



# SQL Server Architecture

Module 2

## Learning Units covered in this Module

- Lesson 1: Introduction to SQL Operating System
- Lesson 2: SQL Server Waits and Queues
- Lesson 3: SQL Server Disk I/O

# Lesson 1: Introduction to SQL Operating System

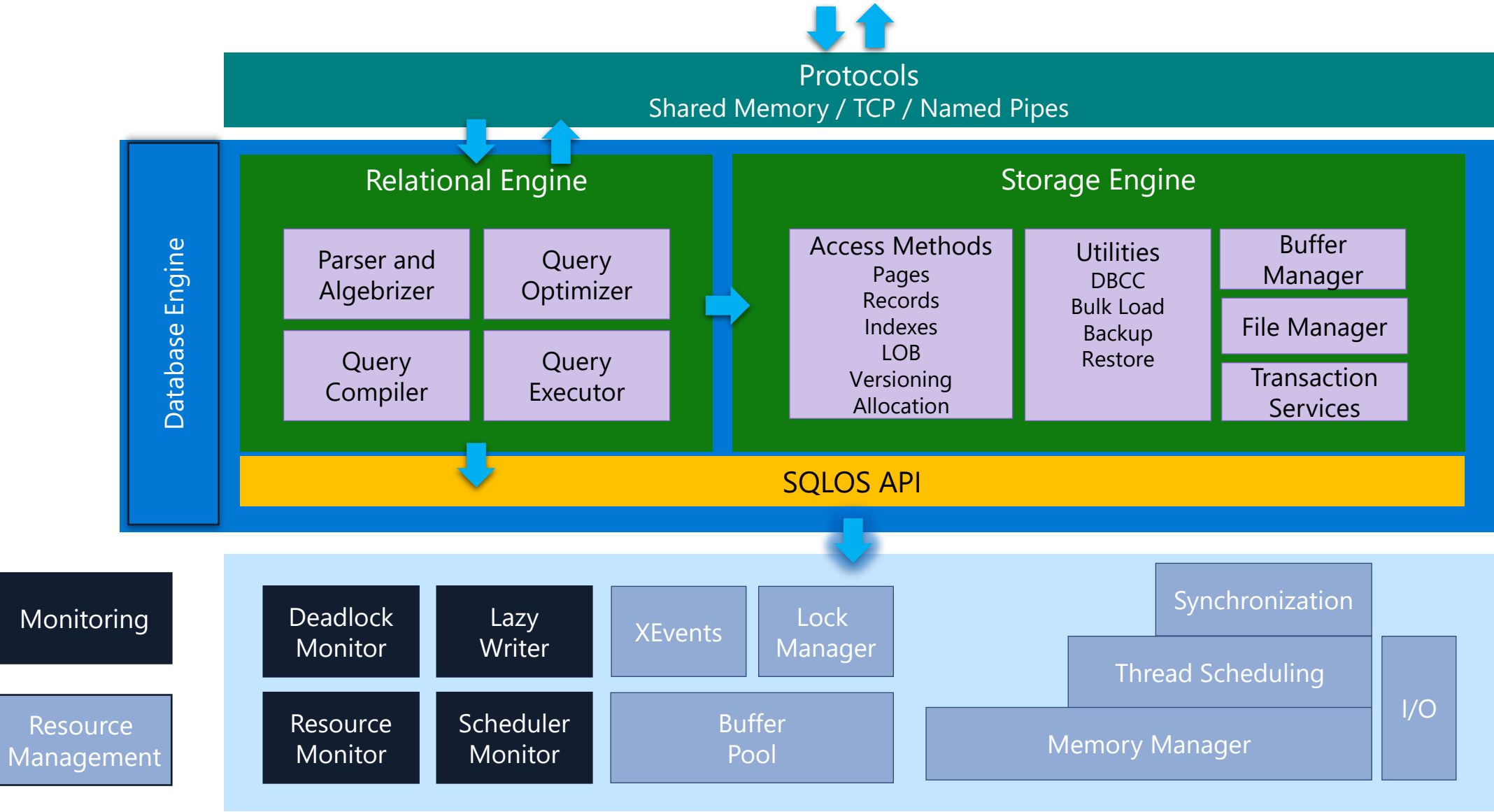
# Objectives

After completing this learning, you will be able to:

- Understand Microsoft SQL Server Architecture Overview.
- Learn SQL Server Operating System (SQLOS) Components.



