Practical guidance to make your tier-1 SQL Server ROAR!

Tejas Shah
tejasaks@Microsoft.com
What’s a tier-1 SQL Server?

In short – a server where performance really counts
Performance considerations at every level

Hardware
Platform
Configuration

Server (OS)
Settings
Pagefile

SQL Server
Settings
Trace flags
TempDB

Application
Query Performance
Database Maintenance
Hardware
Platform

Physical

Virtual

Infrastructure as a Service (IaaS)

Platform as a Service (PaaS)
Physical Server

Know your workload

• OLTP: Large number of fast cores, large amount of memory, disk may not be as critical
• OLAP: Individual core speed not as critical, large amount of memory, high-end disk subsystem

Disk considerations

• Type of storage not as important as the specifications
• Size the storage subsystem based on number of IOPS and latency requirements, not just data size
Virtual Server Considerations

- Same recommendations as physical servers for resource allocation
- Minimal performance penalty, if any
- More flexibility for HA/DR solutions
- Configuration of host server is key
  - Don’t overcommit resources
  - Ensure Power Management is set to High Performance
  - Take care to configure vNUMA appropriately
Infrastructure as a Service Considerations

Know your workload, and tune before you go

- Throwing hardware at the problem no longer cost effective
- Streamline application before migration if possible

Sizing recommendations

- DS3_v2 or higher for SQL Enterprise Edition
- DS2_v2 or higher for SQL Standard Edition

I/O will likely be the biggest challenge

- Use Premium Storage
- Know your IOPS and stripe across multiple disks to achieve higher throughput
- Implement DB I/O optimizations such as page compression and instant file initialization
- Other best practice recommendations can be found here: https://docs.microsoft.com/azure/virtual-machines/windows/sql/virtual-machines-windows-sql-performance
Server (OS)
Server (OS) Configuration

Apply the latest version of the OS if possible

Keep up to date with OS updates

Power plan must be set to High Performance

Windows Paging File

- Don’t waste disk space, huge page file is not required
- If SQL Server memory is being paged out (Error 17890: A significant part of SQL server process memory has been paged out), add more memory

NTFS configuration

- Format data and log drives in NTFS 64-KB blocks
- Alignment is automatic starting with Windows 2008, review if partitions were upgraded
- Turn off NTFS file encryption and compression
SQL Server
Core Licensing vs. CAL Licensing

Make sure you install the correct version of the binaries

For core based licensing, install Enterprise Core to use all the cores on the server

Watch for the following message in the error log:

SQL Server detected 4 sockets with 6 cores per socket and 6 logical processors per socket, **24 total** logical processors; using **20** logical processors based on SQL Server licensing.
Parallelism

MAXDOP

- Default of 0 is rarely the correct setting
- Workload dependent – testing is the only way to know the right setting
- Look for recommendations from vendor if running a third-party application
- No testing? Set to number of physical cores in a single NUMA node, no higher than 8
- Starting with SQL Server 2016, can set this at the database level

Cost threshold for parallelism

- Workload dependent – testing is the only way to know the right setting
- Default value of 5 is fine unless you detect problems
- Consider making a larger number if you want to favor OLTP over OLAP workloads on the same server
Memory

Max server memory

- Common rule of thumb is 85%, but this is not appropriate for all workloads
- Not all SQL Server memory falls within max server memory limit
- If running things like SSIS or SSAS on the same server, adjust accordingly to prevent SQL Server from stepping on other processes and vice-versa

For an x64 server with 32GB RAM and 32 logical CPUs:

2 MB stack size * 960 default worker threads on 32 CPU machine
+ 256 MB (unless -g parameter is set)
+ 4 GB for OS
= roughly 6 GB
Max server memory = 26 GB
Memory

Max server memory

• Common rule of thumb is 85%, but this is not appropriate for all workloads
• Not all SQL Server memory falls within max server memory limit
• If running things like SSIS or SSAS on the same server, adjust accordingly to prevent SQL Server from stepping on other processes and vice-versa

Min server memory

• Default is fine for most workloads
• Consider setting higher for the following conditions
  • Running on a virtual server with an overcommitted host
  • Running with other workloads on the same server
  • Running multiple instances of SQL Server on the same server
Memory – Large Pages

