DBA Services

1. Performance Tuning Overview

Click here to expand...

A. Most Common Performance Issues (64% of all performance problems)

1. Insufficient or poor indexes
   a. Table scans impact disk performance and memory use, as well as lead to blocking
   b. It's possibly to have too many indexes, which lead to performance hits on data modification queries (INSERT, DELETE, or UPDATE operation)
2. Inaccurate or missing statistics
   a. The query optimizer makes choices based on row estimates that come from these statistics
3. Bad T-SQL
   a. Moving too much data, writing overcomplicated code, using wrong object types, etc.
4. Problematic execution plans
   a. Most of times, these are fixed through code changes, statistics updates, or new indexes
   b. Other times, this occurs due to parameter sniffing gone wrong
5. Excessive blocking
   a. A lack of resources, not enough memory, CPU, or fast enough disks can lead to additional blocking
6. Deadlocks
   a. Caused by blocking, but is something separate
   b. If all your queries complete fast enough, the chances of a deadlock are very slim
7. Non-set-based operations
   a. Caused by cursors and other types of loop operations to force a row-by-row style processing
8. Incorrect database design
   a. Ensuring that your database is properly normalized and data is stored properly (ex. dates go into a datetime column)
9. Poor execution plan reuse
   a. Caused by things like dynamic T-SQL or inappropriate parameters, preventing plan reuse or parameterization
10. Frequent recompilation of queries
   a. While recompilation is generally desirable, there can be too much due to volatile data or poor code

B. Overview of (Recursive) Query Performance Tuning Process

1. Set performance target for application
2. Analyze application performance
   a. Ensure servers are not overwhelmed
      i. Process of capturing performance metrics varies depending on if the server is on VMware, Hyper-V, Docker, AWS, Azure, etc.
      ii. In general, focus on collecting metrics on waits and queues, especially around disk I/O, memory, and CPU
      iii. Network (health of the routers, cables, Wi-Fi repeaters, etc.) can also affect performance
3. Identify resource bottlenecks
4. Ensure proper configuration for hardware, OS, platform, SQL Server, database, and applications
5. Identify costliest query associated with bottleneck
6. Optimize query

C. Identifying resource bottlenecks

This is a repetitive process that goes as follows:

1. Identify the bottleneck
2. Fix it
3. Validate the fix
4. Measure the impact and current performance
5. Start again with the next bottleneck

This process should be done for one bottleneck at a time, making one change at a time and validating that one change at a time.

D. In-depth Query Tuning Process Overview

1. Baseline performance and resource use of costliest query
2. Set performance target for query (ex. every query has to meet a three-second minimum operation, with a few exceptions)
3. Analyze and optimize factors (such as statistics) that influence query execution
4. Analyze query for common problems

5. Analyze query execution plan
6. Analyze and prioritize operators to identify bottlenecks
7. If warranted, modify query and/or index. Afterwards:
    a. Measure performance and resource use again
    b. Determine if query performance improved
        i. If not, undo changes!
8. Determine if query performance is acceptable
    a. If not, return to step 4

2. Creating a Baseline

A. Notes on the Performance Monitor Tool

A. Notes on the Performance Monitor Tool

1. On a VM, counter measurements is usually is for the VM, not the physical server. That means some values are not going to accurately reflect physical reality.
2. While real time graphs are available, it’s recommended to captured data into a file (called a *data collector set*) for offline analysis.

B. Memory Performance Analysis

B. Memory Performance Analysis

Memory bottlenecks are a priority because they also affect CPU. More CPU cycles are needed to write memory pages to disk so SQL Server can maintain enough free internal memory (via a process called *lazy writer*).

I. Memory Settings

The following script can be used to determine the *max server memory* setting for SQL Server:

https://github.com/bornsql/scripts/blob/main/max_server_memory.sql

At least 2GB should be available for the OS and another 2GB if MedInformatix is hosted on the same server.

Microsoft recommends that *min server memory* is 0.

