NAV TECH DAYS 2015

mibuso.com

TIPS, TRICKS & TOOLS FOR NAV/SQL TROUBLESHOOTERS

Jörg Stryk
(STRYK System Improvement)

JÖRG A. STRYK

MS Dynamics NAV (Navision) since 1997
Since version 1.2

MS SQL Server since 2003
Since version 2000

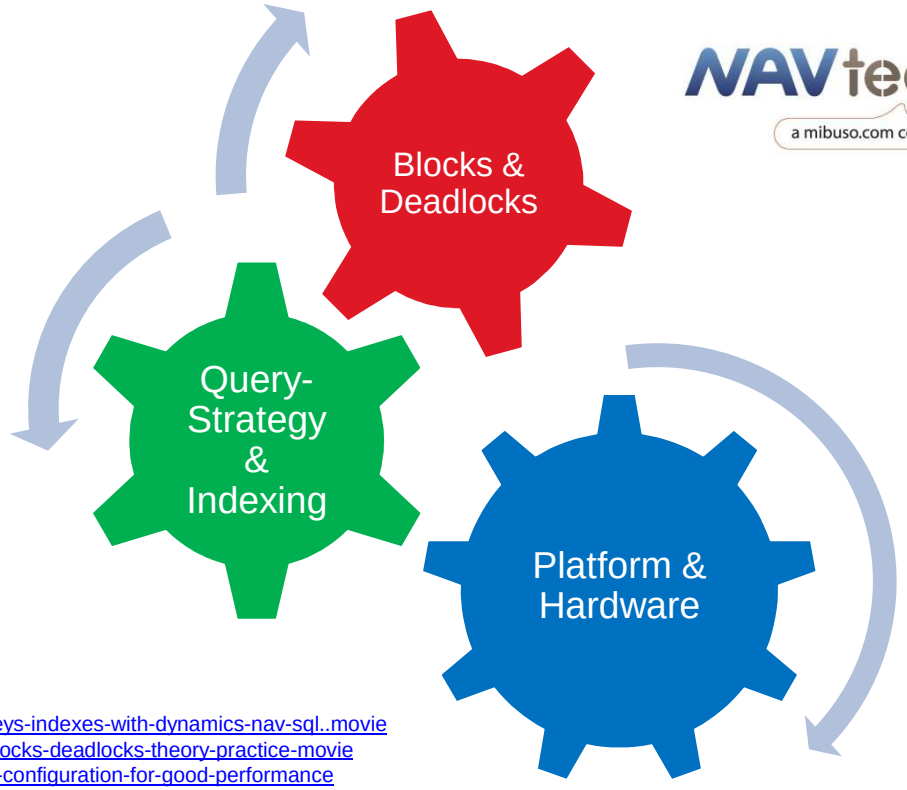
100% focus on **NAV/SQL Performance Optimization**
STRYK System Improvement (since 2006)
Worldwide support of MS Dynamics Partners & Customers

More than 500 projects in about 25 countries on 5 continents – and counting!

Microsoft Most Valuable Professional
MVP MS Dynamics NAV since 2007

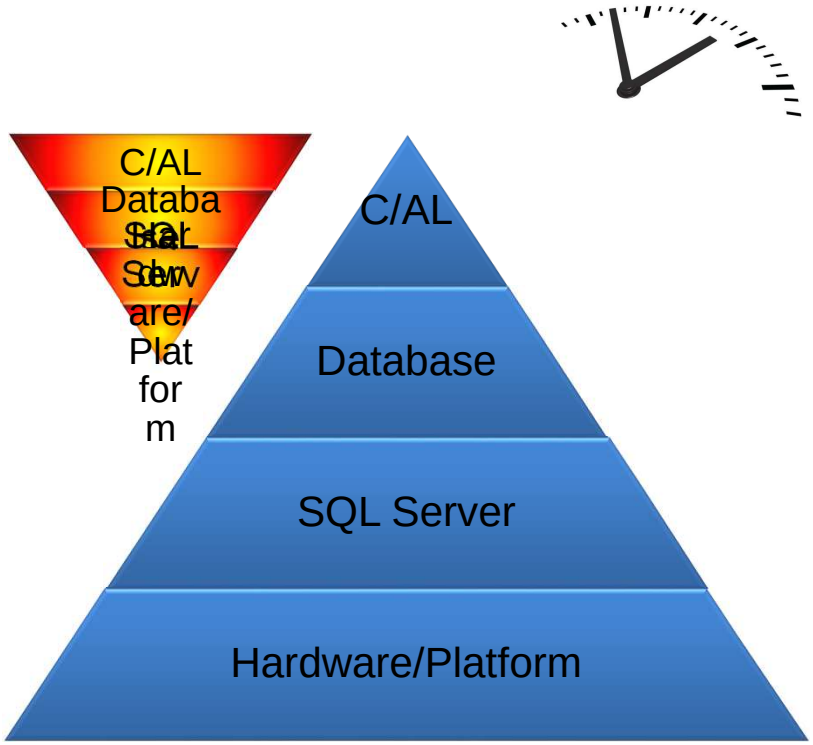
Book:	"*NAV/SQL Performance Field Guide*"	ISBN 978-3-8370-1442-6
Software:	"*SSI Performance Toolbox*"	http://www.stryk.info/toolbox.html
Blog:	"*NAV/SQL Performance – My Two Cents*"	http://dynamicsuser.net/blogs/stryk/

