Aries: A Transaction Recovery Method

Presented by Haoran Song
Discussion by Hoyt

Outline

- Purpose of this paper
- Transaction Review
- Terminology of this paper
- ARIES
- Discussion

Purpose of this paper

- Computer system is crashed as easily as other devices.
  - Disk burned
  - Software Errors
  - Fires or Earthquake
  - Terrorisms
  - Lots of other reasons...

Purpose of this paper

- If data are lost caused by these problems, we need find a recovery scheme to recover the lost data.
- This recovery scheme must
  - Help us retrieve the data
  - Maintain the atomicity and durability
  - High availability: minimize the recovery time

  So ARIES is introduced.

Outline

- Purpose of this paper
- Transaction Review
- Terminology of this paper
- ARIES
- Discussion

Transaction Review

- ***Atomicity:*** Either all actions in the transaction occur, or none occur.
- ***Consistency:*** The isolated execution of a transaction preserve the consistency of database.
- ***Isolation:*** The execution of one transaction is isolated from that of other transactions.
- ***Durability:*** If a transaction commits, then its effects persist, even if there are system failures.

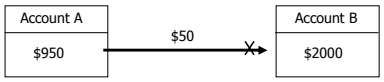
Transaction Review

- Example: Let T be a transaction that transfers $50 from account A to account B.

$1000 - $50 = $950

Transaction Review

- What if this $50 transfer failed? You lose $50!



Therefore we need a recovery scheme to maintain ACID. We do not want to lose our money!

Outline

- Purpose of this paper
- Transaction Review
- Terminology of this paper
- ARIES
- Discussion

Terminology of this paper

- Attention: This paper is not about Transaction and Concurrency. We only care about Recovery.
- Some useful terms
  - Volatile storage: information does not usually survive system crashes. E.g. memory.
  - Nonvolatile storage: information survives system crashes. E.g. disk, magnetic tapes.
  - Stable storage: information is theoretically never lost.

Outline

- Purpose of this paper
- Transaction Review
- Terminology of this paper
- ARIES
- Discussion

ARIES

- *Stands for Algorithm for recovery and Isolation Exploiting Semantics.* A long and confused name!
- Just remember 1 thing
  - It was the best example in recovery technology.

ARIES

- Key points to go over today:
  - main goals of the system (p 105 - 108)
  - general algorithm
  - role of the buffer pool
  - WAL logging
  - notions of fuzzy check points
  - an Recovery Example

ARIES: main goals of the system

1. Simplicity
2. Operation Logging
3. Flexible storage management
4. Partial rollbacks
5. Flexible buffer management
6. Recovery independence
7. Logical undo
8. Parallelism and fast recovery
9. Minimal overhead

Discussion on Goals

As the authors state, some of ARIES' goals are not mutually compatible, thus requiring some balance between them. Discuss how important each of the following goals are in modern DBMSs and the issues underlying them.
- Group 1: Simplicity, Parallelism and fast recovery & Minimal overhead.
- Group 2: Operation Logging, Flexible storage management & Flexible buffer management.
- Group 3: Partial rollbacks, Recovery independence & Logical undo.

ARIES

- Key points to go over today:
  - main goals of the system (p 105 - 108)
  - general algorithm
  - role of the buffer pool
  - WAL logging
  - notions of fuzzy check points
  - an Recovery Example

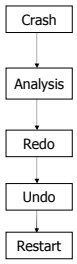
ARIES: General Algorithm

- 3 Phases + 3 Principles

ARIES: General Algorithm

- **3 Phases**

1. Analysis: Identify dirty pages in the buffer pool and active transaction at the time of crash
2. Redo: Repeats all actions and restore the database state to the crash point
3. Undo: Undoes uncommitted actions



ARIES: General Algorithm

- **3 Principles**

1. Write-Ahead Logging: The record in the log must be written to stable storage before the change is written to disk.
2. Repeating History During Redo: Retraces all actions before the crash and brings the system back to the state at the crash point. Then undoes uncommitted transactions
3. Logging Changes During Undo: Changes made to the undoing a transaction are logged to ensure such an action is not repeated in the repeated restarts.

ARIES

- Key points to go over today:
  - main goals of the system (p 105 - 108)
  - general algorithm
  - role of the buffer pool
  - WAL logging
  - notions of fuzzy check points
  - an Recovery Example

ARIES: Role of the buffer pool

- Problems: output every log record to stable storage at the time it is created is high overhead.
- Purpose: impose a minimal amount of overhead on interactions with database.
- Solution: write log records to a log buffer in main memory, and output to stable storage in a single output operation.

ARIES

- Key points to go over today:
  - main goals of the system (p 105 - 108)
  - general algorithm
  - role of the buffer pool
  - WAL logging
  - notions of fuzzy check points
  - an Recovery Example

ARIES: WAL logging

Meaning: write ahead logging. It is a protocol.

Functionality: records the progress of a transaction in a log.

Description: Asserts the log records must already be on stable storage before the previous version of data is replaced by new one.