# Inside the Database Engine



# Two Main Functions of SQLOS

## Management

- Memory Manager
- Process Scheduler
- Synchronization
- I/O
- Support for Non-Uniform Memory Access (NUMA) and Resource Governor

## Monitoring

- Resource Monitor
- Deadlock Monitor
- Scheduler Monitor
- Lazy Writer (Buffer Pool management)
- Dynamic Management Views (DMVs)
- Extended Events
- Dedicated Administrator Connection (DAC)

# Dynamic Management Views and Functions

Category	Description
sys.dm_exec_%	Execution and connection information
sys.dm_os_%	Operating system related information
sys.dm_tran_%	Transaction management information
sys.dm_io_%	I/O related information
sys.dm_db_%	Database information

# Using Dynamic Management Objects (DMOs)

- Must reference using the sys schema
- Two basic types:
  - Real-time state information
  - Historical information

```
SELECT cpu_count, hyperthread_ratio,  
       scheduler_count, scheduler_total_count,  
       affinity_type, affinity_type_desc,  
       softnuma_configuration, softnuma_configuration_desc,  
       socket_count, cores_per_socket, numa_node_count,  
       sql_memory_model, sql_memory_model_desc  
FROM sys.dm_os_sys_info
```



# SQL Server Configuration

## Processor Configuration Settings And Best Practices

### Affinity Mask

- Assigns CPUs for SQL Server use
- Set via `sp_configure` or `Alter Server Configuration`
- Only required in specific scenarios

### Max Degree of Parallelism (MAXDOP)

- Maximum number of processors that are used for the execution of a query in a parallel plan. This option determines the number of threads that are used for the query plan operators that perform the work in parallel.

### Cost Threshold for Parallelism

- Only queries with a cost that is higher than this value will be considered for parallelism
- Only required when dealing with excessive parallelism

### Max Worker Threads

- Number of threads SQL Server can allocate
- Recommended value is 0. SQL Server will dynamically set the Max based on CPUs and CPU architecture

# Demonstration

Dynamic Management Views



Questions?



## Lesson 2: Waits and Queues

# Objectives

After completing this learning, you will be able to:

- Understand SQL Server task scheduling.
- Understand Waits and Queues



# Microsoft SQL Server Scheduling Terminology

## Batch

- A statement or set of statements submitted to SQL Server by the user (a query), also referred to as a request
- Monitor with `sys.dm_exec_requests`

## Task

- A batch will have one or more tasks (aligns with statements)
- Monitor with `sys.dm_os_tasks`

## Worker Thread

- Each task will be assigned to a single worker thread for the life of the task
- Monitor with `sys.dm_os_workers`

# Hierarchy of Common Terms

```
SELECT *  
FROM sys.dm_exec_sessions;  
-- relevant data:  
-- session_id --> spid, memory address of the worker  
-- task_id --> batch_id, request_id, time slice, etc.  
-- waiting_time --> time spent in wait state.  
-- wait_type --> type of wait (e.g., IO, LCK, etc.)  
-- status --> Running, Sleeping, Suspended  
-- data_type, wait_time --> blocking information  
-- open_transaction_count --> blocking identification  
-- cpu_time, total_elapsed_time, reads, writes --> telemetry
```

Connection

Session

Request  
(Batch)