PERFORMANCE AREAS - AGENDA

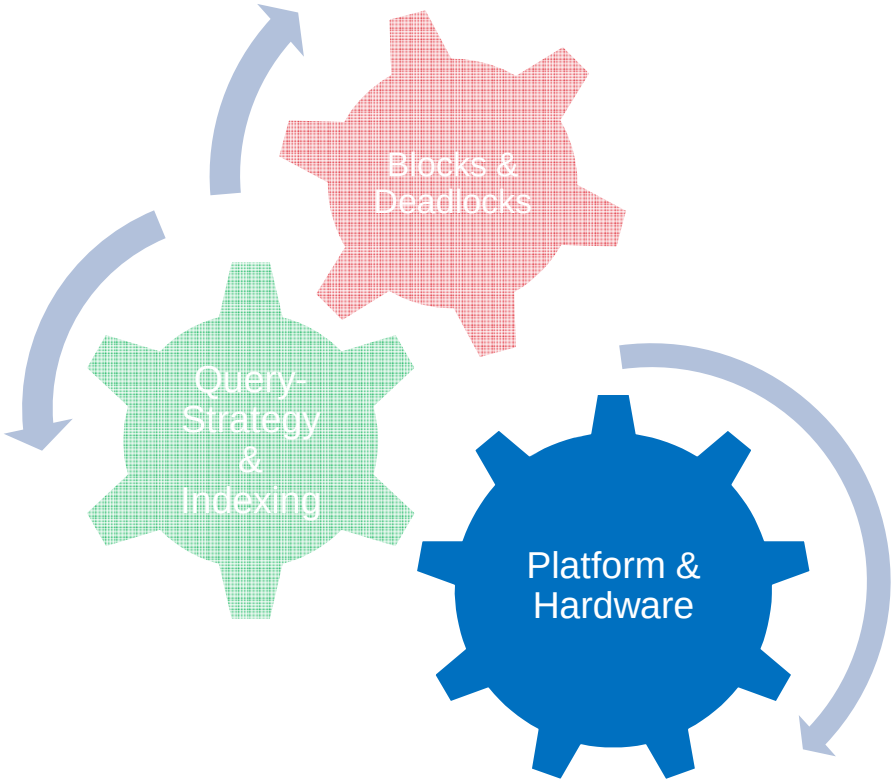


http://mibuso.com/downloads/nav-techdays-2011-understanding-keys-indexes-with-dynamics-nav-sql..movie
http://mibuso.com/downloads/nav-techdays-2012-understanding-blocks-deadlocks-theory-practice-movie
http://mibuso.com/downloads/nav-techdays-2014-sql-server-sizing-configuration-for-good-performance

PERFORMANCE AREAS



PART 1 - PLATFORM



The Perfect Platform!?

It depends …

A common mistake that people make when trying to design something completely foolproof is to underestimate the ingenuity of complete fools.

Douglas Adams

Know the Limits!

Windows

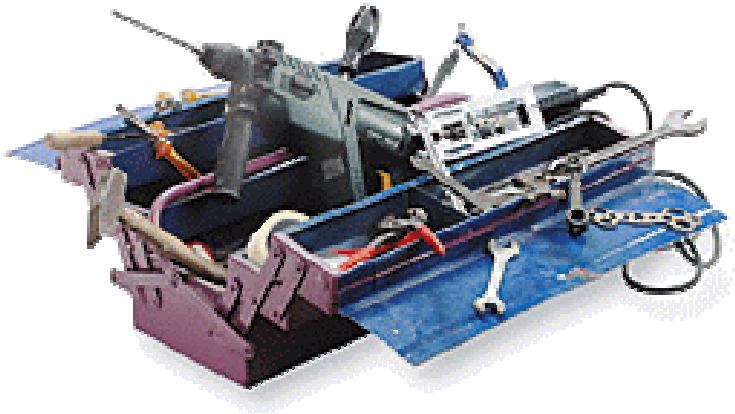
Windows	Foundation		Essential		Standard		Enterprise Data Center	
	CPU	RAM	CPU	RAM	CPU	RAM	CPU	RAM
Server 2008	n/a				4	32 GB	8	1 TB
Server 2008 R2					4	32 GB	8	2 TB
Server 2012	1	32GB	2	64 GB	64	4 TB	64	4 TB
Server 2012 R2	1	32GB	2	64 GB	64	4 TB	64	4 TB
Server 2016								

Microsoft SQL Server

SQL	Standard		Enterprise	
	CPU	RAM	CPU	RAM
Server 2008	4	max	max	max
Server 2008 R2	4	64 GB	max	max
Server 2012	4/16	64 GB	max	max
Server 2014	4/16	128 GB	max	max
Server 2016	4/16	128 GB	max	max

64bit versions only!

Let’s check it out ...



- ✓ **SQL Server Wait Statistics**
- ✓ **Windows Performance Monitor**
- ✓ **Performance Analysis of Logs (PAL)** https://pal.codeplex.com/

CPU – BEST PRACTICES

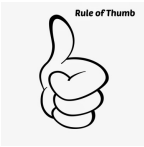
Object	Counter	Instance	Best Practice	Comment
Processor	% Processor Time	Total	15% to 25% (or less)	
Processor	% Privileged Time	Total	Less than 10%	
System	Processor Queue Length	n/a	Less than 2	
SQL Server: General Statistics	User Connections	n/a	n/a	The number of users working
SQL Server: SQL Statistics	Batch Request/sec	n/a	n/a	The number of queries processed
SQL WaitStats	CXPACKET	n/a	n/a	Depends on MAXDOP

1 CPU/Core per 20 to 25 User-Processes

Windows Power Options ➔ High Performance
[BIOS ➔ CPU Power Saving disabled]
Max. Degree of Parallelism ➔ 50% of CPU ➔worst case (NAV!) ➔ MAXDOP = 1
Don‘t change the „Affinity Masks“!
Regard special requirements in virtual environments! [Static vs. Dynamic; NUMA Spanning on/off]

RAM – Best Practices

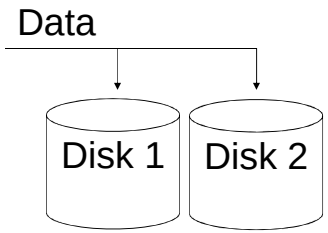
Object	Counter	Instance	Best Practice	Comment
Memory	Available MB	n/a	More than 500 MB	
Paging File	% Usage	n/a	n/a	
SQL Server: Buffer Manager	Buffer Cache Hit Ratio	n/a	Greater than 95%	
SQL Server: Buffer Manager	Free Pages	n/a	Greater than 640	
SQL Server: Buffer Manager	Page Life Expectancy	n/a	Greater than 300	
SQL Server: Memory Manager	Target Server Memory	n/a	n/a	Max. Memory by Configuration
SQL Server: Memory Manager	Total Server Memory	n/a	n/a	Current Memory Usage

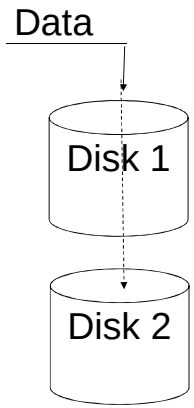


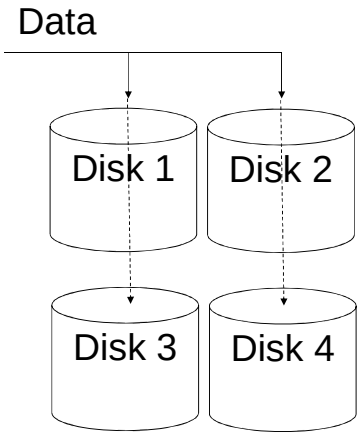
The more the better!
Never less than 16GB
Regard limitations by OS and SQL version/edition

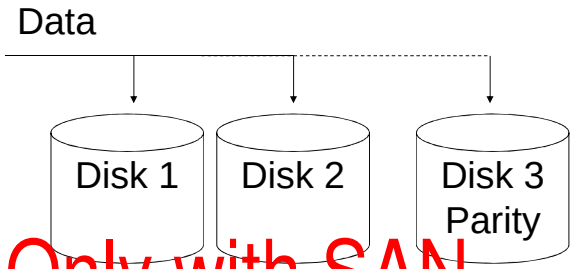
Max. Server Memory = Physical Memory – 2 [to 4]GB [- requirements of other services or applications]
Min. Server Memoyr = 25% to 50% of Max. Server Memory
Local Group Policies ➔ „*Lock Pages in Memory*“ for SQL Server service account [optional: Traceflag 845]

