Performance Analysis
-
SQL Perform Tools
-
DECORA BLIND SYSTEMS

Adriaan Van Bauwel,
Tassos Vassilopoulos
Utrecht, August 2018
Version 1

Table of Contents

Introduction 4
Methodology 4
Hardware 5
Virtual server 5
Processor 6
Memory 6
Storage system 7
Installation 8
SQL Server Setup 8
Blindata database Setup 10
Temp db setup 10
Database Files setup 11
Log fragmentation 11
Database Usage 11
Users/Sessions 12
Auto Options 12
Autostats 13
Heaps 13
Table info 14
Memory 16
Memory distribution 16
Cache Analysis 17
Generic settings 18
Page file 18
Power plan 18
Virus scanner 19
Perform Volume Maintenance Tasks 19

Performance analyses 20
Disk response times & IO requests 20
CPU 26
Waitstats 27
Database Load 28
Indexes 29
Index usage 30
Index Scans 30
Missing Index keys 31
Index Fragmentation 31
Locks & blocks 32
Blocks 32
Deadlocks 39
Timeouts 42
Queries 44
Call to Action 50
Hardware 50
SQL Server 52
Application 55

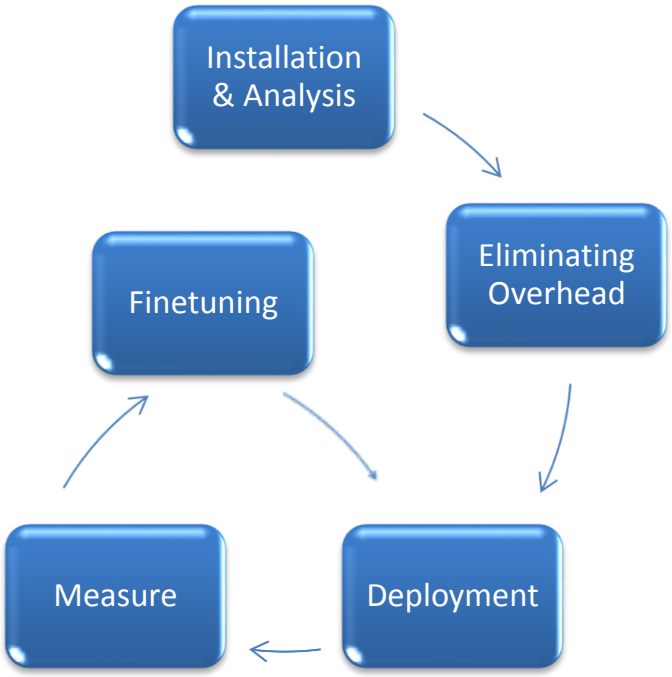
Introduction

DECORA BLIND SYSTEMS LTD has chosen the SQL Perform Tools for optimizing the performance of the system and quickly analyzes potential problems.

DECORA BLIND SYSTEMS LTD has asked SQL Perform Europe to do an analysis of the system using the methodology of SQL Perform.

Methodology

A normal performance project has three phases.



Phase 1

Installing Perform-Tools and measuring performance. This phase usually takes two weeks after which a project plan will be made and recommendations will be given.

Phase 2

Implementing the recommendations. Usually this includes an index tuning.

Phase 3

After care. During this phase the system is monitored and if necessary further adjusted.

Hardware

Virtual server

Windows edition

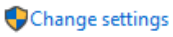
Windows Server 2016 Standard

System

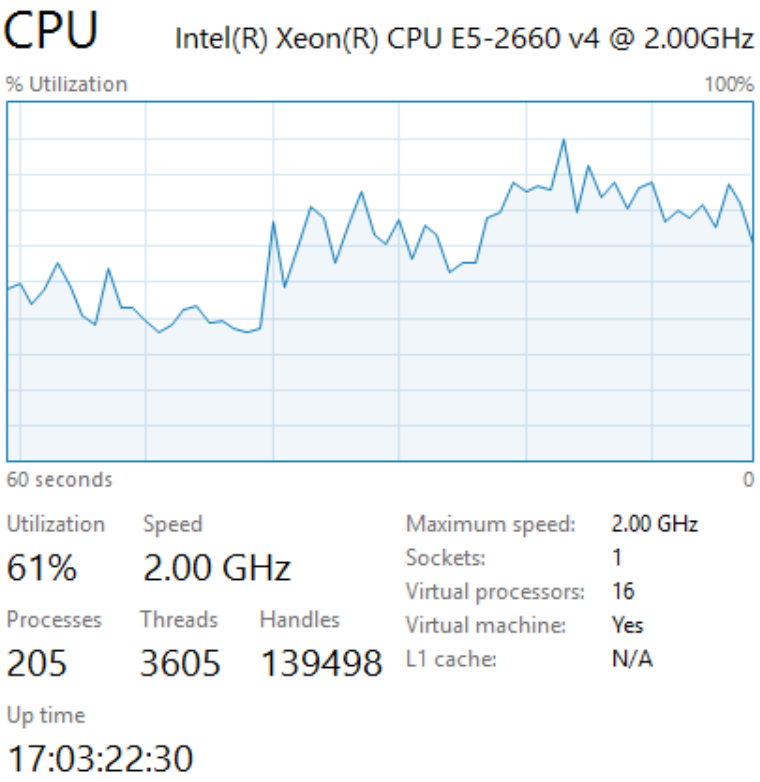
Processor:	Intel(R) Xeon(R) CPU E5-2660 v4 @ 2.00GHz 2.00 GHz
Installed memory (RAM):	145 GB
System type:	64-bit Operating System, x64-based processor
Pen and Touch:	No Pen or Touch Input is available for this Display

Computer name, domain, and workgroup settings

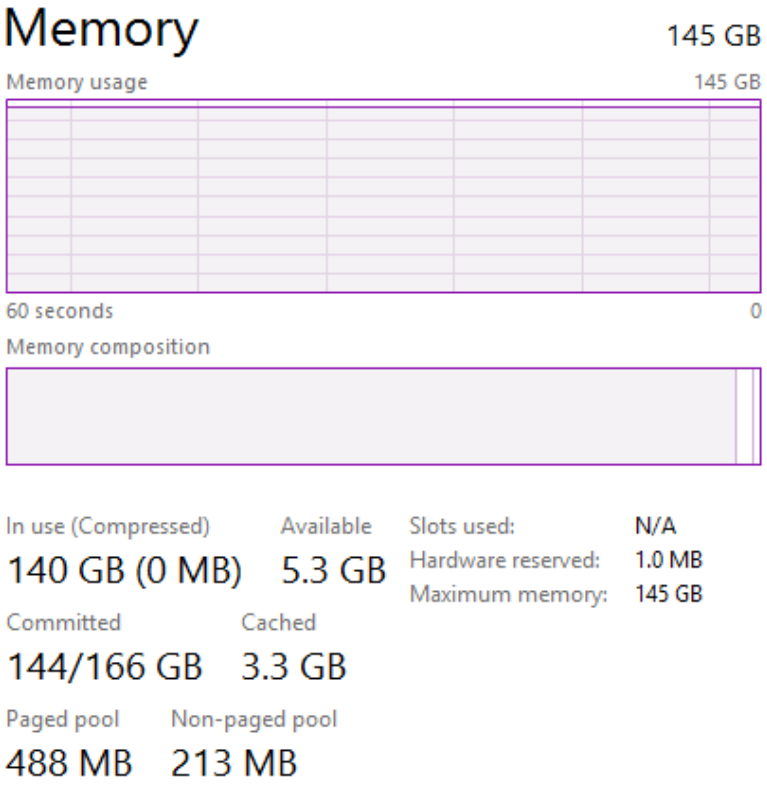
Computer name:	NAVDB
Full computer name:	NAVDB.decora.local
Computer description:	
Domain:	decora.local



Processor

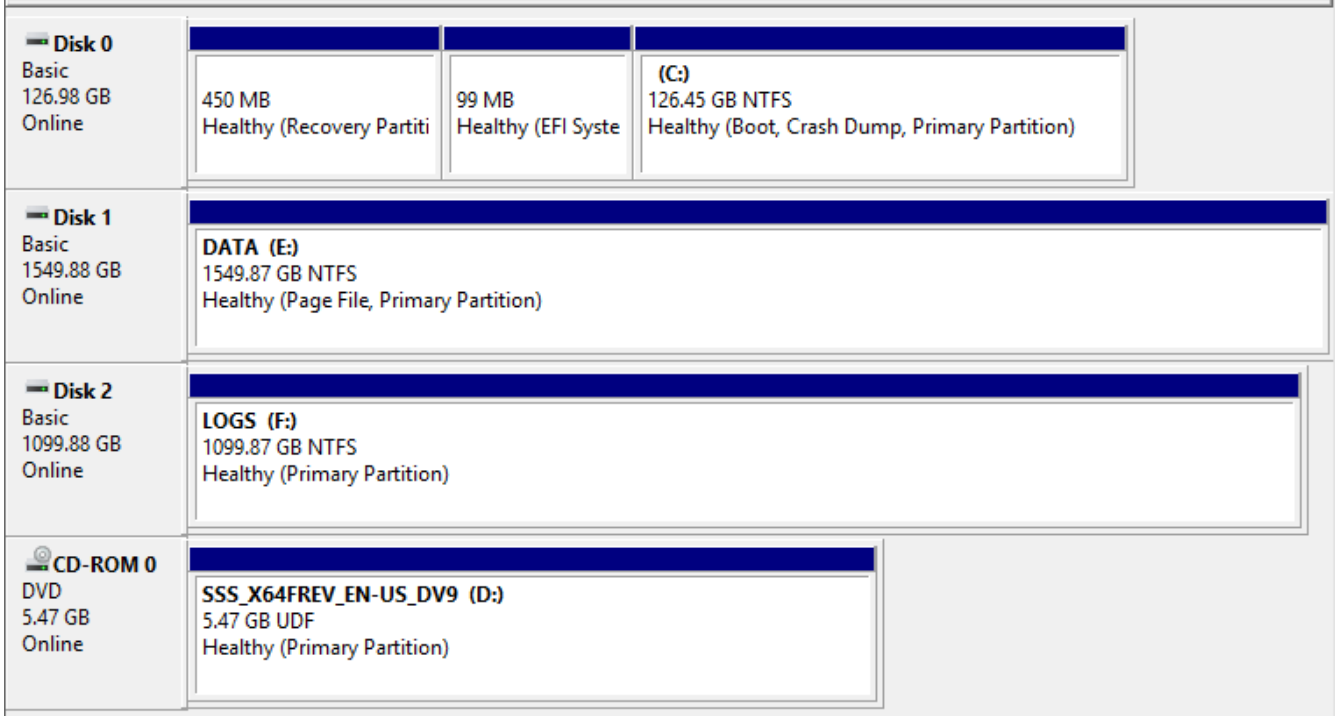


Memory



Storage system

Volume	Layout	Type	File System	Capacity	Free Space	% Free
	Simple	Basic		450 MB	450 MB	100 %
	Simple	Basic		99 MB	99 MB	100 %
(C:)	Simple	Basic	NTFS	126.45 GB	75.69 GB	60 %
DATA (E:)	Simple	Basic	NTFS	1549.87 GB	531.05 GB	34 %
LOGS (F:)	Simple	Basic	NTFS	1099.87 GB	832.11 GB	76 %
SSS_X64FREV_EN-...	Simple	Basic	UDF	5.47 GB	0 MB	0 %



Installation

SQL Server Setup

```
SQL setup for database Blindata on Aug  1 2018

SQL Server:
SQL Server Name ..............  NAVDB
SQL Version ..................  13.0.4502.0
Edition ......................  Standard Edition (64-bit)
Installed on (Y-M-D)..........  2017-09-26
Last re-started on (Y-M-D)....  2018-07-16 05:50:02
Last DBCC CHECKDB date (Y-M-D).. 2018-07-28 03:00:05
Memory dumps.................. 2 , latest memory dump on (Y-M-D) 2018-05-21 09:21:09
Clustered.....................  No
HADR enabled..................  No
Buffer Pool Extension.........  Disabled
Server Authentication ........  SQL Server and Windows Authentication mode
Login Auditing................  Failed logins only
Default Login.................  guest
Default Domain................  DECORA
SQL Server Service Login Acc...  NT AUTHORITY\NetworkService
SQL Agent Service Login Acc....  NT Service\SQLSERVERAGENT
License Type..................  DISABLED
Platform .....................  NT x64
Windows Version ..............  Windows Server 2016 Standard 10.0 <X64> (Build 14393: )
(Hypervisor)
Processor Type ...............  Intel(R) Xeon(R) CPU E5-2660 v4 @ 2.00GHz
Physical Processor Count .....  1
Logical Processor Count .......  16
Physical Memory ..............  148479 (155691532288)
Using Locked Pages ...........  No
Server Collation .............  Latin1_General_CI_AS
There are no trace flags enabled in SQL Server.
SQL Traces running ...........  3. See details below:
  id status  filename
  1  running c:\program files\microsoft sql server\mssql13.mssqlserver\mssql\log\log_701.trc
  2  running e:\tempdb\deadlock.trc
  3  running e:\tempdb\timeout.trc

TempDB Database:
Collation Type ...............  Windows Collation
TempDB Collation .............  Latin1_General_CI_AS
Recovery Model ...............  SIMPLE
Auto Create Statistics........  Yes
Auto Update Statistics........  Yes

SQL Database:
Collation Type ...............  Windows Collation
Collation ....................  Cyrillic_General_CI_AS
Recovery Model ...............  FULL
Auto Create Statistics........  No
Auto Update Statistics........  No

TempDB Database Files.......
                              E:\TempDB\tempdb.mdf
                              F:\TempDB\templog.ldf
                              E:\TempDB\tempdb_mssql_2.ndf
                              E:\TempDB\tempdb_mssql_3.ndf
                              E:\TempDB\tempdb_mssql_4.ndf
                              E:\TempDB\tempdb_mssql_5.ndf
                              E:\TempDB\tempdb_mssql_6.ndf

SQL Database Files..........
                              E:\Databases\Blindata\Blindata_Dat
                              F:\SQL Logs\Blindata\BlinData_log
```

Name	Minimum	Maximum	Default	Changed	Configured	Runtime
access check cache bucket count	0	65536	0	---	0	0
access check cache quota	0	2147483647	0	---	0	0
Ad Hoc Distributed Queries	0	1	0	---	0	0
affinity I/O mask	-2147483648	2147483647	0	---	0	0
affinity mask	-2147483648	2147483647	0	---	0	0
affinity64 I/O mask	-2147483648	2147483647	0	---	0	0
affinity64 mask	-2147483648	2147483647	0	---	0	0
Agent XPs	0	1	1	---	1	1
allow polybase export	0	1	0	---	0	0
allow updates	0	1	0	---	0	0
automatic soft-NUMA disabled	0	1	0	---	0	0
backup checksum default	0	1	0	---	0	0
backup compression default	**0**	**1**	**0**	**Yes**	**1**	**1**
blocked process threshold (s)	0	86400	0	---	0	0
c2 audit mode	0	1	0	---	0	0
clr enabled	0	1	0	---	0	0
contained database authentication	0	1	0	---	0	0
cost threshold for parallelism	0	32767	5	---	5	5
cross db ownership chaining	0	1	0	---	0	0
cursor threshold	-1	2147483647	-1	---	-1	-1
Database Mail XPs	0	1	0	Yes	1	1
default full-text language	0	2147483647	1033	---	1033	1033
default language	0	9999	0	---	0	0
default trace enabled	0	1	1	---	1	1
disallow results from triggers	0	1	0	---	0	0
external scripts enabled	0	1	0	---	0	0
filestream access level	0	2	0	---	0	0
fill factor (%)	0	100	0	---	0	0
ft crawl bandwidth (max)	0	32767	100	---	100	100
ft crawl bandwidth (min)	0	32767	0	---	0	0
ft notify bandwidth (max)	0	32767	100	---	100	100
ft notify bandwidth (min)	0	32767	0	---	0	0
hadoop connectivity	0	7	0	---	0	0
index create memory (KB)	704	2147483647	0	---	0	0
in-doubt xact resolution	0	2	0	---	0	0
lightweight pooling	0	1	0	---	0	0
locks	5000	2147483647	0	---	0	0
max degree of parallelism	**0**	**32767**	**0**	**Yes**	**8**	**8**
max full-text crawl range	0	256	4	---	4	4
max server memory (MB)	**128**	**2147483647**	**2147483647**	**---**	**2147483647**	**2147483647**
max text repl size (B)	-1	2147483647	65536	---	65536	65536
max worker threads	128	65535	0	---	0	0
media retention	0	365	0	---	0	0
min memory per query (KB)	512	2147483647	1024	---	1024	1024
min server memory (MB)	**0**	**2147483647**	**0**	**---**	**0**	**16**
nested triggers	0	1	1	---	1	1
network packet size (B)	512	32767	4096	---	4096	4096
Ole Automation Procedures	0	1	0	---	0	0
open objects	0	2147483647	0	---	0	0
optimize for ad hoc workloads	0	1	0	---	0	0
PH timeout (s)	1	3600	60	---	60	60
polybase network encryption	0	1	1	---	1	1
precompute rank	0	1	0	---	0	0
priority boost	0	1	0	---	0	0
query governor cost limit	0	2147483647	0	---	0	0
query wait (s)	-1	2147483647	-1	---	-1	-1
recovery interval (min)	0	32767	0	---	0	0
remote access	0	1	1	---	1	1
remote admin connections	0	1	0	---	0	0
remote data archive	0	1	0	---	0	0
remote login timeout (s)	0	2147483647	10	---	10	10
remote proc trans	0	1	0	---	0	0
remote query timeout (s)	0	2147483647	600	---	600	600
Replication XPs	0	1	0	---	0	0
scan for startup procs	0	1	1	---	1	1
server trigger recursion	0	1	1	---	1	1
set working set size	0	1	0	---	0	0
show advanced options	0	1	0	Yes	1	1
SMO and DMO XPs	0	1	1	---	1	1
transform noise words	0	1	0	---	0	0
two digit year cutoff	1753	9999	2049	---	2049	2049
user connections	0	32767	0	---	0	0
user options	0	32767	0	---	0	0
xp_cmdshell	0	1	0	---	0	0

Blindata database Setup

```
Database setup of database Blindata on Aug  1 2018

Collation:
Collation Type ....................  Windows Collation
Collation .........................  Cyrillic_General_CI_AS

Settings
Recovery Model ....................  FULL
Restricted access..................  MULTI_USER
Compatibility Level................  100
ANSI NULL Default .................  0
Recursive Triggers ................  0
Page Verify .......................  CHECKSUM
Auto Close ........................  No
Auto Shrink .......................  No
Auto Create Statistics.............  No
Auto Update Statistics.............  No
Auto Update Statistics Asynch......  No
Snapshot Isolation ................  No
Read Committed snapshot ...........  Yes

DR related SQL Configuration
Is Log Shipped.....................  No
Is Published.......................  No
Is Merge Published.................  No
Is Subscribed......................  No
Is Mirrored/Mirroring..............  No

Miscellaneous
Database created (yy.mm.dd)........  2018.01.20
Database owner.....................  DECORA\ChrisWylie
Updateability......................  READ_WRITE

Number of database files...........  2
Number of data files...............  1
Number of log files................  1
Number of filegroups...............  1

Number of tables...................  611
Number of clustered tables.........  593
Number of heaps....................  18
Number of clustered indexed views..  0
Number of clustered indexes........  608
Number of non-clustered indexes....  496
Number of disabled indexes.........  0
Number of hypothetical indexes.....  1
```