Task

Worker

# Scheduling Types

## Non-Preemptive (Cooperative)

- SQL Server manages CPU scheduling for most activity (instead of the operating system).
- SQL Server decides when a thread should wait or get switched out (known as yielding).
- SQL Server developers also programmed some predetermined voluntary yields to avoid starvation of other threads

## Preemptive

- Preemption is the act of an operating system temporarily interrupting an executing task.
- Higher priority tasks can preempt lower priority tasks.
- Preemptive mode used in SQL Server for external code calls, CLR with an UNSAFE assemblies, extended stored procedures



# SQL Server Task Scheduling

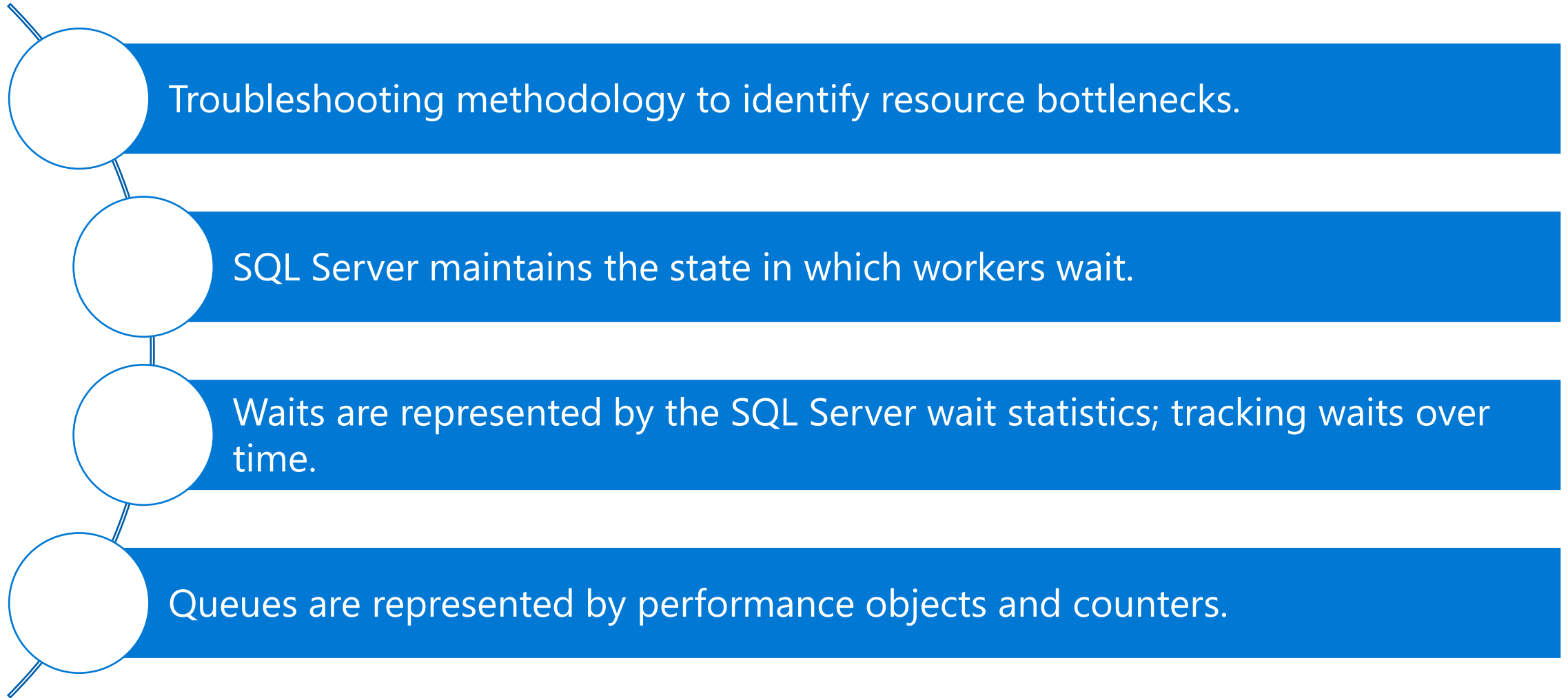
One SQLOS Scheduler  
per core/logical  
processor