II. Memory Perfmon Counters

1. Memory - Available MBytes
    a. Extended periods of time with this value very low and SQL Server memory not changing indicates that the server is under severe memory stress.
        i. Low memory is defined as less than 1GB by Brent Ozar and 10% of the installed memory by PAL
    b. Can also look at Available Bytes/Kbytes for more granularity
2. Memory - Pages/Sec and Memory - Page Faults/Sec
    a. Pages/Sec measures the rate of hard page faults (where data must be retrieved from disk), whereas Page Faults/Sec measures the rate of total page faults, including soft faults (where the data can be retrieved from memory).
    b. A baseline is essential to determine the expected normal behavior, which can range from 0 to 1,000 per second for Page Faults /Sec.
        i. These numbers vary widely based on the amount and type of memory as well as the speed of disk access on the system
    c. Per Scott Whigham, some page faults are normal; look for consistently high numbers
    d. For more granularity, Pages/sec can be broken up into
        i. Pages Input/sec for page reads. This is what causes wait times in application.
        ii. Pages Output/sec for page writes. This can be ignored unless disk load is an issue.
    e. Process:Page Faults/sec can also be used to see specifically which process is causing excessive paging
3. Paging File - %Usage
    a. Paging is used by virtual memory to allows users to execute programs larger than the actual physical memory.
    b. Ideally we shouldn't have to use the Paging File. Brent Ozar recommends a 0 or 1% usage at most, while PAL is fine with up to 70%.
    c. Note that paging may be caused by problems external to SQL Server
    d. Paging File - %Usage Peak can also be used to see peak values
4. SQLServer:Buffer Manager - Buffer Cache Hit Ratio
    a. Per Brent Ozar, it used to be recommended to keep this above 90%, which means the majority of reads are coming out of the buffer cache
        i. This may not be true anymore depending on how reads are done in the system (ex. for a reporting workloads with lots of ad hoc queries)
        ii. A baseline value is needed to determine what’s normal
5. SQLServer:Buffer Manager - Page Life Expectancy

   a. This indicates how long a page will stay in the buffer pool without being referenced
   b. Brent Ozar recommends at least 180 seconds, while SQLwatch.io recommends > 300 seconds
   c. A baseline needs to be established and monitored over time
      i. Reporting systems, as opposed to OLTP systems, are expected to have a lower value
      ii. This number is also expected to dip to very low levels during nightly loads
6. SQLServer:Buffer Manager - Checkpoint Pages/Sec
   a. This represents the number of pages moved to disk by a checkpoint operation
   b. Should be relatively low: less than 30 per second for most systems
   c. Higher values indicate a large number of writes occurring within the system, possibly indicative of I/O problems
7. SQLServer:Buffer Manager - Lazy Writes/Sec
   a. This records how much dirty, aged buffers are removed from the buffer by the lazy write process, a system process that frees up the memory for other uses
   b. Should consistently be less than 20 for the average system
   c. Higher values possibly indicate I/O issues or even memory problems
8. SQLServer:Memory Manager - Memory Grants Pending
   a. Under normal conditions, this should be consistently 0 for most production servers
   b. To retrieve this value on the fly, run queries against the DMV `sys.dm_exec_query_memory_grants` . A `null` value in the column `grant_time` indicates that the process is still waiting for a memory grant.
9. SQLServer:Memory Manager - Target Server Memory (KB) and SQLServer:Memory Manager - Total Server Memory (KB)
   a. Target Server Memory (KB) indicates the total amount of dynamic memory SQL Server is willing to consume, while Total Server Memory (KB) indicates the amount of memory currently assigned to SQL Server
   b. Target Server Memory (KB) should be close to the size of physical memory available, especially if the system is dedicated to SQL Server
   c. Total Server Memory (KB) should be close to Target Server Memory (KB). If it's much less, make sure (1) max server memory is configured properly and (2) the system did not just start up (since there is a *warm-up phase* to expand memory allocation and bring data pages into memory). Ruling those two out, we can conclude SQL Server simply has a low memory requirement.
   d. Systems with a low memory requirement usually have 5,000 free pages or more. The status of memory allocation can also be queried via the DMV `sys.dm_os_ring_buffers` .
10. Process:Private Bytes and SQL Server:Buffer Manager - Total pages
   a. Most memory used by SQL Server goes into the buffer pool. However, there are allocations beyond the buffer pool known as *privat e bytes* that can cause memory pressure
   b. If you suspect this issue, compare Process:Private Bytes with SQL Server:Buffer Manager - Total pages