DISKS – RAID (Redundant Array of Independent Disks)





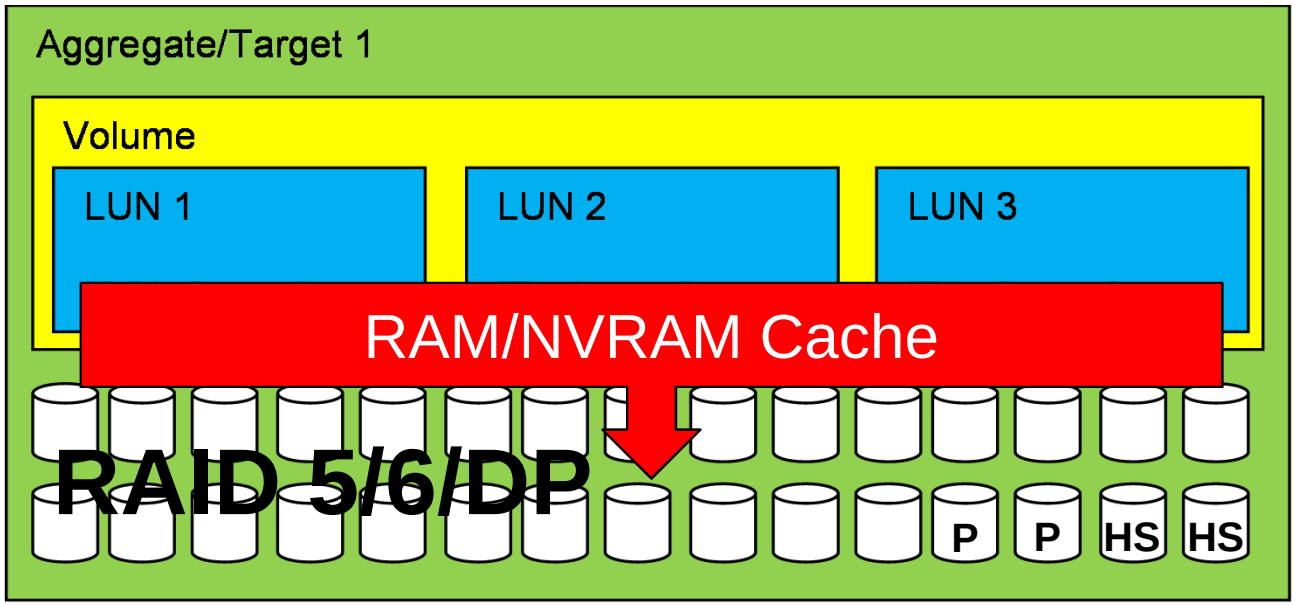




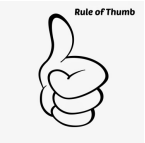
Only with SAN

RAID 0 „Striping“ **RAID 1 „Mirroring“** **RAID 10 „Striping & Mirroring“** **RAID 5 „Striping with Parity“**

DISKS – SAN (Storage Area Network)

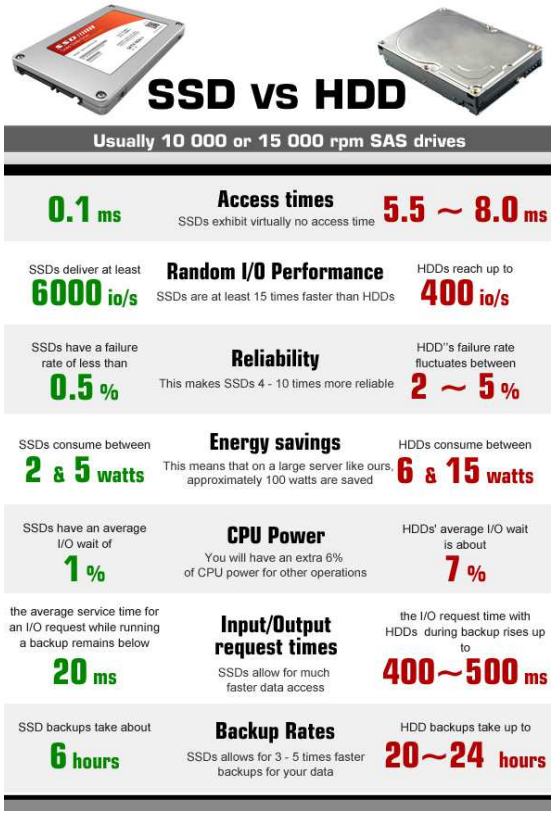


DISKS – Hard Disk Drives & Solid State Disks

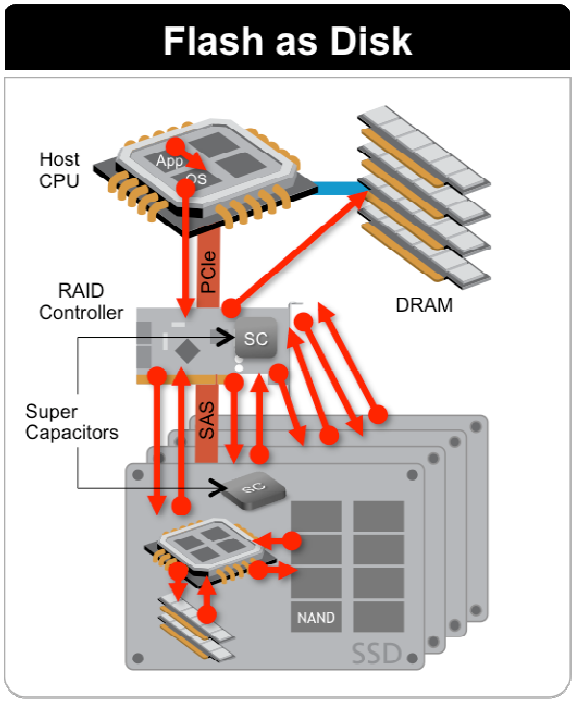


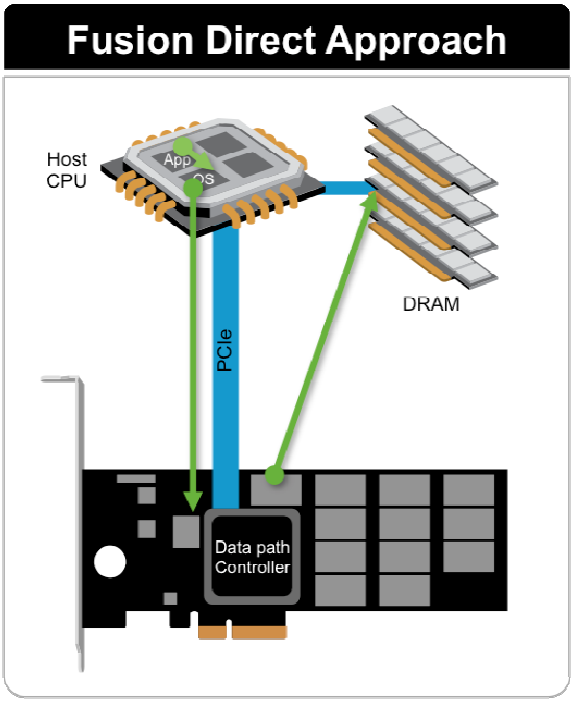
SSD cost per GB is 10 times more expensive than HDD

