

Advanced SQL Server Troubleshooting



Klaus Aschenbrenner

Independent SQL Server Consultant

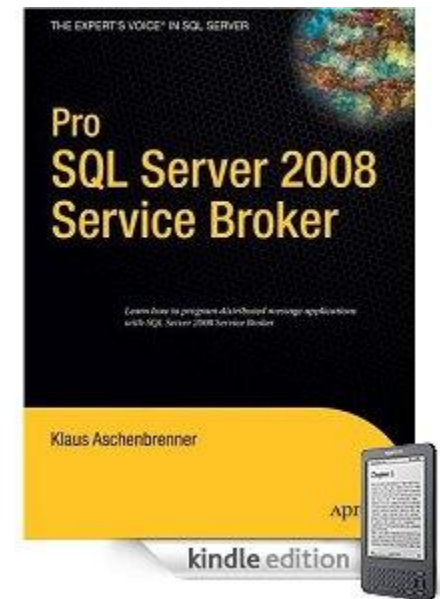
SQLpassion.at

Twitter: @Aschenbrenner

Bring your SQL Server installations to a new level of excellence!

About me

- Independent SQL Server Consultant
- International Speaker, Author
- „Pro SQL Server 2008 Service Broker“
- SQLpassion.at
- Twitter: @Aschenbrenner



Flightdeck Breitenlee

- Based on Microsoft Flight Simulator X
- 6 PCs in a network
- Around 2km cables
- Projection
 - Fully 180 degree curved project surface
 - 6 x 2m Display
 - 3 Beamers
 - 3072 x 768 Pixel
- Get your boarding pass here
 - <http://www.flightdeck-breitenlee.at>



Agenda

- Wait Statistics
- System Health
- Memory Issues
- I/O Issues
- Plan Cache Issues

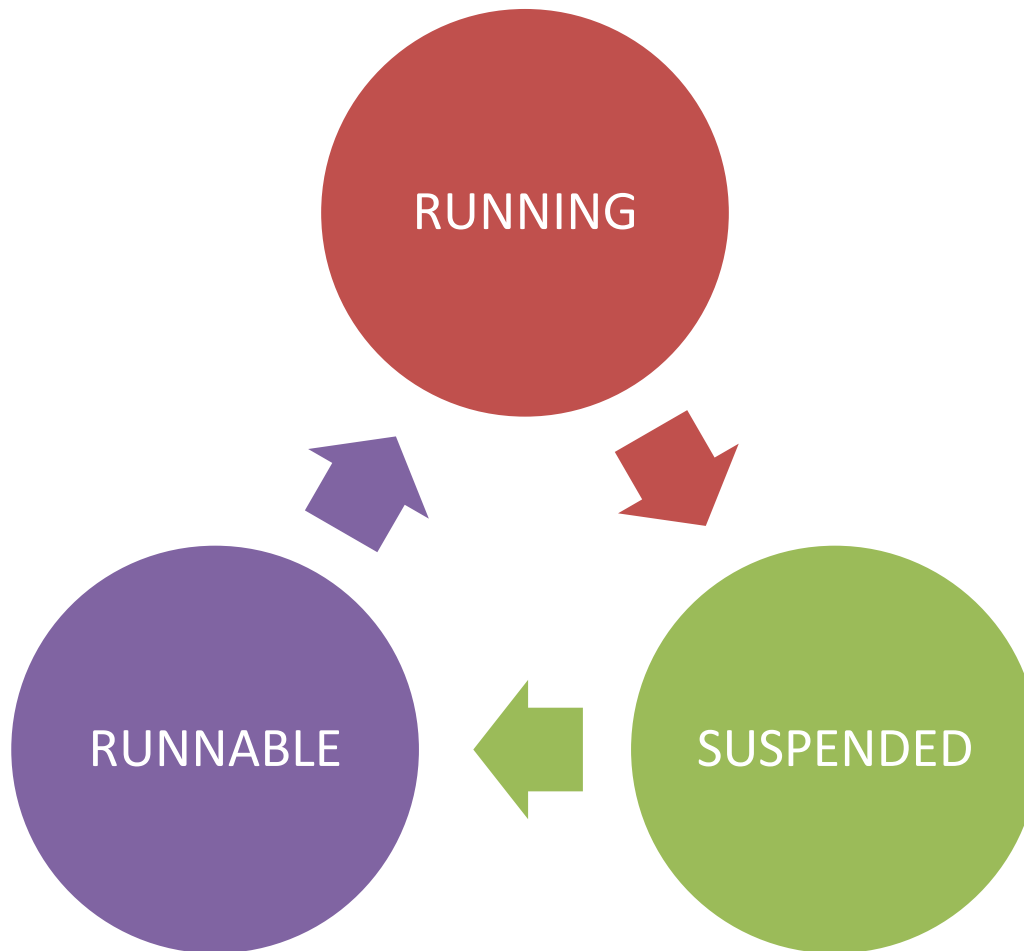
Agenda

- **Wait Statistics**
- System Health
- Memory Issues
- I/O Issues
- Plan Cache Issues