Temp db setup

```
Setup of database tempdb on Aug  1 2018

Collation:
Collation Type ....................  Windows Collation
Collation .........................  Latin1_General_CI_AS

Settings
Recovery Model ....................  SIMPLE
Restricted access..................  MULTI_USER
Compatibility Level................  130
ANSI NULL Default .................  0
Recursive Triggers ................  0
Page Verify .......................  CHECKSUM
Auto Close ........................  No
Auto Shrink .......................  No
Auto Create Statistics.............  Yes
Auto Update Statistics.............  Yes
Auto Update Statistics Asynch......  No
Snapshot Isolation ................  No
Read Committed snapshot ...........  No
```

```
Miscellaneous
Database created (yy.mm.dd).........  2018.07.16
Database owner......................  sa
Updateability.......................  READ_WRITE
Number of database files............  7
Number of data files................  6
Number of log files.................  1
Number of filegroups................  1
```

Database Files setup

Logical Name	File ID	File Name	Filegroup	Size (KB)	Maximum Size (KB)	File Growth	Usage
BlinData_dat	1	E:\Databases\Blindata\Blindata_Dat	PRIMARY	137.193.664	Unlimited	2%	data only
BlinData_log	2	F:\SQL Logs\Blindata\BlinData_log	Not Applicable	79.104.512	Unlimited	1024000 KB	log only

Logical Name	File ID	File Name	Filegroup	Size (KB)	Maximum Size (KB)	File Growth	Usage
tempdev	1	E:\TempDB\tempdb.mdf	PRIMARY	6.758.400	Unlimited	204800 KB	data only
templog	2	F:\TempDB\templog.ldf	Not Applicable	2.355.200	Unlimited	204800 KB	log only
temp2	3	E:\TempDB\tempdb_mssql_2.ndf	PRIMARY	6.553.600	Unlimited	204800 KB	data only
temp3	4	E:\TempDB\tempdb_mssql_3.ndf	PRIMARY	6.553.600	Unlimited	204800 KB	data only
temp4	5	E:\TempDB\tempdb_mssql_4.ndf	PRIMARY	6.553.600	Unlimited	204800 KB	data only
temp5	6	E:\TempDB\tempdb_mssql_5.ndf	PRIMARY	6.553.600	Unlimited	204800 KB	data only
temp6	7	E:\TempDB\tempdb_mssql_6.ndf	PRIMARY	6.758.400	Unlimited	204800 KB	data only

Log fragmentation

ID	Logical Name	File name	Size (MB)	Growth	Max size (MB)	# Fragments	Fragmentation
2	BlinData_log	F:\SQL Logs\Blindata\BlinData_log	77.251	1000 MB	No max	137	High

Database Usage

Logical/DB Name	File/Total	Total Space(MB)	Used Space(MB)	Free Space(MB)	% Free Space	% Used Space
BlinData_dat	1	133.978	119.129	14.849	11	89
Blindata	DATA-TOTAL	133.978	119.129	14.849	11	89
Blindata	LOG-TOTAL	77.250	306	76.944	100	0
Blindata	DB-TOTAL	211.229	119.436	91.793	44	57

Users/Sessions

Program	Connections
	373
Microsoft Dynamics NAV Server	41
2007 Microsoft Office system	11
Microsoft SQL Server Management Studio	8
.Net SqlClient Data Provider	6
Microsoft Dynamics NAV Development Environment client	3
Internet Information Services	3
Microsoft SQL Server Management Studio - Query	3
SQLAgent Job (Perform-Tools - Blocks Monitoring - DecoraERP : Step 1)	1
SQLAgent - Job invocation engine	1
SQLServerCEIP	1
SBOP Crystal Reports	1
SQLAgent Job (Perform-Tools - Blocks Monitoring - Blindata : Step 1)	1
SQLAgent - Alert Engine	1
SQLCMD	1
SQLAgent - Email Logger	1
Microsoft SQL Server	1
SQLAgent - Generic Refresher	1
Grand Total	**458**

Auto Options

The setup options for auto update statistics & auto create statistics are set correctly.

```
Analysis of database Blindata for auto options on Aug  1 2018

 ...... auto update statistics is OFF ...... Correct.
 ...... auto create statistics is OFF ...... Correct.
 ...... auto close             is OFF ...... Correct.
 ...... auto shrink            is OFF ...... Correct.
```

Autostats

No autostats found as the database settings are correct.

Heaps

Heaps are unclustered tables in the database. Unclustered table have no structure for sorting data on the disk, leading to slower data read times.

```
Perform-Tools analysis for clustered indexes in database Blindata on Aug  1 2018  4:19PM.

Total number of clustered tables ..........................   608
Total number of heaps .....................................   28


List of tables that are heaps follows.
```

Table Name	Rows
aSalesleadHistory	1543
BlindType_MaxDiscount	94
ComponentsDiscountByCategory	0
DECORA_Kitting_History	**3375109**
DeliveryRun	7
ExistsComment	188
Measurement	2
pt_queryhistory_cnt_temp	500
pt_queryhistory_cpu_temp	500
pt_queryhistory_reads_temp	500
pt_queryhistory_time_temp	500
pt_queryhistory_writes_temp	500
SageUtilities_Param	52
SalesleadHistory	7020
sys_Postcode	66514
timer_for_delete_Address	1
TNT_Import	243774
TransactionType	2

Table info

SQL Table Name	Reserved (KB)	No. Of Records	Record Size (B)	Index/Data Ratio
aOrderDetail_Fabric	8.837.880	78.213.379	115	45,00%
aStockTransaction	21.436.672	75.039.082	292	33,00%
aOrderStatus_Change	2.296.584	51.015.239	46	0,00%
OrderDetail_Fabric	2.948.352	28.058.619	107	46,00%
StockTransaction	6.880.400	25.312.486	278	32,00%
EDIOptionList	2.279.080	22.781.564	102	17,00%
aOrderAmend	2.217.824	21.841.741	103	44,00%
OrderStatus_Change	1.035.120	14.519.529	72	74,00%
EDILog	38.414.032	10.566.509	3.721	-98,00%
aProcessTransaction	907.152	10.528.217	88	0,00%
aHistoryOrderDetailAmend	991.480	9.700.605	104	0,00%
ProcessTransaction	1.236.296	8.411.636	150	87,00%
aHistoryOrderAmend	891.800	7.898.385	115	38,00%
OrderAmend	722.496	7.164.353	103	42,00%
aOrderDetail	11.974.040	6.276.965	1.953	1,00%
aSerialDetailLine	567.376	6.086.811	95	20,00%
aScheduling	575.912	5.030.495	117	66,00%
aAddress	1.543.712	4.604.684	343	9,00%
EDIOrderDetails	2.801.808	4.393.706	652	2,00%
DECORA_Kitting_History	581.088	3.375.123	164	0,00%
HistoryOrderDetailAmend	352.544	3.364.179	107	16,00%
aCuttingOrderDetail	144.720	3.064.257	48	49,00%
SerialDetailLine	237.408	2.739.367	88	20,00%
HistoryOrderAmend	266.544	2.558.785	106	39,00%
aOrder	2.769.256	2.302.278	1.230	6,00%
OrderDetail	4.561.456	1.955.716	2.289	-12,00%
EDIOrderHeader	363.064	1.772.876	209	16,00%
Scheduling	219.152	1.760.227	127	94,00%
aOrderScheduledDate	74.832	1.743.901	43	56,00%
EDISenderinformation	510.080	1.730.203	301	54,00%
Address	414.768	1.539.391	275	25,00%
pt_sessionhistory	257.920	1.251.214	210	109,00%
CuttingOrderDetail	66.848	1.192.356	57	36,00%
pt_cachehistory	78.368	787.868	101	34,00%
Order	1.079.800	724.777	1.481	-2,00%
OrderScheduledDate	24.736	621.743	40	58,00%
pt_waitstatshistory	140.472	562.819	254	49,00%
Location	47.264	508.334	95	92,00%
TNT_Import	55.880	243.774	234	0,00%
Discount	29.760	224.943	133	49,00%
BlindTypePriceTable	8.592	188.266	45	49,00%
Fabric_BlindType	11.544	167.964	68	93,00%
BlindTypeSync	6.928	166.547	41	84,00%
pt_queryhistory_writes	50.504	132.800	388	36,00%
pt_queryhistory_cnt	51.208	132.743	394	36,00%
pt_queryhistory_cpu	49.096	131.572	380	34,00%
pt_queryhistory_reads	49.544	131.251	384	35,00%
pt_queryhistory_time	48.904	131.194	380	34,00%
pt_sessionhistory_h	10.848	130.382	84	52,00%
HistoryProductionDocument	168.720	126.228	1.367	0,00%

SQL Table Name	Reserved (KB)	Used (KB)	Unusedpages (KB)	Data Used (KB)	Indexes Used (KB)	LOB Used (KB)
EDILog	38.414.032	38.402.784	11.248	38.251.816	150.968	37.527.784
aStockTransaction	21.436.672	21.436.416	256	16.130.408	5.306.008	-
aOrderDetail	11.974.040	11.973.176	864	11.252.920	720.256	596.952
aOrderDetail_Fabric	8.837.880	8.837.600	280	6.074.872	2.762.728	-
StockTransaction	6.880.400	6.879.880	520	5.193.192	1.686.688	-
OrderDetail	4.561.456	4.373.424	188.032	4.136.504	236.920	727.648
OrderDetail_Fabric	2.948.352	2.948.080	272	2.020.528	927.552	-
EDIOrderDetails	2.801.808	2.801.728	80	2.735.528	66.200	-
aOrder	2.769.256	2.766.088	3.168	2.386.328	379.760	243.456
aOrderStatus_Change	2.296.584	2.296.528	56	2.292.824	3.704	-
EDIOptionList	2.279.080	2.278.968	112	1.956.072	322.896	-
aOrderAmend	2.217.824	2.217.592	232	1.542.616	674.976	-
aAddress	1.543.712	1.543.560	152	1.413.768	129.792	-
ProcessTransaction	1.236.296	1.236.152	144	661.056	575.096	-
Order	1.079.800	1.048.616	31.184	924.960	123.656	143.072
OrderStatus_Change	1.035.120	1.034.992	128	594.528	440.464	-
aHistoryOrderDetailAmend	991.480	991.464	16	989.808	1.656	-
aProcessTransaction	907.152	907.144	8	905.656	1.488	-
aHistoryOrderAmend	891.800	891.528	272	647.344	244.184	-
OrderAmend	722.496	722.376	120	510.000	212.376	-
DECORA_Kitting_History	581.088	542.784	38.304	542.560	224	-
aScheduling	575.912	575.760	152	346.936	228.824	-
aSerialDetailLine	567.376	567.232	144	472.768	94.464	-
EDISenderinformation	510.080	509.984	96	331.536	178.448	-
Address	414.768	414.656	112	330.960	83.696	-
EDIOrderHeader	363.064	362.984	80	312.568	50.416	-
HistoryOrderDetailAmend	352.544	352.448	96	304.184	48.264	-
HistoryOrderAmend	266.544	266.504	40	191.864	74.640	-
pt_sessionhistory	257.920	257.816	104	123.472	134.344	-
SerialDetailLine	237.408	237.344	64	198.128	39.216	-
Scheduling	219.152	218.992	160	113.120	105.872	-
HistoryProductionDocument	168.720	168.600	120	168.304	296	-
pt_commands	160.600	160.224	376	154.800	5.424	4.992
aCuttingOrderDetail	144.720	144.648	72	97.280	47.368	-
pt_waitstatshistory	140.472	139.936	536	93.968	45.968	-
pt_cachehistory	78.368	78.336	32	58.280	20.056	-
aOrderScheduledDate	74.832	74.728	104	47.784	26.944	-
CuttingOrderDetail	66.848	66.760	88	49.032	17.728	-
TNT_Import	55.880	55.736	144	55.728	8	-
pt_queryhistory_cnt	51.208	51.080	128	37.560	13.520	-
pt_queryhistory_writes	50.504	50.424	80	37.208	13.216	-
pt_queryhistory_reads	49.544	49.296	248	36.584	12.712	-
EDIReceivedTransactionsLog	49.512	41.176	8.336	41.128	48	34.208
pt_queryhistory_cpu	49.096	48.864	232	36.472	12.392	-
pt_queryhistory_time	48.904	48.688	216	36.376	12.312	-
Location	47.264	47.232	32	24.640	22.592	-
User	46.840	45.376	1.464	43.992	1.384	31.416
Fabric	46.648	46.128	520	42.544	3.584	6.272
Discount	29.760	29.248	512	19.592	9.656	-
OrderScheduledDate	24.736	24.704	32	15.608	9.096	-

Memory

Counter	Allocated (MBs)
Total machine memory	148.479
Available machine memory	4.463
Target server memory	137.582
Total server memory	137.584
Database buffers	121.547
Lock manager buffers	5.838
Stolen (incl. Procedure cache) buffers	15.838
Free buffers	198

Memory distribution

Database Name	Buffers	Size (KB)
DecoraERP	10.509.928	84.079.424
Blindata	4.706.673	37.653.384
tempdb	229.686	1.837.488
Decora2017CU14	66.244	529.952
msdb	3.933	31.464
MsSqlSystemResource	2.335	18.680
master	235	1.880
model	30	240

Cache Analysis

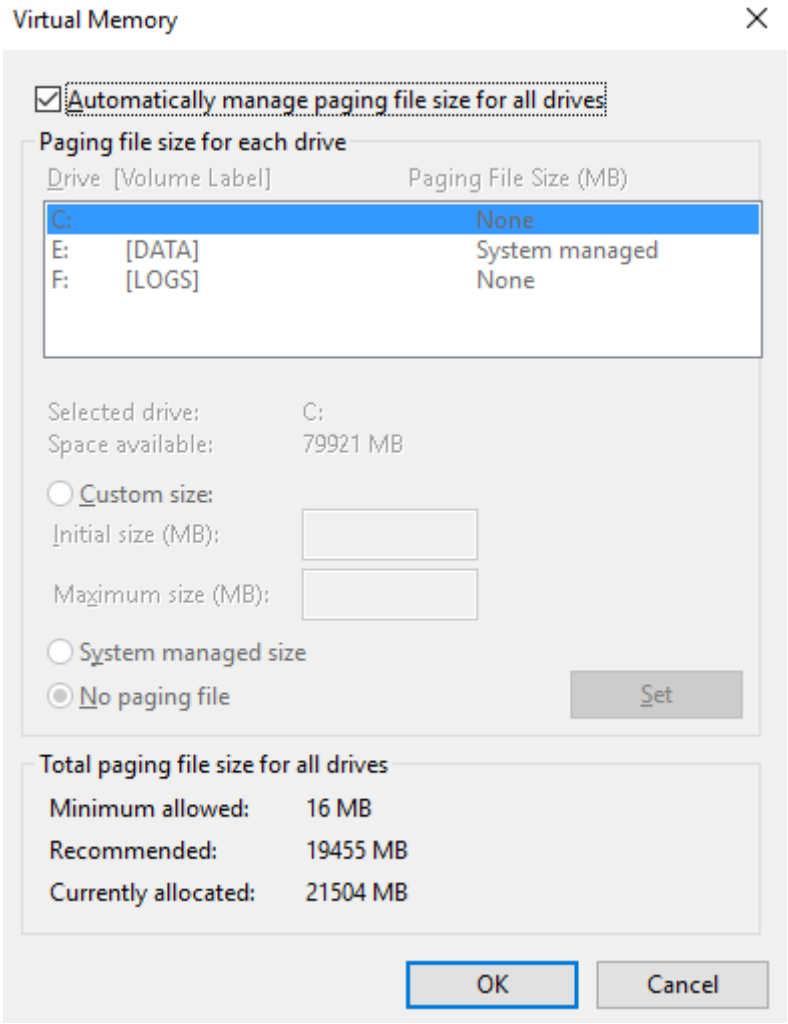
This table shows how often a table can be read from cache instead of from disk. A table read from cache receives more points than a table read from disk.