III. Additional Memory Monitoring Tools

1. DBCC MEMORYSTATUS
   a. Returns two result sets, one showing allocations of memory and counts of occurrences, another showing different memory managers and the amount of memory they have consumed at the moment.
   b. If Memory Manager "Target Committed" (the system required memory) is higher than "Current Committed" (the memory currently provided), there may be buffer cache problems.
      i. Figure out which process is currently using the most memory using `sys.dm_os_performance_counters`
2. Dynamic Management Views
   a. Sys.dm_os_memory_brokers
      i. While most memory is allocated to the buffer cache, this DMV exposes other processes that also consume memory
   b. Sys.dm_os_memory_clerks
      i. If perfmon counter Process:Private Bytes is high, use this to determine how memory is being allocated
      ii. For in-memory OLTP storage, use `sys.dm_db_xtp_table_memory_stats` instead
   c. Sys.dm_os_ring_buffers
      i. Not documented in Books Online, so it's subject to change or removal
      ii. Outputs as XML, so XQuery may be needed to digest
      iii. A ring buffer is simply a recorded response to a notification. We can use it to see changes in memory usage
      iv. Main ring buffers associated with memory:
         1. RING_BUFFER_RESOURCE_MONITOR
            a. Records changes in memory allocation
            b. Useful for identifying external memory pressure
         2. RING_BUFFER_OOM (out of memory)
            a. Records out-of-memory issues
            b. Can see what kind of memory action failed
         3. RING_BUFFER_MEMORY_BROKER
            a. Identifies internal memory pressure, which forces processes to release memory for the buffer
         4. RING_BUFFER_BUFFER_POOL
            a. Records when the buffer pool itself is running out of memory
               i. This is a general indication of memory pressure
   d. Sys.dm_db_xtp_table_memory_stats
      i. Shows memory used by tables and indexes created in-memory
      ii. Will need to query `sys.objects` to get the actual names of tables or indexes

e. Sys.dm_xtp_system_memory_consumers
    i. Shows system structures used to manage the internals of the memory engine (as opposed to the amount of data loaded into memory)
    ii. Look for the allocated and used bytes shown for each of the management structures

IV. Memory Bottleneck Resolutions

1. When there is high memory stress, indicated by a large number of hard page faults, this general flowchart can be used to resolve it:
    a. Check if Memory:Available Mbytes is low
        i. If yes, it's possible there are external memory problems. Troubleshoot in operating system.
    b. In `DBCC MEMORYSTATUS`, check if COMMITTED is above TARGET
        i. If yes, there is internal memory pressure. Identify large consumers using `sys.dm_os_memory_brokers`
    c. Check if Process:Private Bytes is high
        i. If yes, there is internal memory pressure other than the buffer. Identify with `sys.dm_os_memory_clerks`
    d. Finally, check for memory errors in the OS log and SQL Server log
2. Optimizing Application Workload
    a. Capture all SQL queries using Extended Events (Chapter 6) or Query Store (Chapter 11), then group the output on the `Reads` column
        i. Queries with the highest number of logical reads contribute most often to memory stress
    b. `sys.dm_exec_query_stats` can also be used for a quick and easy analysis, but since it's based on cache it may not be as accurate as EE
3. Allocating More Memory to SQL Server
    a. Hard page faults mean that the memory requirement of SQL Server is more than the max server memory value
    b. In this case, the max server memory configuration should be increased accordingly (assuming enough physical memory is available in the system)
    c. If using in-memory OLTP storage, the memory percentages allocated to the resource pools defined may need to be adjusted
4. Moving In-Memory Tables Back to Standard Storage
    a. Keep an eye on general query performance metrics for in-memory tables since not all tables or workloads benefit from them
    b. See Chapter 24 for more details
5. Increasing System Memory
    a. To identify which queries are using more memory, query `sys.dm_exec_query_memory_grants` to see I/O use
        i. Be careful running this query on systems already under memory stress
    b. If no queries stand out, the easiest and quickest resolution may be to simply increase system memory by purchasing and installing more
6. Changing from a 32-Bit to a 64-Bit Processor
    a. Since SQL Server 2012, a 32-bit instance of SQL Server is limited to accessing only 3GB of memory
        i. Because of this limitation, only small systems should be running 32-bit versions
    b. 64-bit processors can access up to 24TB depending on the version of the OS and the specific processor type
    c. SQL Server 2017 does not support the x86 chip set and must be run on a 64-bit processor
7. Compressing Data
    a. Less system memory is used because compressed data remains compressed in memory
    b. There is a CPU cost, so keep an eye on that to be sure you're not just transferring stress
    c. Depending on the nature of your data, there may not be much compression
8. Enabling 3GB of Process Address Space
    a. For 32-bit OS systems with 16GB of physical memory or less, less memory can be allocated to the OS and more to applications. This is done by specifying a /3GB switch in the `boot.ini` file
        i. Not applicable to SQL Server 2017+
9. Addressing Fragmentation
    a. While fragmentation may affect storage, it also can affect memory
        i. Fragmented pages have lots of empty space, all of which are retrieved from disk and into memory
    b. See Chapter 17 for details

