The Most Prominent Wait Types of your SQL Server 2016

Sergio Govoni

@segovoni

www.sqlblog.com/blogs/sergio_govoni

PASS

PASS SQL saturday

#495 | PORDENONE 2016

Organizers

CONSORZIO
UNIVERSITARIO
DI PORDENONE

getlatestversion

inn0va
community

UGISS

PASS

Gold Sponsors

Microsoft

Real Comm
Easy for real

beanTech
IT moves your business

OverNet
EDUCATION

thinkIT®
Information Technology

Bronze Sponsors

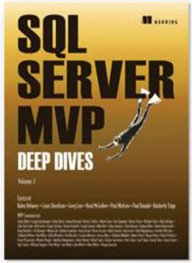
Speaker info: Sergio Govoni

SQL Server MVP

sqlblog.com/blogs/sergio_govoni

@segovoni

www.ugiss.org


SQL SERVER MVP

DEEP DIVES

manning.com/delaney

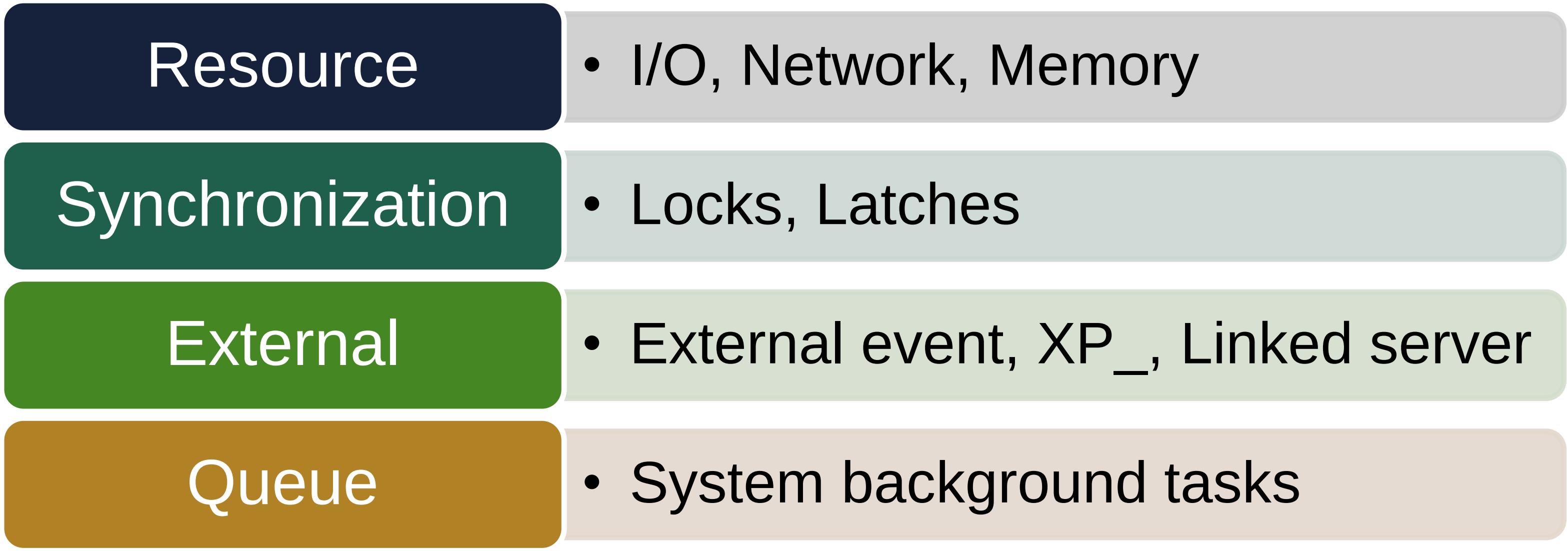
segovoni@gmail.com

Agenda

- Wait Types Fundamentals
  - What are they?
  - Where do you see them?
- Wait Types Scenario
  - PAGEIOLATCH_*
  - RESOURCE_SEMAPHORE
  - CXPACKET
  - LCK_M_*
  - SOS_SCHEDULER_YIELD