**Lock pages in memory (LPIM)**
- Good idea to set on most servers (and another reason to set Max Server Memory)
- Needed for large pages

**Large page allocations**
- Enable with TF 834
- Good for large memory machines with analytical workloads
- Not supported with columnstore indexes at this time
I/O Considerations

Turn on instant file initialization

• Starting with SQL Server 2016, leverage setup to enable

File placement

• Still recommend separate logical drives for data, transaction log and tempdb, even on a SAN
• No need for multiple data files in the same filegroup unless spreading across multiple VHDs, or hitting allocation contention on a user database (similar to TempDB)
• If using multiple files per filegroup, ensure all are of equal size
• For IaaS, use Storage Spaces to create logical partitions across multiple disks rather than database files
**TempDB**

**Fast storage**
- No need for redundancy, optimize for performance (avoid RAID 5 and 6)
- Consider flash or SSD

**Multiple data files**
- Start with one data file per logical processor up to 8
- Add more in multiples of 4 files until you no longer see contention
- Files must be equally sized
- If adding new files, be sure to restart SQL Server to rebalance across all the files

**Upgrade to the latest CU to get all the new fixes and enhancements**

**Avoid explicitly dropping temp tables in code**
NUMA

Important for SQL Server memory object management and thread scheduling

If running in virtualized environment, emulate host NUMA settings
  • Avoid uneven and/or partially overlapping NUMA nodes

Auto Soft-NUMA
  • Enabled by default on SQL Server 2016 and later, for servers with more than 8 processors in a single node
  • SQL Server 2014 SP2 needs trace flag 8079
Trace Flags

For SQL Server 2016 and 2017

• 174: SOS_CACHESTORE spinlock contention, KB 3026083
• 4199: Optimizer fixes (can also be set via a database-level configuration)
• 3468: Disable indirect checkpoints on tempdb, KB 4040276
• 7471: Parallel statistics update jobs, KB 3156157
• 2566: Faster CHECKDB command, KB 945770
• 7412: Enabled lightweight profiling, KB 3170113

For SQL Server 2012 and 2014, all of the above plus

• 1117: Autogrow all files in the same filegroup at the same time (tempdb, KB 2154845)
• 1118: Turn off mixed page allocations (tempdb, KB 2154845)
• 2371: Enable linear recompilation threshold for large tables, KB 2754171
• 6498: Avoid compilation waits for concurrent large queries, KB 3024815
• 6532, 6533, 6534: Spatial query performance, KB 3107399
• 8048, 8079: CMEMTHREAD waits for systems with more than 8 cores per NUMA node
• 8075: VAS fragmentation, KB 3074434
Application
Partitioning?

A perfect fit for VLDBs

- Sliding windows scenarios
- Fast archiving
- Allows partition switching and optimal filegroup backup/restore (read-only data)

Find an optimal partitioning scheme based on the existing business requirements

- Examine the partitioning function, the data in the table, and the business uses of the data
- When partitioning any index (including clustered), the index key must contain the partitioning column
- If a natural partitioning key is available, use it (date, incremental ID)

Leverage partition-aligned indexes

- Partitioned indexes, functions, and schemes should have the same partitioning key and ranges
- To enable partition switching, all indexes on the table must be aligned
- Use the partitioning key in your query predicates (WHERE / JOIN) to encourage partition elimination
- Beware of queries with TOP, MIN/MAX as these must scan all partitions
Collations

• UTF-8 is enabled when creating or changing an object’s collation to a collation with the “UTF8” suffix, such as LATIN1 GENERAL 100_CI_AS_SC to LATIN1 GENERAL 100_CI_AS_SC_UTF8
• UTF-8 is only available to Windows collations that support supplementary characters, as introduced in SQL Server 2012.
• CHAR and VARCHAR datatypes store UTF-8, NCHAR and NVARCHAR continue to use UTF-16 encoding only, and remain unchanged.