Total Count	5.497

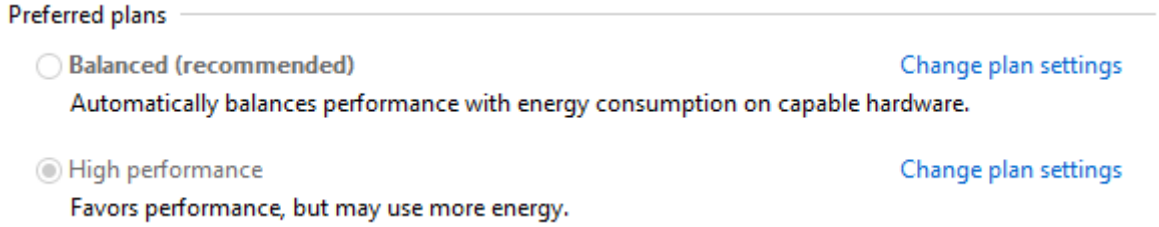
% occurrance	No. of occurrances	Object Name	Importance
96,74%	5.318	aOrderDetail$1	50.132
100,00%	5.497	OrderDetail$1	43.092
96,13%	5.284	aOrder$1	31.112
99,05%	5.445	EDILog$1	26.500
97,74%	5.373	StockTransaction$1	24.331
94,00%	5.167	EDIOptionList$1	20.897
98,80%	5.431	EDIOrderDetails$1	20.145
35,46%	1.949	aStockTransaction$1	15.634
52,36%	2.878	aOrderStatus_Change$1	13.070
100,00%	5.497	Order$1	12.898
70,55%	3.878	aAddress$1	10.841
100,00%	5.497	ProcessTransaction$1	8.154
49,48%	2.720	aStockTransaction$2	4.624
96,87%	5.325	OrderDetail_Fabric$1	3.749
30,76%	1.691	aOrderDetail_Fabric$5	3.204
41,22%	2.266	aHistoryOrderDetailAmend$1	2.405
99,87%	5.490	OrderStatus_Change$1	2.274
93,52%	5.141	OrderAmend$1	1.961
94,20%	5.178	OrderDetail_Fabric$2	1.362
99,18%	5.452	EDISenderinformation$1	1.050
79,48%	4.369	aSerialDetailLine$1	766
69,89%	3.842	ProcessTransaction$2	758
99,93%	5.493	Address$1	585
93,25%	5.126	StockTransaction$4	554
99,20%	5.453	EDIOrderHeader$1	507
55,98%	3.077	DECORA_Kitting_History$0	500
100,00%	5.497	SerialDetailLine$1	408
9,77%	537	aProcessTransaction$1	206
100,00%	5.497	Scheduling$1	137
70,80%	3.892	StockTransaction$30	137
29,62%	1.628	aOrderDetail_Fabric$1	90
99,16%	5.451	EDISenderinformation$9	75
92,45%	5.082	aOrderDetail$2	40
100,00%	5.497	Fabric$1	29
100,00%	5.497	OrderDetail$102	27
99,09%	5.447	OrderStatus_Change$6	24
89,61%	4.926	HistoryOrderAmend$1	18
99,98%	5.496	Scheduling$12	15
100,00%	5.497	sysobjvalues$1	13
100,00%	5.497	User$1	6
87,17%	4.792	sysmultiobjrefs$1	3
90,79%	4.991	syscolpars$1	2
12,01%	660	aScheduling$1	0
95,27%	5.237	EDIOrderDetails$43	0
90,65%	4.983	HistoryOrderDetailAmend$1	0
96,43%	5.301	Order$89	0

Generic settings

Page file

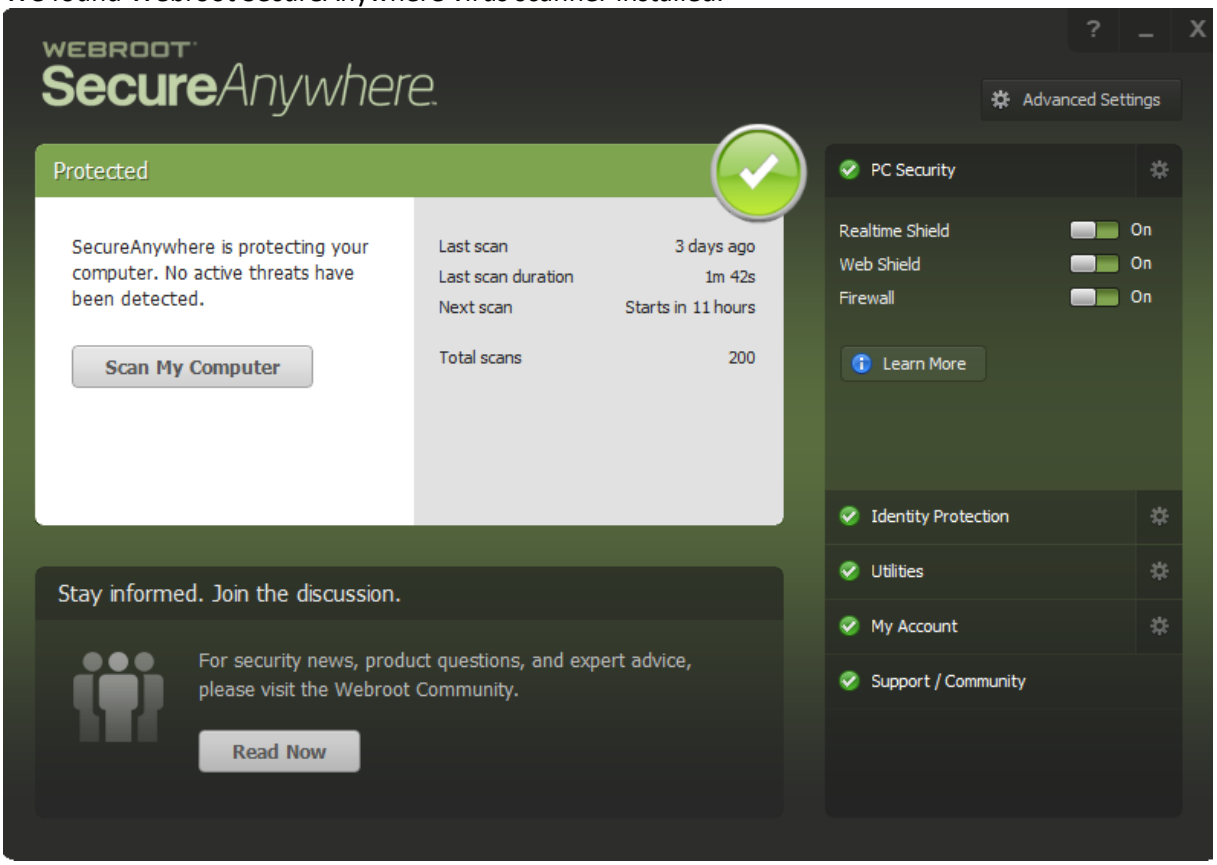


Power plan



Virus scanner

We found Webroot SecureAnywhere virus scanner installed.



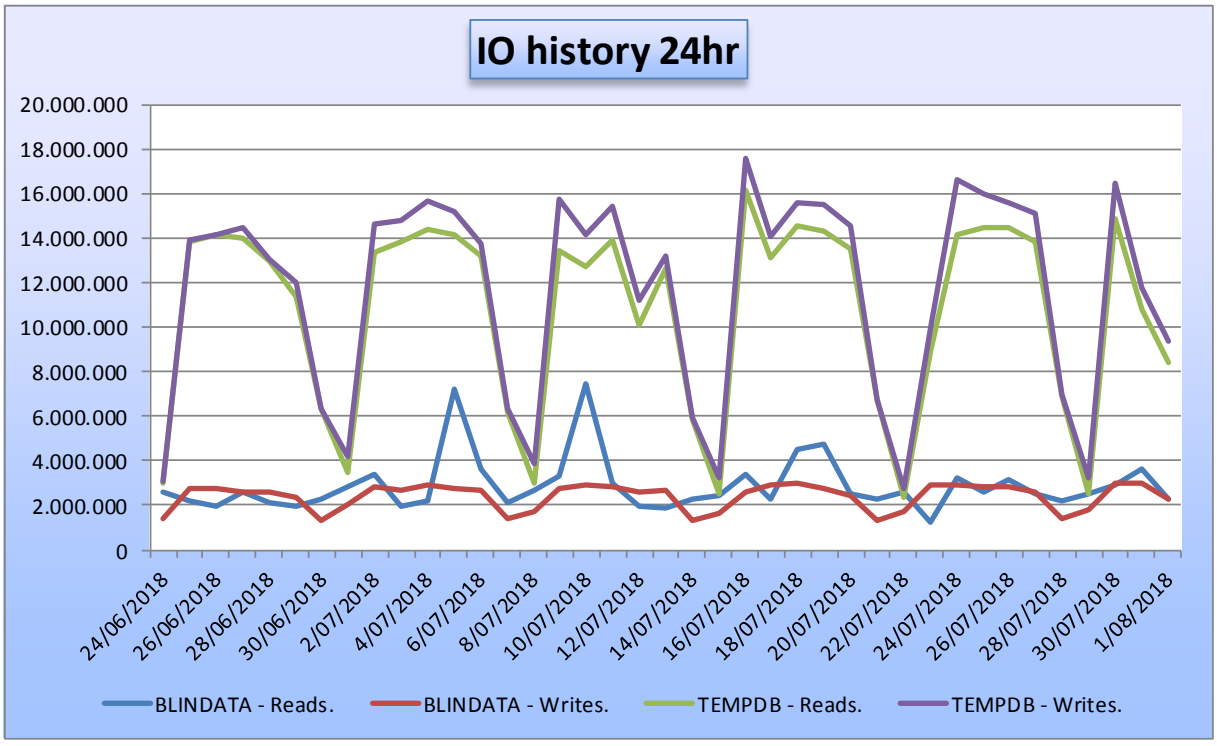
Perform Volume Maintenance Tasks

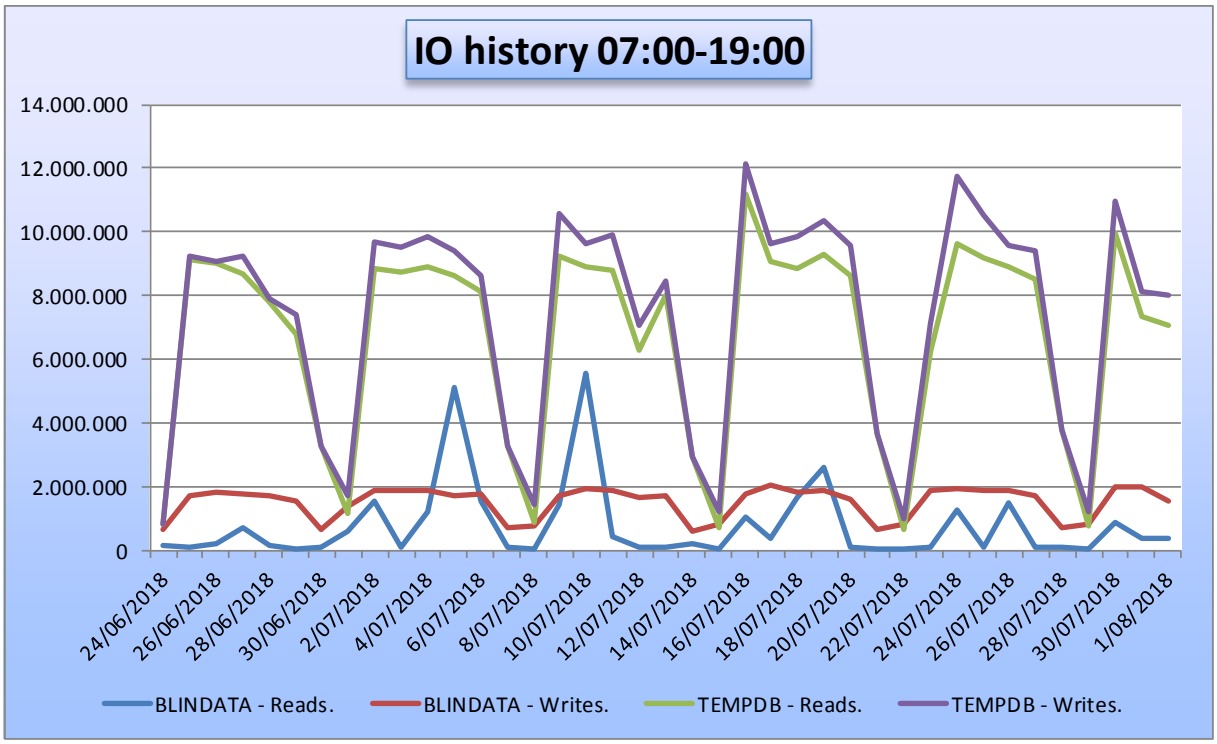
Policy	Security Setting
Deny log on as a service	
Deny log on locally	
Deny log on through Remote Desktop Services	
Enable computer and user accounts to be trusted for delega...	
Force shutdown from a remote system	Administrators
Generate security audits	LOCAL SERVICE,NETWORK SERVICE,IIS APPPOOL\.NET v4.5,IIS APPPOOL\DefaultAppPo...
Impersonate a client after authentication	LOCAL SERVICE,NETWORK SERVICE,Administrators,IIS_IUSRS,SERVICE
Increase a process working set	Users
Increase scheduling priority	Administrators
Load and unload device drivers	Administrators
Lock pages in memory	DECORA\sqladmin
Log on as a batch job	Administrators,Backup Operators,Performance Log Users,IIS_IUSRS
Log on as a service	DECORA\Citrix Users,DECORA\besadmin,DECORA\Administrator,Administrators,NT SER...
Manage auditing and security log	Administrators
Modify an object label	
Modify firmware environment values	Administrators
Obtain an impersonation token for another user in the same...	Administrators
Perform volume maintenance tasks	DECORA\sqladmin,Administrators
Profile single process	Administrators
Profile system performance	Administrators,NT SERVICE\WdiServiceHost
Remove computer from docking station	Administrators
Replace a process level token	LOCAL SERVICE,NETWORK SERVICE,NT SERVICE\SQLSERVERAGENT,*S-1-5-80-347704441...
Restore files and directories	Administrators,Backup Operators
Shut down the system	Administrators,Backup Operators