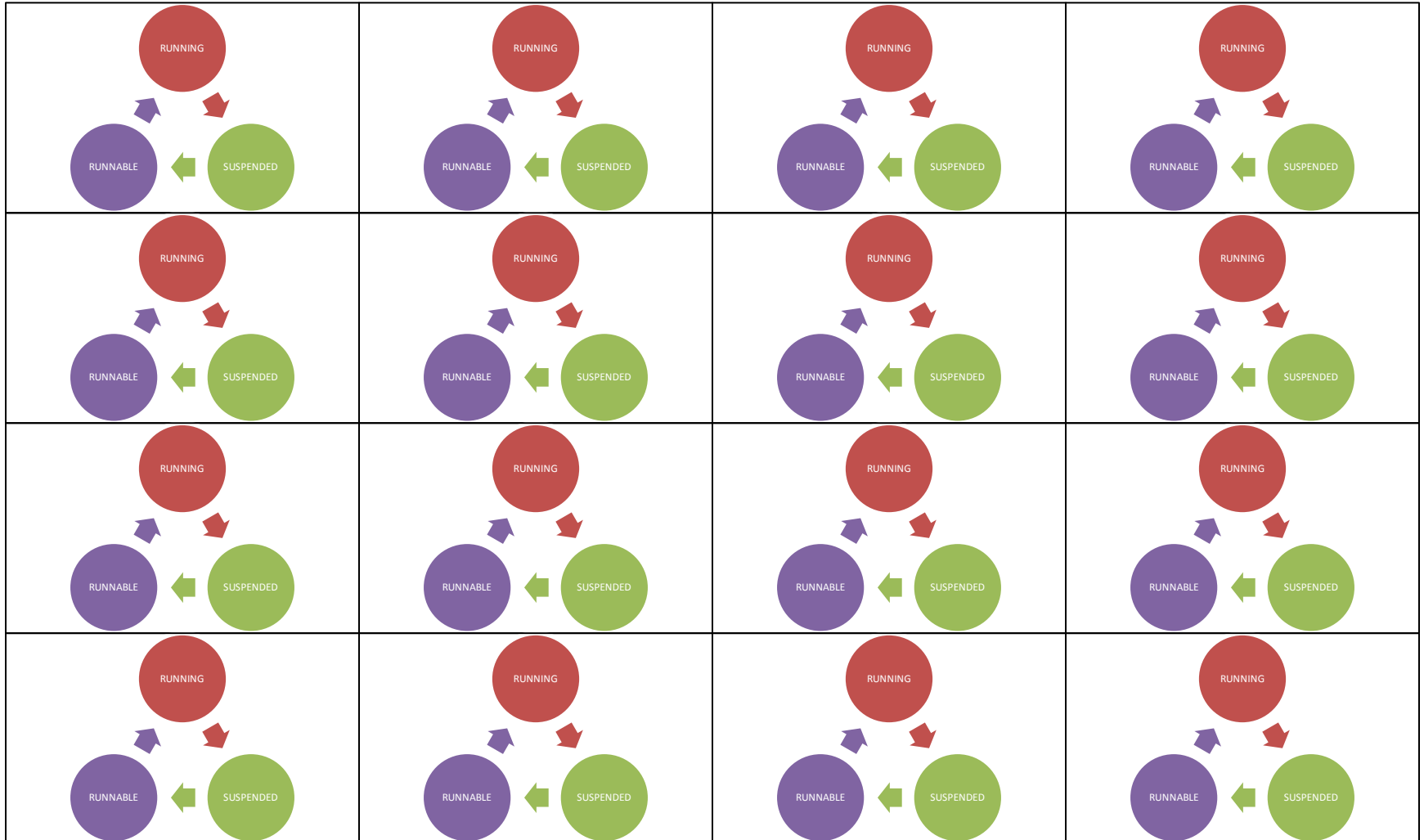
Wait Statistics

- Tracks wait information on the instance level
 - Wait Reason
 - Wait Time
- Waits occur ALWAYS because of
 - Cooperative Scheduling
 - Asynchronous Resource Waiting
- First place to check why SQL Server MIGHT have a performance bottleneck
 - `sys.dm_os_wait_stats`

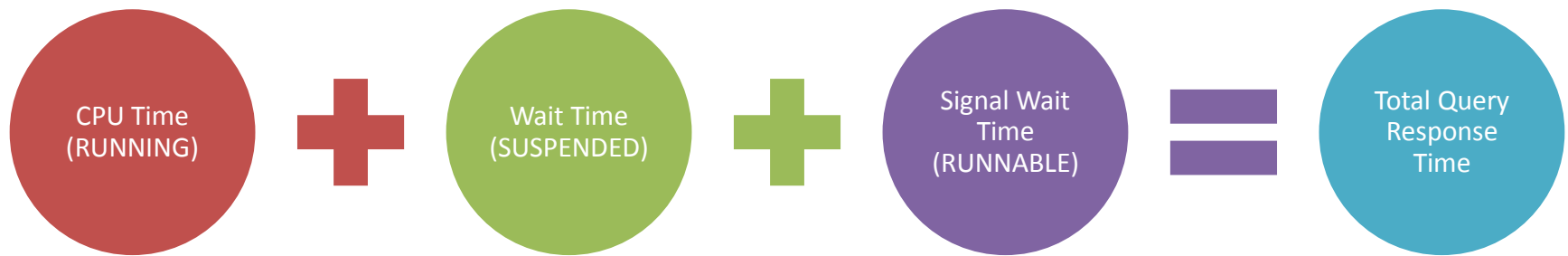
Query Lifecycle



Query Lifecycle – 16 CPUs



Total Query Response Time



sys.dm_os_wait_stats

- `wait_type`
 - Wait Reason
- `waiting_tasks_count`
 - Number of waits for this wait type
- `wait_time_ms`
 - Total wait time for this wait type
- `max_wait_time_ms`
 - Maximum wait time for this wait type
- `signal_wait_time_ms`
 - Time in RUNNABLE state for this wait type

sys.dm_os_waiting_tasks

- Shows waiter lists for the current executing requests
 - Snapshot of what is happening NOW
 - Includes the wait type (column „wait_type“)
- Join it together with
 - sys.dm_exec_requests
 - sys.dm_exec_sessions
 - sys.dm_exec_connections
 - sys.dm_exec_sql_text
 - sys.dm_exec_query_plan

Analyzing Wait Statistics

- Wait Statistics are cumulative since the last instance start
- There are no deltas available
- You must do it at your own
 - Capture Wait Statistics in a regular interval (with date information)
 - Store everything in a central table
 - Graph everything in Excel