- https://learning.oreilly.com/library/view/sql-server-2017/9781484238882/html/323849_5_En_5_Chapter.xhtml

C. Disk Performance Analysis

C. Disk Performance Analysis

Although memory and CPUs have become faster and faster, disks and the disk subsystem remains one of the slowest parts of any computing system. Additionally, I/O is affected by many factors such as:

- Virtual environments with shared disks
- Antivirus programs
- Filter drivers acting as bottlenecks in I/O paths

I. Disk Counters

1. PhysicalDisk - Disk Transfers/sec
    a. Although disc activity can be divided into LogicalDisk partitions, bottlenecks ultimately occur on the physical disk
    b. Note that RAID (redundant array of independent disks) and SAN (storage area network) systems are treated as a single physical disk
    c. Numbers are more in line with platter-style disk drives that are fast becoming obsolete
        i. Modern setups will use solid-state drives, SSD arrays, or iSCSI interfaces
    d. Disk Transfers/sec monitors the rate of read and write operations on the disk
        i. A typical hard disk drive can do about 180 disk transfers per second for sequential I/O (IOPS) and 100 disk transfers per second for random I/O
            1. Random I/O is lower because more disk arm and head movements are involved
            2. OLTP workloads rely heavily on random access for its singleton/small operations and thus is typically constrained more by disk transfers/sec than disk bytes/sec
            3. For comparison, an SSD can be anywhere from around 5,000 IOPS to as much as 500,000 IOPS
2. PhysicalDisk - Disk Bytes/Sec
    a. Monitors the rate at which bytes are transferred to or from the disk during read or write operations
        i. A typical disk spinning at 7200RPM can transfer about 1000MB per second
        ii. If the amount of data transfer exceeds capacity, there will be a backlog shown by the Disk Queue Length counters
        iii. This is usually a nonissue for OLTP applications, which access small amounts of data in individual database requests
3. PhysicalDisk - Avg. Disk Sec/Read and Avg. Disk Sec/Write
    a. These track the average amount of time it takes in milliseconds to read from or write to a disk

3. SQL Server Health Check Tools

Click here to expand...

