ARIES: A Transaction Recovery Method Supporting Fine Granularity Locking and Partial Rollbacks Using Write-Ahead Logging

--- C. MOHAN, DON HADERLE, BRUCE LINDSAY

Summary

The main ideas addressed in this paper are

- Efficient recovery algorithm: Aries, which supports partial rollbacks of transactions, fine-granularity locking and recovery using write-ahead logging.
- ARIES uses a log sequence number in each page to correlate the state of a page with respect to logged updates of that page. All updates of a transaction are logged, including those performed during rollbacks.
- ARIES is applicable not only to database management systems but also to persistent object-oriented languages, recoverable file systems and transaction-based operating systems
- Basic concepts of recovery, concurrency control and Buffer Management and the difficulties involved in designing a recovery method.
- Overview of ARIES and its functionalities.

It is Important to make undo & redo operations during roll back such that the transaction is completely recovered. ARIES works well with all problems like recovery, concurrency, buffer management etc. by assigning a log sequence number to every log. ARIES uses Write Ahead logging protocol which is considered to be better than shadow page technique.

The Main Ideas discussed are

- The concepts of logging, failures and recovery methods and how ARIES handles them. The concepts discussed are:
    - **Forward Processing**: updates performed when system is in normal processing mode
    - **Partial Rollback**: ability to set up save points during transaction, and request roll back of changes performed by transaction to save point
    - **Total Rollback**: all updates of transaction are undone
    - **Normal Undo**: total or partial rollback when system is in normal operation
    - **Restart Undo**: transaction rollback during restart recovery after system failure
    - **CLRs**: Compensation log records are logs of updates performed during rollback in ARIES
    - Page-oriented redo occurs if log record whose update is being redone describes the page of the database that was modified. Logical redo is higher-level, performing a redo may require accessing several pages.

- Latches and how they are implemented, Locks, Granularity and types of locks and modes, and conditional & Unconditional lock requests.
- Buffer Management policies for different systems and the problems in Buffer Management.
- The main goals aimed to compare metrics of recovery methods are simplicity, operation Logging, flexible storage management, partial roll backs, flexible buffer management, recovery independence, logical undo, parallelism & fast recovery, and minimal overhead.
- Overview of ARIES and how it satisfies the goals reasonably. Data structures used by ARIES are:
    - **Log Records**: Every log has a unique LSN. Fields in the log include PrevLSN, TransID, type, PageID, UndoNxtLSN and other data fields. Log is written when any actions of updating a page, commit, abort, end and undoing an update happens.
    - **Page Structure**: Every page has a page_LSN field, describing the latest update to the page.
    - **Transaction Table**: Entry for each active transaction. Each entry contains TransID, Status and LastLSN and UndoNxtLSN.
    - **Dirty_Pages Table**: Entry for each dirty page in the buffer pool. Each entry contains pageID and recLSN.
- Actions performed during Normal Transaction processing like Updates, Total or Partial rollbacks, Transaction Termination, and checkpoints
- When the transaction system restarts after a failure, recovery needs to be performed to bring the data to a consistent state and ensure the atomicity and durability properties of transactions. The steps in Restart processing are:
    - **Analysis Pass**: It begins at the most recent checkpoint and then scan forward until reaching the end of logs. If a log record for a transaction T is encountered, and entry for T is added to the table; if the end log record for T is encountered, T is removed; if a log record is encountered for a page P whose identity does already appear in the dirty-pages table, then an entry is made in the table with the current log record's LSN as the page's RecLSN.
    - **Redo Pass**: Redo begins with the log record that has he smallest recLSN of all pages in the dirty page table, and scans forwards until the end of the log. The action must be redone unless the affected page is not in the dirty page table, or The affected page is in the dirty page table, but the recLSN for the entry is greater than the LSN of the log record being checked, or
      The pageLSN is greater than or equal to the LSN of the log record being checked. If the logged action must be done, then the logged action is reapplied and the pageLSN on the page is set to the LSN of the redone log record.
    - **Undo Pass**: Undo scans backward from the end of the log and undo all the loser transactions in the reverse order in which they appear in the log that were active at crash. This is done by continually taking the maximum of the LSNs of the next log record to be processed for each of the yet-to-be-completely-undone loser transaction, until no loser transaction remains to be done.
- Impact of failures on CPU processing and I/O can be reduced by taking checkpoints at end of Analysis pass, beginning of Redo pass and beginning of Undo pass.

- Using the concept of Nested Top Action whenever some updates of transaction needs to be committed irrespective of whether the transaction commits or not.
- Recovery Paradigms like Selective Redo, Rollback State, Space Management, and Multiple LSNs.
- The properties of other Recovery methods which use Write Ahead Logging are Buffer Management, Normal Check pointing, Partial roll backs, Compensation log records, Log record contents, Page overload, Log passes during restart recovery, Page forces during restart, Restart checkpoints, and Restrictions on data.
- Interesting and Useful Properties of ARIES

The final results are

- Overview of ARIES recovery method and Recovery paradigms of System-R are inappropriate in Write Ahead Logging.
- Variety of features that are important in building and operating transaction processing system.
- Several issues regarding operation logging, fine granularity locking, space management and flexible recovery are discussed.
- ARIES accomplishes the goals that we set out with by logging all updates on a per-page basis, using an LSN on every page for tracking page state, repeating history during restart recovery before undoing the loser transactions, and chaining the CLRS to the predecessors of the log records that they compensated.

Question: Is ARIES successful in handling all recovery problems in RDBMS? Are there any other recovery options available now?

Submitted By

Lakshman Madhav Kollipara

SID- 932655504

Submitted To

Arash Termehchy

CS540- Database Management Systems