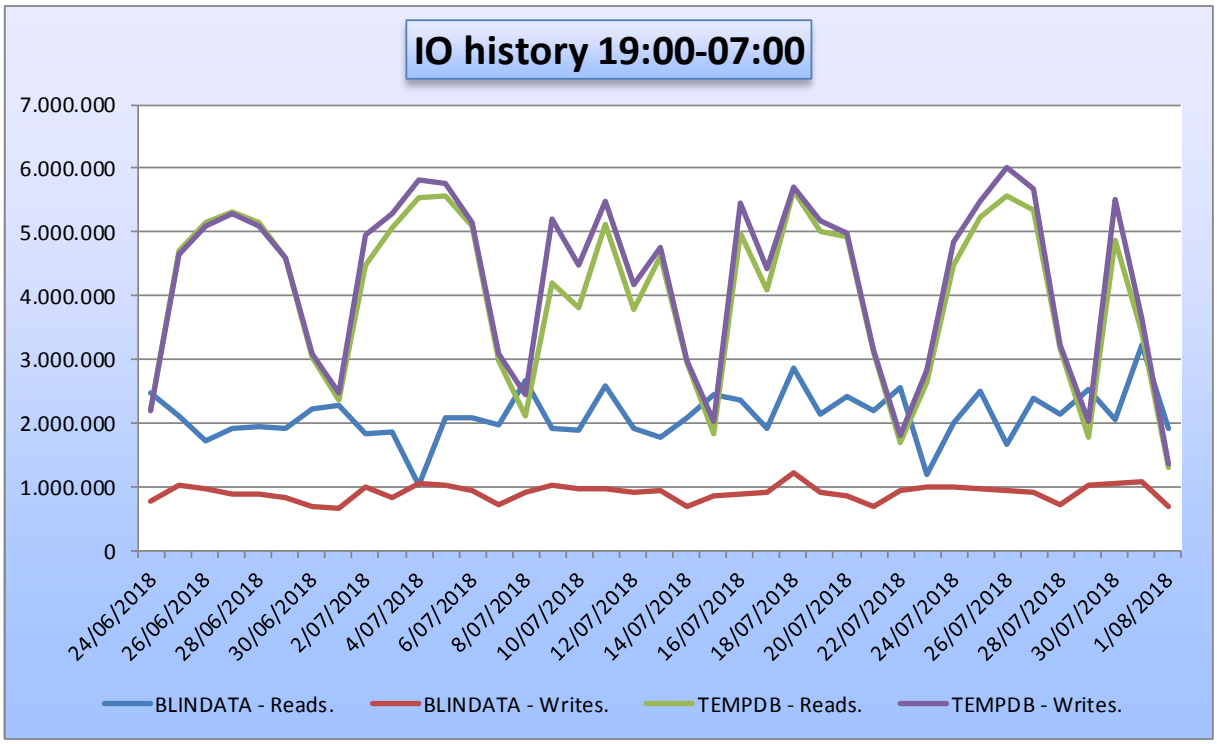
Performance analyses

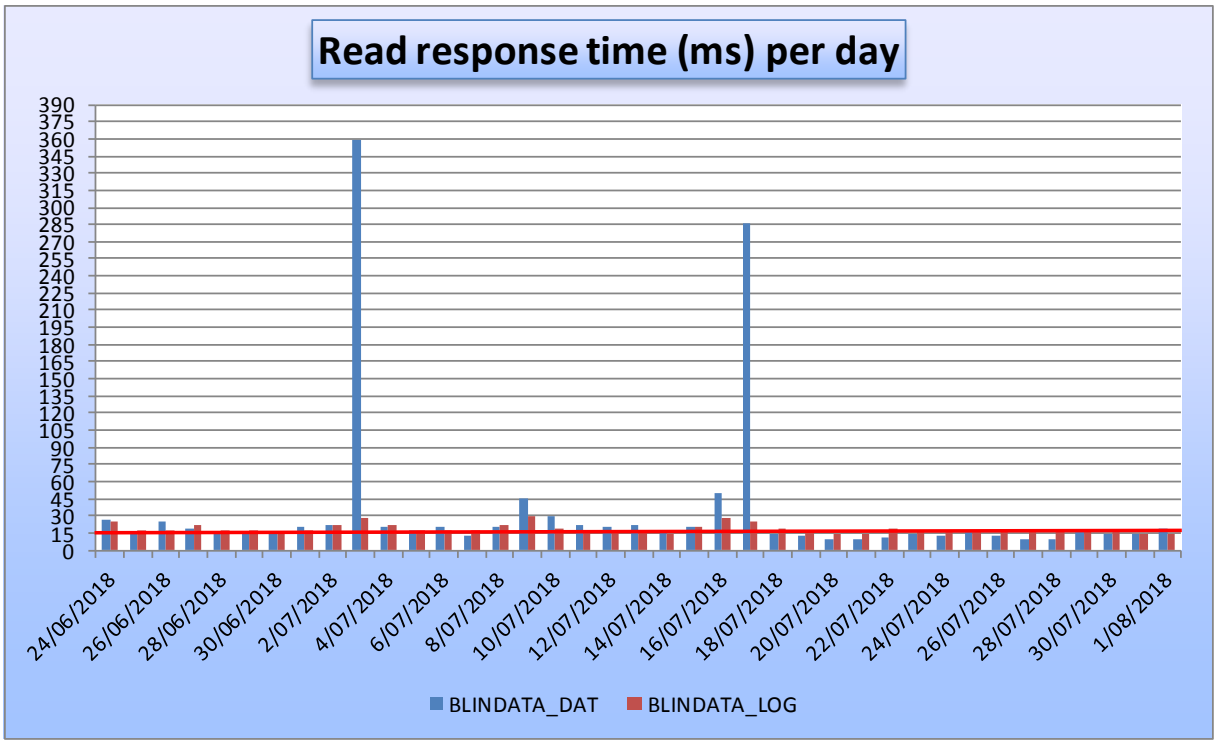
Disk response times & IO requests

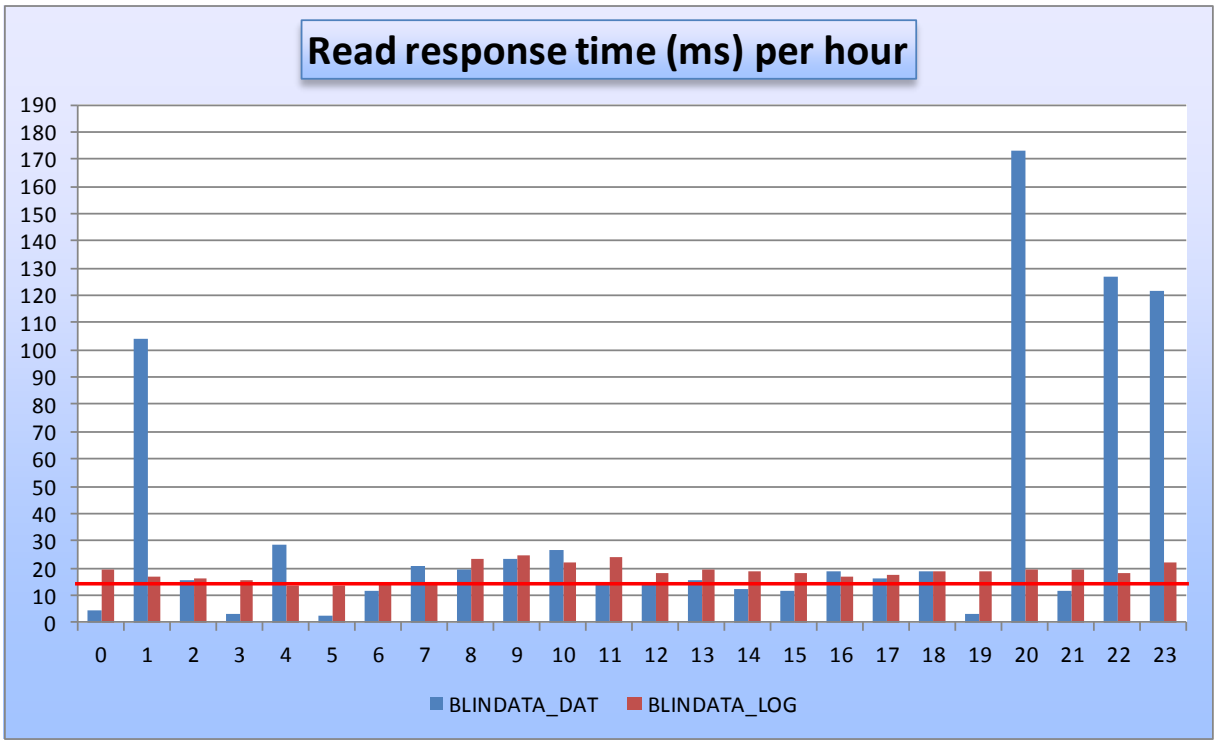
DB Name	Logical Name	%IOs	%Reads	%Writes	Avg. Read Wait (ms)	Avg. Write Wait (ms)	Total IOs	Read %	Write %
master	master	0	0	0	4,25	32,56	7.569,00	42	57
master	mastlog	0	0	0	3,85	11,82	40.264,00	0	99
tempdb	**tempdev**	**10**	**10**	**11**	**9,65**	**171,24**	**65.571.626,00**	**47**	**52**
tempdb	**templog**	**0**	**0**	**0**	**1,69**	**21,63**	**2.853.772,00**	**0**	**99**
tempdb	**temp2**	**10**	**10**	**11**	**9,61**	**171,83**	**65.388.572,00**	**48**	**51**
tempdb	**temp3**	**10**	**10**	**11**	**9,59**	**170,73**	**65.364.453,00**	**48**	**51**
tempdb	**temp4**	**10**	**10**	**11**	**9,66**	**170,82**	**65.385.806,00**	**48**	**51**
tempdb	**temp5**	**10**	**10**	**11**	**9,74**	**172,48**	**65.390.580,00**	**48**	**51**
tempdb	**temp6**	**10**	**10**	**11**	**9,66**	**172,28**	**65.415.133,00**	**48**	**51**
model	**modeldev**	**0**	**0**	**0**	**5,23**	**14,67**	**775,00**	**95**	**4**
model	modellog	0	0	0	3,52	8,72	178,00	18	81
msdb	MSDBData	0	0	0	6,29	31,43	670.940,00	31	68
msdb	MSDBLog	0	0	0	1,20	8,71	2.104.082,00	0	99
Blindata	**BlinData_dat**	**9**	**15**	**2**	**8,03**	**22,70**	**55.896.832,00**	**86**	**13**
Blindata	**BlinData_log**	**5**	**0**	**11**	**17,45**	**6,74**	**34.416.276,00**	**0**	**99**
DecoraERP	DecoraERP_Data	0	0	0	6,56	82,42	608.991,00	31	68
DecoraERP	DecoraERP_Log	3	0	6	14,78	6,84	20.743.960,00	1	98
DecoraERP	DecoraERP_1_Data	10	15	5	16,25	218,54	64.526.656,00	76	23
DecoraERP	DecoraERP_2_Data	4	6	2	16,62	215,92	26.423.073,00	74	25
Decora2017CU14	DecoraERP_Data	0	0	0	2,13	53,76	183.097,00	97	2
Decora2017CU14	DecoraERP_Log	0	0	0	2,86	15,89	129.452,00	0	99
Decora2017CU14	DecoraERP_1_Data	0	0	0	8,22	51,74	2.160.676,00	96	3
Decora2017CU14	DecoraERP_2_Data	0	0	0	7,61	50,85	821.343,00	95	4

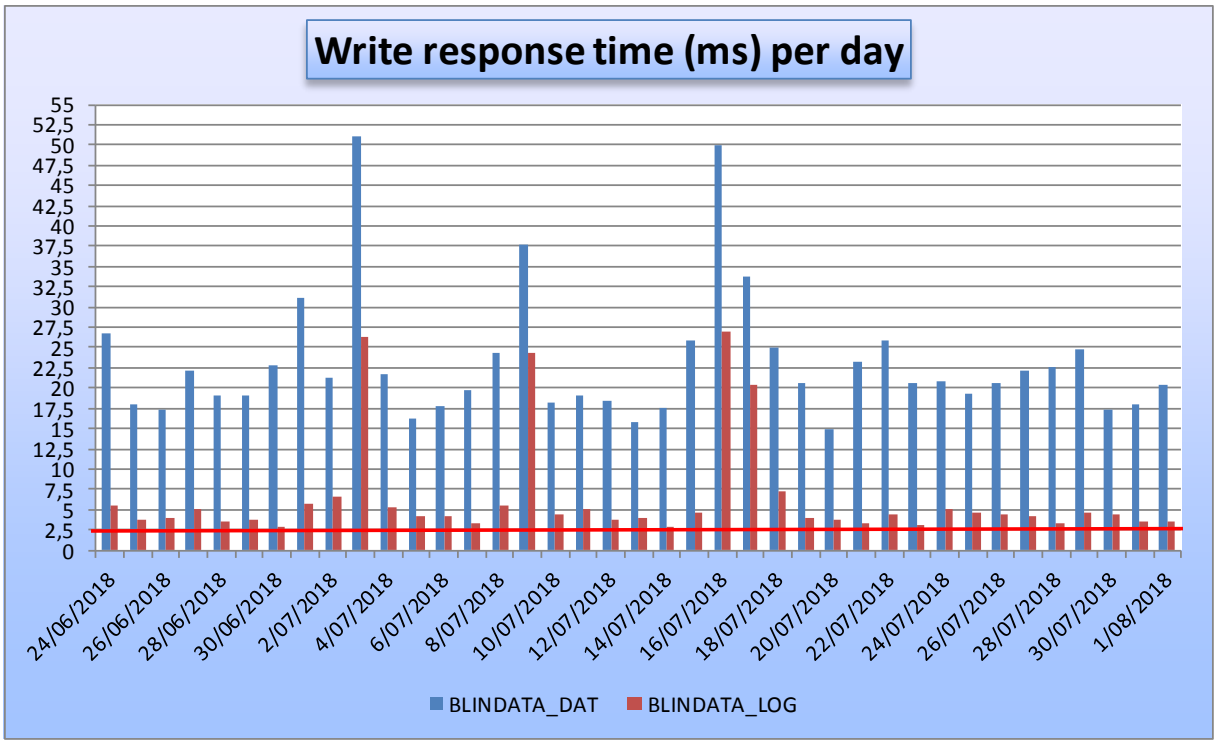


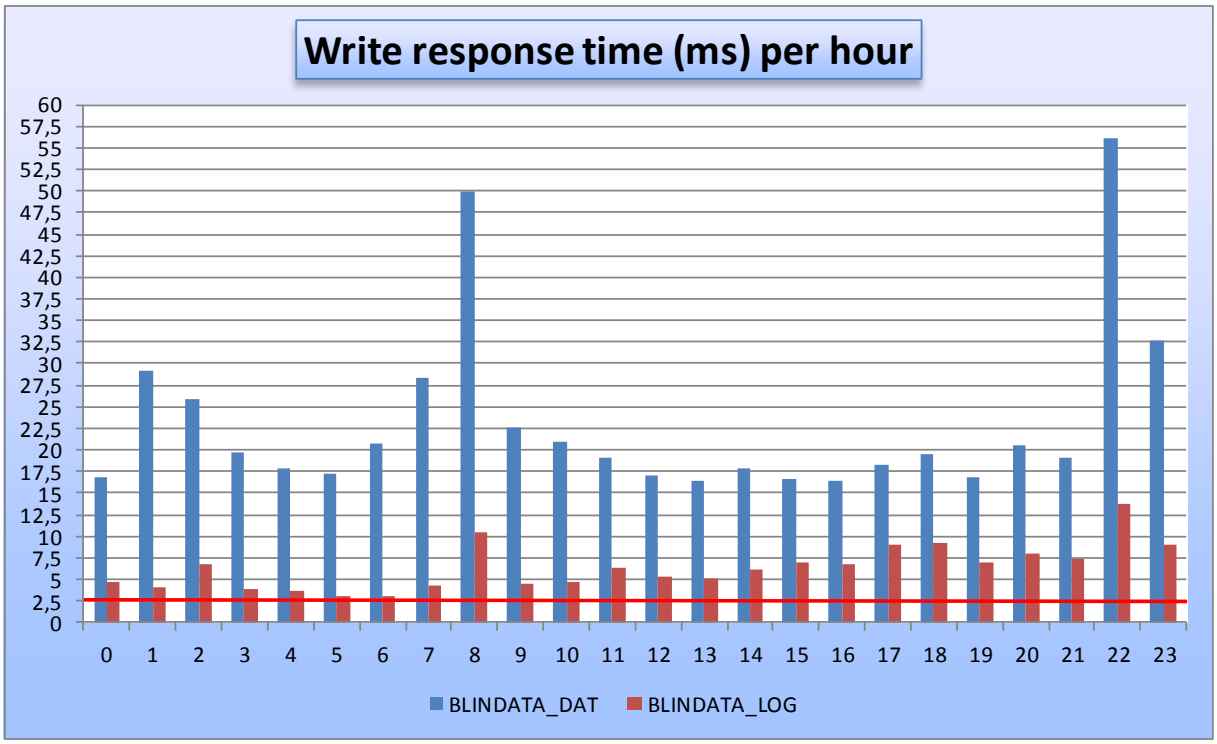

IO history 07:00-19:00
14.000.000
12.000.000
10.000.000
8.000.000
6.000.000
4.000.000
2.000.000
0
24/06/2018
26/06/2018
28/06/2018
30/06/2018
2/07/2018
4/07/2018
6/07/2018
8/07/2018
10/07/2018
12/07/2018
14/07/2018
16/07/2018
18/07/2018
20/07/2018
22/07/2018
24/07/2018
26/07/2018
28/07/2018
30/07/2018
1/08/2018
BLINDATA - Reads.
BLINDATA - Writes.
TEMPDB - Reads.
TEMPDB - Writes.

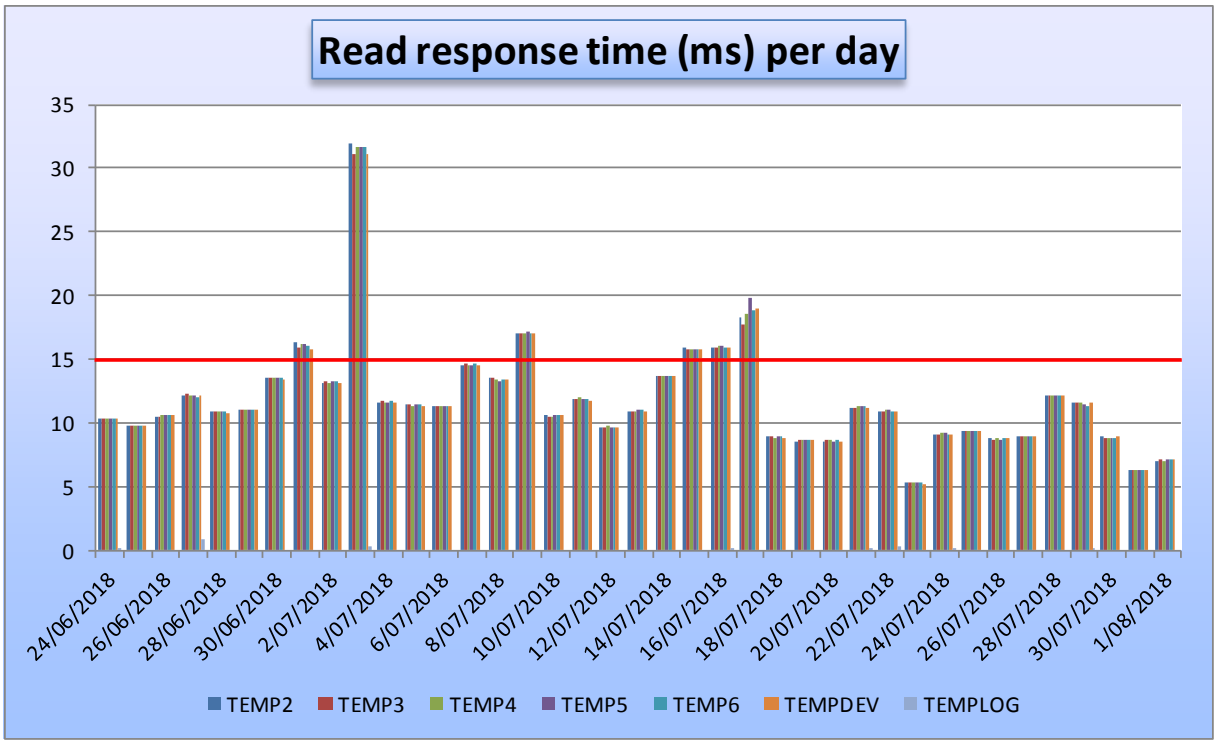

IO history 19:00-07:00
7.000.000
6.000.000
5.000.000
4.000.000
3.000.000
2.000.000
1.000.000
0
24/06/2018
26/06/2018
28/06/2018
30/06/2018
2/07/2018
4/07/2018
6/07/2018
8/07/2018
10/07/2018
12/07/2018
14/07/2018
16/07/2018
18/07/2018
20/07/2018
22/07/2018
24/07/2018
26/07/2018
28/07/2018
30/07/2018
1/08/2018
BLINDATA - Reads.
BLINDATA - Writes.
TEMPDB - Reads.
TEMPDB - Writes.

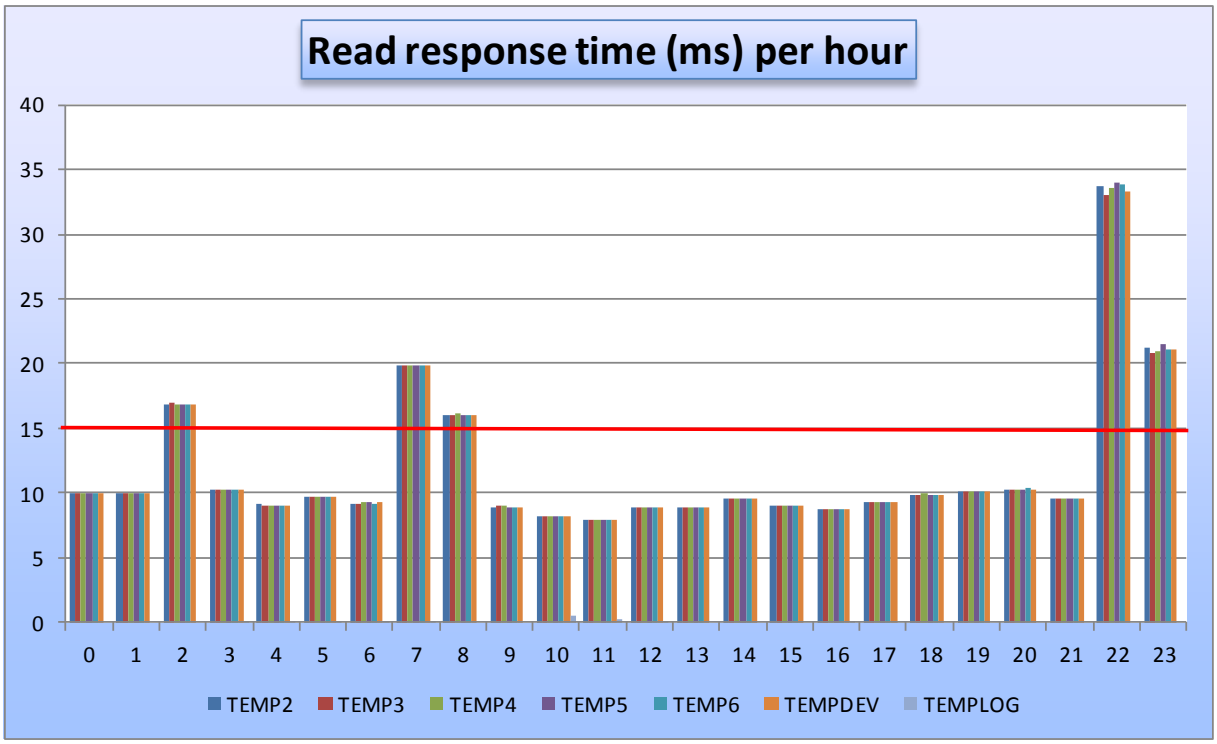

Read response time (ms) per day
390
375
360
345
330
315
300
285
270
255
240
225
210
195
180
165
150
135
120
105
90
75
60
45
30
15
0
24/06/2018
26/06/2018
28/06/2018
30/06/2018
2/07/2018
4/07/2018
6/07/2018
8/07/2018
10/07/2018
12/07/2018
14/07/2018
16/07/2018
18/07/2018
20/07/2018
22/07/2018
24/07/2018
26/07/2018
28/07/2018
30/07/2018
1/08/2018
BLINDATA_DAT
BLINDATA_LOG

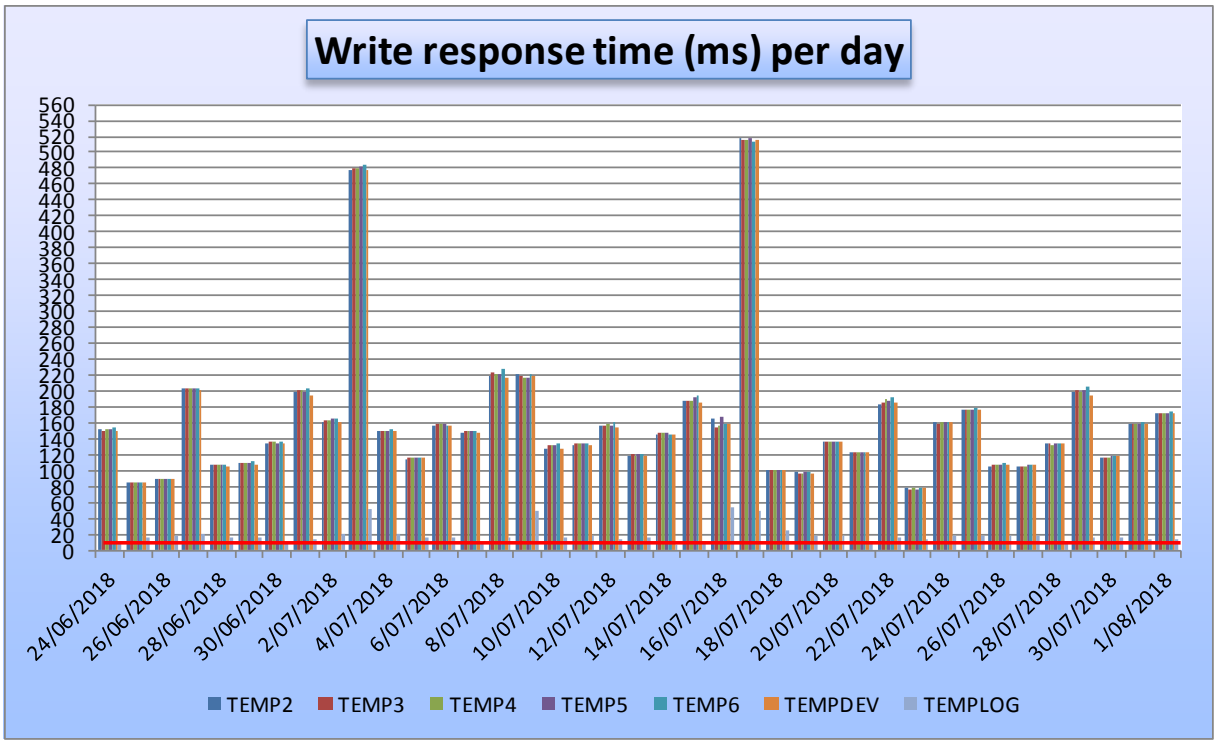

Read response time (ms) per hour
190
180
170
160
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
BLINDATA_DAT
BLINDATA_LOG