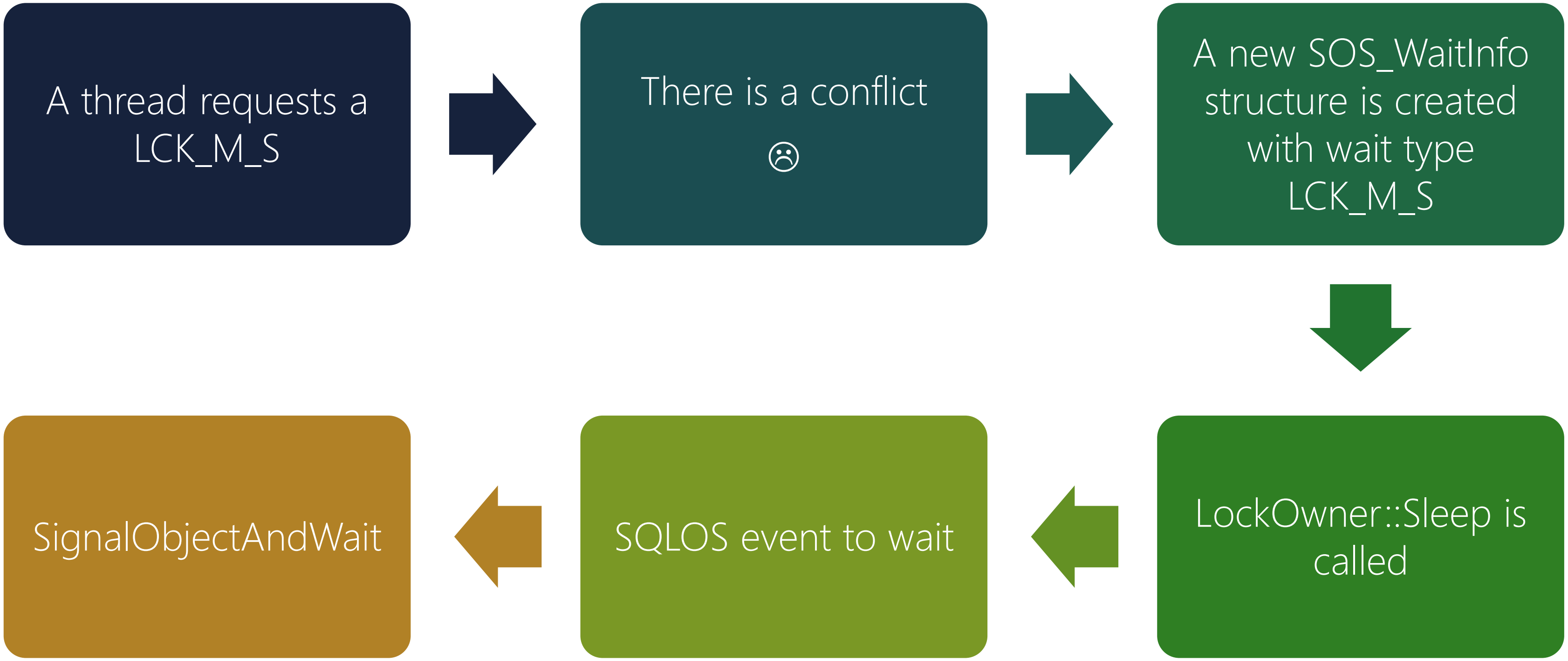
What is a Wait Type?

- It's a "tool" created to help us find bottlenecks
- Wait Types are recorded through a common mechanism



Resource	• I/O, Network, Memory
Synchronization	• Locks, Latches
External	• External event, XP_, Linked server
Queue	• System background tasks

How does wait type work?