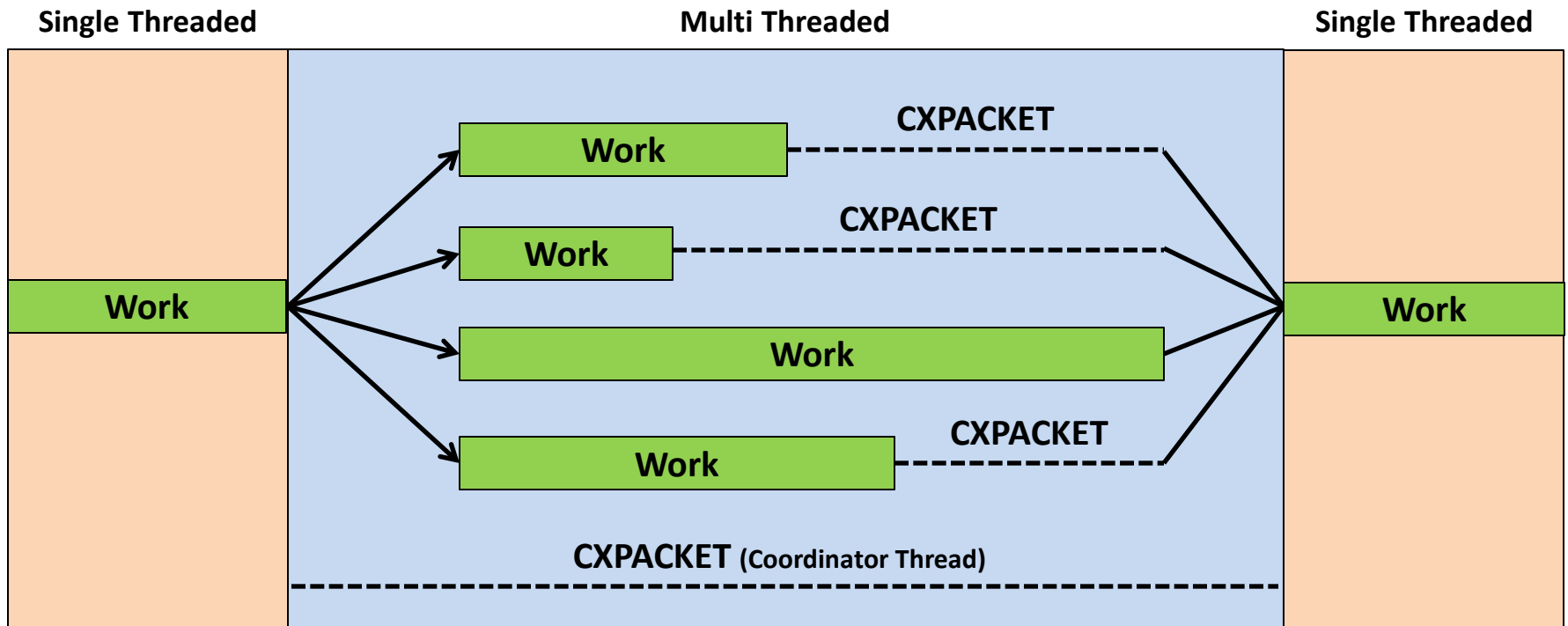
Most Prominent Wait Types

- CXPACKET
- ASYNC_NETWORK_IO
- SOS_SCHEDULER_YIELD
- WRITELOG
- PAGEIOLATCH_XX

CXPACKET

- Parallelism issue through Parallel Execution Plans
- Parallel threads are not given equal amount of work to do
- MAXDOP option
 - By default: 0 – parallelism allowed
 - Disable parallelism: 1 – only one CPU for a query
- Cost threshold for parallelism
 - By default: 5 – queries above that cost are executed with parallelism
 - Set a much higher cost threshold

CXPACKET



ASYNC_NETWORK_IO

- SQL Server is waiting until a client is consuming the retrieved data
- Poor client programming
 - Retrieving data in a loop, where row-by-row processing is done inside the loop
 - SELECT without any WHERE predicate (retrieving too much data)
- Poor network connection between SQL Server and the client applications

SOS_SCHEDULER_YIELD

- Indication of CPU pressure
- Cross check with sys.dm_os_schedulers
 - Column „runnable_tasks_count“ > 5 to 10
- Check for
 - Compilations/Recompilations?
 - Ad-hoc SQL statements (must be compiled EVERY time)?

WRITELOG

- Log management system waits for a log flush to the disk
- Indicates a problem with the I/O subsystem
- Cross check with `sys.dm_io_virtual_file_stats`
 - Column „io_stall_write_ms“
- Check for
 - Data and Log Files on separate physical disks?
 - Raid level of the physical disk (preferred RAID 10)?
 - Properly configured Auto Growth settings for the Log File?

PAGEIOLATCH_XX

- SQL Server waits for a data page to be read from disk into memory
- Indication
 - Problem with the I/O subsystem
 - Buffer Pool pressure – not enough memory for the workload
- Check for
 - Low Page Life Expectancy (PLE)?
 - Large Table/Index Scans (bad indexing)?
 - Combined with CXPACKET?

Demo

Wait Statistics

Agenda

- Wait Statistics
- **System Health**
- Memory Issues
- I/O Issues
- Plan Cache Issues

System Health

- `sys.dm_os_ring_buffers`
 - Contains notifications from various ring buffers
 - XML data
 - Must be converted through XQuery

RING_BUFFER_RESOURCE_MONITOR

- Contains information about memory status
- Incorrect ,max memory setting'
 - RESOURCE_MEMPHYSICAL_LOW
 - RESOURCE_MEMPHYSICAL_HIGH
- Should constantly contain
 - RESOURCE_MEMPHYSICAL_STEADY

RING_BUFFER_SCHEDULER_MONITOR

- Contains CPU utilization
 - By the SQL Server Process
 - Outside of SQL Server
- Notifications posted every minute
 - For the last 2 hours
- Useful for Multi-Instance Monitoring

RING_BUFFER_EXCEPTION

- Contains exceptions during query execution
- Join to sys.messages to get exact error message
- Can be used for further troubleshooting

RING_BUFFER_CONNECTIVITY

- Contains information about connection details
- Used to troubleshoot connectivity problems
 - Resource Governor Classifier Functions
 - Login Triggers
 - ThreadPool waits
 - SSPI Validation Delays

RING_BUFFER_MEMORY_BROKER

- Contains information about internal memory pressure
- There should be NO notifications
- Startup notification will exist
- Oldest event time
 - Last instance restart

RING_BUFFER_OOM

- Contains information about Out-Of-Memory exceptions
- There should be NO notifications

