SQL Server: Detecting and Correcting Database Corruption

Module 3: Detecting Page Corruption

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

Introduction

- **SQL Server has mechanisms to automatically detect and alert you when it finds corruption during I/O operations**

- **In this module we'll cover:**
  - Torn-page detection
  - Page checksums
  - I/O errors
  - Monitoring for I/O errors

Page Protection Options

- **SQL Server allows pages to be “protected” on disk**
- **This allows fast detection of corruption when a page is read into memory**
- **Set using**
  - ALTER DATABASE SET PAGE_VERIFY <option>
- **Three options:**
  - NONE (don’t do this…)
  - TORN_PAGE_DETECTION
  - CHECKSUM
- **All page protection operations are performed by the *buffer pool* (also known as the buffer manager or buffer cache)**

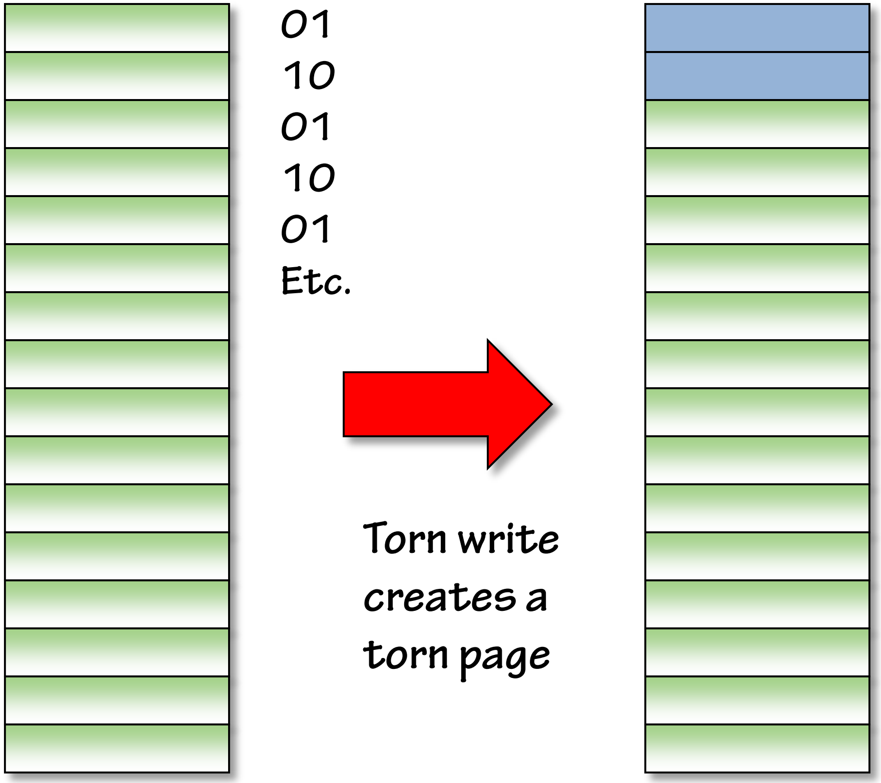
Torn-Page Detection (1)

- **An 8 KB data file page is really 16 x 512-byte disk sector**
- **It is possible for a page to be partially written in the event of a power failure**
- **Torn-page detection allows SQL Server to detect incomplete writes**
  - Takes two-bits from each sector, stores them in the page header (a 32-byte metadata structure in each data file page) and writes an alternating bit pattern in each sector
    - 01 in the first, 10 in the second, 01 in the third, etc.
    - The bit pattern flips each time the page is written to disk
  - On a subsequent read, if the pattern is disrupted, the page is torn
- **This does not detect corruptions within a disk sector though**

Torn-Page Detection (2)

Alternating
bit pattern

16 x 512-byte
Blocks



01
10
01
10
01
Etc.

Torn write
creates a
torn page

10
01
01
10
01
Etc.

This is detected by
the buffer pool the
next time the page is
read from disk

Page Checksums (1)

- **SQL Server 2005 introduced per-page checksums**
  - On by default for new databases from SQL Server 2005 onward
  - Added to *tempdb* from SQL Server 2008 onward
- **Performed by the buffer pool**
  - Calculated as a four-byte value and stored in the page header as the very last thing SQL Server does on a physical write
  - Recalculated and checked against the stored value as the very first thing SQL Server does on a physical read
- **Upgraded databases must enable it**
  - Switching it on doesn't do anything until pages are written
    - No easy method to force all pages to get a page checksum
  - Switching it on does not erase existing torn page detection
- **Negligible CPU overhead as simple checksum algorithm**
  - Error detecting, not error correcting

Page Checksums (2)