Handles scheduling  
tasks, I/O, and  
synchronization of  
resources

Work requests are  
balanced across  
schedulers based on  
the number of active  
tasks

Monitor using  
`sys.dm_os_schedulers`

# Waits and Queues



## Thread States and Queues

**Runnable:** The thread is currently in the Runnable Queue waiting to execute. (First In, First Out).

**Running:** One active thread executing on a processor.

**Suspended:** Placed on a Waiter List waiting for a resource other than a processor. (No specific order).

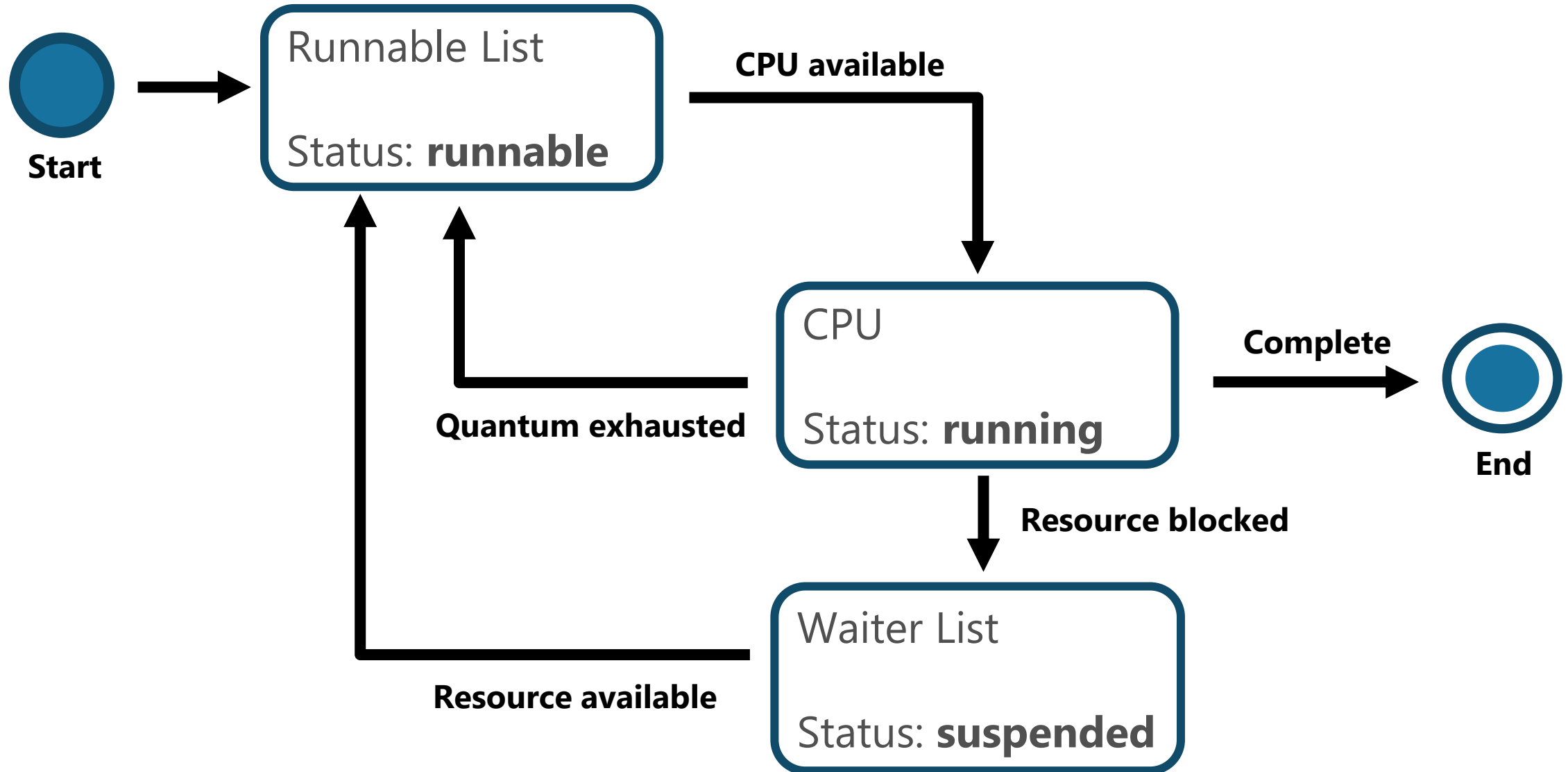
# Yielding

In SQL Server, each thread is assigned a quantum (duration 4ms), with SQL Server using a cooperative model to ensure its CPU resources are shared amongst all the threads that are in a runnable state, preventing the 'starving' condition of any individual thread.

By design, a worker owns the scheduler until it yields to another worker on the same scheduler.

When no worker is currently on the Runnable list, the **yielding worker is allowed another quantum** or performs the necessary idle scheduler maintenance.

# Yielding



# Task Execution Model

Status: Running

session_id 51	Running
---------------	---------



Runnable Queue (Signal Waits)