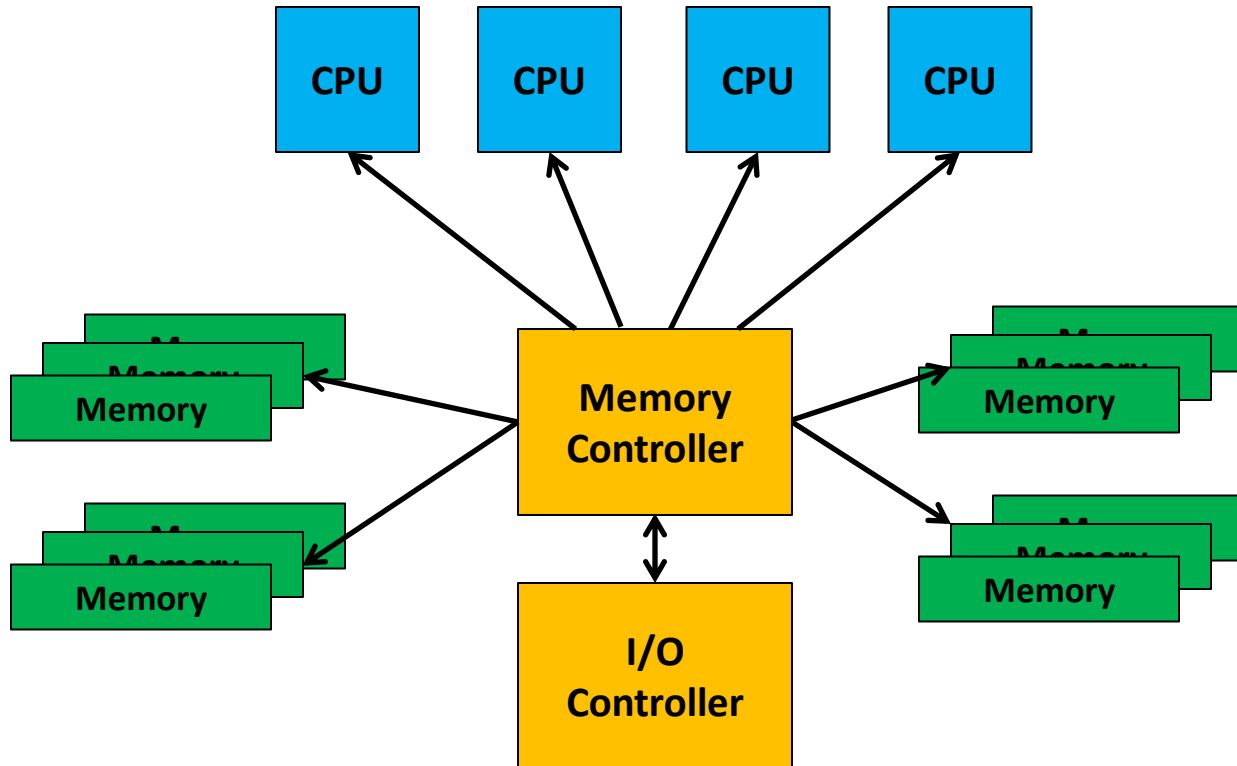
Demo

System Health

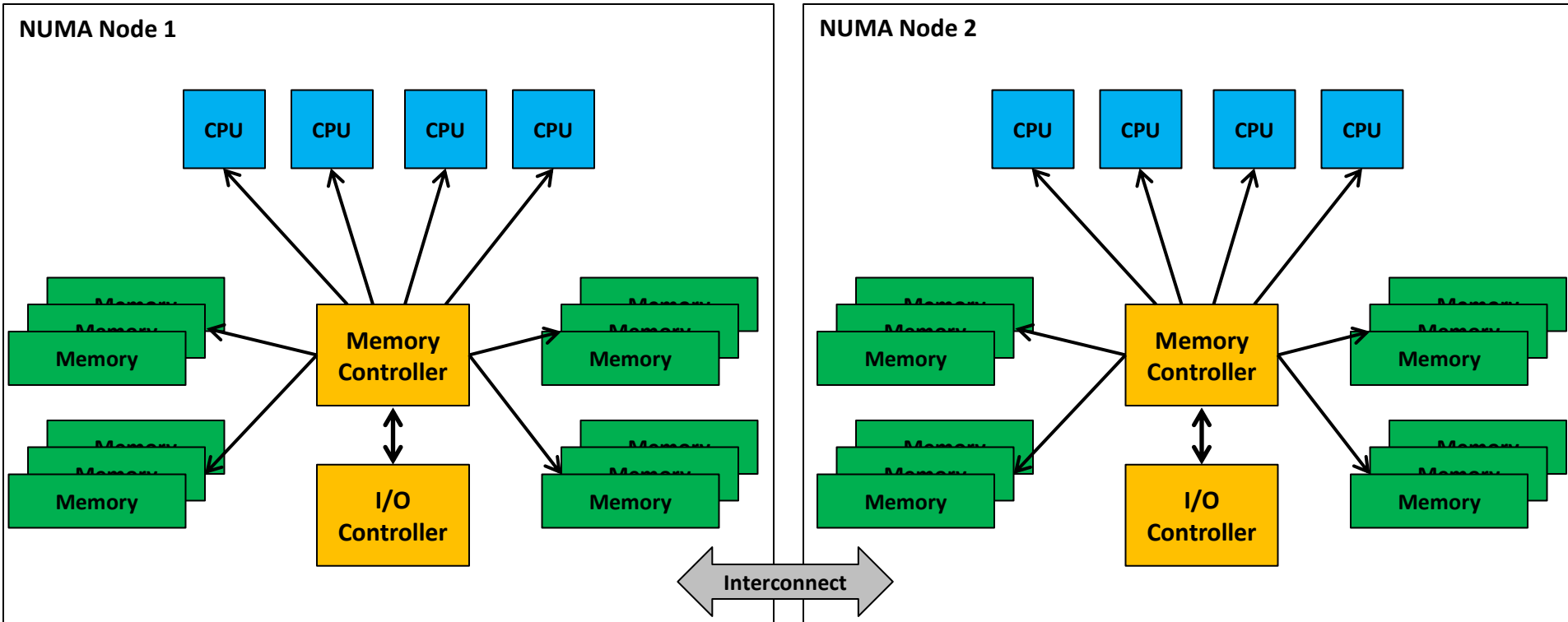
Agenda

- Wait Statistics
- System Health
- **Memory Issues**
- I/O Issues
- Plan Cache Issues