- **Provides the “smoking gun” that the error is not due to SQL Server**
  - Helps in hardware vendor vs. Microsoft arguments
- **Checked when:**
  - Page is read normally
  - Page is read during consistency checks
  - Page is read during BACKUP … WITH CHECKSUM
  - Page is read from within a checksum’d backup

I/O Errors

- **Three types**
  - 823: a hard I/O error
  - 824: a soft I/O error
  - 825: a read-retry error (we'll discuss later)
- **823 and 824 are severity 24 errors**
  - Connection will be broken
- **Logged in the *msdb.dbo.suspect_pages* table**
  - Can be used as input into single-page restore operations
- **Written to the SQL Server error log and the Windows Application event log**

I/O Error Example

```
Msg 824, Level 24, State 2, Line 1
SQL Server detected a logical consistency-based I/O error: incorrect
checksum (expected: 0x610d44d2; actual: 0x3e067a44). It occurred
during a read of page (1:3324) in database ID 31 at offset
0x000000019f8000 in file ‘D:\Pluralsight\corrupt.mdf'. Additional
messages in the SQL Server error log or system event log may provide
more detail. This is a severe error condition that threatens database
integrity and must be corrected immediately. Complete a full
database consistency check (DBCC CHECKDB). This error can be
caused by many factors; for more information, see SQL Server Books
Online.
```

Read-Retry and Error 825

- **Present in SQL Server 2000 for limited circumstances**
- **Extended in SQL Server 2005 for data file pages**
- **Reads that fail because of I/O errors are retried 4 times**
- **Eventual failure results in an I/O error**
- **Eventual "success" results in an 825 error in the error log:**
  - A read of the file <<FILE NAME>> at offset <<PHYSICAL OFFSET>> succeeded after failing <<RETRY COUNT>> time(s) with error: <<DETAILED ERROR INFORMATION>>. Additional messages in the SQL Server error log and system event log may provide more detail. This error condition threatens database integrity and must be corrected. Complete a full database consistency check (DBCC CHECKDB). This error can be caused by many factors; for more information, see SQL Server Books Online.
- **Early warning of impending doom as read-retry means the I/O subsystem is returning incorrect data to SQL Server**
- **KB article on this is at http://support.microsoft.com/kb/2015757**

Automatic Page Repair

- **Database mirroring and availability groups allow some corruptions to be automatically repaired, where possible**
  - Works for 824 "soft-I/O" errors, some 823 "hard-I/O" errors, 829 "in restore" errors
  - Errors are hit by queries reading the page or by consistency checks
- **Corrupt pages on the principal/mirror and primary/secondary replicas can be repaired**
- **Repairs are asynchronous, corrupt pages are unusable until fixed**
  - Subsequent reads of the page will return an 829 "in restore" error until the page is repaired, but will not trigger another repair attempt
- **NOTE: Only meant to be a band-aid to prevent downtime. It is NOT a substitute for having alerts for high-severity errors and taking action to rectify/prevent them.**
- **More information in Books Online at http://bit.ly/13J50y6**

Memory Corruption

- **The buffer pool periodically validates page checksums of clean pages in the buffer pool**
  - Failures results in an 832 error (do not confuse with 823)
    - 2013-06-18 11:41:45.94 spid4s Error: 832, Severity: 24, State: 1.
    - 2013-06-18 11:41:45.94 spid4s A page that should have been constant has changed (expected checksum: fecdb2d5, actual checksum: f04214c5, database 5, file 'C:\SQLskills\Data\AdventureWorks2008R2_Data.mdf', page (1:745)). This usually indicates a memory failure or other hardware or OS corruption.
- **SQL Server 2012 on Windows Server 2012 also supports automatic correction of some memory corruptions of buffer pool memory**
  - Corrupt clean pages are re-read from disk
  - Corrupt dirty pages cannot be repaired
  - See Glenn Berry's post at http://bit.ly/SmZLLA for more details

Monitoring for I/O Errors

- **Manual monitoring is time-consuming and prone to being forgotten**
- **Much better to use an automated monitoring process**
  - SQL Agent alerts
  - Microsoft SCOM
  - 3rd-party monitoring tool
- **Create alerts for:**
  - Severity 19 errors and above
  - Anything else you're interested in
- **Glenn Berry has a comprehensive blog post and detailed Transact-SQL script for creating SQL Agent alerts at http://bit.ly/1d86rcv**

Summary

- **It's very important that you utilize SQL Server's built-in corruption detection mechanism**
  - Use page checksums instead of torn page detection
  - On by default for new databases from SQL Server 2005 onward
- **Remember that the sooner you know you have corruption, the more likely you can recover with minimal downtime and data loss**

- **In the next module, we'll discuss:**
  - Consistency checking
  - Backup checksums
  - Frequently asked questions