SSD cost per IOPS is 10 times cheaper than HDD



DISKS – „Fusion IO“





http://www.fusionio.com

DISKS – Best Practices 1

Object	Counter	Instance	Best Practice	Comment
Physical Disk	Avg. Disk Sec/Read	Each drive	Less than 0.015	HDD; SSD less than 0.005
Physical Disk	Avg. Disk Sec/Write	Each drive	Less than 0.015	HDD; SSD less than 0.005
SQL WaitStats	PAGEIOLATCH	n/a	Less than 0.015	HDD; SSD less than 0.005
SQL WaitStats	WRITELOG	n/a	Less than 0.015	HDD; SSD less than 0.005
SQL WaitStats	IO_COMPLETION	n/a	Less than 0.015	HDD; SSD less than 0.005

Use separate drives for

- OS, Programms, etc.
- SQL Server tempdb
- NAV DB Data (mdf/ndf)
- NAV DB Log (ldf)
- Local Backups & Misc

Local Group Policies ➔ „*Perform Volume Maintenance Tasks*“ for SQL Server service account
Format DB drives with 64KB (Caution: regard special SAN requirements!)
SAN: adjust Host Bus Adapter (HBA) queue depth (depends, something between 32 and 128)

DISKS – Best Practices 2

tempdb

Data Size ➔ equal to estimated maximum (e.g. 1000 MB to 5000 MB)
Auto Growth Data ➔fix value (e.g. 100 MB to 500 MB)
1 data-file per CPU; no more than 8 to 12 data-files in total
Log Size ➔ 100 MB to 500 MB
Auto Growth Log ➔ fix value (e.g. 100 MB to 500 MB)
Maybe Traceflag 1118

NAV database

Data Size ➔ 10% free space minimum
Auto Growth ➔fix value „Data Filegroup 1“ (e.g. 1000 MB to 5000 MB)
Log Size ➔ depends on Log Backup frequency; max. 20% of net data-size
Auto Growth Log ➔ fix value (e.g. 500 MB to 1000 MB)
NEVER EVER USE „Auto Shrink“!

LAN – Best Practices

Object	Counter	Instance	Best Practice	Comment
Network Interface	Current Bandwith	each adapter	n/a	Should equal desired bandwidth
Network Interface	Output Queue Length	each adapter	0	
Network Interface	Packets Outbound Errors	each adapter	0	
SQL WaitStats	OLEDB	n/a	Less than 1 msec	
SQL WaitStats	ASYNC_NETWORK_IO	n/a	Less than 1 msec	

Gigabit
Regard LAN/WAN requirements
Dedicated Server-to-Server connections

Regard special requirements in virtual environments!
[TCP Offloading? UDP Offloading? Checksum Offloading? Receive Side Scaling (RSS)?]

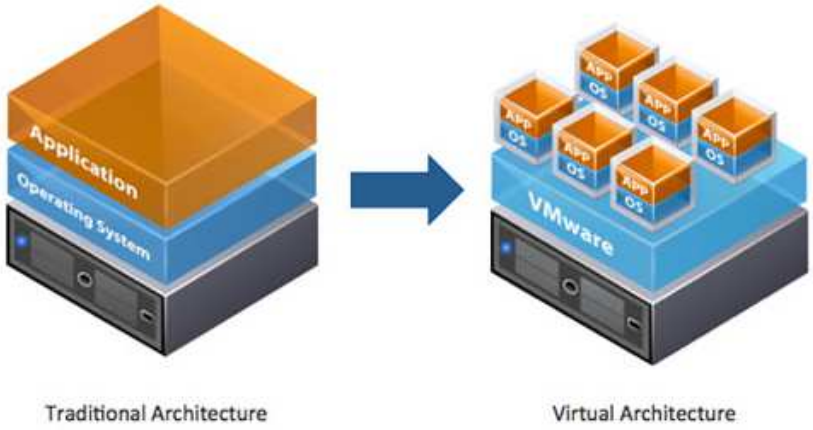
Dynamics NAV Service (NST) (NAV2013+)

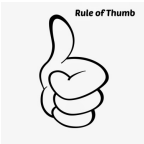
Object	Counter	Instance	Best Practice	Comment
Microsoft Dynamics NAV	*	all	???	
Microsoft Dynamics NAV	% Primary key cache hit rate	all	Greater than 90	Indicates data caching issues
Microsoft Dynamics NAV	Heartbeat time (ms)	all	Less than 10	Indicates LAN issues
Microsoft Dynamics NAV	Average server operation time (ms)	all	Less than 100	[Indicates RAM/CPU issues ?]
SQL WaitStats	ASYNC_NETWORK_IO	n/a	~~Less than 1 msec~~	Will be always high with NAV 2013+

One NST per 50 to 100 Users ➔ one CPU per 20 to 25 Users
RAM: 500 MB per NST + 20 MB per Session + DataCache + MetaDataCache (apx. 100 MB)
At least two NST for the Users (for failover)
One separate NST for debugging/troubleshooting

Data Cache ➔ 10 [= 1024 MB] to 12 [= 4096 MB]
Metadata Provider Cache ➔ 5000 [objects]
Max. Concurrent Calls ➔ equal to number of actual Sessions connected to this NST
“Debugging NST” ➔ different service account!
“Debugging NST” ➔ “*Disable SmartSql*” = TRUE

