SQL Server: Myths and Misconceptions

Module 3: Transaction Log

Paul S. Randal

Paul@SQLskills.com

pluralsight
hardcore developer training

Introduction

- **Logging and the transaction log are two of the most misunderstood parts of relational database management systems**
  - How logging works and which operations are logged differently
  - How transaction log backups work and what affects them

- **In this module:**
  - Twelve myths around logging and the transaction log

Transaction Log Myth #1

UNTRUE!!

- **Myth: When the transaction log is cleared, it gets zeroed out and/or changes size**

- **Log clearing and log truncation are both misnomers**
  - Simply mean that zero or more portions of the log are marked as reusable
  - Log records are never zeroed out
  - The log never changes size except from an explicit shrink operation
- **In the FULL and BULK_LOGGED recovery models, only a log backup can clear the log**
- **In the SIMPLE recovery model, only a checkpoint can clear the log**
- **There are no exceptions**

Transaction Log Myth #2

UNTRUE!!

- **Myth: It's not possible to analyze the transaction log using SQL Server**

- **DBCC LOG and fn_dblog both exist to dump the log contents**
  - Use the function as it's easier to use than the DBCC command
  - They both give the same results, but the DBCC command gives text output
- **Also fn_dump_dblog to allow dumping log contents from a backup**
- **See http://bit.ly/KhL1Hv for some examples**
- **SQL Server does not contain a full-blown log analysis tool**
  - 3rd-party tools exist to fill that need

Transaction Log Myth #3

UNTRUE!!

- **Myth: TRUNCATE TABLE is a non-logged operation**

- **There are no non-logged user database operations**
- **The only non-logged operations are the version stores and workfiles in tempdb**
- **TRUNCATE TABLE is efficiently, but fully logged**
  - Metadata changes are always logged at TRUNCATE time
  - Extent/page deallocations may be logged at TRUNCATE time or later by the deferred-drop background task
  - Not a true minimally-logged operation

Transaction Log Myth #4

UNTRUE!!

- **Myth: “Fully logged” means you’ll always see one log record for each part of an operation**

- **Some operations can be efficiently logged without one log record per table row**
- **Consider a rebuild of a 100-thousand row index**
  - You would expect to see 100 thousand LOP_INSERT_ROW log records
  - However, it will log LOP_FORMAT_PAGE log records instead with full page images with the net effect of all the inserts on
  - This is still fully logged!
- **“Fully logged” simply means the transaction log contains enough information to reconstitute the transaction after a crash or restore**

Transaction Log Myth #5

UNTRUE!!

- Myth: Multiple transaction log files will help performance

- SQL Server will always use log files sequentially
- You may see them all having I/Os, but that's just updating the file header pages
- The only time another log file is needed is if the first one fills up and cannot grow, you cannot take a log backup, and you do not want to break the log backup chain
- Remove additional log files once you don't need them

Transaction Log Myth #6

UNTRUE!!

- Myth: The transaction log can use instant file initialization

- The log must always be zero initialized when first created or grown
- This allows crash recovery to work when the log has not yet wrapped around
  - Crash recovery knows where to start but not where to stop
  - Crash recovery keeps reading until it finds a log block with zeroes in

- Beware restoring a database with a giant log file
  - The log file must be created and zero initialized before restore can continue

Transaction Log Myth #7

UNTRUE!!

- **Myth: The transaction log should always be as small as possible**

- **The log needs to be as big as it needs to be**
- **Do not regularly shrink the log**
  - It'll just have to grow again, and can't use instant file initialization
- **How big should the log be?**
  - Single largest transaction
  - Database mirroring SEND queue
  - Index rebuild of largest index
  - How long is the longest data backup?
  - Transactional replication
- **Beware of anyone that gives an arithmetic formula for log size…**

Transaction Log Myth #8

UNTRUE!!

- **Myth: Log records can be removed from the transaction log**

- **Log records are never removed from the log**
  - There are only ever overwritten when portions of the log are reused
- **Rollback of a transaction does not remove log records**
  - Any log records generated by a transaction being rolled back must be compensated for by ‘anti-operations’ – which are logged
  - So rollback actually causes *more* log records to be generated

Transaction Log Myth #9

UNTRUE!!

- Myth: Log records can move in the transaction log

- Just like log records cannot be removed from the log, they also cannot be moved
- A log record has a fixed Log Sequence Number (LSN) which cannot change
  - LSN is <VLF Sequence#><:Log Block #>:<Log Record #>
- Moving to a different VLF would invalidate the LSN, breaking all kinds of things

Transaction Log Myth #10

UNTRUE!!

- **Myth: The BULK_LOGGED recovery model reduces the size of log backups**

- **Using BULK_LOGGED doesn't change log backup size**
- **BULK_LOGGED only reduces the amount of transaction log generated**
- **After a minimally-logged operation in the BULK_LOGGED recovery model, the next log backup must also back up all data extents changed by the operation**
  - Otherwise the restore would result in a corrupt database with a bunch of empty pages
  - Minimally-logged operations cause the buffer pool to 'eager write' the pages being changed so all data pages changed by the operation are flushed to disk before the operation completes

Transaction Log Myth #11

UNTRUE!!

- **Myth: Log backups will be the same size as the log itself**

- **Log backups only back up the log generated since last log backup**
- **Even if the log is growing a huge amount, the log backup still won't be the same size**
  - The log manager reserves empty space in the log so that all active transactions can roll back without the log having to grow
  - This 'log space reservation' can account for up to (roughly) 50% of the size of the log at any time

Transaction Log Myth #12

UNTRUE!!

- **Myth: Using the BULK_LOGGED recovery model has no effect on disaster recovery**

- **Using BULK_LOGGED can be very detrimental to disaster recovery**
- **It can have a disastrous effect if a crash occurs after a minimally-logged operation but before the next log backup and the data files are damaged**
  - The tail-of-the-log backup will not be possible
- **Do not use BULK_LOGGED if you have concurrent user transactions that you cannot regenerate**