A thread requests a LCK_M_S

There is a conflict ☹

A new SOS_WaitInfo structure is created with wait type LCK_M_S

LockOwner::Sleep is called

SQLOS event to wait

SignalObjectAndWait

Windbg

DEMO

Wait Statistics Analysis

- Wait Types and Statistics are stored into the DMVs
- Cumulative values since the last restart
- No deltas are available
  - You must do it by yourself
- Don't restart your SQL Server when you have performance problem
  - You will loose all information about Wait Types

The DMV sys.dm_os_wait_stats

- The sys.dm_os_wait_stats is the first place you have to go to check why SQL Server might have a performance bottleneck
- In SQL Server 2016 there are 831 reasons why a query can wait
  - 769 in SQL Server 2014
  - 653 in SQL Server 2012

The DMV sys.dm_os_wait_stats

- wait_type
  - Name of the wait type, wait reason
- wait_tasks_count
  - Number of waits for wait_type
- wait_time_ms
  - Total wait time for wait_type
- max_wait_time_ms
  - Maximum wait time for wait_type
- signal_wait_time_ms
  - Time spent in the runnable state for wait_type

Environment and Baseline

DEMO

The Most Prominent Wait Types

- PAGEIOLATCH_*
- RESOURCE_SEMAPHORE
- CXPACKET
- LCK_M_*
- SOS_SCHEDULER_YIELD
- ASYNC_NETWORK_IO

PAGEIOLATCH_*

- Occurs when SQL Server waits for an I/O request
- Symptom
  - Long waits may indicate problems with the I/O
  - Buffer Pool pressure
- Check for
  - Low Page Life Expectancy (PLE)
  - Bad indexing strategy: Scans
  - Combined with CXPACKET?

The Most Prominent Wait Types

- PAGEIOLATCH_*
- RESOURCE_SEMAPHORE
- CXPACKET
- LCK_M_*
- SOS_SCHEDULER_YIELD
- ASYNC_NETWORK_IO

RESOURCE_SEMAPHORE

- SQL Server waits when a query memory request can not be granted immediately due to other concurrent queries or memory pressure
  - sys.dm_exec_query_memory_grants
  - sys.dm_exec_query_resource_semaphores
- It is increased when a query waits in the "cost-based queue"
- High number of simultaneous requests