Virtualization

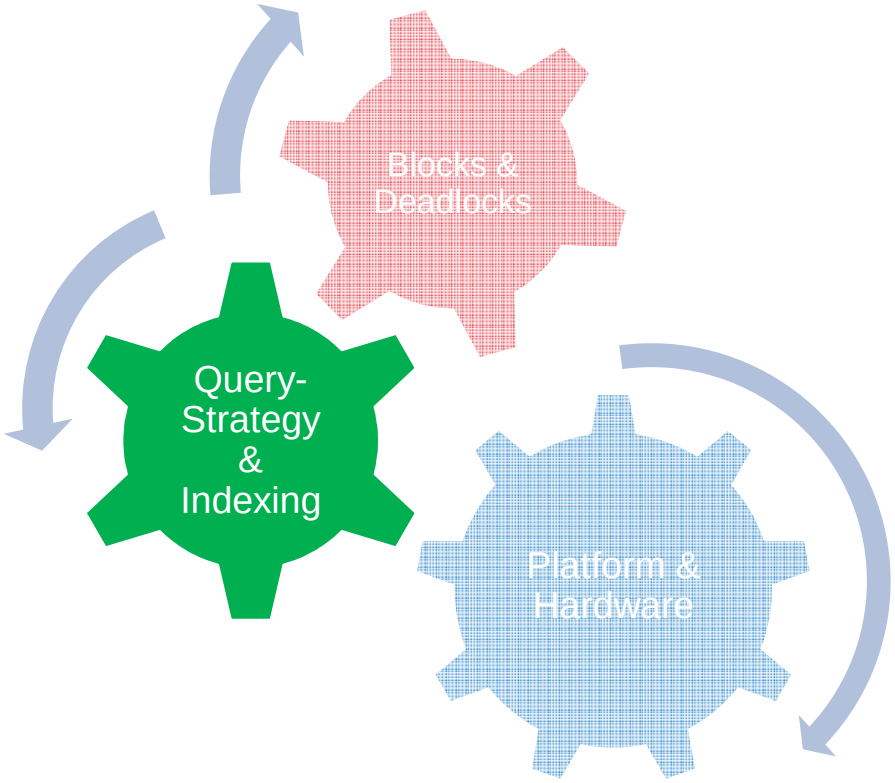




Resource-Requirements are the same as with physical Servers
NEVER EVER OVERBOOK THE VIRTUAL HOSTS!

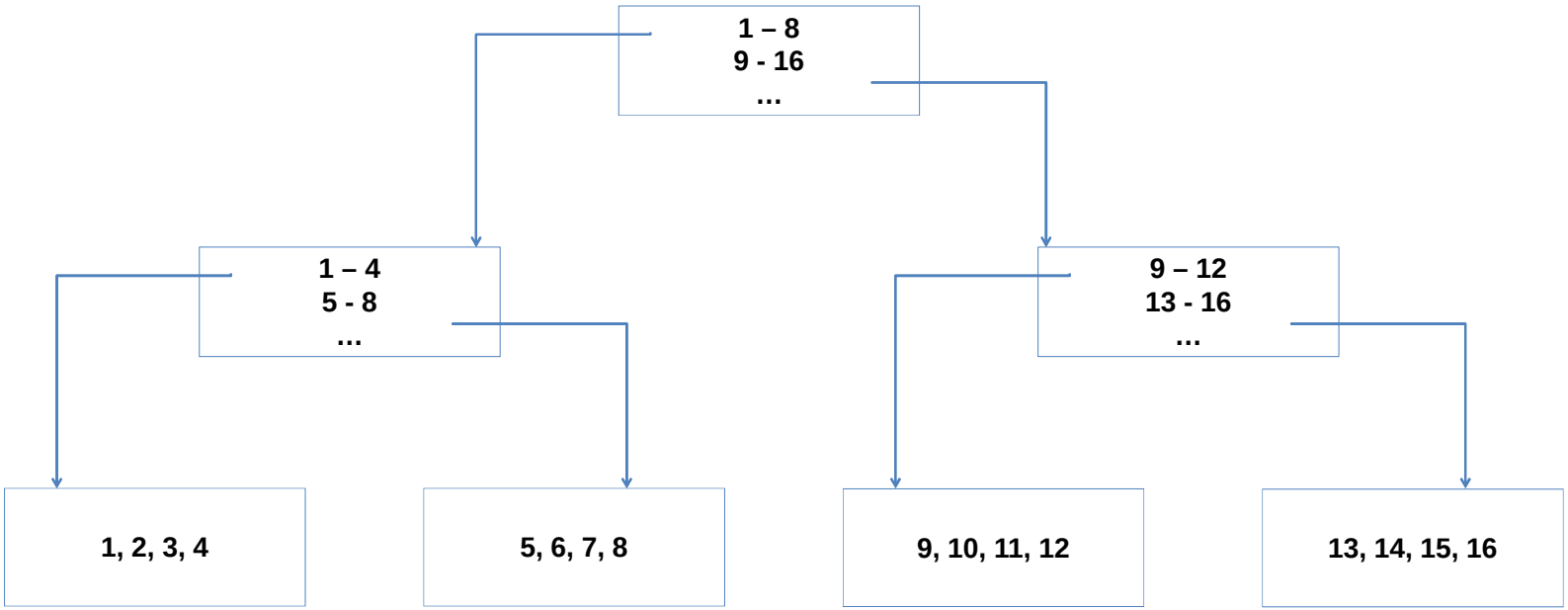
Refer to “Best Practice Guides” of Hypervisor
Regard special requirements regarding CPU (Static vs. Dynamic; NUMA Spanning on/off)
Regard special requirements regarding LAN (TCP Offloading? UDP Offloading? Checksum Offloading? Receive Side Scaling?)