Beware of user databases with collation different from system databases

In SQL Server 2019, leverage the new UTF-8 collations
Database Maintenance

Maintain your indexes

• This may mean Rebuild or Reorganize depending on current level of fragmentation
• Workloads that scan are more vulnerable to performance issues with fragmented indexes
• Leverage intelligent management. Ex: Adaptive Index Defrag (http://aka.ms/AID)

Update statistics

• As needed, so leverage intelligent management. Ex: Adaptive Index Defrag (http://aka.ms/AID)
• Even more relevant if not using AUTO_UPDATE_STATISTICS

Implement Integrity checking

• DBCC CHECKDB runs depends on the individual business needs and the importance of the information in the database
• At minimum, run against all production databases at least once a week and review results
• Not running means not knowing of early signs of corruption, and can lead to data loss
Query Optimizer / Cardinality Estimation

Every new version introduces functional changes to the QO under the latest Database Compatibility Level

- Upgrading and staying in source database compatibility mode is fully supported.
- Still plan to certify ASAP on latest
- Evaluate use of TF 4199 / DB scoped config “Query Optimizer Fixes”

A new version introduces a new CE version

- Mechanism for estimating required number of rows to satisfy query, using statistical models and heuristics
- No more “Old CE” vs “New CE” only
  - “Old CE” = CE 70, default for SQL Server 7.0 to 2012
  - “New CE” = CE 120 (SQL Server 2014) through CE 150 (SQL Server 2019)
- It’s recommended to follow documented upgrade process to avoid increased risk of regression!
Query Store

Comprehensive query-performance information when you need it most!
Query Store

Includes waits in SQL Server 2017
Query Store and Automatic Plan Correction

Identifies the problematic query plan and “fixes” it to be optimal.

In the scope of a Database Compatibility Level upgrade process was followed.
Lightweight Profiling: query progress anytime, anywhere

Starting with SQL Server 2016 SP1* and 2017, the new lightweight query execution statistics profile infrastructure allows continuous collection of per-operator query execution statistics

- Using global TF 7412
- Enabling query_thread_profile extended event
- Using query hint `USE HINT('query_plan_profile')` in SQL Server 2017 CU11 and 2016 SP2 CU3 (KB 4458593)
- Default ON in SQL Server 2019

When lightweight profiling is on, `sys.dm_exec_query_profiles` is also populated for all sessions

- Enables new LQS feature in SSMS (including Activity Monitor)
- New DMF `sys.dm_exec_query_statistics_xml`

The following still use regular profiling infra

- SET STATISTICS XML (or Include Actual Plan)
- query_post_execution_showplan extended event
# What is the impact of Lightweight Profiling?

Query Execution Statistics Profiling Infrastructure tests with TPC-C like workloads

<table>
<thead>
<tr>
<th>Infra Type</th>
<th>Overhead percent (up to)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>no active xEvents</td>
</tr>
<tr>
<td>Regular</td>
<td>75.5</td>
</tr>
<tr>
<td>Lightweight in SQL Server 2014 SP2/2016</td>
<td>3.5</td>
</tr>
<tr>
<td>Lightweight in SQL Server 2016 SP1 and above</td>
<td>2</td>
</tr>
</tbody>
</table>
Tiger Toolbox

The SQL Server team shares a number of scripts and tools you can (argue should) use every day in your SQL Server estate:

- Adaptive Index Defrag
- BPCheck (SQL Best Practices and Performance checks)
- Fixing transaction log VLF count
- Maintenance solution that includes integrity checking
- Waits and Latches information
- Running and Blocked processes
- System Health Session PowerBI
- Various other support scripts

Find these and more at http://aka.ms/TigerToolbox
Query progress anytime, anywhere

Demo
Automatic Tuning

Demo
Bookmarks

SQL Server Team (Tiger) Blog
  http://aka.ms/sqlserverteam
Tiger Toolbox GitHub
  http://aka.ms/tigertoollbox
SQL Server Release Blog
  http://aka.ms/sqlreleases
BP Check
  http://aka.ms/bpcheck
SQL Server Standards Support
  http://aka.ms/sqlstandards
Trace Flags
  http://aka.ms/traceflags
SQL Server Support lifecycle
  http://aka.ms/sqlilifecycle
SQL Server Updates
  http://aka.ms/sqluupdates
SQL Server Guides
  http://aka.ms/sqlserverguides
SQL Feedback (New “Connect”)
  http://aka.ms/sqlfeedback
T-SQL Syntax Conventions
  http://aka.ms/sqlconventions
Twitter
  @mssqltiger
Thank you