

Module 2

SQL Server I/O

Copyright Tibor Karaszi Consulting and Cornerstone Group AB

Module Overview

- Core concepts of I/O
- Storage solutions
- I/O setup and testing

Lesson: Core concepts of I/O

- I/O patterns for data files (mdf and ndf)
- I/O patterns for transaction log files (ldf)
- Recovery related aspects
- Input/Output Operations Per Second
- Throughput
- Latency Factor
- Magnetic Disk and SSD

I/O patterns for data files (mdf and ndf)

- Reads
 - Synchronous, when data is requested
 - Read-ahead (EE has deeper RA)
 - Buffer ramp-up (EE)
 - Merry-go-round scanning (EE)
 - One scan operation jumps in on some other on-going scan
 - Only for allocation order scans (IAM)
 - Requires heap and NOLOCK
- Writes
 - Asynchronous
 - Checkpoint
 - Lazywriter
 - Exception is minimally logged operations which are synchronous

Investigating I/O using Extended Events

<https://techcommunity.microsoft.com/t5/core-infrastructure-and-security/dissecting-sql-server-physical-reads-with-extended-events-and/ba-p/370393>

I/O patterns for transaction log files (ldf)

- Some operations reads the transaction log
 - Backup
 - Log backup
 - Other backup types (includes log records produced while backup was running)
 - Rollback
 - Recovery
 - Start-up
 - Fail-over
 - Restore
 - Transactional replication and Change Data Capture
 - Create a database snapshot
 - DBCC CHECKDB uses snapshot internally
 - Availability Groups / Database mirroring (if falls behind)
 - CHECKPOINT in simple recovery (need to read the log in order to clear the log)
- Writes
 - Synchronous sequential writes

SQL Server Transaction Log Architecture and Management Guide

<https://docs.microsoft.com/en-us/sql/relational-databases/sql-server-transaction-log-architecture-and-management-guide>

Recovery related aspects

- Keep number of virtual log files to a reasonable number
 - Shortens recovery, failover etc.
- Turn on accelerated database recovery (2019)
 - Shortens recovery, failover etc
 - “Immediate” rollback
 - Less space used for transaction log
- Turn on indirect checkpoints
 - Default for new databases as of 2016
 - Lots of databases were born in an earlier version
 - Shorter checkpoints
 - Faster backups

SQL Server Transaction Log Architecture and Management Guide

<https://docs.microsoft.com/en-us/sql/relational-databases/sql-server-transaction-log-architecture-and-management-guide>

A Busy/Accidental DBA's Guide to Managing VLFs

<http://adventuresinsql.com/2009/12/a-busyaccidental-dbas-guide-to-managing-vlfs/>

Virtual Log Files

<https://dallasdbas.com/virtual-log-files/>

Accelerated Database Recovery in SQL Server 2019

<https://www.mssqltips.com/sqlservertip/5971/accelerated-database-recovery-in-sql-server-2019/>

More Reasons to Enable SQL Server Indirect Checkpoints

<https://www.mssqltips.com/sqlservertip/7108/sql-server-indirect-checkpoints-why-enable/>

Input/Output Operations Per Second

- A measure of the number of operations that a storage device can support in a second
- Can be further divided into read/write and random/sequential IOPS:
 - Random Read
 - Random Write
 - Sequential Read
 - Sequential Write
- As we have discussed, different type of database files have different read/write characteristics

Throughput

- A measure of the amount of data that a device can transfer in a fixed time period
 - Normally expressed in seconds per byte
- A device may have different figures for burst throughput and sustained throughput
- Sustained throughput is generally more important for SQL Server systems

Latency Factor

- A measure of how quickly a storage device can complete a read or write request
 - Normally measured in seconds per read or seconds per write
- A lower figure is always more desirable

- Magnetic Disk
 - Electro-mechanical technology—many moving parts
 - Performance and lifespan limited by mechanical restrictions
 - Slower, cheaper
- SSD
 - Electronic technology—no moving parts
 - Performance and lifespan limited by integrated circuit design and firmware
 - Faster, more expensive

Lesson: Storage solutions

- Direct-Attached Storage
- Storage Area Network
- Windows Storage Spaces
- SQL Server data files in Microsoft Azure
- Selecting a storage solution

Direct-Attached Storage

- One or more storage devices is directly attached to the server through a host bus adapter
- Dedicated to a specific server
- Advantages:
 - Easy and fast to provision
 - Easy monitoring and troubleshooting
- Disadvantages:
 - Expansion may be limited by hardware design
 - Lacks flexibility—unused capacity cannot be shared with other servers