PART 2 – QUERY STRATEGY



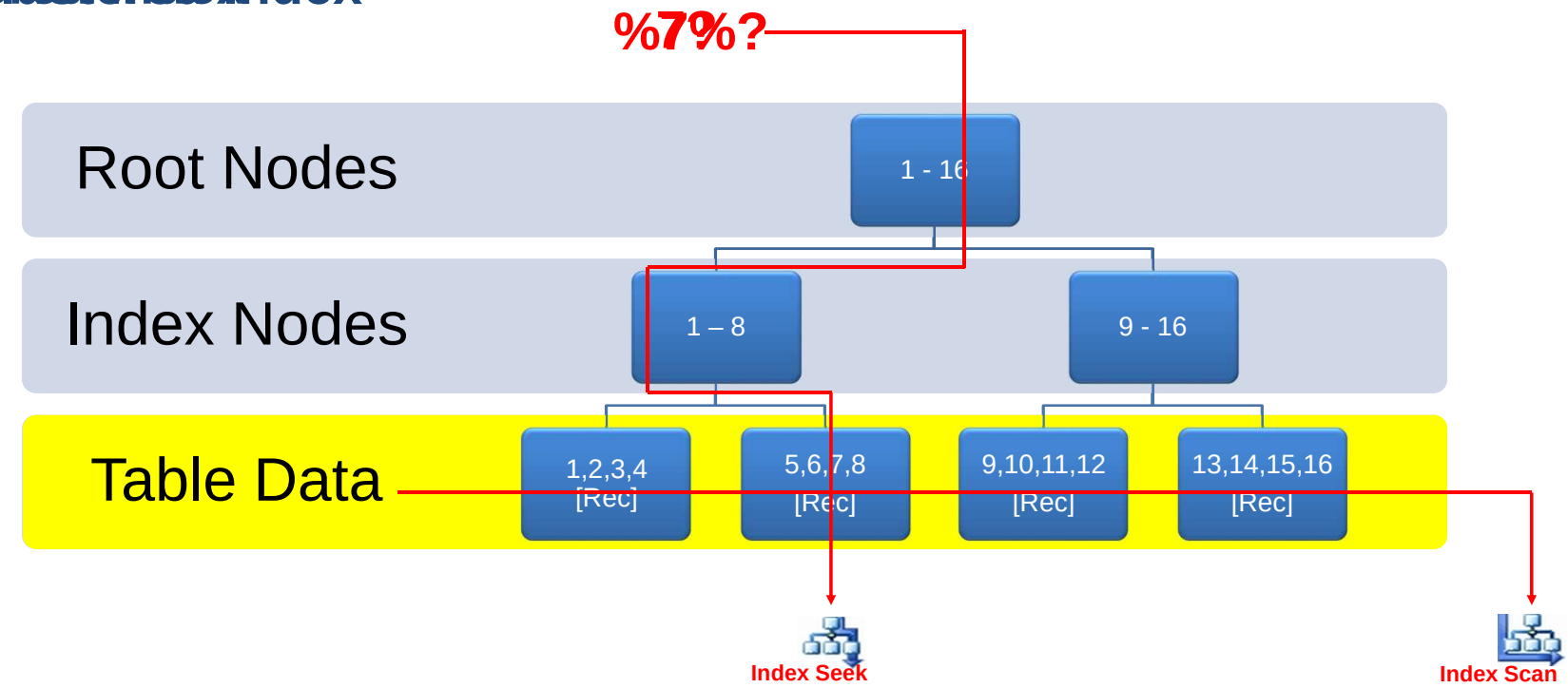
Indexes

Balanced Trees



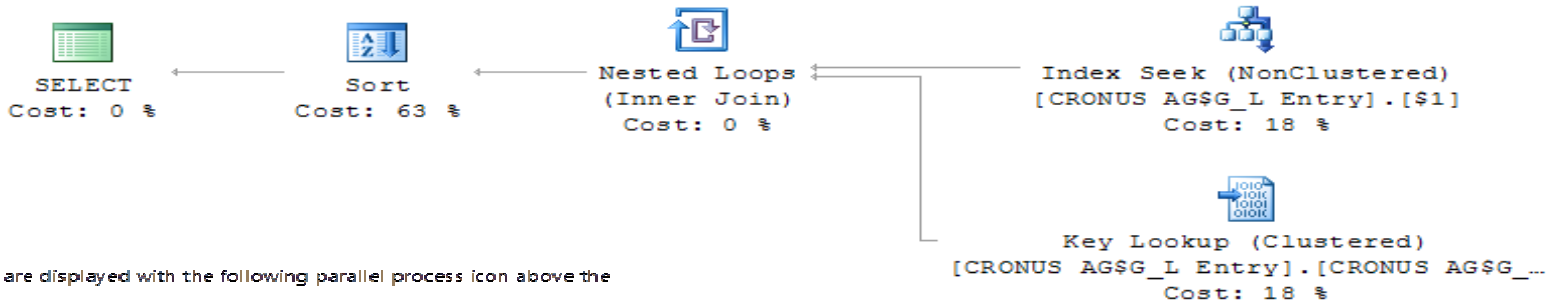
Indexes

Clustered Index



Indexes

```
SELECT * FROM [CRONUS AG$G_L Entry]
WHERE ([G_L Account No_] = @P1)  AND (([Posting Date] >= @P2) AND ([Posting Date] <= @P3))
ORDER BY [Entry No_]
```



Operators that are executed in parallel are displayed with the following parallel process icon above the operator.

Icon	Operator
	Bookmark Lookup
	Clustered Index Scan
	Clustered Index Seek
	Constant Scan
	Filter

Icon	Operator
	Nonclustered Index Scan
	Nonclustered Index Seek
	Sort
	Table Scan
	Top

Indexes

C/SIDE	SQL Server	Example (Customer)	
Primary Key	Primary Key Clustered Index	`No.`	`…$Customer$0: No_`
Secondary Key	Unique Non-Clustered Index	`Search Name`	`$1 (UNIQUE): Search Name, No_`
Table Relation	Foreign Key (Maintain Relationships = TRUE)	`Country Code`	`…$Customer$FK$T18_F35$T9`
n/a	Included Columns		
n/a	Filtered Indexes		
n/a	Descending Indexes		
n/a	Partitioned Indexes		

Caution:
Using the „*SQL Index*“ property in a wrong way might cause severe performance degradation!

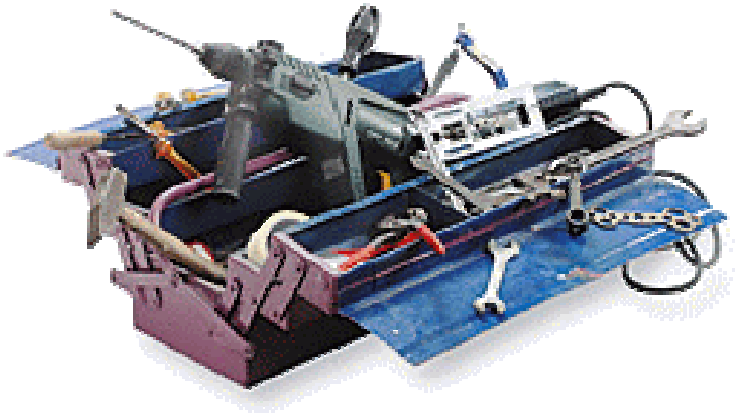
FIND Commands (before NAV 2013)

Old	New (?)	Cursor	Comment
FIND(-)	FIND(-)	**Dynamic**	• No TOP clause • Regard WHERE and ORDER BY clause • Required with ASCENDING(FALSE)
	FINDSET	n/a	• TOP clause based on „Cache Record Set“ • **Cursor** if actual result set exceeds TOP size • If with modification (MODIFY) then FINDSET(TRUE) or FINDSET(TRUE, TRUE) ➔ causing **LOCKING**! • Cannot be used with ASCENDING(FALSE)
	FINDFIRST	n/a	• TOP 1 clause; returns one record • **Cursor** if actual result size exceeds 1 record • Don‘t use with loops
	ISEMPTY	n/a	• TOP 1 NULL; returns 0 or 1 • Don‘t use if record data is needed
FIND(+)	FIND(+)	**Dynamic**	• No TOP clause • DESCENDING order • Regard WHERE and ORDER BY clause
	FINDLAST	n/a	• TOP 1 clause; returns one record • DESCENDING order • **Cursor** if actual result size exceeds 1 record • Don‘t use with loops

FIND Commands (since NAV 2013)

Old	New (?)	Cursor	Comment
FIND(-)	FIND(-)	n/a	• Automatic TOP clause • Required with ASCENDING(FALSE) • Small “Rollback” effort
	FINDSET	n/a	• No TOP clause („Cache Record Set“ is void) • If with modification (MODIFY) then FINDSET(TRUE) or FINDSET(TRUE, TRUE) ➔ causing **LOCKING**! • Cannot be used with ASCENDING(FALSE) • Higher “Rollback” effort
	FINDFIRST	n/a	• TOP 1 clause; returns one record • Don‘t use with loops
	ISEMPTY	n/a	• TOP 1 NULL; returns 0 or 1 • Don‘t use if record data is needed
FIND(+)	FIND(+)	n/a	• Automatic TOP clause • DESCENDING order
	FINDLAST	n/a	• TOP 1 clause; returns one record • DESCENDING order • Don‘t use with loops

Let’s check it out ...