SMP System



NUMA System



Memory Nodes

- Depends on hardware architecture
- More nodes are available through NUMA
 - Non Uniform Memory Access
 - Each CPU has local memory attached
- Each node has its own
 - Memory Clerks
 - Memory Caches
- `sys.dm_os_memory_nodes`

Memory Clerks

- Each memory consumer **MUST** use a memory clerk to allocate some memory inside SQL Server
 - Track and control amount of allocated memory
- Major components have its own memory clerk
 - Buffer Pool: MEMORYCLERK_SQLBUFFERPOOL
 - Execution Plans: MEMORYCLERK_SQLQUERYPLAN
- `sys.dm_os_memory_clerks`

Buffer Pool

- Most memory is consumed by the buffer pool
- Consists of 8kb pages
- Controlled through
 - „Min Server Memory“ setting
 - „Max Server Memory“ setting
- Mostly data cache
- Single page allocations use Buffer Pool
 - Log Manager
 - Procedure Cache
 - „Stolen Pages“
- `sys.dm_os_buffer_descriptors`

Buffer Pool Allocation

- Buffer Pool stays between min and max server memory
- Commits and releases physical memory as needed
- Performance Counters
 - Memory Manager: Total Server Memory
 - Current Buffer Pool Size
 - Memory Manager: Target Server Memory
 - Max Server Memory setting

Min/Max Server Memory Settings

- Controls the size of the Buffer Pool
 - Single Page Allocations are taken from the BP
 - Multi Page Allocations are done outside of the BP
- Min Server Memory Setting
 - Floor value
- Max Server Memory Setting
 - Ceiling value
- Recommendation
 - 2 GB reservation for OS
 - 10% reservation for OS
 - What ever comes first

Query Memory

- Temporarily stores results during
 - Hash operations
 - Sort operations
- Allocated from the Buffer Pool
- Managed dynamically
- Has its own Memory Clerk
 - MEMORYCLERK_SQLQERESERVATIONS
- Query memory must be granted before execution
 - `sys.dm_exec_query_memory_grants`
- Can be controlled through Resource Governor

Hash/Sort Warnings

- Traceable through SQL Server Profiler
- Generated when Hash/Sort operations are not fit into memory
- Spilled to disk
 - TempDb
 - Physical I/O involved
 - Performance decreases!

Demo

Hash/Sort Warnings

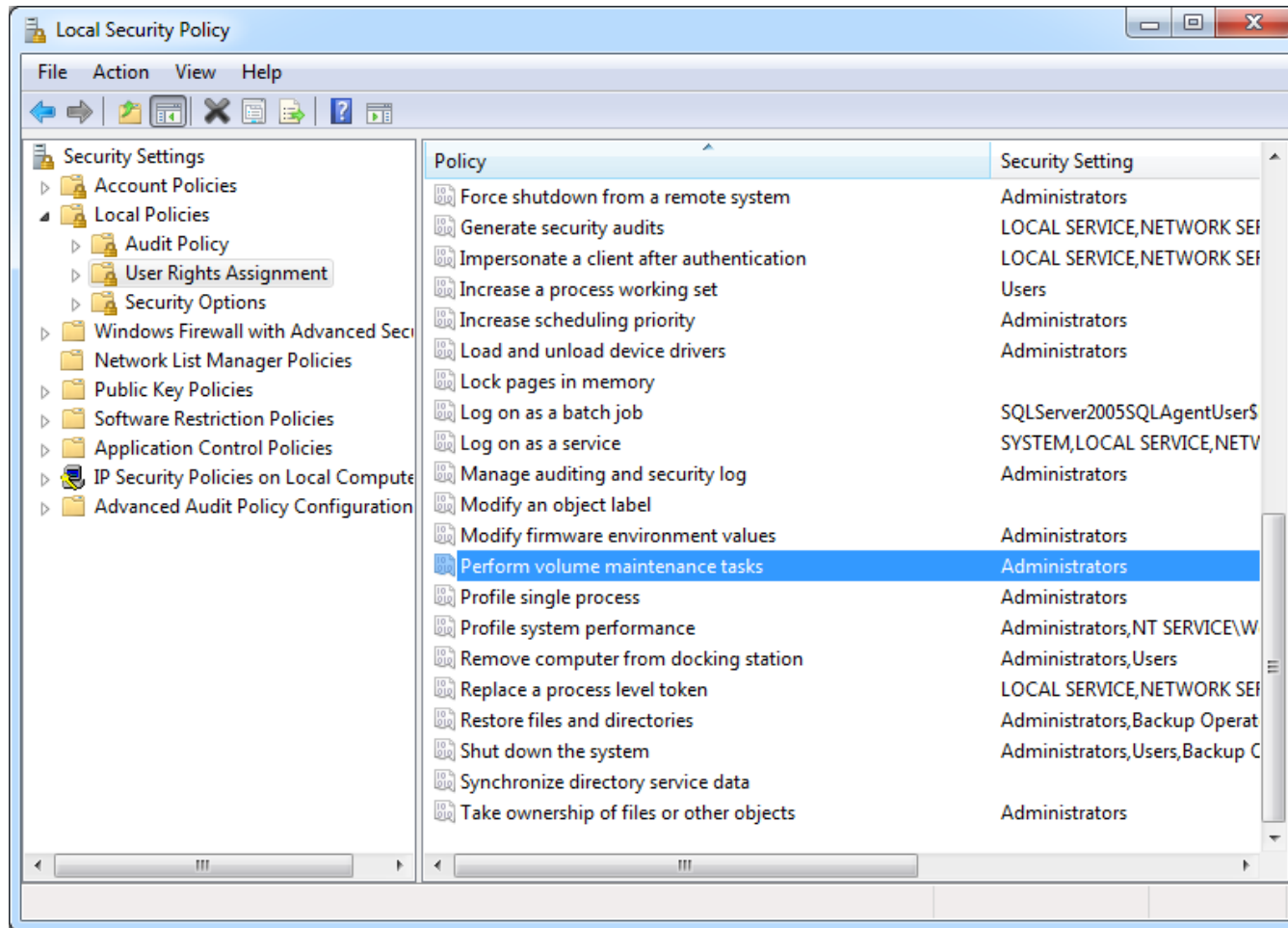
Agenda

- Wait Statistics
- System Health
- Memory Issues
- **I/O Issues**
- Plan Cache Issues