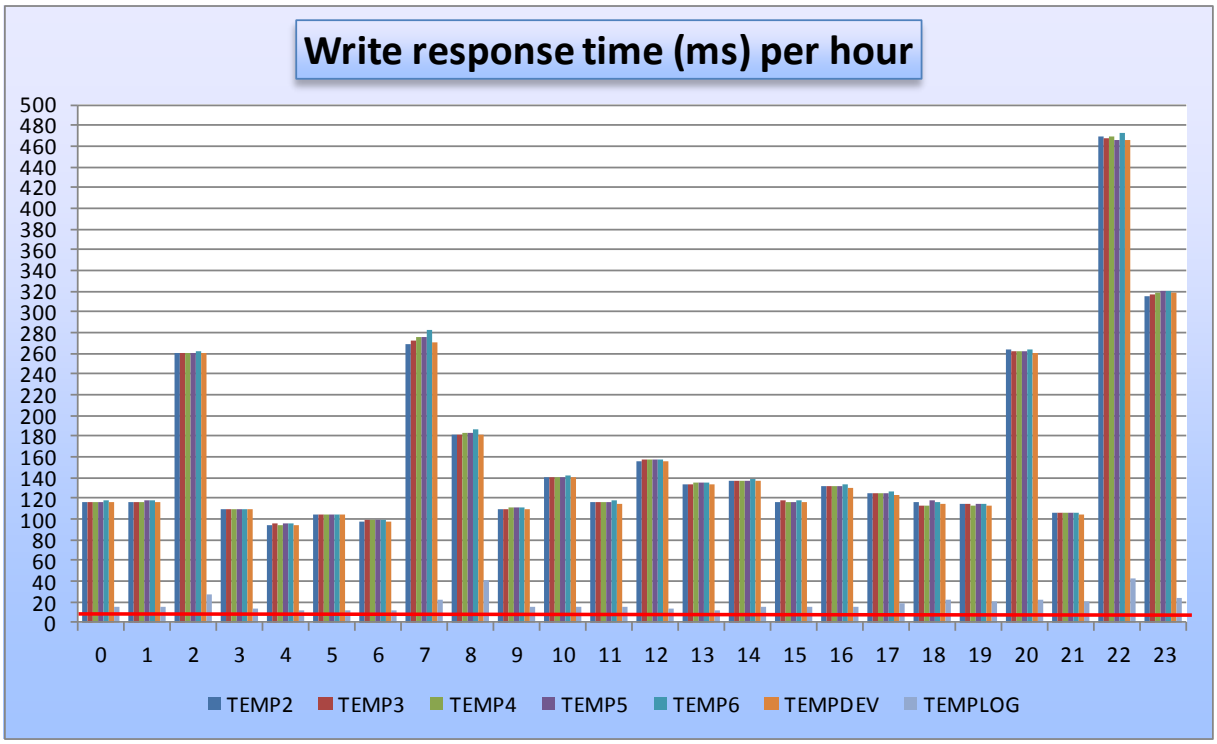

Write response time (ms) per day
55
52,5
50
47,5
45
42,5
40
37,5
35
32,5
30
27,5
25
22,5
20
17,5
15
12,5
10
7,5
5
2,5
0
24/06/2018
26/06/2018
28/06/2018
30/06/2018
2/07/2018
4/07/2018
6/07/2018
8/07/2018
10/07/2018
12/07/2018
14/07/2018
16/07/2018
18/07/2018
20/07/2018
22/07/2018
24/07/2018
26/07/2018
28/07/2018
30/07/2018
1/08/2018
BLINDATA_DAT
BLINDATA_LOG


Write response time (ms) per hour
60
57,5
55
52,5
50
47,5
45
42,5
40
37,5
35
32,5
30
27,5
25
22,5
20
17,5
15
12,5
10
7,5
5
2,5
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
BLINDATA_DAT
BLINDATA_LOG

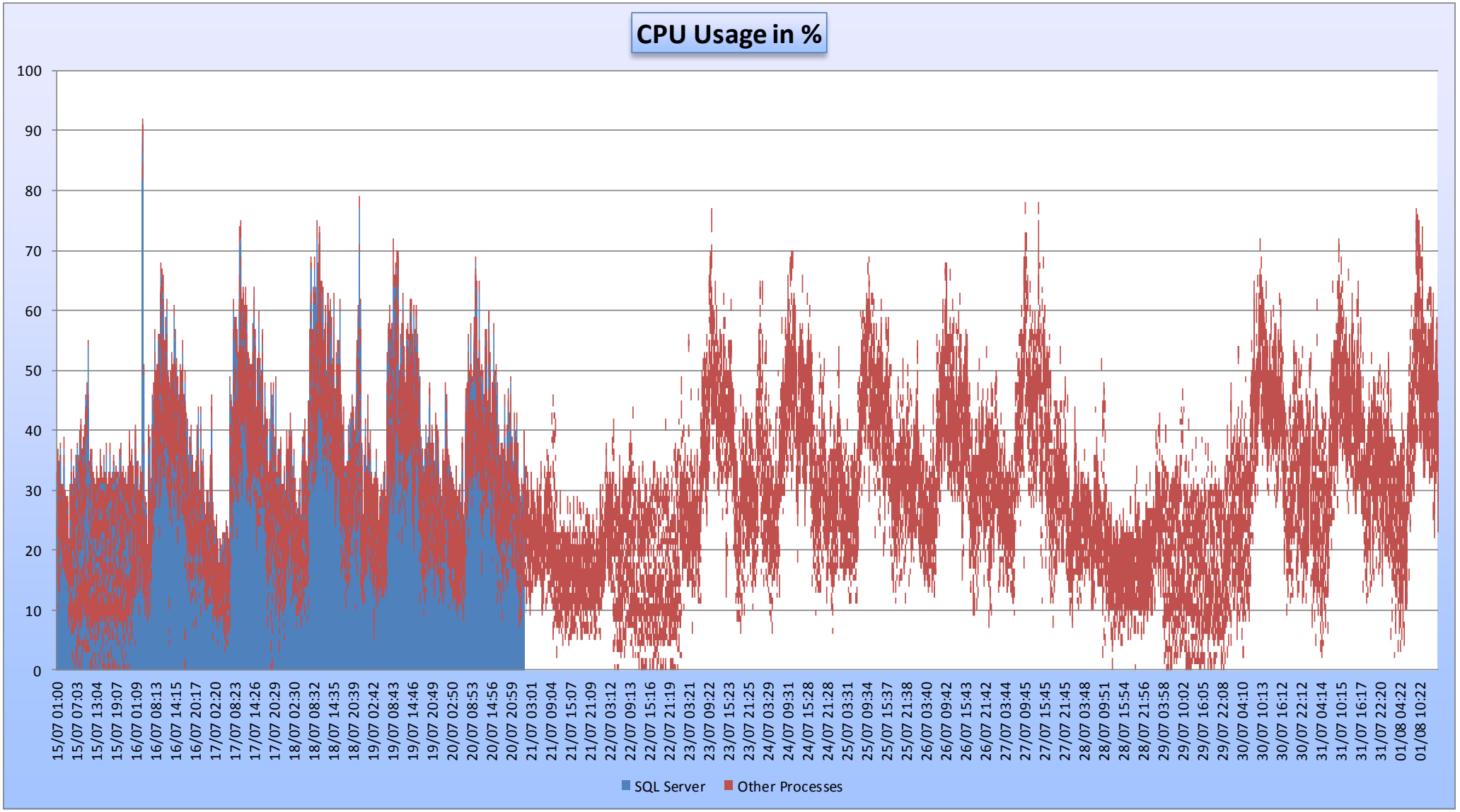

Read response time (ms) per day
35
30
25
20
15
10
5
0
24/06/2018
26/06/2018
28/06/2018
30/06/2018
2/07/2018
4/07/2018
6/07/2018
8/07/2018
10/07/2018
12/07/2018
14/07/2018
16/07/2018
18/07/2018
20/07/2018
22/07/2018
24/07/2018
26/07/2018
28/07/2018
30/07/2018
1/08/2018
TEMP2
TEMP3
TEMP4
TEMP5
TEMP6
TEMPDEV
TEMPLOG


Read response time (ms) per hour
40
35
30
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
TEMP2
TEMP3
TEMP4
TEMP5
TEMP6
TEMPDEV
TEMPLOG


Write response time (ms) per day
560
540
520
500
480
460
440
420
400
380
360
340
320
300
280
260
240
220
200
180
160
140
120
100
80
60
40
20
0
24/06/2018
26/06/2018
28/06/2018
30/06/2018
2/07/2018
4/07/2018
6/07/2018
8/07/2018
10/07/2018
12/07/2018
14/07/2018
16/07/2018
18/07/2018
20/07/2018
22/07/2018
24/07/2018
26/07/2018
28/07/2018
30/07/2018
1/08/2018
TEMP2
TEMP3
TEMP4
TEMP5
TEMP6
TEMPDEV
TEMPLOG


Write response time (ms) per hour
500
480
460
440
420
400
380
360
340
320
300
280
260
240
220
200
180
160
140
120
100
80
60
40
20
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
TEMP2
TEMP3
TEMP4
TEMP5
TEMP6
TEMPDEV
TEMPLOG

CPU


CPU Usage in %
100
90
80
70
60
50
40
30
20
10
0
15/07 01:00
15/07 07:03
15/07 13:04
15/07 19:07
16/07 01:09
16/07 08:13
16/07 14:15
16/07 20:17
17/07 02:20
17/07 08:23
17/07 14:26
17/07 20:29
18/07 02:30
18/07 08:32
18/07 14:35
18/07 20:39
19/07 02:42
19/07 08:43
19/07 14:46
19/07 20:49
20/07 02:50
20/07 08:53
20/07 14:56
20/07 20:59
21/07 03:01
21/07 09:04
21/07 15:07
21/07 21:09
22/07 03:12
22/07 09:13
22/07 15:16
22/07 21:19
23/07 03:21
23/07 09:22
23/07 15:23
23/07 21:25
24/07 03:29
24/07 09:31
24/07 15:28
24/07 21:28
25/07 03:31
25/07 09:34
25/07 15:37
25/07 21:38
26/07 03:40
26/07 09:42
26/07 15:43
26/07 21:42
27/07 03:44
27/07 09:45
27/07 15:45
27/07 21:45
28/07 03:48
28/07 09:51
28/07 15:54
28/07 21:56
29/07 03:58
29/07 10:02
29/07 16:05
29/07 22:08
30/07 04:10
30/07 10:13
30/07 16:12
30/07 22:12
31/07 04:14
31/07 10:15
31/07 16:17
31/07 22:20
01/08 04:22
01/08 10:22
SQL Server
Other Processes

Waitstats

Wait Type	Requests	Signal Wait (ms)	Avg. Signal Wait (ms)	Resource Wait (ms)	Avg. Resource Wait (ms)	Total Wait (ms)	Avg. Total Wait (ms)	% Waits
CXPACKET	5.477.731.394	3.287.492.879	0,60	32.184.311.402	5,88	35.471.804.281	6,48	64,57
LATCH_EX	2.347.463.314	1.144.492.143	0,49	8.827.677.046	3,76	9.972.169.189	4,25	18,15
EXECSYNC	1.860.520	481.397	0,26	6.265.638.600	3.367,68	6.266.119.997	3.367,94	11,41
SOS_SCHEDULER_YIELD	1.011.950.694	1.494.300.688	1,48	861.292	0,00	1.495.161.980	1,48	2,72
WRITELOG	54.993.201	20.372.059	0,37	373.594.797	6,79	393.966.856	7,16	0,72
IO_COMPLETION	51.145.254	2.323.231	0,05	371.344.990	7,26	373.668.221	7,31	0,68
LCK_M_U	62.708	43.136	0,69	218.101.479	3.478,05	218.144.615	3.478,74	0,40
PAGEIOLATCH_SH	59.765.782	2.327.748	0,04	142.072.138	2,38	144.399.886	2,42	0,26
OLEDB	654.329.211	-	-	62.128.766	0,09	62.128.766	0,09	0,11
PAGEIOLATCH_EX	8.143.057	483.960	0,06	53.725.193	6,60	54.209.153	6,66	0,10
LCK_M_IS	2.612	1.553	0,59	52.489.408	20.095,49	52.490.961	20.096,08	0,10
PAGELATCH_EX	55.948.378	22.800.268	0,41	24.921.665	0,45	47.721.933	0,85	0,09
BACKUPIO	17.596.834	887.319	0,05	41.431.788	2,35	42.319.107	2,40	0,08
LCK_M_IX	15.230	9.050	0,59	38.365.648	2.519,08	38.374.698	2.519,68	0,07
SESSION_WAIT_STATS_CHILDREN	21.909.448	6.496.907	0,30	27.498.922	1,26	33.995.829	1,55	0,06
PAGEIOLATCH_UP	205.631	76.095	0,37	28.935.650	140,72	29.011.745	141,09	0,05
ASYNC_IO_COMPLETION	292	96	0,33	28.340.955	97.058,07	28.341.051	97.058,39	0,05
BACKUPBUFFER	18.129.170	1.674.080	0,09	26.529.905	1,46	28.203.985	1,56	0,05
SQLTRACE_FILE_BUFFER	64.664	33.297	0,51	18.488.587	285,92	18.521.884	286,43	0,03
TRANSACTION_MUTEX	1.628.626	316.753	0,19	18.015.313	11,06	18.332.066	11,26	0,03
RESERVED_MEMORY_ALLOCATION_EXT	13.904.510.221	-	-	17.060.175	0,00	17.060.175	0,00	0,03
LATCH_SH	243.284	170.520	0,70	15.150.844	62,28	15.321.364	62,98	0,03
MSQL_DQ	52.862	-	-	14.629.688	276,75	14.629.688	276,75	0,03
PREEMPTIVE_COM_QUERYINTERFACE	53.554	-	-	14.629.592	273,17	14.629.592	273,17	0,03
LCK_M_X	68.902	28.605	0,42	13.149.862	190,85	13.178.467	191,26	0,02
PAGELATCH_SH	845.604	312.944	0,37	11.573.641	13,69	11.886.585	14,06	0,02
PREEMPTIVE_OS_WRITEFILE	833.624	-	-	10.455.392	12,54	10.455.392	12,54	0,02
BACKUPTHREAD	24.668	10.640	0,43	9.696.990	393,10	9.707.630	393,53	0,02
LCK_M_S	543	283	0,52	6.886.692	12.682,67	6.886.975	12.683,20	0,01
LCK_M_SCH_S	364	81	0,22	6.492.564	17.836,71	6.492.645	17.836,94	0,01
CXROWSET_SYNC	4.353.439	754.105	0,17	4.471.210	1,03	5.225.315	1,20	0,01
LCK_M_IU	378	209	0,55	3.586.488	9.488,06	3.586.697	9.488,62	0,01
PREEMPTIVE_OS_AUTHENTICATIONOPS	8.356.218	-	-	3.209.371	0,38	3.209.371	0,38	0,01
PREEMPTIVE_OLEDBOPS	50.293	-	-	1.966.219	39,10	1.966.219	39,10	0,00
WRITE_COMPLETION	481.412	1.902.951	3,95	3.876	0,01	1.906.827	3,96	0,00
MSQL_XP	281.297	-	-	1.717.457	6,11	1.717.457	6,11	0,00
PAGELATCH_UP	522.901	257.572	0,49	1.413.987	2,70	1.671.559	3,20	0,00
SQLTRACE_FILE_READ_IO_COMPLETION	98.359	5.063	0,05	1.502.692	15,28	1.507.755	15,33	0,00
LCK_M_SCH_M	75.477	15.324	0,20	1.444.722	19,14	1.460.046	19,34	0,00
PREEMPTIVE_XE_GETTARGETSTATE	75.352	-	-	1.292.591	17,15	1.292.591	17,15	0,00
PREEMPTIVE_OS_DELETESECURITYCONTEXT	1.460.874	-	-	819.603	0,56	819.603	0,56	0,00
PREEMPTIVE_OS_WRITEFILEGATHER	184	-	-	600.795	3.265,19	600.795	3.265,19	0,00
CMEMTHREAD	2.593.409	540.049	0,21	56.216	0,02	596.265	0,23	0,00
PREEMPTIVE_OS_DISCONNECTNAMEDPIPE	1.276.012	-	-	579.351	0,45	579.351	0,45	0,00
PREEMPTIVE_OS_AUTHORIZATIONOPS	1.571.956	-	-	553.679	0,35	553.679	0,35	0,00
PREEMPTIVE_OS_REVERTTOSELF	1.570.958	-	-	362.448	0,23	362.448	0,23	0,00
PREEMPTIVE_OS_FILEOPS	15.382	-	-	330.033	21,46	330.033	21,46	0,00
LOGBUFFER	5.990	2.351	0,39	228.091	38,08	230.442	38,47	0,00
PREEMPTIVE_OS_CRYPTOPS	355.989	-	-	186.997	0,53	186.997	0,53	0,00
PREEMPTIVE_OS_CREATEFILE	290.236	-	-	184.182	0,63	184.182	0,63	0,00

Database Load

Database Name	Data Size (MB)	Data Size %	Log Size (MB)	Log Size %	Total # of I/Os	I/O %	Total I/O MBs	I/O MBs %	Total CPU (s)	Total CPU %	Cache Size (MB)	Cache %
DecoraERP	402673	42,72	26799	20,74	112299880	18,59	10378374	26,9	2123	13,28	82107	**67,73**
Decora2017CU14	366859	38,92	22799	17,65	3294567	0,55	179493	0,47	0	0	518	0,43
Blindata	133978	14,22	77250	59,8	90312960	14,95	3373843	8,75	13857	86,64	36772	**30,34**
tempdb	38800	4,12	2300	1,78	395363389	65,45	24630112	63,85	0	0	1787	1,47
msdb	179	0,02	28	0,02	2774984	0,46	14537	0,04	1	0,01	34	0,03
model	8	0	8	0,01	953	0	16	0	0	0	0	0
master	5	0	2	0	47833	0,01	242	0	12	0,08	2	0

Indexes

During posting routines the tables and indexes need updating. The more indexes and index keys, the heavier the transaction.

