

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 capture 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 allow 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`.

III. Additional Memory Monitoring Tools

- 1. DBCC MEMORYSTATUS
- 2. Dynamic Management Views
 - a. `sys.dm_os_memory_brokers`
 - b. `sys.dm_os_memory_clerks`
 - c. `sys.dm_os_ring_buffers`
 - d. `sys.dm_db_xtp_table_memory_stats`
 - e. `sys.dm_xtp_system_memory_consumers`

IV. Memory Bottleneck Resolutions

- 1. Optimizing Application Workload
- 2. Allocating More Memory to SQL Server
- 3. Moving In-Memory Tables Back to Standard Storage
- 4. Increasing System Memory
- 5. Changing from a 32-Bit to a 64-Bit Processor
- 6. Compressing Data
- 7. Enabling 3GB of Process Address Space
- 8. Addressing Fragmentation

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

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].[SignalsS]) 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].[SignalsS]) / 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>