Instant File Initialization

- Data- and Log files are initialized with zero values when
 - Creating a database
 - Adding files, log or data, to an existing database
 - Increasing the size of an existing file (incl. Autogrowth!)
 - Restoring a database or file group
- Instant File Initialization
 - File Initialization is done **WITHOUT** writing the zero values to the file
 - Available on Windows Server 2003 onwards for data files only
 - Disabled by default
 - Needs special permission „Perform volume maintenance tasks“ in secpol.msc
 - SQL Server service must be restarted

secpol.msc



IFI & Log Files

- Instant File Initialization not supported for Log Files
- Log Files are always initialized with 0...
 - During creation
 - During AutoGrowth
 - Takes some time
- Multiple Log Files across physical disks
 - Not recommended, no performance improvement

Time Comparison

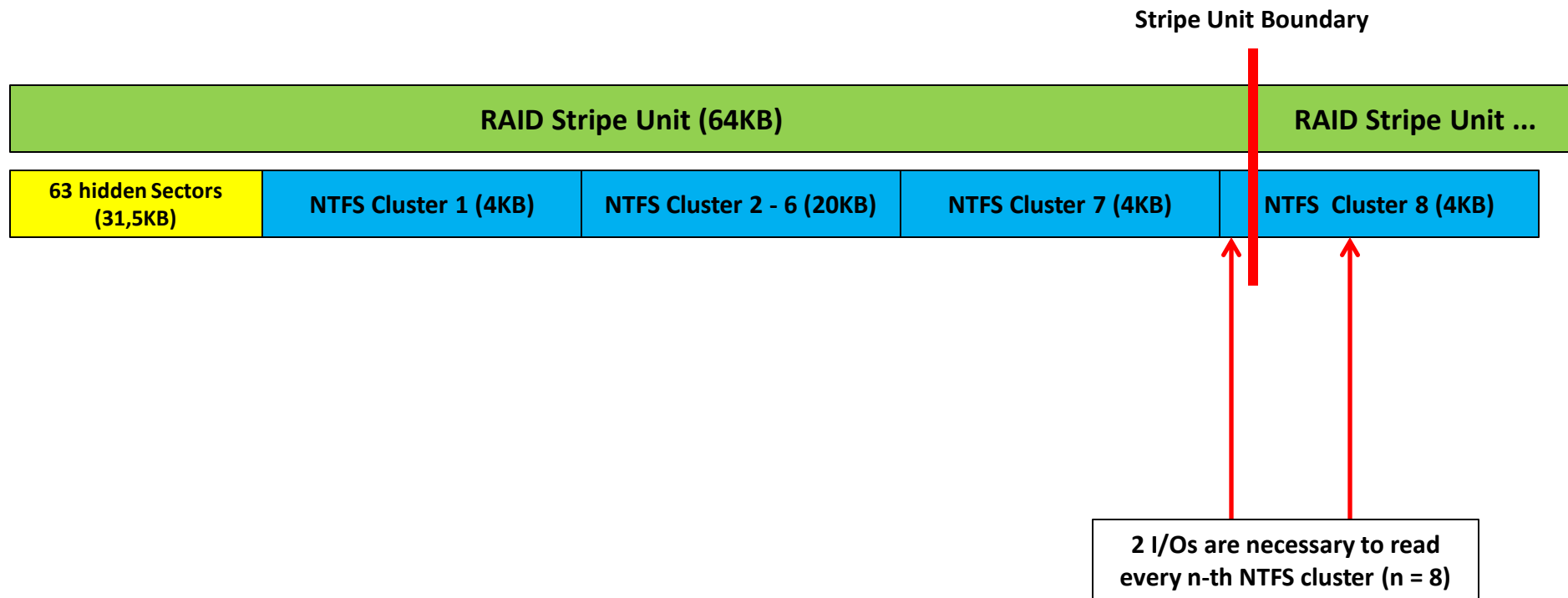
- Benchmarking was done on a SSD

	Without IFI	With IFI
1 GB	0:05min	153ms
10 GB	0:50min	171ms
50 GB	4:04min	1072ms

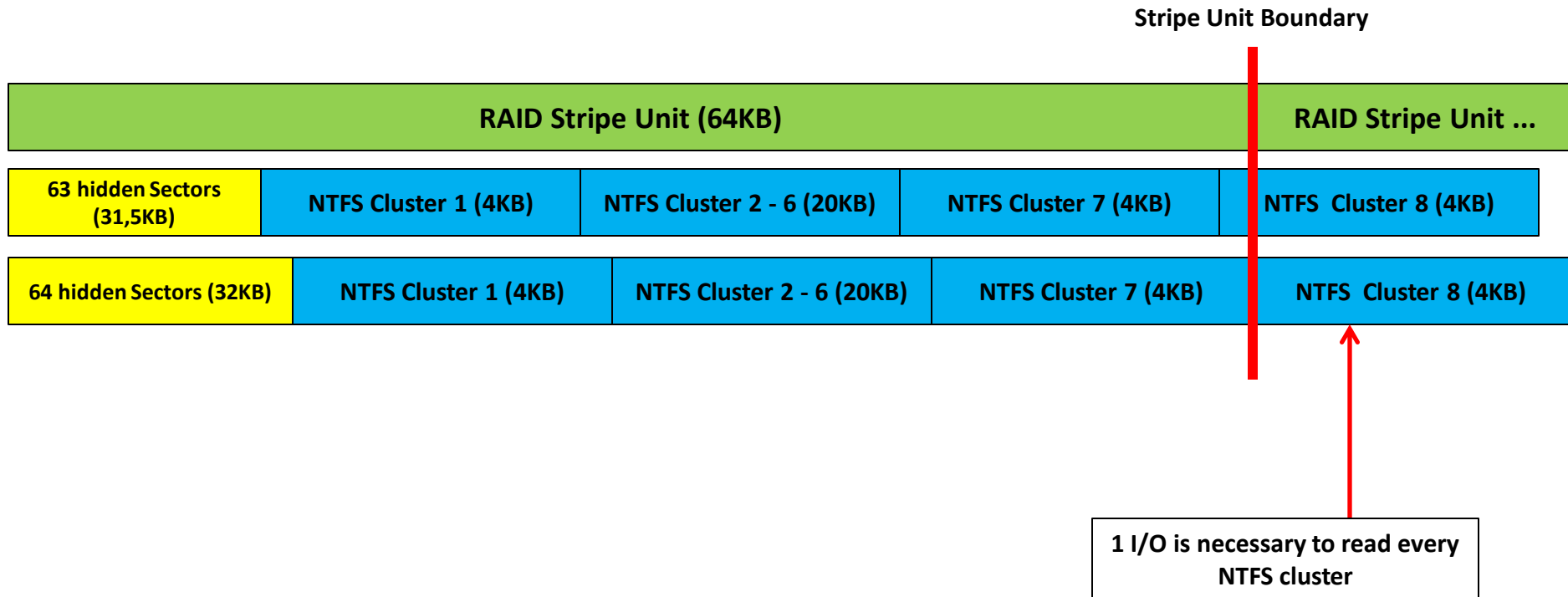
Disk Partition Alignment

- Can improve SQL Server IO performance up to 100%
 - RAID Stripe Unit is misaligned with NTFS Cluster Units
 - Accessing an NTFS Cluster Unit needs access to more than one RAID Stripe Unit
- Before Windows Server 2008
 - 63 hidden Sectors ($63 * 512 \text{ bytes} / 1024 = 31,5\text{KB}$)
 - Default Offset: 31,5KB
- Windows Server 2008 and higher
 - Default Offset: 1MB
 - Alignment by default

