SQL Server: Advanced Corruption Recovery Techniques

Module 4: Dealing with Transaction Log Problems

Paul S. Randal
http://www.SQLskills.com/blogs/paul/
Paul@SQLskills.com

pluralsight
hardcore developer training

Introduction

- **The transaction log can be damaged just like a data file**
- **The path to recovering from a damaged log depends on the state of the database and whether it is attached or not**

- **In this module we'll cover:**
  - Damaged or missing transaction log
  - EMERGENCY mode
  - EMERGENCY-mode repair
  - Reattaching a detached SUSPECT database

Database States (1)

- **More information in Books Online at http://bit.ly/1ddqven**
- **ONLINE**
  - Not a problem
- **OFFLINE**
  - The database is offline and unavailable
  - Use ALTER DATABASE dbname SET ONLINE to bring it online
- **RESTORING**
  - The database is unavailable because a restore is in progress
  - Complete the restore sequence and then use RESTORE DATABASE dbname WITH RECOVERY to bring it online
- **RECOVERING**
  - The database is unavailable because recovery has not completed
  - Allow the database to complete recovering and the state will change to ONLINE if successful or SUSPECT if unsuccessful

Database States (2)

- **EMERGENCY**
  - The database has been set to EMERGENCY mode to allow data export or EMERGENCY-mode repair, which we'll discuss later
- **RECOVERY PENDING**
  - Recovery could not start for some reason, possibly missing or damaged transaction log
- **SUSPECT**
  - Recovery started but could not complete for some reason, possibly damaged transaction log or corrupt data files
- **For either SUSPECT or RECOVERY_PENDING:**
  - You could try setting the database OFFLINE and then ONLINE again but it likely will not work
  - You will likely need to restore from backups or use EMERGENCY mode

Damaged/Missing Log With Clean Shutdown

- **The transaction log is essential for crash recovery to work**
- **However, a damaged or missing transaction log does not matter if the database had a clean shutdown**
  - The transaction log is not required as there is no crash recovery to run
- **In that case, a new transaction log can be created**
  - Requires re-attaching the database using CREATE DATABASE and using either the FOR ATTACH or FOR ATTACH_REBUILD_LOG options
  - FOR ATTACH creates a single 0.5 MB transaction log file, but only works if there was previously one transaction log file
  - FOR ATTACH_REBUILD_LOG also creates a single 0.5 MB transaction log file, but works no matter how many log files there were previously
- **Make sure to provision the correct size and auto-growth afterwards**
  - See *SQL Server: Understanding Logging, Recovery, and the Transaction Log*

Damaged/Missing Log Without Clean Shutdown

- **If the database did not have a clean shutdown, crash recovery must be performed before the database can be brought online**
- **If the transaction log is damaged or missing, the database will be in one of two states:**
  - RECOVERY_PENDING: crash recovery has to run but could not start
  - SUSPECT: crash recovery started but could not complete
- **Recovering from this situation requires restoring from backups, or using EMERGENCY mode to access the database**

EMERGENCY Mode

- **EMERGENCY mode is also known as "bypass recovery" mode, as this is exactly what it instructs the Storage Engine to do**
  - ALTER DATABASE mydb SET EMERGENCY
- **Although this allows access to the database, the data is transactionally inconsistent**
  - Furthermore, the database itself may be structurally inconsistent
- **You can try to export data out into a new database**
- **Without backups, the only way to recover the existing database is to try EMERGENCY-mode repair**

EMERGENCY-Mode Repair

- **This was added to SQL Server 2005 as a documented and supported way to attempt recovery with a damaged transaction log**
  - It used to be possible using DBCC REBUILD_LOG in SQL Server 2000
- **The database must be set to EMERGENCY and SINGLE_USER**
- **Running DBCC CHECKDB (mydb, REPAIR_ALLOW_DATA_LOSS) does the following in EMERGENCY mode:**
  - Run as much crash recovery as possible, skipping damaged log records
  - Build a new transaction log file
  - Run a full DBCC CHECKDB with REPAIR_ALLOW_DATA_LOSS
  - Try to bring the database online
- **This is not guaranteed to work in all scenarios**
  - Severe damage to data files may prevent the database being brought online
  - File system damage may prevent the log file being recreated

Detached SUSPECT Database

- **When attaching a database, crash recovery must be completed before the database will attach**
- **If a detached database requires crash recovery and the transaction log is damaged or missing, the attach will fail**
- **Prior to SQL Server 2008 it was possible to detach a SUSPECT database**
- **Reattaching a database in this state requires the following:**
  - Create a database with the same name and file IDs as the detached database
  - Set the dummy database offline and delete the log and data files
  - Copy in the files from the detached database
  - Try to bring the database online again
- **Once the database is attached, you can proceed to EMERGENCY mode**

Damaged Log of Attached Database

- **If the Storage Engine encounters transaction log corruption, for instance while rolling back a transaction, the database will be set offline with a status of SUSPECT**
  - Recover from this using backups or EMERGENCY mode
- **If there are no active transactions that could encounter the corruption, DBCC CHECKDB or a transaction log backup may fail**
- **Usually possible to recover from this:**
  - Switch the database to the SIMPLE recovery model
  - Issue a CHECKPOINT to force the transaction log to clear
  - Switch back to the original recovery model
- **This should hopefully clear the damaged log records**
  - Note: this breaks the log backup chain

Summary

- **Recovering from transaction log corruption usually requires restoring from backups or using EMERGENCY mode**
- **Detached databases with transaction log problems can be attached using the dummy database technique**

- **In the next module, we'll discuss:**
  - Tail-of-the-log backups when the original server is not available
  - Page and partial restores
  - Examining log backup contents
  - System databases