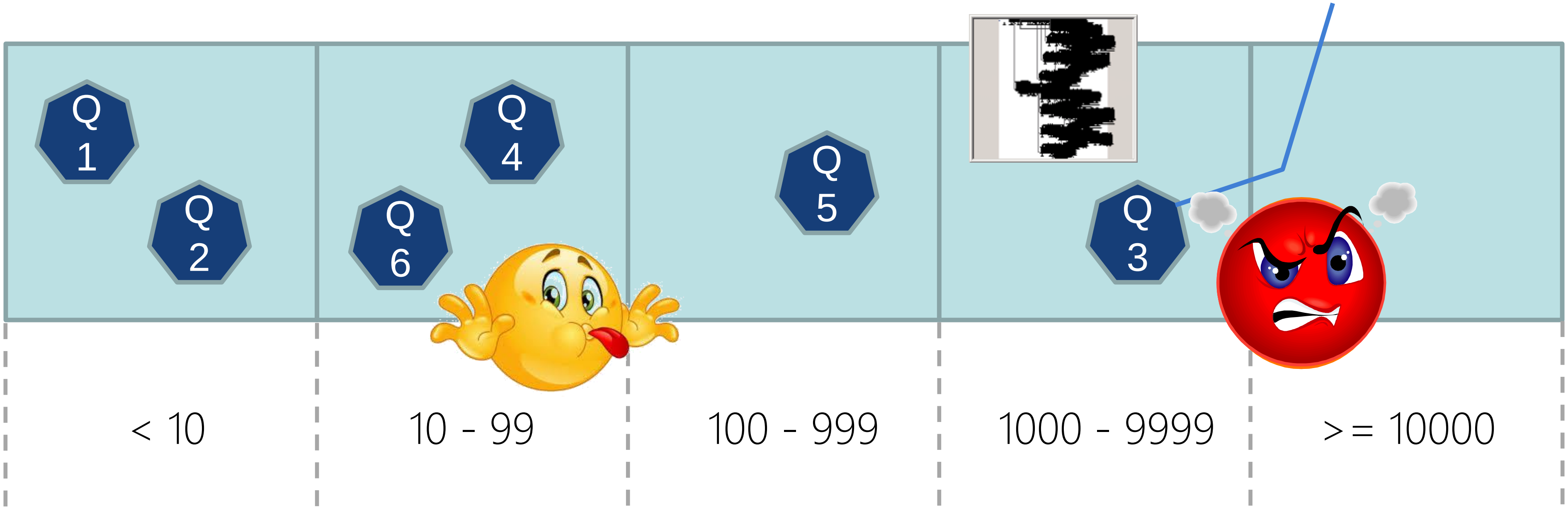
RESOURCE_SEMAPHORE

- Why a query needs more memory than the initial size?
  - Because the Execution Plan may contain Sort or Hash operations and these tasks require memory
  - Because the cardinality is not estimated correctly
- When the traffic-light is green the query can enter into the Memory Manager, when the traffic light is red (this traffic-light has only two states ☺), the query can only wait

RESOURCE_SEMAPHORE

The Cost-based Queue

Q3 is waiting the end of execution of Q1, Q2 and Q4.. and..