By reducing the number of indexes and index keys, better performance of posting routines can be achieved.

SQL Name	Rows	Updates	Indexes	IndexKeys	IndexCost1	IndexCost	IndexKeyCost1	IndexKeyCost
Fabric	56281	7.355.768	4	4	225.124	29.423.072	225.124	29423072
OrderDetail	1955715	661.483	9	9	17.601.435	5.953.347	17.601.435	5953347
pt_waitstatshistory	562819	843.539	3	7	1.688.457	2.530.617	3.939.733	5904773
OrderDetail_Fabric	28058619	1.299.001	3	4	84.175.857	3.897.003	112.234.476	5196004
OrderDetailOptions	1629	1.550.875	2	3	3.258	3.101.750	4.887	4652625
StockTransaction	25312478	762.256	5	6	126.562.390	3.811.280	151.874.868	4573536
pt_sessionhistory	1251214	228.446	4	18	5.004.856	913.784	22.521.852	4112028
Table_Block	29	756.586	3	4	87	2.269.758	116	3026344
EDIOrderDetails	4393706	1.105.827	2	2	8.787.412	2.211.654	8.787.412	2211654
Order	724777	154.033	12	13	8.697.324	1.848.396	9.422.101	2002429
EDIOrderHeader	1772876	595.547	3	3	5.318.628	1.786.641	5.318.628	1786641
ProcessTransaction	8411625	200.872	4	8	33.646.500	803.488	67.293.000	1606976
pt_sessionhistory_temp	82299	159.245	2	8	164.598	318.490	658.392	1273960
TransactionLogForOnline	56547	338.336	3	3	169.641	1.015.008	169.641	1015008
UserActivityDetail	98	670.171	1	1	98	670.171	98	670171
Scheduling	1760226	90.589	5	5	8.801.130	452.945	8.801.130	452945
User	6116	49.842	8	9	48.928	398.736	55.044	448578
SerialDetailLine	2739366	198.508	2	2	5.478.732	397.016	5.478.732	397016
OrderStatus_Change	14519518	124.160	3	3	43.558.554	372.480	43.558.554	372480
pt_cachehistory	787868	50.300	2	5	1.575.736	100.600	3.939.340	251500
OrderAmend	7164350	69.553	3	3	21.493.050	208.659	21.493.050	208659
pt_queryhistory_time_temp	500	196.992	1	1	500	196.992	500	196992
pt_queryhistory_writes_temp	500	196.956	1	1	500	196.956	500	196956
pt_queryhistory_cpu_temp	500	196.951	1	1	500	196.951	500	196951
pt_queryhistory_reads_temp	500	196.884	1	1	500	196.884	500	196884
ManLocationDatesAndTimes	22619	55.486	2	3	45.238	110.972	67.857	166458
pt_queryhistory_cnt	132743	8.400	5	16	663.715	42.000	2.123.888	134400
pt_queryhistory_writes	132800	8.400	5	16	664.000	42.000	2.124.800	134400
EDIOptionList	22781564	67.148	2	2	45.563.128	134.296	45.563.128	134296
pt_queryhistory_cpu	131572	8.393	5	16	657.860	41.965	2.105.152	134288
pt_queryhistory_time	131194	8.388	5	16	655.970	41.940	2.099.104	134208
pt_queryhistory_reads	131251	8.198	5	16	656.255	40.990	2.100.016	131168
EDILog	10566509	61.032	2	2	21.133.018	122.064	21.133.018	122064
pt_queryhistory_cnt_temp	500	113.696	1	1	500	113.696	500	113696
pt_logsizehistory	39699	7.901	3	11	119.097	23.703	436.689	86911
HistoryOrderAmend	2558785	26.896	3	3	7.676.355	80.688	7.676.355	80688
HistoryOrderDetailAmend	3364178	36.081	2	2	6.728.356	72.162	6.728.356	72162
pt_sessionhistory_h	130382	9.063	2	7	260.764	18.126	912.674	63441
Address	1539391	18.029	3	3	4.618.173	54.087	4.618.173	54087
user_activity	175	53.676	1	1	175	53.676	175	53676
pt_indexusehistory	63143	3.399	4	15	252.572	13.596	947.145	50985
EDISenderinformation	1730203	9.991	4	5	6.920.812	39.964	8.651.015	49955
CuttingOrderDetail	1192356	16.331	2	2	2.384.712	32.662	2.384.712	32662
pt_blockhistory	82307	3.740	3	8	246.921	11.220	658.456	29920
pt_commands	115605	9.867	2	3	231.210	19.734	346.815	29601
pt_dbsizehistory	12692	2.684	3	11	38.076	8.052	139.612	29524
NextNumber	8	5.238	4	5	32	20.952	40	26190
pt_missingindexhistory	652	1.006	4	26	2.608	4.024	16.952	26156
pt_filestatshistory2	35178	2.231	3	10	105.534	6.693	351.780	22310
OrderScheduledDate	621743	7.583	2	2	1.243.486	15.166	1.243.486	15166

Index usage

This table shows the number of unused indexes on updated tables.

Table	# Updates	Unused Indexes
TransactionLogForOnline	20.461.916	1
StockTransaction	12.821.612	1
pt_sessionhistory	7.962.426	1
Order	2.861.507	3
EDIMappingChildData	2.544.754	1
EDILog	2.155.368	1
CuttingTableDetails	2.053.329	1
Address	1.503.021	2
OrderDetail	1.125.817	1
NextNumber	257.161	2
pt_logsizehistory	248.336	1
pt_filestatshistory2	246.144	2
HardwareOptimisationDetail	203.644	1
pt_waitstatshistory	109.236	2
pt_dbsizehistory	84.959	1
pt_dbloadhistory	64.610	2
Attachment	48.532	2
Order_Letter	39.078	2
pt_cachehistory	27.309	1
pt_memoryhistory	27.309	1

Index Scans

When SQL Server does not find a matching index, a seek is performed on either the clustered index or another index matching as close as possible.

Table	# Updates	# Scans
Fabric	356.649.007	10.977.488
Table_Block	101.074.032	19.770.354
StockTransaction	50.218.245	65.356
Order	46.217.039	6.731.019
ProcessTransaction	44.869.049	1.696.689
TransactionLogForOnline	40.923.832	392.710.390
OrderDetail_Fabric	32.016.380	23.027.488
OrderDetail	30.655.368	6.998.784
OrderDetailOptions	29.533.443	14.059.856
UserActivityDetail	29.469.709	29.470.115
Scheduling	22.126.013	2.982.923
SerialDetailLine	10.483.624	3.962.457
pt_sessionhistory	7.962.426	560.850
EDIOrderHeader	7.253.736	925.878
EDIOptionList	7.052.990	763.306
EDIMappingData	6.367.900	3.977.315
OrderStatus_Change	6.093.421	1.087
pt_sessionhistory_temp	5.080.666	556.235
EDIOrderDetails	4.028.337	931.157
user_activity	4.000.632	51.364.453

Missing Index keys

Whenever SQL Server detects a certain query cannot be executed optimally, it records what index fields are missing. These are the tables with the most recorded missing index keys.

Table	Missing Index keys
Order	66
StockTransaction	19
OrderDetail_Fabric	10
aOrder	10
Fitter	10
OrderDetail	7
User	6
EDISenderinformation	5
Scheduling	5
aOrderStatus_Change	4
aOrderDetail	4
aOrderDetail_Fabric	3
EDIOrderHeader	3
OrderStatus_Change	2
Discount	2
OrderDetailOptions	2
Fabric_BlindType	2
OrderAmend	2
Address	2
DECORA_Kitting_History	2

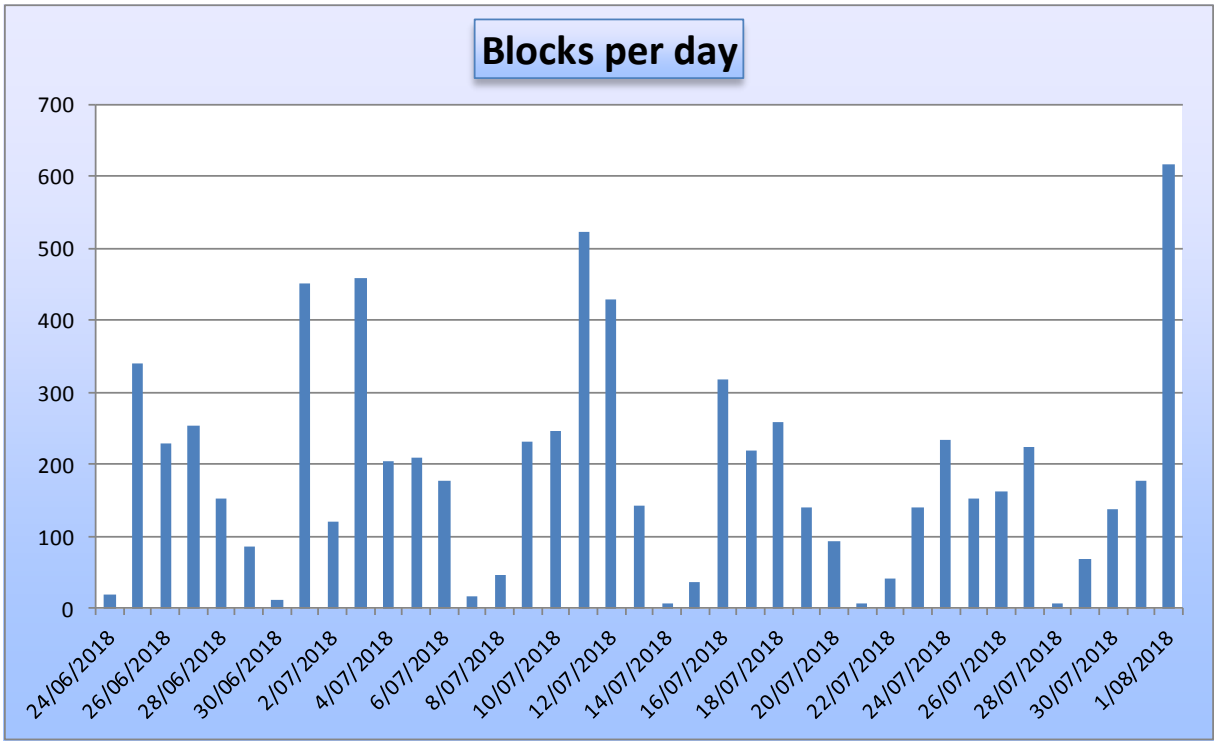
Index Fragmentation

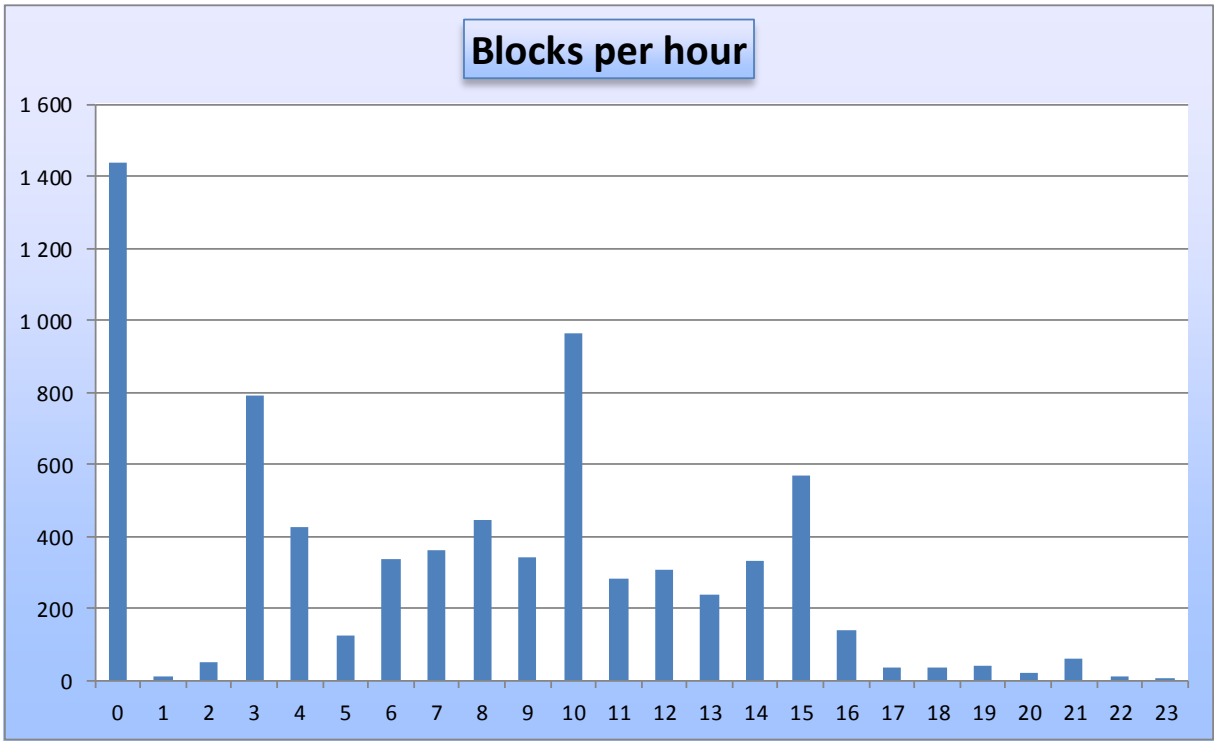
SQL Object Name	SQL Index Name	Index Type	Pages	Rows	Fragments	Avg. Fragm
DECORA_Kitting_History		heap	67598	3363418	8.971	99
TNT_Import		heap	6966	243774	306	35
SalesleadHistory		heap	93	6979	23	96
OrderDetailOptions	PKOrderDetailOptionsID	clustered	44	1759	21	45
OrderDetailOptions	IOrderDetailOptions	nonclustered	31	1759	31	97

Locks & blocks

Every database has locks. Locks occur when users change or create records. Too many locks or locks that are kept active too long can result in blocks; when users start waiting for each other.
During the test period the SQL Perform Tools lock and Block monitors were running. This shows the following locking processes.

Blocks





Blocking Users	Blocks
BLINDATA	3.955
NT SERVICE\SQLSERVERAGENT	3.362
DECORA\SQLADMIN	16
DECORA\CHRISWYLIE	12
SA	10
SYSCO	5
DECORA\SERVERADMIN	4
POWERBI	4

Blocked Users	Blocks
BLINDATA	7.187
NT SERVICE\SQLSERVERAGENT	109
DECORA\CHRISCAIRNS	26
DECORA\SERVERADMIN	23
DECORA\CHRISWYLIE	18
POWERBI	4
SYSCO	1

Blocking users/blocked users	Blocks
BLINDATA	**3.955**
BLINDATA	3.920
NT SERVICE\SQLSERVERAGENT	27
DECORA\CHRISWYLIE	6
SYSCO	1
DECORA\SERVERADMIN	1
NT SERVICE\SQLSERVERAGENT	**3.362**
BLINDATA	3.221
NT SERVICE\SQLSERVERAGENT	80
DECORA\CHRISCAIRNS	26
DECORA\SERVERADMIN	20
DECORA\CHRISWYLIE	11
POWERBI	4
DECORA\SQLADMIN	**16**
BLINDATA	15
DECORA\SERVERADMIN	1
DECORA\CHRISWYLIE	**12**
BLINDATA	11
DECORA\CHRISWYLIE	1
SA	**10**
BLINDATA	9
NT SERVICE\SQLSERVERAGENT	1
SYSCO	**5**
BLINDATA	5
DECORA\SERVERADMIN	**4**
BLINDATA	4
POWERBI	**4**
BLINDATA	2
NT SERVICE\SQLSERVERAGENT	1
DECORA\SERVERADMIN	1

Tables	Blocks
Order	2.101
Table_Block	1.765
Fabric	1.558
StockTransaction	574
user_activity	393
OrderDetail	195
Hardware AM	118
User	78
aOrderDetail	73
Address	51
TAB:	45
OrderDetail_ai	42
OrderDetail_Fabric	28
PowerBI_MainScreen_K3_Section33	23
ProcessTransaction	22
vOrderDetail	22
EDIOrderDetails	20
Blindata.Address	19
EDISenderinformation	17
EDIOrderHeader	16
Indexen	16
OrderAmend	16
EDIOptionList	14
OrderStatus_Change	13
CalcScheduling	13
ACCESS_METHODS_ACCESSOR_CACHE:	11
ManLocationDatesAndTimes	11
EDILog	10
Order_ai	7
UserActivityDetail	7
Blindata.Fabric	5
PowerBI_MainScreen_K3_Line1	5
CalculateEfficiencyProcess	5
DB:	4
Scheduling	4

Tables	Blocks
HistoryOrderAmend	4
HistoryOrderDetailAmend	4
aOrder	4
Blindata.TransactionLogForOnline	3
User2	3
PurchaseOrder	3
NeedToSend	3
Order_Calc_Sub_Options2	3
Get_Count3ForWOLabel	3
StockTransaction_DeleteExisting	2
CalcAndUpdateScheduledDates	2
Batch	2
OptionCostsCalculateForOrder	2
SerialDetailLine	2
Get_Count2ForWOLabel	2
TransactionLogForOnline	2
BlindType	2
Blindata.User	2
Despatch	2
Order_Calc_Sock_Dimension2	2
Blindata.Order	1
Blindata.aOrderDetail	1
OrderGetLeadTimeForOrderDetail	1
Blindata.PurchaseOrder	1
GetSchedulingDataForOrderDetail	1
Order_CalcStockForItem	1
APPEND_ONLY_STORAGE_INSERT_POINT:	1
Fabric_BlindType_GetOptionsCostPrice	1
Blindata.OrderStatus_Change	1
CalculateManLocationDatesAndTimes	1
fnCheckMakeType	1
FitterSheet	1
ACCESS_METHODS_HOBT_VIRTUAL_ROOT:	1
OrderDetailOptions	1
OptionsCalc_Get_Options_Difference	1

Blocking Users / Tables (Top 10)	Blocks
BLINDATA	**3.955**
Table_Block	1.764
Order	557
StockTransaction	556
user_activity	393
Fabric	342
Hardware AM	95
OrderDetail_ai	38
OrderDetail	29
User	18
Address	16
NT SERVICE\SQLSERVERAGENT	**3.362**
Order	1.527
Fabric	1.215
OrderDetail	166
aOrderDetail	72
User	60
TAB:	45
Address	31
OrderDetail_Fabric	28
PowerBI_MainScreen_K3_Section33	23
vOrderDetail	22
DECORA\SQLADMIN	**16**
Blindata.Address	14
Hardware AM	1
Blindata.PurchaseOrder	1
DECORA\CHRISWYLIE	**12**
Order	11
Blindata.OrderStatus_Change	1
SA	**10**
Order	6
Fabric	1
Table_Block	1
Hardware AM	1
Indexen	1
SYSCO	**5**
Blindata.TransactionLogForOnline	3
Hardware AM	1
APPEND_ONLY_STORAGE_INSERT_POINT:	1
DECORA\SERVERADMIN	**4**
Hardware AM	4
POWERBI	**4**
Address	4

