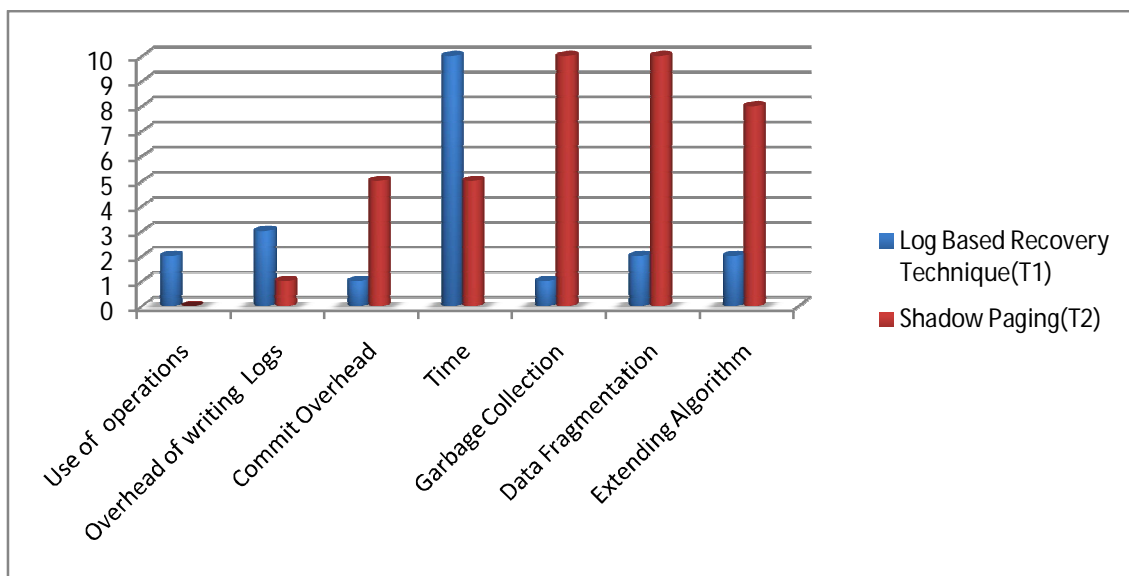


Figure-4
Graphical Representation of Recovery Techniques

Suggestion and Recommendation

From the figure-5 it is clear that Log Based Recovery Technique use two operation (UNDO and REDO) but Shadow paging does not use them. In each transaction Log Based Technique uses Logs but Shadow Paging does not use which increases its performance. But Shadow Paging needs to be garbage collection of each new modified data in old table and also data gets fragmented which are the drawbacks of Shadow Paging. Shadow paging algorithm is also hard to extend to allow transaction to run concurrently but here, Log Based scheme is easier to extend. Hence we recommend that Log Based Technique should be extending to recover fast.

Conclusion

We have analyzed that in some cases Shadow Paging can be used to recover easily but it is restricted to use only in case of single user environment. In case of multi-user environment Shadow paging uses Logs for concurrency Control method. Hence we can say that Log Based Recovery should be used to recover data in the database. It also needs to be modified to perform trivial recovery as in Shadow Paging to decrease the time of execution of the process of Log Based Recovery Technique. But there arises condition of discomfiting that which technique should be used to recover in the transaction failure. So in future our aim is to develop a new technique that solves this problem efficiently.

References

1. Akanshika (2013). Analysis of Rollback Recovery Techniques in Distributed Database Management System”, *International Journal of Modern Engineering Research (IJMER)*, 3(3), 1353-1356 ISSN: 22496645.
2. Morgan Davis (2014). What Is a Database Management System. <http://www.hostway.com/blog/database-management-system.25.06.2015>.
3. Dinesh Thakur (2015). What is Transaction in DBMS. <http://ecomputernotes.com/database-system/rdbms/transaction.29/06/2015>.
4. Gursharan Singh Tatla (2015). Transaction Management. <https://www.coursehero.com/file/10173083/transaction-management.30/06/2015>.
5. Ashish Srivastava, Udai Shankar and Sanjay Kumar Tiwari (2012). Transaction Management in Homogenous Distributed Real-Time Replicated Database Systems. *International Journal of Advanced Research in Computer Science and Software Engineering*. 2(6), ISSN: 2277 128X.
6. WIKIPEDIA (2015) Data Recovery. https://en.wikipedia.org/wiki/Data_recovery.04/07/2015.
7. Ramez Elmasri, Shamkant B. Nava the (2015). Fundamentals of Database System (5th Edition). ISBN-13:978-0321369574, ISBN-10: 0321369572.
8. Hector Garcia-Molina et. Al. (2008). Database Systems: The Complete Book (2nd Edition), ISBN-13:978-0131873254, ISBN-10: 0131873253.
9. Philip Bohannon and at all (2003). Detection and Recovery Techniques for Database Corruption. *IEEE: Transactions on Knowledge and Data Engineering*, 15(5).
10. Mohan C., Haderle D., Lindsay B., Pirahesh H. and Schwarz P. (1992). ARIES: A Transaction Recovery Method Supporting Fine-Granularity Locking and Partial Rollbacks Using Write-Ahead Logging. *ACM Trans. Database Systems*, 17(1) 94-162.