< 10

10 - 99

100 - 999

1000 - 9999

> = 10000

Cost factor: Subtree Cost

RESOURCE_SEMAPHORE

DEMO

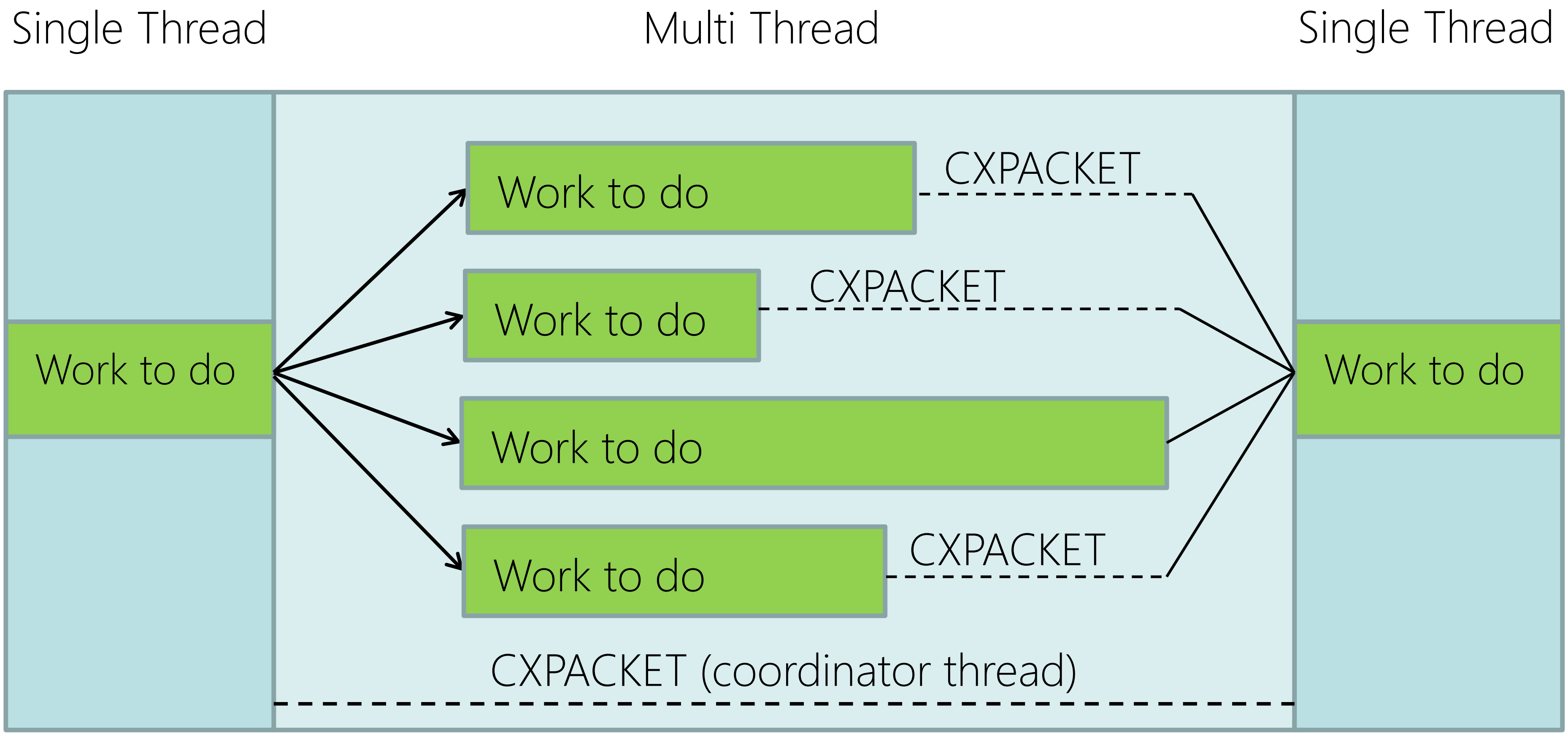
The Most Prominent Wait Types

- PAGEIOLATCH_*
- RESOURCE_SEMAPHORE
- CXPACKET
- LCK_M_*
- SOS_SCHEDULER_YIELD
- ASYNC_NETWORK_IO

CXPACKET

- What exactly is CXPACKET?
  - **C**lass E**X**change **Packet**s
- It occurs with Parallel Query Plans when trying to synchronize the query processor exchange iterator
- MAXDOP option
  - The default value is 0: Parallelism enabled
- Cost threshold for parallelism option
  - The default value is 5

CXPACKET


Single Thread
Multi Thread
Single Thread
Work to do
Work to do
CXPACKET
Work to do
CXPACKET
Work to do
Work to do
CXPACKET
CXPACKET (coordinator thread)
Work to do

CXPACKET

- Troubleshoot the underlying root cause!
- sys.dm_os_waiting_tasks
  - Shows the wait queue of tasks that are waiting (now) on some resource, includes the column wait_type
- Check for
  - Bad indexing strategy! Large scans?
  - Are the statistics updated?
  - Are there locking/blocking issues?
  - Is combined with PAGEIOLATCH_* ?

CXPACKET: Summary

- Parallelism in an OLTP system is not so good!
- MAXDOP
  - Set it to 1 **is not the "wizard solution"**
  - Limit it to the max core in the NUMA node
  - KB 2806535
- Cost threshold for parallelism
  - Set it to a much higher value than the default
  - Usually, values from 10 to 30 are good values
    - Test, test, test ☺