Blocking Tables / Users	Blocks
Order	**2.101**
NT SERVICE\SQLSERVERAGENT	1.527
BLINDATA	557
DECORA\CHRISWYLIE	11
SA	6
Table_Block	**1.765**
BLINDATA	1.764
SA	1
Fabric	**1.558**
NT SERVICE\SQLSERVERAGENT	1.215
BLINDATA	342
SA	1
StockTransaction	**574**
BLINDATA	556
NT SERVICE\SQLSERVERAGENT	18
user_activity	**393**
BLINDATA	393
OrderDetail	**195**
NT SERVICE\SQLSERVERAGENT	166
BLINDATA	29
Hardware AM	**118**
BLINDATA	95
NT SERVICE\SQLSERVERAGENT	16
DECORA\SERVERADMIN	4
SA	1
SYSCO	1
DECORA\SQLADMIN	1
User	**78**
NT SERVICE\SQLSERVERAGENT	60
BLINDATA	18
aOrderDetail	**73**
NT SERVICE\SQLSERVERAGENT	72
BLINDATA	1
Address	**51**
NT SERVICE\SQLSERVERAGENT	31
BLINDATA	16
POWERBI	4
TAB:	**45**
NT SERVICE\SQLSERVERAGENT	45
OrderDetail_ai	**42**
BLINDATA	38
NT SERVICE\SQLSERVERAGENT	4
OrderDetail_Fabric	**28**
NT SERVICE\SQLSERVERAGENT	28
PowerBI_MainScreen_K3_Section33	**23**
NT SERVICE\SQLSERVERAGENT	23

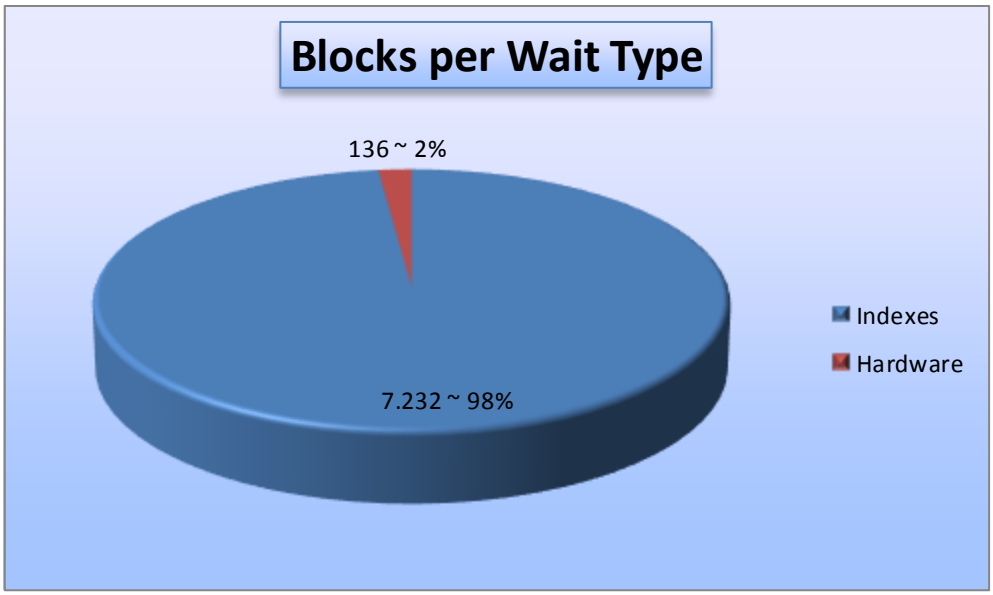
Tables	Waiting Time (ms)
Order	9.700.500
Fabric	8.063.925
Table_Block	5.658.387
user_activity	1.472.314
StockTransaction	1.064.541
OrderDetail	838.977
Hardware AM	618.036
User	336.036
aOrderDetail	315.711
TAB:	233.195
Address	212.936
PowerBI_MainScreen_K3_Section33	134.412
OrderDetail_Fabric	118.807
ProcessTransaction	88.409
vOrderDetail	84.170
Blindata.Address	83.396
EDIOrderDetails	81.285
OrderStatus_Change	67.677
OrderAmend	66.846
EDIOptionList	66.606
EDISenderinformation	55.583
EDILog	54.632
Blindata.Fabric	39.937
PowerBI_MainScreen_K3_Line1	37.014
EDIOrderHeader	35.772
OrderDetail_ai	31.705
Indexen	28.695
DB:	26.268
aOrder	23.208
ManLocationDatesAndTimes	22.547
User2	14.255
NeedToSend	12.219
Blindata.User	10.859
HistoryOrderAmend	10.788
HistoryOrderDetailAmend	8.542

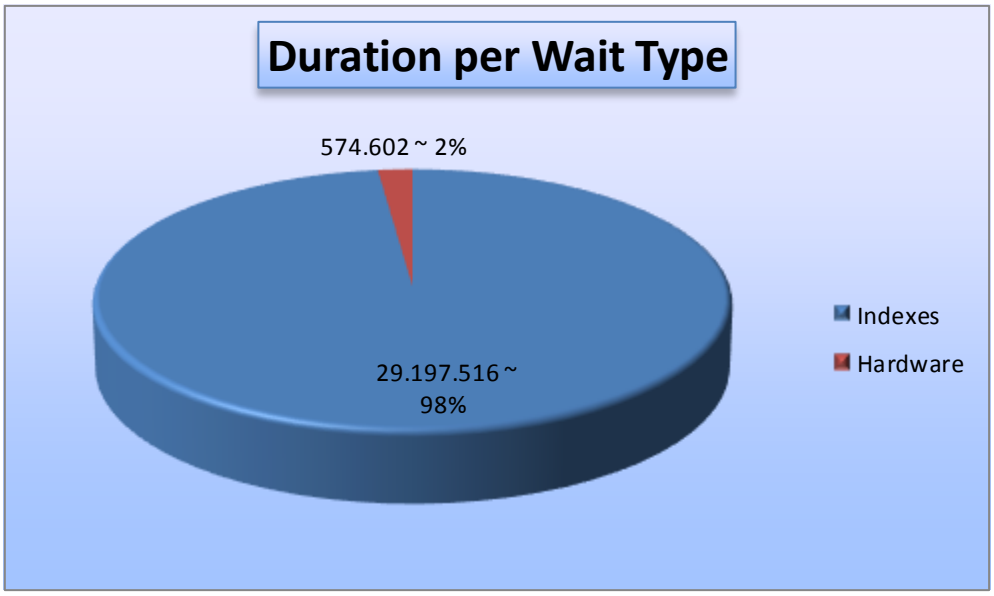
Tables	Waiting Time (ms)
Despatch	7.522
Batch	6.516
CalcScheduling	5.562
BlindType	5.075
SerialDetailLine	4.289
UserActivityDetail	4.144
Scheduling	3.645
FitterSheet	3.073
Blindata.aOrderDetail	2.543
Blindata.OrderStatus_Change	2.406
PurchaseOrder	2.146
Get_Count3ForWOLabel	1.310
Blindata.PurchaseOrder	1.265
Order_ai	1.058
Blindata.Order	723
Order_Calc_Sub_Options2	618
APPEND_ONLY_STORAGE_INSERT_POINT:	558
OptionsCalc_Get_Options_Difference	266
Get_Count2ForWOLabel	200
CalcAndUpdateScheduledDates	155
CalculateEfficiencyProcess	155
Order_Calc_Sock_Dimension2	112
OptionCostsCalculateForOrder	104
OrderDetailOptions	102
Fabric_BlindType_GetOptionsCostPrice	97
StockTransaction_DeleteExisting	79
Order_CalcStockForItem	66
Blindata.TransactionLogForOnline	45
TransactionLogForOnline	34
ACCESS_METHODS_ACCESSOR_CACHE:	23
CalculateManLocationDatesAndTimes	19
ACCESS_METHODS_HOBT_VIRTUAL_ROOT:	5
GetSchedulingDataForOrderDetail	5
fnCheckMakeType	4
OrderGetLeadTimeForOrderDetail	4

Blocking Query	Blocks
CREATE PROCEDURE dbo.Order_Calc_Stock_For_Additional @ORDERDETAIL_ID int, @OrderID INT = NULL, @OperatorID INT = 0 AS BEGIN DECLARE @STOCKTRANS_ID int DECLARE @FABRIC_ID int DECLARE Stock_Trans_U_cur INSENSITIVE CURSOR FOR (SELECT [OrderDetail_Fabric].[fabric_id] FROM [OrderDetail_Fabric] WHERE [OrderDetail_Fabric].[orderdetail_id] = @ORDERDETAIL_ID UNION ..... DEALLOCATE Stock_Trans_U_cur END	477
CREATE PROCEDURE TransactionLogForOnlineInsertRecord @TableName NVARCHAR(50), @TableRowID INT, @Operation TINYINT AS BEGIN --Cheching Parameter - if do not need create web log then nothing doing IF (SELECT TOP 1 [Parameter].[IsWebSiteLog] FROM [Parameter] ORDER BY [Parameter].[id]) = 0 RETURN IF (@Operation = 2 -- Delete then delete all previous record by this id AND ..... (ID, TableName, TableRowID, Operation) VALUES (@MaxID + 1, @TableName, @TableRowID, @Operation) COMMIT TRANSACTION END	264
create procedure sys.sp_procedure_params_rowset (@procedure_name sysname, @group_number int = 1, @procedure_schema sysname = null, @parameter_name sysname = null) as select PROCEDURE_CATALOG = s_pp.PROCEDURE_CATALOG, PROCEDURE_SCHEMA	89
CREATE PROCEDURE [dbo].[DECORA_Import_Express_Nos] AS SET ANSI_NULLS ON SET ANSI_WARNINGS ON update [blindata].dbo.[order] set consignmentnoteno = (SELECT max(ConsignmentNo) FROM SQL.CourierConsignments.dbo.Express WHERE [OrderNo] like '%' + cast((Order_ID)as varchar(20)) +'%') where dat_delivery >=(GETDATE()-5) and consignmentnoteno is null and order_id is not null and order_id in (select [order2].order_id from [order] as [order2] inner join SQL.CourierConsignments.dbo.Express as [express2] on express2.[OrderNo] like '%' + cast((order2.Order_ID)as varchar(20)) +'%' where dat_delivery >=(GETDATE()-5) and consignmentnoteno is null and order_id is not null)	85
update [fabric] set ExpectedDeliveryDate = (select [Delivery Date] from decoraerp.dbo.DECORA_Next_Delivery_Date where [No_]=code collate cyrillic_general_CI_AS) where is_enable=0 and display_stock_message=1	79

Blocked Query	Blocks
CREATE PROCEDURE dbo.Blind_Made @Order_ID INT, @Order_Detail_ID INT AS BEGIN --Declare variabless DECLARE @Fabric_ID INT DECLARE @Quantity REAL DECLARE @IsCutLength BIT DECLARE @ResultGetCutLength INT DECLARE @StockTransactionID INT DECLARE @IsEnable BIT DECLARE @IsDiscontinued BIT DECLARE @Stock INT	1.469
CREATE TRIGGER dbo.OrderDetail_ai ON dbo.OrderDetail WITH EXECUTE AS CALLER FOR UPDATE AS BEGIN SET NOCOUNT ON -- Declaring Variables ---------- DECLARE @ORDER_ID INT, @OPERATOR_ID INT, ...	1.053
CREATE procedure Function_CheckBlock @table_name nvarchar (50), @row_id int, @comp_name nvarchar (20), @login_name nvarchar (20), @operator_id int AS BEGIN DECLARE @count_existsrecord INT, @result bit, @Table_BlockIDForCurrentOperator INT SET @Table_BlockIDForCurrentOperator = 0 ..... END END	1.020
CREATE PROCEDURE dbo.[Function_UnCheckBlock] @table_name NVARCHAR(50), @row_id INT, @OperatorID INT = 0, @IsRemoveAllRecordsForOperator BIT = 0 AS IF @IsRemoveAllRecordsForOperator = 0 BEGIN DELETE FROM [Table_Block] WHERE	640
CREATE PROCEDURE dbo.Order_Calc_Stock_For_Additional @ORDERDETAIL_ID int, @OrderID INT = NULL, @OperatorID INT = 0 AS BEGIN DECLARE @STOCKTRANS_ID int DECLARE @FABRIC_ID int DECLARE Stock_Trans_U_cur INSENSITIVE CURSOR FOR (SELECT [OrderDetail_Fabric].[fabric_id] FROM [OrderDetail_Fabric] ..... DEALLOCATE Stock_Trans_U_cur END	435

Blocked Query	Waiting Time (ms)
CREATE PROCEDURE dbo.Blind_Made @Order_ID INT, @Order_Detail_ID INT AS BEGIN --Declare variabless DECLARE @Fabric_ID INT DECLARE @Quantity REAL DECLARE @IsCutLength BIT DECLARE @ResultGetCutLength INT DECLARE @StockTransactionID INT DECLARE @IsEnable BIT DECLARE @IsDiscontinued BIT DECLARE @Stock INT	7.147.506
CREATE TRIGGER dbo.OrderDetail_ai ON dbo.OrderDetail WITH EXECUTE AS CALLER FOR UPDATE AS BEGIN SET NOCOUNT ON -- Declaring Variables ---------- DECLARE @ORDER_ID INT, @OPERATOR_ID INT, ...	5.187.477
CREATE procedure Function_CheckBlock @table_name nvarchar (50), @row_id int, @comp_name nvarchar (20), @login_name nvarchar (20), @operator_id int AS BEGIN DECLARE @count_existsrecord INT, @result bit, @Table_BlockIDForCurrentOperator INT SET @Table_BlockIDForCurrentOperator = 0 ..... END END	3.623.093
CREATE PROCEDURE dbo.[Function_UnCheckBlock] @table_name NVARCHAR(50), @row_id INT, @OperatorID INT = 0, @IsRemoveAllRecordsForOperator BIT = 0 AS IF @IsRemoveAllRecordsForOperator = 0 BEGIN DELETE FROM [Table_Block] WHERE	1.869.257
CREATE PROCEDURE User_Login_is_active @ID int, @SessionMainID INT = 0, @SessionGUID UNIQUEIDENTIFIER = NULL AS BEGIN UPDATE [User_activity] SET [active_time] = GetDate(), [SessionID] = @SessionMainID, [SessionGUID] = @SessionGUID WHERE [User_activity].[User_id] = @ID END	1.248.842


Blocks per Wait Type
136 ~ 2%
7.232 ~ 98%
Indexes
Hardware

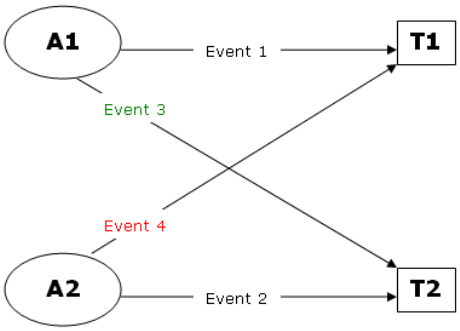

Duration per Wait Type
574.602 ~ 2%
29.197.516 ~ 98%
Indexes
Hardware

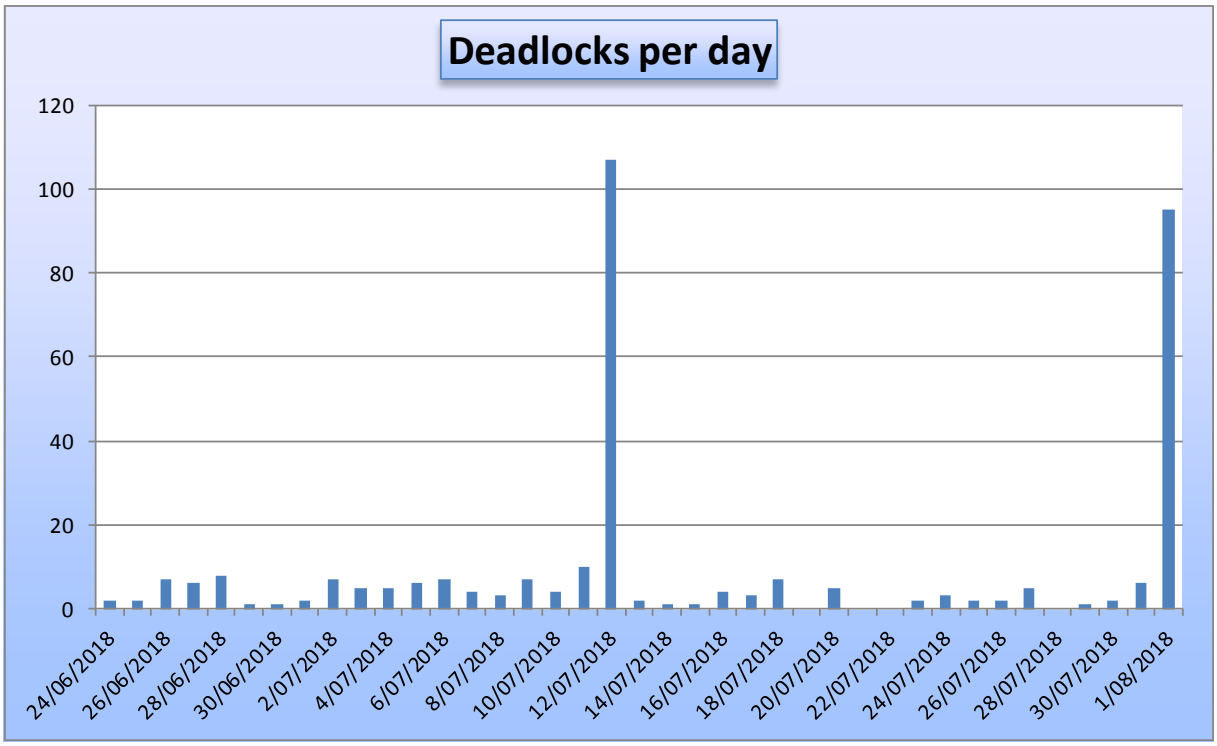
Deadlocks

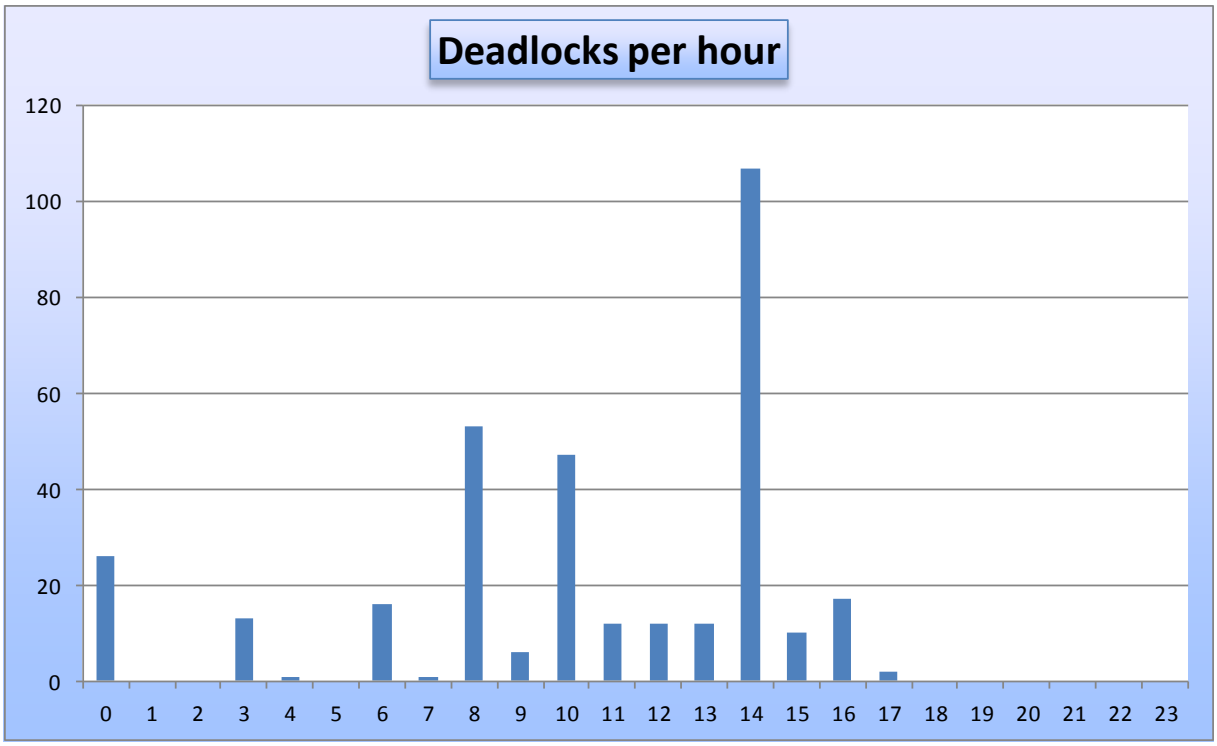
Lock timeouts and deadlocks are experienced by end users as the most frustrating.