A. Brent Ozar's First Responder Kit

What_to_Do_Whe...Brent_Ozar.pdf

B. Paul Randal's Wait Statistics Query

```
-- Last updated October 1, 2021
WITH [Waits] AS
    (SELECT
        [wait_type],
        [wait_time_ms] / 1000.0 AS [WaitS],
        ([wait_time_ms] - [signal_wait_time_ms]) / 1000.0 AS
[ResourceS],
        [signal_wait_time_ms] / 1000.0 AS [SignalS],
        [waiting_tasks_count] AS [WaitCount],
        100.0 * [wait_time_ms] / SUM ([wait_time_ms]) OVER() AS
[Percentage],
```

```
        ROW_NUMBER() OVER(ORDER BY [wait_time_ms] DESC) AS [RowNum]
    FROM sys.dm_os_wait_stats
    WHERE [wait_type] NOT IN (
        -- These wait types are almost 100% never a problem and so
they are
        -- filtered out to avoid them skewing the results. Click on
the URL
        -- for more information.
        N'BROKER_EVENTHANDLER', -- https://www.sqlskills.com/help
/waits/BROKER_EVENTHANDLER
        N'BROKER_RECEIVE_WAITFOR', -- https://www.sqlskills.com/help
/waits/BROKER_RECEIVE_WAITFOR
        N'BROKER_TASK_STOP', -- https://www.sqlskills.com/help/waits
/BROKER_TASK_STOP
        N'BROKER_TO_FLUSH', -- https://www.sqlskills.com/help/waits
/BROKER_TO_FLUSH
        N'BROKER_TRANSMITTER', -- https://www.sqlskills.com/help/waits
/BROKER_TRANSMITTER
        N'CHECKPOINT_QUEUE', -- https://www.sqlskills.com/help/waits
/CHECKPOINT_QUEUE
        N'CHKPT', -- https://www.sqlskills.com/help/waits/CHKPT
        N'CLR_AUTO_EVENT', -- https://www.sqlskills.com/help/waits
/CLR_AUTO_EVENT
        N'CLR_MANUAL_EVENT', -- https://www.sqlskills.com/help/waits
/CLR_MANUAL_EVENT
        N'CLR_SEMAPHORE', -- https://www.sqlskills.com/help/waits
/CLR_SEMAPHORE

        -- Maybe comment this out if you have parallelism issues
        N'CXCONSUMER', -- https://www.sqlskills.com/help/waits
/CXCONSUMER

        -- Maybe comment these four out if you have mirroring issues
        N'DBMIRROR_DBM_EVENT', -- https://www.sqlskills.com/help/waits
/DBMIRROR_DBM_EVENT
        N'DBMIRROR_EVENTS_QUEUE', -- https://www.sqlskills.com/help
/waits/DBMIRROR_EVENTS_QUEUE
        N'DBMIRROR_WORKER_QUEUE', -- https://www.sqlskills.com/help
/waits/DBMIRROR_WORKER_QUEUE
        N'DBMIRRORING_CMD', -- https://www.sqlskills.com/help/waits
/DBMIRRORING_CMD
        N'DIRTY_PAGE_POLL', -- https://www.sqlskills.com/help/waits
/DIRTY_PAGE_POLL
        N'DISPATCHER_QUEUE_SEMAPHORE', -- https://www.sqlskills.com
/help/waits/DISPATCHER_QUEUE_SEMAPHORE
        N'EXECSYNC', -- https://www.sqlskills.com/help/waits/EXECSYNC
        N'FSAGENT', -- https://www.sqlskills.com/help/waits/FSAGENT
        N'FT_IFTS_SCHEDULER_IDLE_WAIT', -- https://www.sqlskills.com
/help/waits/FT_IFTS_SCHEDULER_IDLE_WAIT
        N'FT_IFTSHC_MUTEX', -- https://www.sqlskills.com/help/waits
```

```
/FT_IFTSHC_MUTEX

       -- Maybe comment these six out if you have AG issues
        N'HADR_CLUSAPI_CALL', -- https://www.sqlskills.com/help/waits
/HADR_CLUSAPI_CALL
        N'HADR_FILESTREAM_IOMGR_IOCOMPLETION', -- https://www.
sqlskills.com/help/waits/HADR_FILESTREAM_IOMGR_IOCOMPLETION
        N'HADR_LOGCAPTURE_WAIT', -- https://www.sqlskills.com/help
/waits/HADR_LOGCAPTURE_WAIT