CXPACKET

DEMO

The Most Prominent Wait Types

- PAGEIOLATCH_*
- RESOURCE_SEMAPHORE
- CXPACKET
- LCK_M_*
- SOS_SCHEDULER_YIELD
- ASYNC_NETWORK_IO

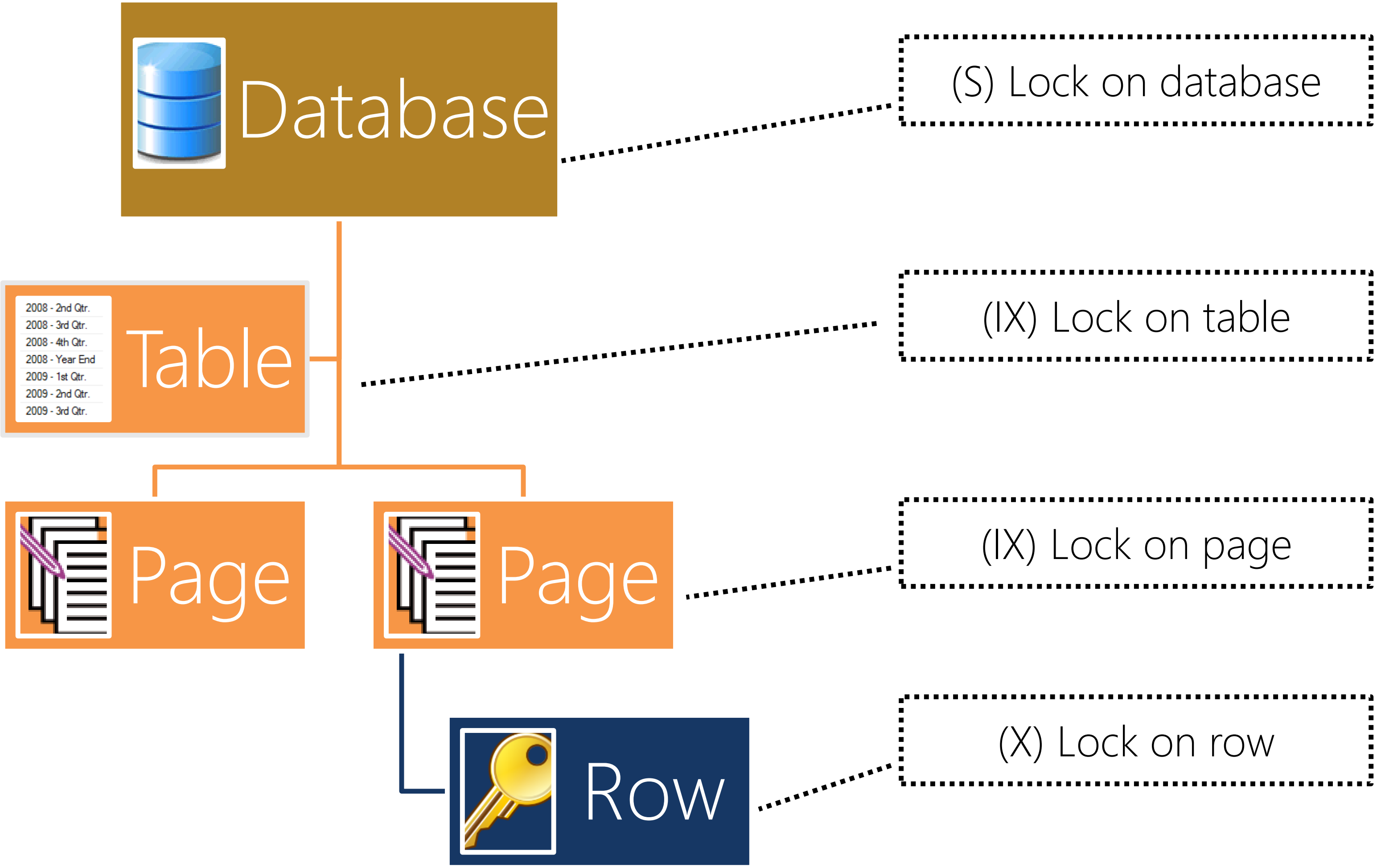
Lock Modes

- Shared (S)
  - Used for read operations that do not change data
- Update (U)
  - Reading with the intention to modify
- Exclusive (X)
  - Used for data-modification operations, no multiple updates to the same resource at the same time
- Intent (I)
  - Used to establish a lock hierarchy (IS, IX, SIX)