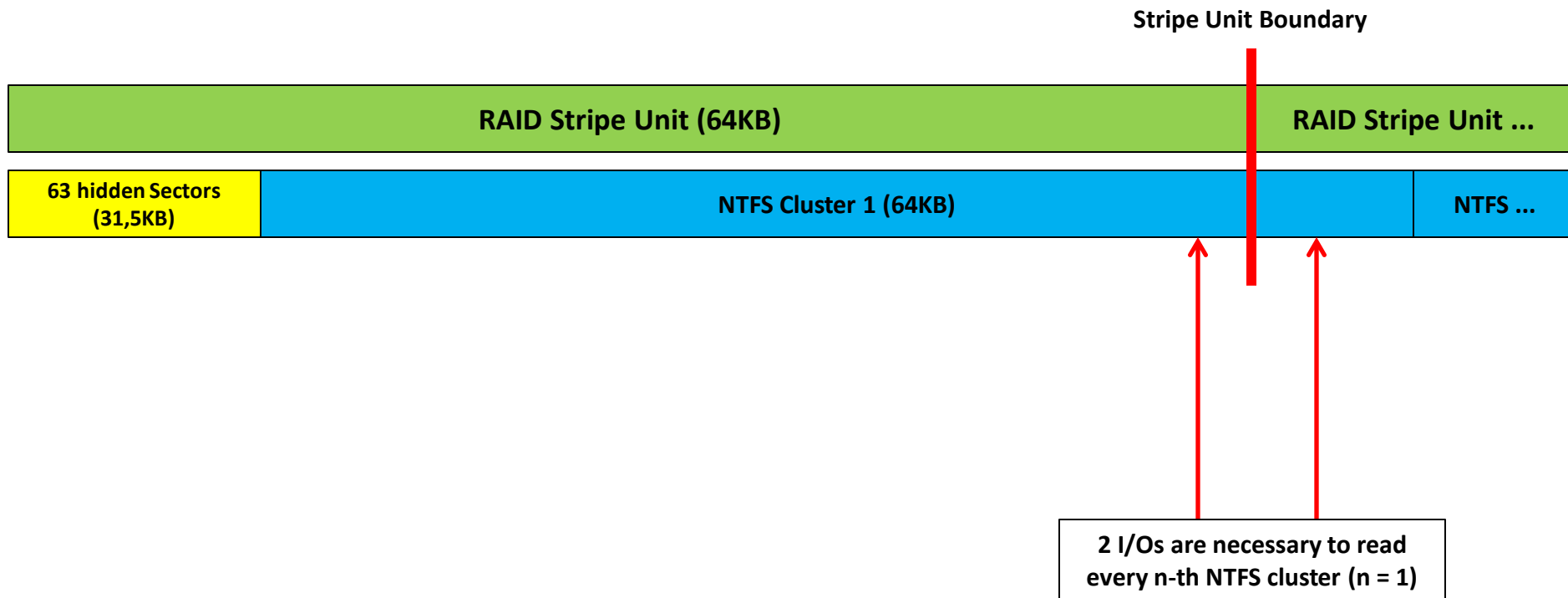
Misalignment – 4KB



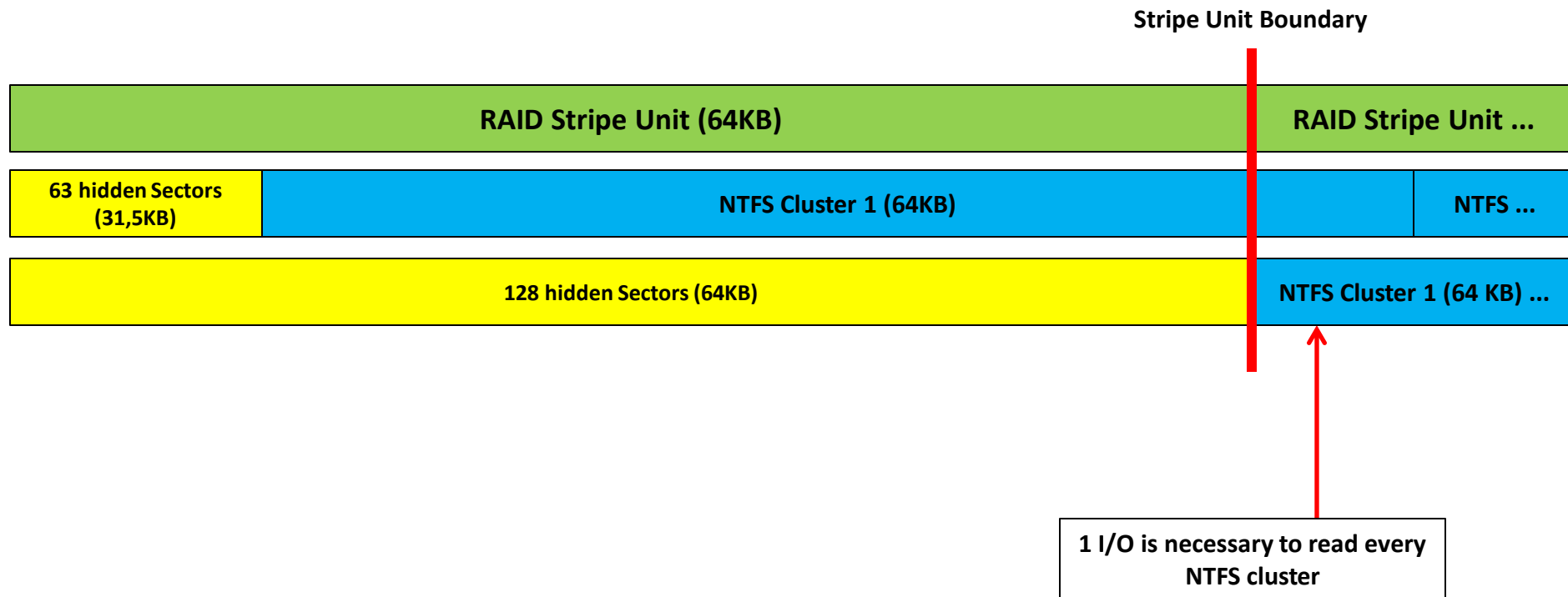
Alignment – 4KB



Misalignment – 64KB



Alignment – 64KB



Disk Partition Alignment

- Must be done
 - Before the disk is formatted
 - Before any data is stored on the disk
- Depends on the OS
 - Windows Server 2000: diskpar.exe
 - Windows Server 2003: diskpart.exe
 - Windows Server 2008 (R2): automatic
- Recommendation
 - RAID Stripe Size: 64KB (1 Extent)
 - NTFS Cluster Size: same as RAID Stripe Size
 - Partition_Offset / Stripe_Unit_Size must be a whole integer, like
 - $1048576 / 65536 = 16$

Disk Partition Alignment

- Further information
 - <http://blogs.msdn.com/b/jimmymay/archive/2008/10/14/disk-partition-alignment-for-sql-server-slide-deck.aspx>
 - <http://blogs.msdn.com/b/jimmymay/archive/2008/11/04/disk-partition-alignment-sector-alignment-for-sql-server-part-2-adding-hp-eva-8000-to-veritas-enterprise-administrator-track-alignment-settings-dialog.aspx>
 - <http://blogs.msdn.com/b/jimmymay/archive/2008/11/25/disk-partition-alignment-sector-alignment-for-sql-server-part-3-pass-2008.aspx>
 - <http://blogs.msdn.com/b/jimmymay/archive/2008/12/04/disk-partition-alignment-sector-alignment-for-sql-server-part-4-essentials-cheat-sheet.aspx>

Demo

Disk Partition Alignment

Agenda

- Wait Statistics
- System Health
- Memory Issues
- I/O Issues
- **Plan Cache Issues**

Adhoc Query Caching

- Each unique query gets cached
 - Only reused for the identical query
 - Exact text match necessary
- `sys.dm_exec_cached_plans`
 - `cacheobjtype` „Compiled Plan“
 - `objtype` „Adhoc“

Auto Parametrization

- Safe Plans can be reused