- ✓ SQL Profiler
- ✓ Management Studio & DMV
- ✓ NAV SQL Trace
- ✓ Smart Query Tuning
- ✓ SQL Refactoring

Indexing Rules

NEVER EVER change or drop a NAV standard index from SQL site!

```
CREATE INDEX <Name> -- stay away from NAV naming conventions (never name $0, $1 etc.)!
ON <Table_or_IndexedView>  -- e.g. tables or VSIFT
(
  -- Fields to support WHERE clause
  <equality_columns, ordered by selectivity>   -- regard AND OR LIKE etc. operators
  <inequality_columns, ordered by selectivity> -- regard AND OR LIKE etc. operators

  -- optional (e.g. to support Dynamic Cursors or similar): Fields to support ORDER BY clause
  <fields as taken from ORDER BY, double values (see WHERE) removed>
)
INCLUDE  -- optional: e.g. to support SELECT SUM queries or some JOINS etc.
(
  <included_columns> -- excerpt from table; never copy all/too many fields!
)
WHERE -- optional: to reduce index size filtering according to query requirements
  <filters>  -- filtering according to query requirements
WITH  -- optional: create index with small impact during creation
(
  MAXDOP = 64, -- using all CPU
  ONLINE = ON  -- ONLINE Indexing with Enterprise Edition to reduce blocks
)
```

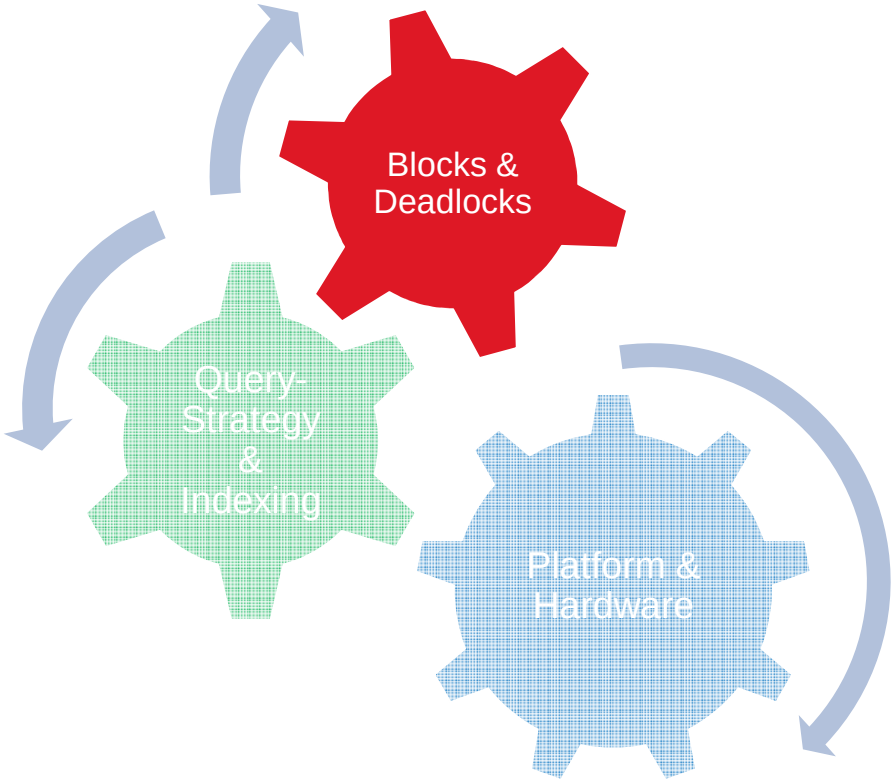
Troubleshooting Expensive Queries

- ✓ Avoid “Dynamic Cursors” (before NAV 2013)
- ✓ Regard Key/Sorting and Filter (more important before NAV 2013)
- ✓ Identify “bad queries” with SQL Profiler
- ✓ Analyze TRC file and QEP (using various Tools & Scripts etc.)
- ✓ Implement optimized indexes (NAV or SQL)
- ✓ Verify Index Usage
- ✓ Remove unused indexes
- ✓ Maintain Indexes & Statistics

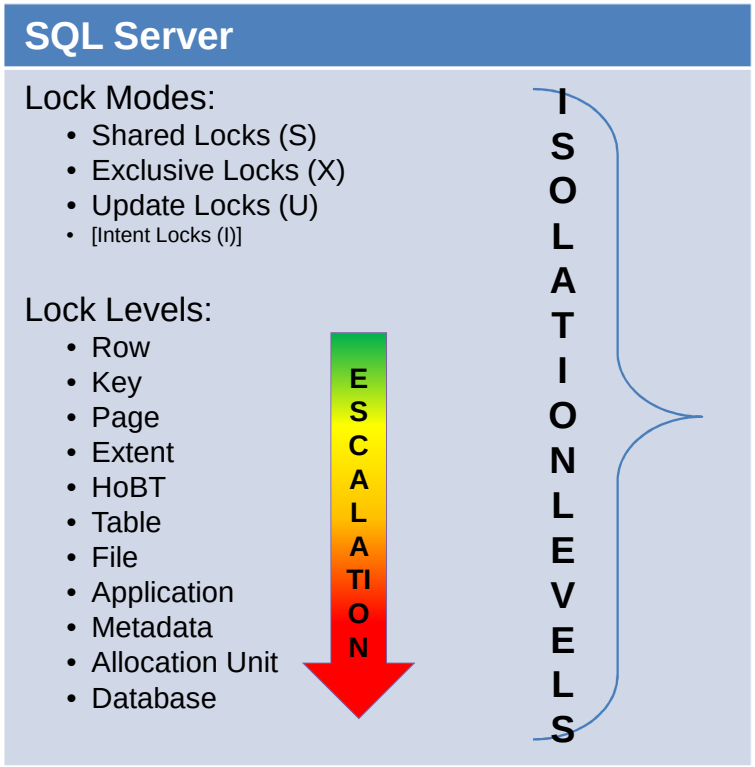
Expensive Queries – „Black Belt“: SmartQuery