Status: Runnable

session_id 51	Runnable
session_id 64	Runnable
session_id 87	Runnable
session_id 52	Runnable
session_id 56	Runnable

SPID56 moved to the bottom of the Runnable queue.

Wait Queue (Resource Waits)

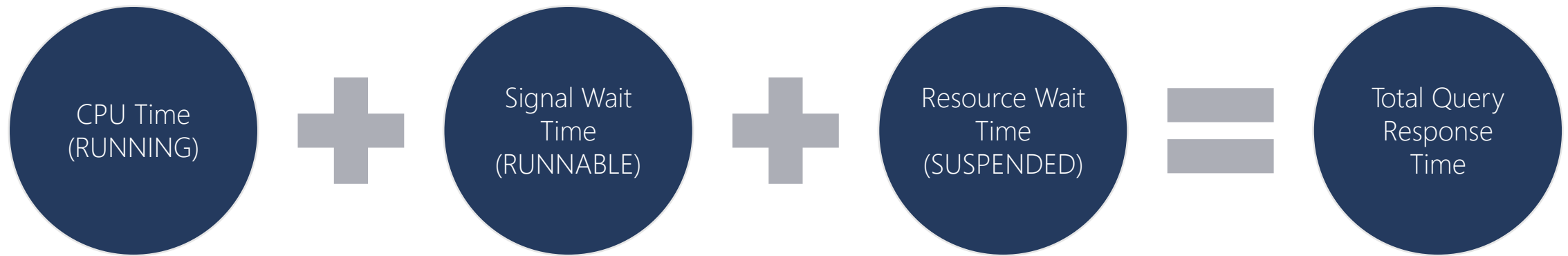
Status: Suspended

session_id 73	LCK_M_S
session_id 59	NETWORKIO
<del>session_id 56</del>	<del>Runnable</del>
session_id 55	RESOURCE_SEMAPHORE
session_id 60	IO_Completion



# Task Execution Model

- The full cycle between the several task states, for how many times it needs to cycle, is what we experience as the total query response time.



# Relevant Dynamic Management Views (DMVs)

## `sys.dm_os_wait_stats`

- Returns information about all the waits encountered by threads that ran.
- Includes wait type, number of tasks that waited in the specific wait type, total and max wait times, and the amount of signal waits.

## `sys.dm_os_waiting_tasks`

- Returns information about the wait queue of tasks actively waiting on some resource.

## `sys.dm_exec_requests`

- Returns information about each request that is in-flight.
- Includes session owning the request and status of the request, which will reflect the status of one or more tasks assigned to the request.