- SQL Server parametrizes them automatically
- Statistics are used to determine if a plan is safe
- Each individual query gets also cached
 - „Shell Query“
 - Cached to make it easier to find the parametrized version
- `sys.dm_exec_cached_plans`
 - `cacheobjtype` „Compiled Plan“
 - `objtype` „Prepared“

Restrictions

- JOIN
- IN
- BULK INSERT
- UNION
- INTO
- DISTINCT
- TOP
- GROUP BY, HAVING, COMPUTE
- Sub Queries
- ...

Forced Parametrization

- Database Option
 - ALTER DATABASE <db_name> SET PARAMETERIZATION FORCED
- Forces Auto Parametrization
 - Constants are treated as parameters
 - Plans are considered as safe... are they?
- Only a few exceptions
 - INSERT ... EXECUTE
 - Prepared Statements
 - RECOMPILE
 - COMPUTE
 - ...

Optimize for Adhoc Workloads

- Available on SQL Server 2008 and higher
- Server Option
- Adhoc Query Plans are not cached on the first use
 - Stub is put into the Plan Cache (~ 344 bytes)
 - On subsequent reuse the whole Execution Plan is cached
- Better Memory Management
- 2nd Recompile necessary!
- `sys.dm_exec_cached_plans`
 - `cacheobjtype` „Compiled Plan Stub“
 - `objtype` „Adhoc“

Query Hash Analysis

- Exposed through `sys.dm_exec_query_stats`
 - `query_hash`
 - `query_plan_hash`
- Can be used to determine if Forced Parametrization should be enabled, or not
 - Each query without constants gets a hash value
 - Each generated Execution Plan gets a hash value
- Goal
 - Each `query_hash` (without constants) should have the SAME `query_plan_hash`
 - Consistent, safe plan across different input parameters

Demo

Plan Caching

Summary

- Wait Statistics
- System Health
- Memory Issues
- I/O Issues
- Plan Cache Issues