- ✓ Option 1:
  - ✓ Identify problematic SmartQuery (Profiler etc.)
  - ✓ Format Statement; e.g. using third party tools
  - ✓ Investigate “Query Execution Plan”
  - ✓ Identify high-cost operators
  - ✓ Pick related sub-SELECT and apply optimized index
- ✓ Option 2:
  - ✓ Disable “SmartQuery” on NST
  - ✓ Identify problematic (sub-)SELECT (Profiler etc.)
  - ✓ Apply optimized index

PART 3 - BLOCKING



Locking Mechanisms



Lock Modes

Lock mode	Description
Shared (S)	Used for **read operations** that do not change or update data, such as a SELECT statement.
Update (U)	Used on resources that can be **updated**. Prevents a common form of deadlock that occurs when multiple sessions are reading, locking, and potentially updating resources later.
Exclusive (X)	Used for **data-modification** operations, such as INSERT, UPDATE, or DELETE. Ensures that multiple updates cannot be made to the same resource at the same time.
Intent	Used to establish a **lock hierarchy**. The types of intent locks are: intent shared (IS), intent exclusive (IX), and shared with intent exclusive (SIX).

Lock Modes

	Existing granted mode					
Requested mode	IS	S	U	IX	SIX	X
Intent shared (IS)	✓	✓	✓	✓	✓	☠
Shared (S)	✓	✓	✓	☠	☠	☠
Update (U)	✓	✓	☠	☠	☠	☠
Intent exclusive (IX)	✓	☠	☠	✓	☠	☠
Shared with intent exclusive (SIX)	✓	☠	☠	☠	☠	☠
Exclusive (X)	☠	☠	☠	☠	☠	☠

Lock Granularity

Avoid SQL Server Lock Escalation

- *Enable „Always Rowlock“* in NAV (before NAV 2013)

Caution: Do not use with 32bit systems or with little RAM!

```
SELECT  * FROM "NAV"."dbo"."CRONUS AG$Purchase Header"
WITH (UPDLOCK, ROWLOCK)
WHERE (("Document Type"=@P1)) AND (("No_"=@P2))
OPTION (OPTIMIZE FOR UNKNOWN)
```

- *Optional: disable Lock Escalation globally using Traceflag 1224*

Caution: Do not use Traceflag 1211!

Transaction Isolation Levels

READ UNCOMMITTED (NAV Default)
Implements dirty read, or isolation level 0 locking, which means that no shared locks are issued and no exclusive locks are honored. When this option is set, it is possible to read uncommitted or dirty data; values in the data can be changed and rows can appear or disappear in the data set before the end of the transaction. This option has the same effect as setting NOLOCK on all tables in all SELECT statements in a transaction. **This is the least restrictive of the four isolation levels**.

READ COMMITTED (SQL Default)
Specifies that shared locks are held while the data is being read to avoid dirty reads, but the data can be changed before the end of the transaction, resulting in nonrepeatable reads or phantom data. **This option is the SQL Server default**.

REPEATABLE READ (NAV Default "Hardlocking" since NAV 2013)
Locks are placed on **all data** that is used in a query, preventing other users from updating the data, but new phantom rows can be inserted into the data set by another user and are included in later reads in the current transaction. Because concurrency is lower than the default isolation level, use this option only when necessary.

SERIALIZABLE (NAV Default "Hardlocking" before NAV 2013)
Places a range lock on the data set, preventing other users from updating or inserting rows into the data set until the transaction is complete. **This is the most restrictive of the four isolation levels**. Because concurrency is lower, use this option only when necessary. This option has the same effect as setting HOLDLOCK on all tables in all SELECT statements in a transaction.

Transaction Isolation Levels

Using REPEATABLE READ Isolation with NAV (before NAV 2013)

Requires NAV 5.0 SP1 Build 30482 (or higher) or NAV 2009 SP1 Build 30609 (or higher)
Theoretical risk of “Phantom Reads”.

```
USE [<DatabaseName>]  -- select NAV database here
GO

SELECT diagnostics FROM [$ndo$dbproperty]
GO

-- set NAV isolation to REPEATABLE READ
UPDATE [$ndo$dbproperty] SET diagnostics = diagnostics + 4194304
GO
```

Read Committed Snapshot Isolation

Using “RCSI” ???

This has no benefit for NAV internal queries but may support “external” queries.

Caution: enabling RCSI will increase the pressure on "**tempdb**“;
thus, it is mandatory to have it really optimized!

```
USE [master]
GO

ALTER DATABASE [<DatabaseName>]
SET ALLOW_SNAPSHOT_ISOLATION ON
GO

ALTER DATABASE [<DatabaseName>]
SET READ_COMMITTED_SNAPSHOT ON
GO
```

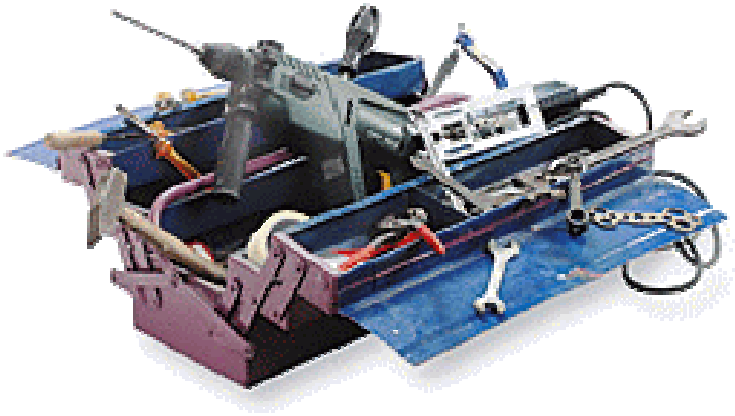
Avoiding Blocks & Deadlocks

- ✓ Optimize Workflow / Business Processes
- ✓ Avoid Lock Escalation
- ✓ Avoid Serialization (before NAV 2013)
- ✓ Regard Data-Structure; e.g. “Clustered Index”
- ✓ Avoid “overlapping” Resultsets
- ✓ Short Transactions, short Locking-Times

Deadlocks:

- ✓ [Lock Resources in same sequence ???]
- ✓ [Using “Semaphore” ???]

Let’s check it out ...



- ✓ **Block Detection Events**
- ✓ **Deadlock Tracing**
- ✓ **NAV SQL Trace**

Any Questions?

THANK YOU

WHEN YOU ARE PASSIONATE ABOUT MICROSOFT DYNAMICS NAV | *www.navtechdays.com*

NAV TECH DAYS 2015

mibuso.com

COFFEE BREAK

see you back in **30** min.