# Waiting Tasks DMV

```
SELECT w.session_id, w.wait_duration_ms, w.wait_type,
       w.blocking_session_id, w.resource_description,
       s.program_name, t.text, t.dbid, s.cpu_time, s.memory_usage
FROM sys.dm_os_waiting_tasks as w
     INNER JOIN sys.dm_exec_sessions as s
         ON w.session_id = s.session_id
     INNER JOIN sys.dm_exec_requests as r
         ON s.session_id = r.session_id
     OUTER APPLY sys.dm_exec_sql_text (r.sql_handle) as t
WHERE s.is_user_process = 1;
```

session_id	wait_duration_ms	wait_type	blocking_session_id	resource_description
58	8563	LCK_M_S	62	keylock hobtid=72057594047365120 dbid=5 id=lock1...

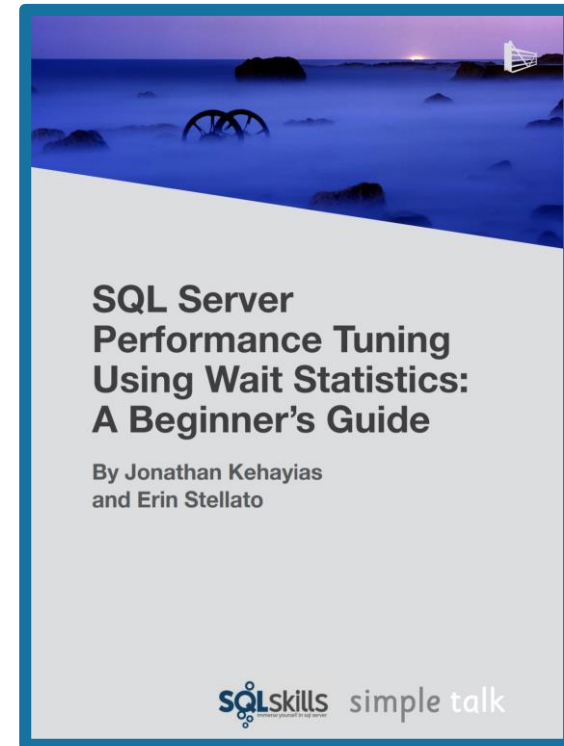
# Troubleshooting Wait Types

Aaron Bertrand – Top Wait Types

<https://sqlperformance.com/2018/10/sql-performance/top-wait-stats>

Paul Randal – SQL Skills Wait Types Library

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



# Lesson 3: SQL Server Disk I/O

# SQL Server disk I/O patterns:

## Data Files

- One .mdf file per database
- May have one or more .ndf file
- Random reads and writes
  - Read activity during backups; other activity varies depending on query activity and buffer pool size
  - Write activity during checkpoints, recovery, and lazy writes

## Log Files

- One\* .ldf file per database
- Sequential reads and writes
- Write activity during the log buffer flush operations
- Read activity during checkpoints, backups, and recovery
- Features such as database mirroring and replication will increase read and write activity

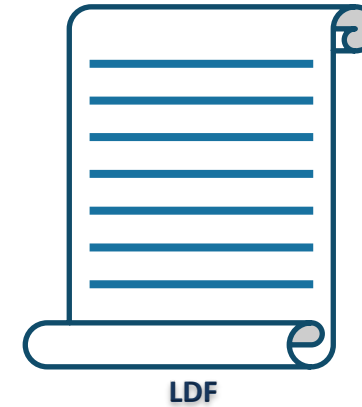
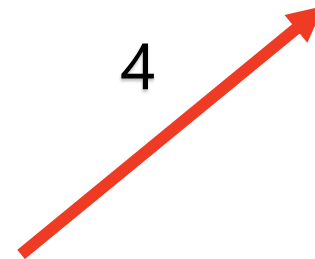
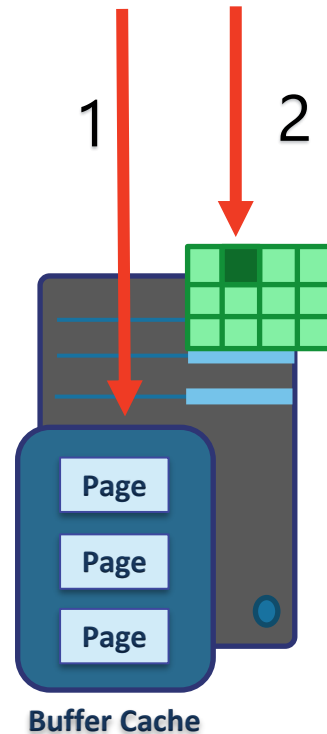
# SQL Server Disk I/O (Write-Ahead Logging)

```
UPDATE Accounting.BankAccounts  
SET Balance -= 200  
WHERE AcctID = 1
```