        N'HADR_NOTIFICATION_DEQUEUE', -- https://www.sqlskills.com
/help/waits/HADR_NOTIFICATION_DEQUEUE
        N'HADR_TIMER_TASK', -- https://www.sqlskills.com/help/waits
/HADR_TIMER_TASK
        N'HADR_WORK_QUEUE', -- https://www.sqlskills.com/help/waits
/HADR_WORK_QUEUE

        N'KSOURCE_WAKEUP', -- https://www.sqlskills.com/help/waits
/KSOURCE_WAKEUP
        N'LAZYWRITER_SLEEP', -- https://www.sqlskills.com/help/waits
/LAZYWRITER_SLEEP
        N'LOGMGR_QUEUE', -- https://www.sqlskills.com/help/waits
/LOGMGR_QUEUE
        N'MEMORY_ALLOCATION_EXT', -- https://www.sqlskills.com/help
/waits/MEMORY_ALLOCATION_EXT
        N'ONDEMAND_TASK_QUEUE', -- https://www.sqlskills.com/help
/waits/ONDEMAND_TASK_QUEUE
        N'PARALLEL_REDO_DRAIN_WORKER', -- https://www.sqlskills.com
/help/waits/PARALLEL_REDO_DRAIN_WORKER
        N'PARALLEL_REDO_LOG_CACHE', -- https://www.sqlskills.com/help
/waits/PARALLEL_REDO_LOG_CACHE
        N'PARALLEL_REDO_TRAN_LIST', -- https://www.sqlskills.com/help
/waits/PARALLEL_REDO_TRAN_LIST
        N'PARALLEL_REDO_WORKER_SYNC', -- https://www.sqlskills.com
/help/waits/PARALLEL_REDO_WORKER_SYNC
        N'PARALLEL_REDO_WORKER_WAIT_WORK', -- https://www.sqlskills.
com/help/waits/PARALLEL_REDO_WORKER_WAIT_WORK
        N'PREEMPTIVE_OS_FLUSHFILEBUFFERS', -- https://www.sqlskills.
com/help/waits/PREEMPTIVE_OS_FLUSHFILEBUFFERS
        N'PREEMPTIVE_XE_GETTARGETSTATE', -- https://www.sqlskills.com
/help/waits/PREEMPTIVE_XE_GETTARGETSTATE
        N'PVS_PREALLOCATE', -- https://www.sqlskills.com/help/waits
/PVS_PREALLOCATE
        N'PWAIT_ALL_COMPONENTS_INITIALIZED', -- https://www.sqlskills.
com/help/waits/PWAIT_ALL_COMPONENTS_INITIALIZED
        N'PWAIT_DIRECTLOGCONSUMER_GETNEXT', -- https://www.sqlskills.
com/help/waits/PWAIT_DIRECTLOGCONSUMER_GETNEXT
        N'PWAIT_EXTENSIBILITY_CLEANUP_TASK', -- https://www.sqlskills.
com/help/waits/PWAIT_EXTENSIBILITY_CLEANUP_TASK
        N'QDS_PERSIST_TASK_MAIN_LOOP_SLEEP', -- https://www.sqlskills.
com/help/waits/QDS_PERSIST_TASK_MAIN_LOOP_SLEEP
```

```
        N'QDS_ASYNC_QUEUE', -- https://www.sqlskills.com/help/waits
/QDS_ASYNC_QUEUE
        N'QDS_CLEANUP_STALE_QUERIES_TASK_MAIN_LOOP_SLEEP',
            -- https://www.sqlskills.com/help/waits
/QDS_CLEANUP_STALE_QUERIES_TASK_MAIN_LOOP_SLEEP
        N'QDS_SHUTDOWN_QUEUE', -- https://www.sqlskills.com/help/waits
/QDS_SHUTDOWN_QUEUE
        N'REDO_THREAD_PENDING_WORK', -- https://www.sqlskills.com/help
/waits/REDO_THREAD_PENDING_WORK
        N'REQUEST_FOR_DEADLOCK_SEARCH', -- https://www.sqlskills.com
/help/waits/REQUEST_FOR_DEADLOCK_SEARCH
        N'RESOURCE_QUEUE', -- https://www.sqlskills.com/help/waits
/RESOURCE_QUEUE
        N'SERVER_IDLE_CHECK', -- https://www.sqlskills.com/help/waits
/SERVER_IDLE_CHECK
        N'SLEEP_BPOOL_FLUSH', -- https://www.sqlskills.com/help/waits
/SLEEP_BPOOL_FLUSH
        N'SLEEP_DBSTARTUP', -- https://www.sqlskills.com/help/waits
/SLEEP_DBSTARTUP
        N'SLEEP_DCOMSTARTUP', -- https://www.sqlskills.com/help/waits
/SLEEP_DCOMSTARTUP
        N'SLEEP_MASTERDBREADY', -- https://www.sqlskills.com/help
/waits/SLEEP_MASTERDBREADY
        N'SLEEP_MASTERMDREADY', -- https://www.sqlskills.com/help