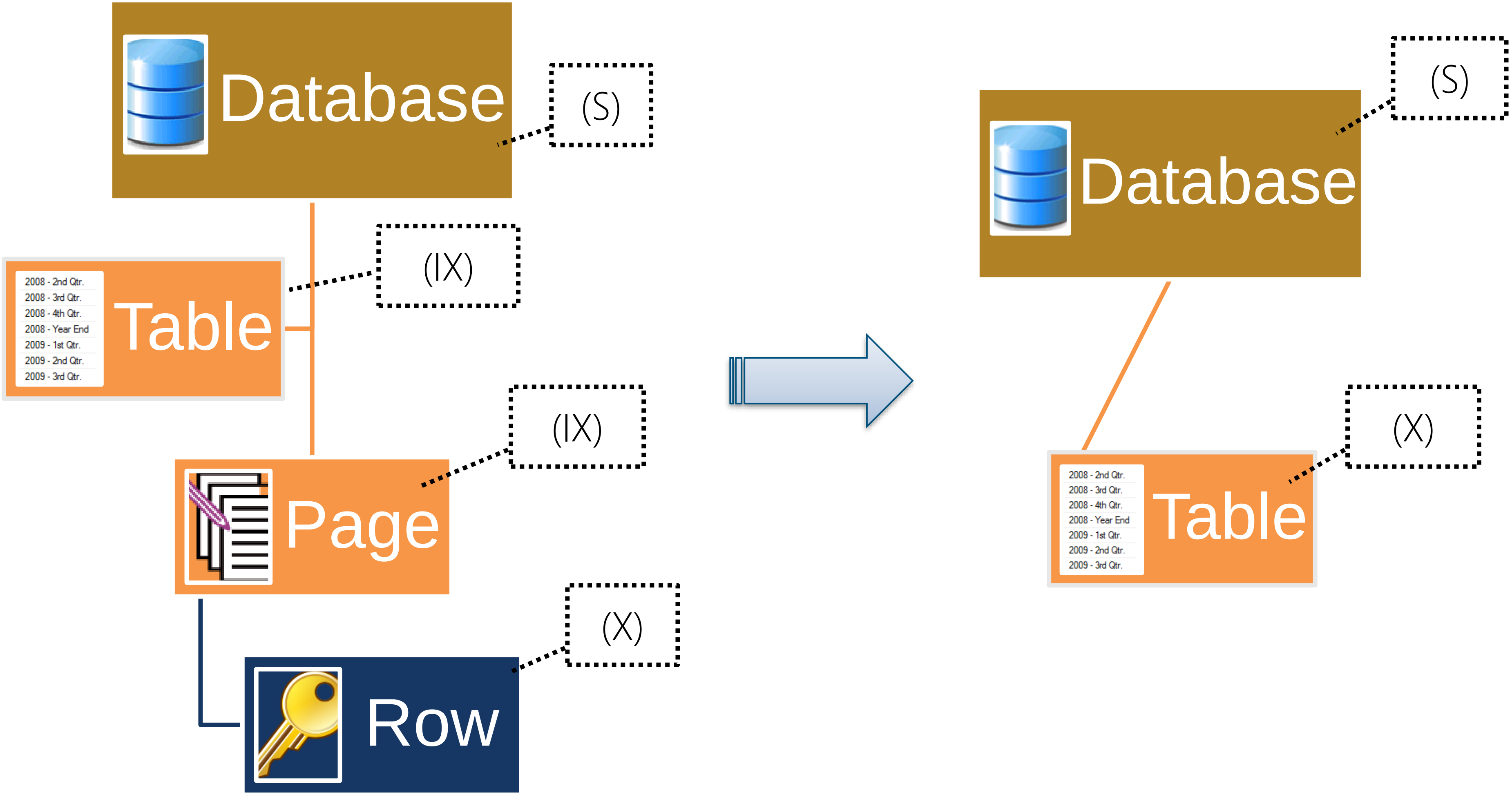
Lock Modes

Lock	Held		
Requested	Shared (S)	Update (U)	Exclusive (X)
Shared (S)	Allowed	Allowed	Wait
Update (U)	Allowed	Wait	Wait
Exclusive (X)	Wait	Wait	Wait

Lock Hierarchy



Lock Escalation



Lock Escalation

- “Lock escalation is the process of converting many fine-grain locks into fewer coarse-grain locks, reducing system overhead while increasing the probability of concurrency contention”

LCK_M_S

DEMO

Waits that are not waits

- MISCELLANEOUS
- Background task waits
  - LAZYWRITER_SLEEP
  - SQLTRACE_BUFFER_FLUSH
  - LOGMGR_QUEUE
  - CHECKPOINT_QUEUE
  - REQUEST_FOR_DEADLOCK_SEARCH
  - HADR_WORK_QUEUE
- CLR_AUTO_EVENT

Summary

- Every query can waits on SQL Server
  - Check the most prominent wait types in your workload
  - Troubleshoot the underlying root cause!
- Don’t restart your SQL Server if you have performance issue!
- Establish a baseline, it’s important!
- Make one change at a time.. and good luck ☺

Resources

- Tools
  - Benchmark Factory for Databases (freeware)
  - Performance Analysis of Logs (PAL) Tool
  - Debugging Tools for Windows (WinDbg, ...)
- Books and Whitepapers
  - SQL Server 2005 Waits and Queues
  - Performance Tuning with SQL Server DMVs

Q&A

- Questions?

#sqlsatPordenone
#sqlsat495
@segovoni

THANKS!

Thanks for attending this session. If you have questions, please post them to me (@segovoni) on twitter with the hashtag #sqlsat495