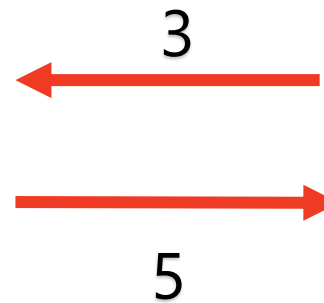
1. Data modification is sent to buffer cache in memory.

2. Modification is recorded in the log cache.

3. Data pages are located or read into the buffer cache and then modified.



4. Log cache record is flushed to the transaction log



5. At checkpoint, dirty data pages are written to the database file.

# Log Buffer Flushing

SQL Server will flush the log buffer to the log file

- SQL Server gets a commit request of a transaction that changes data.
- The log buffer fills up. (Max size 60kb.)
- SQL Server needs to harden dirty data pages (checkpoints)
- Manually request a log buffer flush using the `sys.sp_flush_log` procedure

Log buffer flushing results in a WRITELOG wait type.

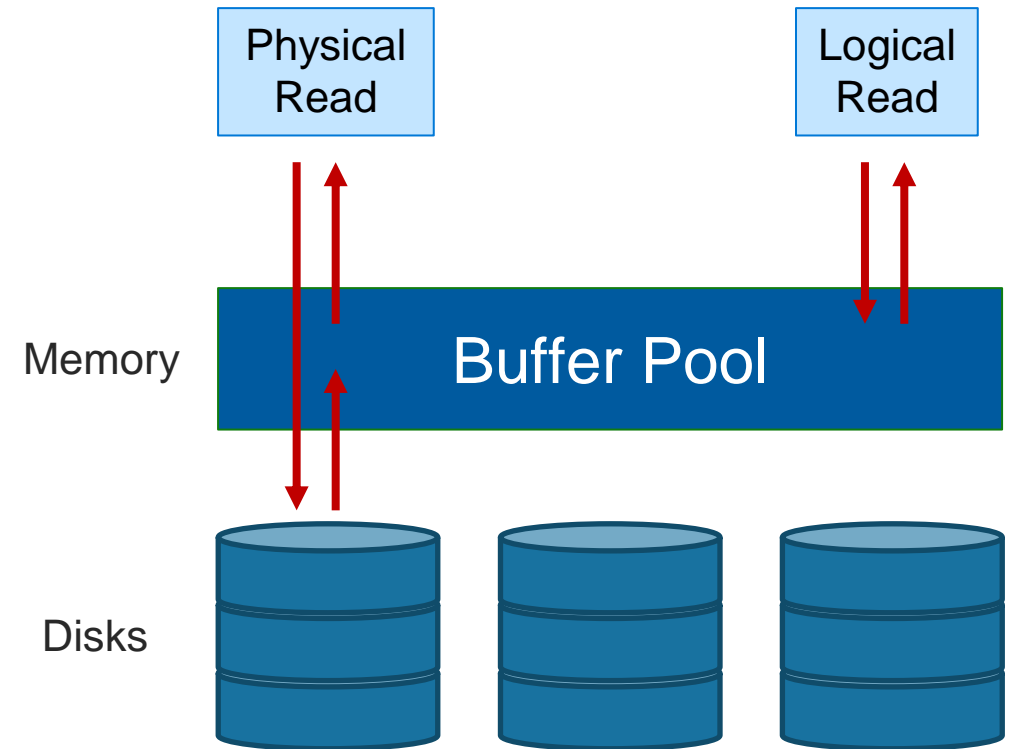
# SQL Server Buffer Pool

Stores 8 kilobytes (KB) pages of data to avoid repeated disk I/O.