Storage Area Network

- Pools of drive arrays linked together with a network
- Each server connects into this network and can share drives, cache, and throughput with many more servers
- Advantages:
 - Increases disk utilization and reduces management
 - Mirroring, snapshots, continuous data protection, clustering, and geocustering only offered by SANs
- Disadvantages:
 - Unpredictable performance, higher latency, limited bandwidth, and high cost

Windows Storage Spaces

- Software RAID managed by Windows
- Physical disks are grouped into storage pools
- Storage Spaces enables volumes to be created from storage pools
- Storage Spaces supports resiliency:
 - Mirroring: similar to RAID 1
 - Parity: similar to RAID 5
 - Simple: similar to RAID 0, no resiliency

Storage Spaces overview

<https://docs.microsoft.com/en-us/windows-server/storage/storage-spaces/overview>

Database files in Microsoft Azure

- SQL Server instance connects to database files held as Page Blobs in Azure Blob storage
 - Instance may be on-premises or on an Azure virtual machine
- Advantages:
 - Unlimited storage—pay for what you use
 - Easy migration
 - Centralized storage
 - Azure snapshot backup
- Disadvantages:
 - Maximum file size 1 TB
 - Difficult to predict costs
 - Not all SQL Server features available
 - Filestream
 - In-Memory OLTP

SQL Server data files in Microsoft Azure

<https://docs.microsoft.com/en-us/sql/relational-databases/databases/sql-server-data-files-in-microsoft-azure>

Selecting a storage solution

- No one-size-fits-all solution
- Many factors to consider:
 - Performance requirements
 - Organizational cloud strategy
 - Existing solutions
 - Budget
 - Urgency

Lesson: I/O Setup and testing

- NTFS Allocation Unit size
- Storage performance testing
- I/O related wait types

NTFS Allocation Unit size

- Sets the smallest unit of consumption on NTFS volumes for buffered I/O
- SQL Server mainly does unbuffered I/O
- Set during formatting
- Default value is 4 KB
- For SQL Server, a 64-KB allocation unit size is recommended:
 - Only affects a few operations because most SQL Server I/O is unbuffered
 - Improves read-ahead performance
 - NTFS compression may not be used
 - Less meta-data

SQL Server I/O block size

<https://blog.purestorage.com/purely-technical/what-is-sql-servers-io-block-size/>

More on SQL Server I/O, including elaboration on Allocation Unit size

<https://www.nocentino.com/posts/2021-12-10-sqlserver-io-size/>

Example of how you can measure I/O size. This example is for a log records, but can be adapted to other operations

<https://www.brentozar.com/archive/2012/05/how-big-your-log-writes-spying-on-sql-server-transaction-log/>

Storage performance testing

- Diskspd
 - A general-purpose load generator for I/O subsystems
 - Can be configured to mimic SQL Server I/O
 - Suitable for use as a performance-tuning tool
 - Replaces SQLIO
- ChrystalDiskMark
 - GUI on top of DISKSPD
 - You might want to change default values so it matches SQL Server I/O sizes
- Or, run BACKUP and DBCC CHECKDB and compare the new with the old
 - Backup to NUL to test reading of data only
 - Note that encryption (backup or database) and compression will add CPU load
 - CHECKDB does more than raw reading of source data

```
DiskSpd.exe -c500G -d600 -r -w20 -t8 -o8 -b8K -h -L H:\testfile.dat
```

Demo Diskspd

Testing I/O performance for SQL Server using ChrystalDiskMark

<https://www.brentozar.com/archive/2012/03/how-fast-your-san-or-how-slow/>

DISKSPD, github

<https://github.com/microsoft/diskspd>

Using Diskspd to test SQL Server Storage Subsystems

<https://www.sqlshack.com/using-diskspd-to-test-sql-server-storage-subsystems/>

- PAGEIOLATCH

- A task waits for a data page to be retrieved from disk to memory
- Reduce waits by
 - Improving I/O subsystem
 - Tune query so it does less I/O

- WRITELOG

- Writing to log is at the latest done synchronously as end of transaction
- Reduce waits by
 - Fewer, larger transactions
 - Fewer indexes
 - Improving I/O subsystem

SQL Server Wait Types

<https://www.sqlskills.com/help/waits/>

Optional Lab 2: SQL Server I/O

- Ex 1: Determine if a query is I/O bound
- Ex 2: If time permits, improve the performance of the query

Estimated Time: 30 minutes