Comparison: WAL VS Shadow page technique

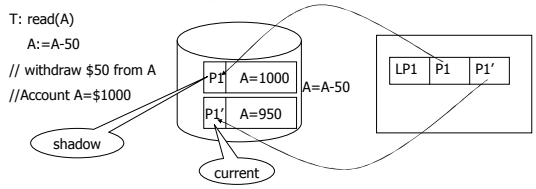
ARIES: WAL logging

Before We start WAL, Let us firstly review 2 things

1. Shadow Page
2. The Log.

ARIES: WAL logging

- Quick Review on Shadow page
- Consider Following Transaction T



On a failure, use shadow version to do recovery

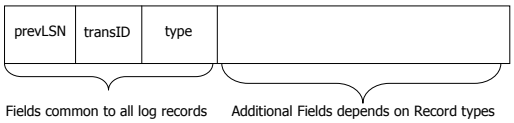
ARIES: WAL logging

Quick Review on Log

- It is a history of actions executed by the DBMS
- It is a file of records stored in stable storage
- Every log record is given a unique ID called LSN
- There are different types of Log records: such as *Update Log Record, Commit type Log Record, Abort type Log Record, End type log Record, CLR*

ARIES: WAL logging

Quick Review on Log (continue…)



Fields common to all log records Additional Fields depends on Record types

ARIES: WAL logging

Quick Review on Log (continue…)

- ***Update :***
  - Be appended to the log tail when modifying a page.
  - pageLSN of that changed page is set to the LSN of the record.
- ***Commit :***
  - When a transaction commits, a commit-type record is force-written to stable storage.
- ***Abort :***
  - An about record is appended to the log tail when transaction is aborted
- ***End :***
  - End-type log is appended to log tail when transaction has completed all work (after commit or abort)
- ***Compensation Log Records (CLR) :***
  - When a transaction's updates are undone, CLR is written
  - It happens during aborting or a recovery from crash.
  - Contains **undoNextLSN** field: LSN of next log record to be undone.

ARIES: WAL logging

Other Recovery-Related Structures

- Transaction Table: records the entry for each active transaction.
- Dirty Page Table: records the pages with changes in the buffer pool but not yet reflected on disk.

pageID	recLSN
p500	
p600	

Points to LSN of the first log record that caused the page dirty

Dirty Page Table

transID	lastLSN
T1000	
T2000	

Points to LSN of the most recent log record for this transaction

ARIES: fuzzy check points

- Key points to go over today:
  - main goals of the system (p 105 - 108)
  - general algorithm
  - role of the buffer pool
  - WAL logging
  - fuzzy check points
  - an Recovery Example

ARIES: Fuzzy Point

- Periodically ***checkpoint***, to minimize recovery time in system crash. Write to log:
  - ***begin_checkpoint*** record: when checkpoint began
  - ***end_checkpoint*** record: current *transaction table* and *dirty page table*.
- Aries uses a **fuzzy checkpoint**: Transactions continue to run; so these tables are accurate only as of time of begin_checkpoint; Store LSN of checkpoint record in a safe place (***master*** record).
- How to use it?

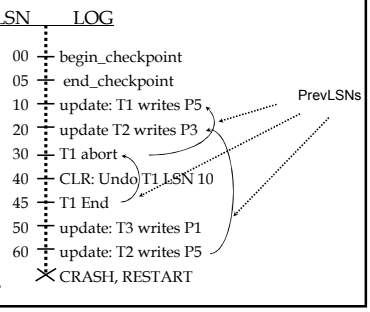
  When restart from crash, analysis initializes the dirty page table and transaction table by examining the most recent begin_checkpoint.

ARIES: an Recovery Example

- Key points to go over today:
  - main goals of the system (p 105 - 108)
  - general algorithm
  - role of the buffer pool
  - WAL logging
  - fuzzy check points
  - an Recovery Example

ARIES: an Recovery Example

After CRASH?



Slides from Ramakrishnan & Gehrke

ARIES: an Recovery Example

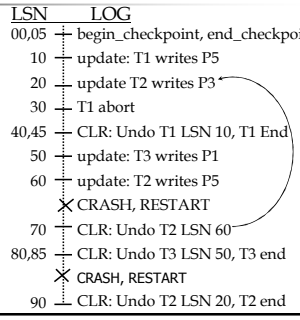
1. Analysis identifies P1, P3, P5 as dirty pages. 45 shows T1 is completed, so only T2 and T3 were active before crash.

2. Redo begins with 10 reapplies all actions.

3. Undo set consists of LSN 60 and 50. The update is undone and 2 CLRs(70,80) are written to the log.

4. When Crash again, Analysis only finds T2 is active, then Redo from 10 to 85. Undo only considers T2 and processes undoNextLSN=20.

5.CLR(LSN=90) is written.



- **Thanks!**
- **Discussion**

Discussion on Checkpoints

- It seems one could design a system that doesn't require the use of checkpoints, handling the same functionality with a queue-type approach. Is such a system possible, given ARIES' goals? What would be the complications? What would be the advantages?