- Pages held in the buffer until the space is needed by something else.

Lazy Writer searches for eligible buffers.

- If the buffer is dirty, an asynchronous write (lazy write) is posted so that the buffer can later be freed.
- If the buffer is not dirty, it is freed.



# SET STATISTICS IO

```
SET STATISTICS IO ON
GO
SET STATISTICS TIME ON
SELECT SOH.SalesOrderID, SOH.CustomerID,
OrderQty, UnitPrice, P.Name
FROM Sales.SalesOrderHeader AS SOH
JOIN Sales.SalesOrderDetail AS SOD
ON SOH.SalesOrderID = SOD.SalesOrderID
JOIN Production.Product AS P
ON P.ProductID = SOD.ProductID
SET STATISTICS IO, TIME OFF
```

Used to identify physical reads and logical reads for a query

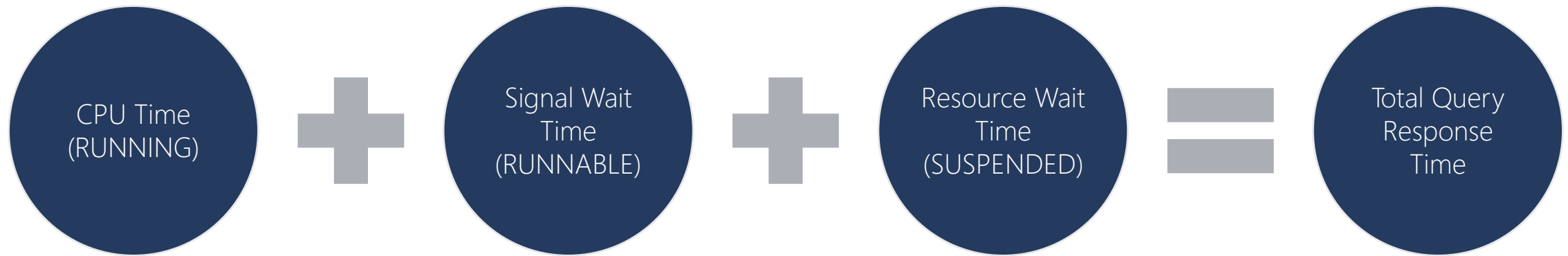
```
(121317 rows affected)
Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server r
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server
Table 'SalesOrderDetail'. Scan count 1, logical reads 428, physical reads 0, pag
Table 'Product'. Scan count 1, logical reads 15, physical reads 0, page server r
Table 'SalesOrderHeader'. Scan count 1, logical reads 57, physical reads 0, page

SQL Server Execution Times:
    CPU time = 94 ms,  elapsed time = 1653 ms.
```



# Total Query Response Time

- The full cycle between the several task states, for how many times it needs to cycle, is what we experience as the total query response time.



# Checkpoints

Flushes dirty pages from the buffer pool to the disk. Frequency of checkpoints varies based on the database activity and recovery interval.

**Automatic** (default) – Database engine issues checkpoints automatically based on the server level “recovery interval” configuration option

**Indirect** (new in SQL Server 2012) – Database engine issues checkpoints automatically based on the database level TARGET\_RECOVERY\_TIME

```
ALTER DATABASE [AdventureWorksPTO] SET TARGET_RECOVERY_TIME = 60 SECONDS
```

**Manual** – Issued in the current database for your connection when you execute the T-SQL CHECKPOINT command

**Internal** – Issued by various server operations