/waits/SLEEP_MASTERMDREADY
        N'SLEEP_MASTERUPGRADED', -- https://www.sqlskills.com/help
/waits/SLEEP_MASTERUPGRADED
        N'SLEEP_MSDBSTARTUP', -- https://www.sqlskills.com/help/waits
/SLEEP_MSDBSTARTUP
        N'SLEEP_SYSTEMTASK', -- https://www.sqlskills.com/help/waits
/SLEEP_SYSTEMTASK
        N'SLEEP_TASK', -- https://www.sqlskills.com/help/waits
/SLEEP_TASK
        N'SLEEP_TEMPDBSTARTUP', -- https://www.sqlskills.com/help
/waits/SLEEP_TEMPDBSTARTUP
        N'SNI_HTTP_ACCEPT', -- https://www.sqlskills.com/help/waits
/SNI_HTTP_ACCEPT
        N'SOS_WORK_DISPATCHER', -- https://www.sqlskills.com/help
/waits/SOS_WORK_DISPATCHER
        N'SP_SERVER_DIAGNOSTICS_SLEEP', -- https://www.sqlskills.com
/help/waits/SP_SERVER_DIAGNOSTICS_SLEEP
        N'SQLTRACE_BUFFER_FLUSH', -- https://www.sqlskills.com/help
/waits/SQLTRACE_BUFFER_FLUSH
        N'SQLTRACE_INCREMENTAL_FLUSH_SLEEP', -- https://www.sqlskills.
com/help/waits/SQLTRACE_INCREMENTAL_FLUSH_SLEEP
        N'SQLTRACE_WAIT_ENTRIES', -- https://www.sqlskills.com/help
/waits/SQLTRACE_WAIT_ENTRIES
        N'VDI_CLIENT_OTHER', -- https://www.sqlskills.com/help/waits
/VDI_CLIENT_OTHER
        N'WAIT_FOR_RESULTS', -- https://www.sqlskills.com/help/waits
```

```
/WAIT_FOR_RESULTS
        N'WAITFOR', -- https://www.sqlskills.com/help/waits/WAITFOR
        N'WAITFOR_TASKSHUTDOWN', -- https://www.sqlskills.com/help
/waits/WAITFOR_TASKSHUTDOWN
        N'WAIT_XTP_RECOVERY', -- https://www.sqlskills.com/help/waits
/WAIT_XTP_RECOVERY
        N'WAIT_XTP_HOST_WAIT', -- https://www.sqlskills.com/help/waits
/WAIT_XTP_HOST_WAIT
        N'WAIT_XTP_OFFLINE_CKPT_NEW_LOG', -- https://www.sqlskills.com
/help/waits/WAIT_XTP_OFFLINE_CKPT_NEW_LOG
        N'WAIT_XTP_CKPT_CLOSE', -- https://www.sqlskills.com/help
/waits/WAIT_XTP_CKPT_CLOSE
        N'XE_DISPATCHER_JOIN', -- https://www.sqlskills.com/help/waits
/XE_DISPATCHER_JOIN
        N'XE_DISPATCHER_WAIT', -- https://www.sqlskills.com/help/waits
/XE_DISPATCHER_WAIT
        N'XE_TIMER_EVENT' -- https://www.sqlskills.com/help/waits
/XE_TIMER_EVENT
        )
    AND [waiting_tasks_count] > 0
    )
SELECT
    MAX ([W1].[wait_type]) AS [WaitType],
    CAST (MAX ([W1].[WaitS]) AS DECIMAL (16,2)) AS [Wait_S],
    CAST (MAX ([W1].[ResourceS]) AS DECIMAL (16,2)) AS [Resource_S],
    CAST (MAX ([W1].[SignalS]) AS DECIMAL (16,2)) AS [Signal_S],
    MAX ([W1].[WaitCount]) AS [WaitCount],
    CAST (MAX ([W1].[Percentage]) AS DECIMAL (5,2)) AS [Percentage],
    CAST ((MAX ([W1].[WaitS]) / MAX ([W1].[WaitCount])) AS DECIMAL
(16,4)) AS [AvgWait_S],
    CAST ((MAX ([W1].[ResourceS]) / MAX ([W1].[WaitCount])) AS
DECIMAL (16,4)) AS [AvgRes_S],
    CAST ((MAX ([W1].[SignalS]) / MAX ([W1].[WaitCount])) AS DECIMAL
(16,4)) AS [AvgSig_S],
    CAST ('https://www.sqlskills.com/help/waits/' + MAX ([W1].
[wait_type]) as XML) AS [Help/Info URL]
FROM [Waits] AS [W1]
INNER JOIN [Waits] AS [W2] ON [W2].[RowNum] <= [W1].[RowNum]
GROUP BY [W1].[RowNum]
HAVING SUM ([W2].[Percentage]) - MAX( [W1].[Percentage] ) < 95; --
percentage threshold
GO
```

4. In-depth Query Tuning

https://learning.oreilly.com/library/view/t-sql-querying/9780133986631/ch02.html