Lock timeouts and deadlocks can be the result of a blocking situation. The following lists will give an impression of the damage.

SQL Server checks for deadlocks every 5 seconds.







Users	Deadlocks
BLINDATA	264
NT SERVICE\SQLSERVERAGENT	64
DECORA\CHRISWYLIE	5
SYSCO	1
DECORA\SQLADMIN	1

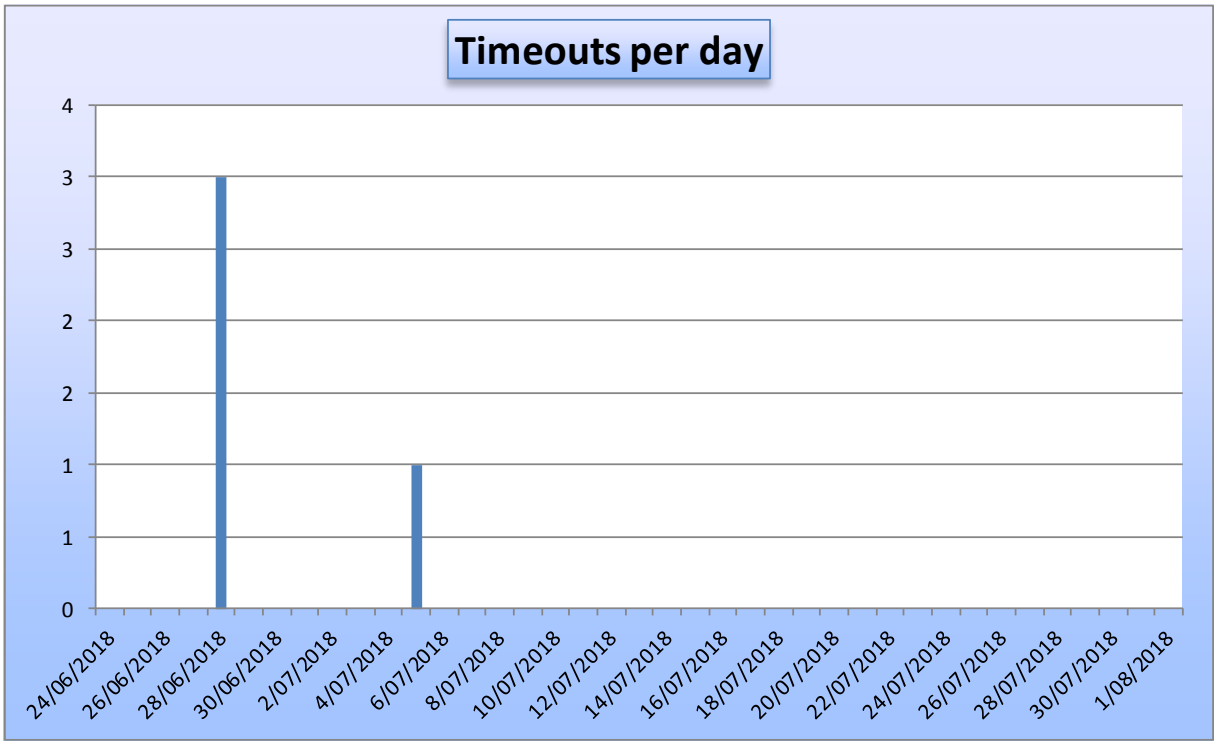
Tables	Deadlocks
Order	165
Table_Block	87
User	63
Fabric	13
EDIOrderDetails	2
OrderDetail	2
PurchaseOrder	1
OrderDetail_Fabric	1
Address	1

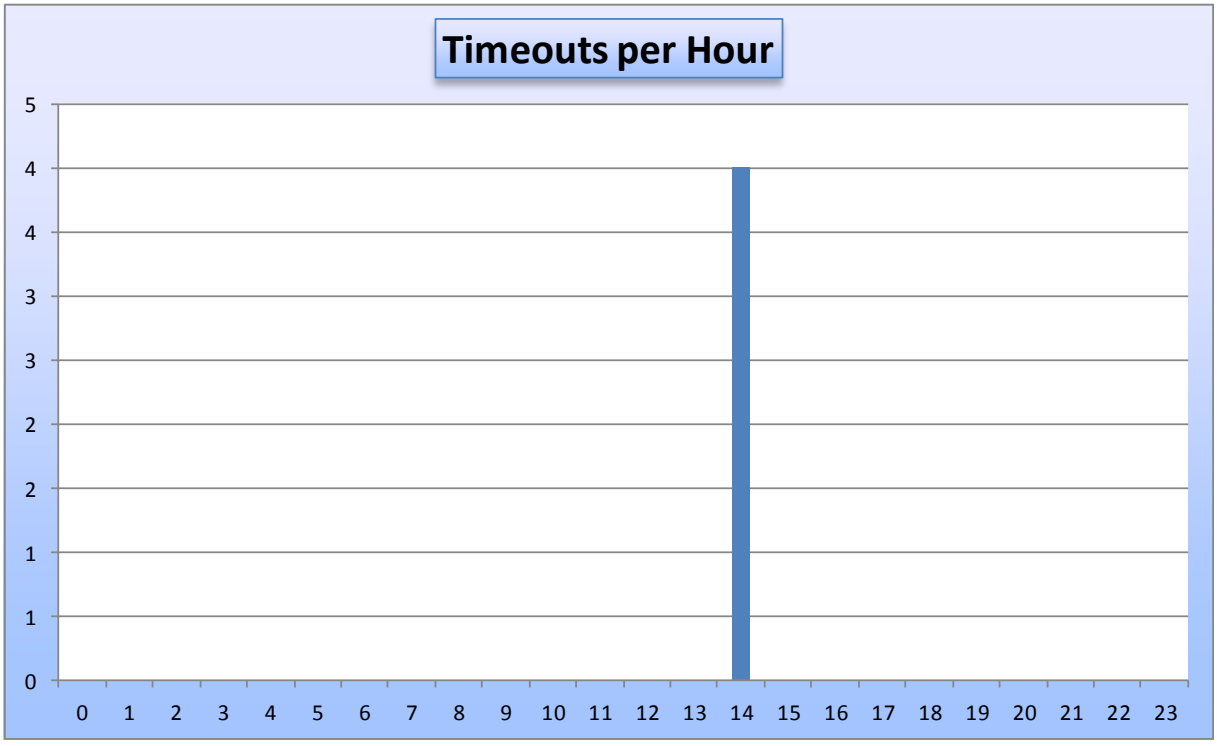
Deadlocks Users / Tables	Deadlocks
BLINDATA	**264**
Order	118
Table_Block	87
User	55
EDIOrderDetails	2
PurchaseOrder	1
OrderDetail	1
NT SERVICE\SQLSERVERAGENT	**64**
Order	45
Fabric	13
User	5
OrderDetail	1
DECORA\CHRISWYLIE	**5**
User	3
Order	1
OrderDetail_Fabric	1
SYSCO	**1**
Order	1
DECORA\SQLADMIN	**1**
Address	1

Deadlocks Tables / Users	Deadlocks
Order	**165**
BLINDATA	118
NT SERVICE\SQLSERVERAGENT	45
SYSCO	1
DECORA\CHRISWYLIE	1
Table_Block	**87**
BLINDATA	87
User	**63**
BLINDATA	55
NT SERVICE\SQLSERVERAGENT	5
DECORA\CHRISWYLIE	3
Fabric	**13**
NT SERVICE\SQLSERVERAGENT	13
EDIOrderDetails	**2**
BLINDATA	2
OrderDetail	**2**
NT SERVICE\SQLSERVERAGENT	1
BLINDATA	1
PurchaseOrder	**1**
BLINDATA	1
OrderDetail_Fabric	**1**
DECORA\CHRISWYLIE	1
Address	**1**
DECORA\SQLADMIN	1

Timeouts

When one user locks a table, a second user waiting to access this table can get a message that the resources is in use. The second user then will lose any data entered and has to start over. This table shows the users and tables this occurs for.





Users	Timeouts
DECORA\CHRISCAIRNS	3
BLINDATA	1

Tables	Timeouts
PowerBI_MainScreen_K3_Section33	3
ProcessTransaction	1

Timeouts Users / Tables	Timeouts
DECORA\CHRISCAIRNS	**3**
PowerBI_MainScreen_K3_Section33	3
BLINDATA	**1**
ProcessTransaction	1

Timeouts Tables / Users	Timeouts
PowerBI_MainScreen_K3_Section33	**3**
DECORA\CHRISCAIRNS	3
ProcessTransaction	**1**
BLINDATA	1

Queries

The SQL Perform tools analyses transactions that take the longest and happen most often. Possible solutions are new or better indexes. Also user training might solve issues.

Query sorted by execution count	Count	Tot time	Avg time	Tot reads	Avg reads
SELECT TOP 1 @ResrictionTypeCodeDesktopID = [ID] FROM [RestrictionType] WITH(NOLOCK) WHERE [Code] = dbo.fnRestrictionTypeCodeDesktop()	960.295.404	16.958.119	-	1.973.821.877	2
SELECT TOP 1 @IsOffice = ISNULL([IsOffice], 1) FROM [Version_Blindata] WITH(NOLOCK)	960.294.954	7.986.970	-	1.920.590.062	2
SELECT TOP 1 @ResrictionTypeCodeBothID = [ID] FROM [RestrictionType] WITH(NOLOCK) WHERE [Code] = dbo.fnRestrictionTypeCodeBoth()	960.294.953	16.490.663	-	1.973.821.031	2
IF EXISTS (SELECT TOP 1 [ID] FROM [OptionRestrictions] WITH(NOLOCK) WHERE ([OptionRestrictions].[OptionID] = @OptionID) AND ([OptionRestrictions].[CustomerID] = @CustomerID) AND ([OptionRestrictions].[BlindTypeID] = @BlindTypeID OR [OptionRestrictions].[BlindTypeID] IS NULL) AND ([OptionRestrictions].[IsAvailable] = 1) AND ((@IsOffice = 0 AND [OptionRestrictions].[RestrictionTypeID] = @ResrictionTypeCodeOnlineID) OR (@IsOffice = 1 AND [OptionRestrictions].[RestrictionTypeID] = @ResrictionTypeCodeDesktopID) OR ([OptionRestrictions].[RestrictionTypeID] = @ResrictionTypeCodeBothID)))	960.290.550	47.860.018	-	2.992.606.458	3
SELECT TOP 1 @ResrictionTypeCodeOnlineID = [ID] FROM [RestrictionType] WITH(NOLOCK) WHERE [Code] = dbo.fnRestrictionTypeCodeOnline()	960.290.329	22.703.154	-	1.973.811.935	2
IF EXISTS (SELECT TOP 1 [ID] FROM [OptionRestrictions] WITH(NOLOCK) WHERE ([OptionRestrictions].[OptionID] = @OptionID) AND ([OptionRestrictions].[CustomerID] <> @CustomerID) AND ([OptionRestrictions].[BlindTypeID] = @BlindTypeID OR [OptionRestrictions].[BlindTypeID] IS NULL) AND ([OptionRestrictions].[IsAvailable] = 1) AND ((@IsOffice = 0 AND [OptionRestrictions].[RestrictionTypeID] = @ResrictionTypeCodeOnlineID) OR (@IsOffice = 1 AND [OptionRestrictions].[RestrictionTypeID] = @ResrictionTypeCodeDesktopID) OR ([OptionRestrictions].[RestrictionTypeID] = @ResrictionTypeCodeBothID)))	955.314.717	39.922.632	-	2.970.677.137	3
IF EXISTS (SELECT TOP 1 [ID] FROM [OptionRestrictions] WITH(NOLOCK) WHERE ([OptionRestrictions].[OptionID] = @OptionID) AND ([OptionRestrictions].[CustomerID] = @CustomerID) AND ([OptionRestrictions].[BlindTypeID] = @BlindTypeID OR [OptionRestrictions].[BlindTypeID] IS NULL) AND ([OptionRestrictions].[IsAvailable] = 0) AND ((@IsOffice = 0 AND [OptionRestrictions].[RestrictionTypeID] = @ResrictionTypeCodeOnlineID) OR (@IsOffice = 1 AND [OptionRestrictions].[RestrictionTypeID] = @ResrictionTypeCodeDesktopID) OR ([OptionRestrictions].[RestrictionTypeID] = @ResrictionTypeCodeBothID)))	955.314.317	39.984.500	-	2.962.749.172	3
SELECT TOP 1 @IsImperialSetup = [Parameter3].[IsImperialSetup] FROM [Parameter3] WITH(NOLOCK) WHERE [CompanyID] = @CompanyID	792.549.668	5.808.767	-	1.585.099.336	2
SELECT @Result = COUNT(DISTINCT [ID]) FROM [vOrder] WHERE [RemakeOriginalOrderID] = @OrderID	212.376.769	6.942.404	-	1.224.658.213	5
SELECT TOP 1 @IsBougthIn = CASE WHEN @MakeType = 2 THEN CASE WHEN [OrderDetail].[ManufactureOrBuyInChoiceCode] = 2 AND [BlindType].[is_stock] = 1 THEN 1 WHEN [OrderDetail].[ManufactureOrBuyInChoiceCode] = 0 AND [BlindType].[is_madeup] = 1 ..... INNER JOIN [BlindType] WITH(NOLOCK) ON [OrderDetail].[blindtype_id] = [BlindType].[id] WHERE [OrderDetail].[id] = @OrderDetailID	114.676.972	2.352.660	-	688.064.744	6

Query sorted by execution count	Count	Tot time	Avg time	Tot reads	Avg reads
FETCH NEXT FROM CurDetails INTO @OrderDetailID	38.888.756	399.067	-	77.777.512	2
SELECT TOP 1 @StatusName = [OrderStatus].[name], @SupplierStatusDesc = [OrderStatus].[SupplieStatusDesc], @IsOrderBoughtIn = 0 FROM [OrderStatus] WITH(NOLOCK) WHERE [OrderStatus].[id] = @OrderStatusID	38.374.597	687.980	-	76.749.194	2
IF (SELECT TOP 1 [Parameter3].[IsUseSupplierManufactureStatusDesc] FROM [Parameter3] WITH(NOLOCK) WHERE [Parameter3].[CompanyID] = @CompanyID) = 0	38.328.130	522.738	-	76.656.260	2
SELECT @Result = @Result + CAST(ROUND(SUM([OrderDetail].[quantity]), 0) AS NVARCHAR(10)) + ' ' + [BlindType].[Code] + ', ' FROM [OrderDetail] WITH(NOLOCK) INNER JOIN [BlindType] WITH(NOLOCK) ON [OrderDetail].[blindtype_id]=[BlindType].[id] INNER JOIN [Fabric] WITH(NOLOCK) ON [OrderDetail].[fabric_id] = [Fabric].[id] LEFT JOIN [FieldName] WITH(NOLOCK) ON [Fabric].[fieldname_id] = [FieldName].[id] WHERE ([OrderDetail].[order_id] = @OrderID) AND dbo.IsCarriage([FieldName].[code]) = 0 GROUP BY [BlindType].[Code] ORDER BY [BlindType].[Code]	29.848.742	4.816.942	-	1.138.037.460	38
FETCH NEXT FROM CurDetails INTO @OrderDetailID	24.228.148	356.316	-	48.456.296	2
DECLARE CurDetails CURSOR STATIC FOR SELECT [OrderDetail].[ID] FROM [OrderDetail] WITH(NOLOCK) INNER JOIN [Blind] WITH(NOLOCK) ON [OrderDetail].[blind_id] = [Blind].[id] WHERE [OrderDetail].[order_id] = @OrderID AND [Blind].[code] <> 32 -- Miscellaneous AND [Blind].[code] <> 4096 -- Fitting AND [Blind].[code] <> 2	24.187.886	157.342	-	1.102	-
SELECT TOP 1 @StatusName = [DetailStatus].[name], @SupplierStatusDesc = [DetailStatus].[SupplieStatusDesc] FROM [DetailStatus] WHERE [DetailStatus].[id] = @DetailStatusID	20.908.771	455.414	-	41.817.542	2
IF (SELECT TOP 1 [Parameter3].[IsUseSupplierManufactureStatusDesc] FROM [Parameter3] WHERE [Parameter3].[CompanyID] = @CompanyID) = 0	20.888.299	332.888	-	41.776.598	2
IF UPDATE([on_stock]) AND (@IsEnableBatchNumberint = 1) AND (dbo.isBatchRoutine() = 0)	16.797.513	160.600	-	20	-
UPDATE [Fabric] SET [LastUpdated] = GETDATE() WHERE [ID] = @ID	16.776.792	1.045.389	-	55.394.041	3

Query sorted by Total reads	Count	Tot time	Avg time	Tot reads	Avg reads
SELECT @CountProcessTransaction = COUNT([ProcessTransaction].[ID]) FROM [SerialDetailLine] WITH(NOLOCK) INNER JOIN [ProcessTransaction] WITH(NOLOCK) ON [SerialDetailLine].[id] = [ProcessTransaction].[Serial_id] INNER JOIN [OrderDetail] WITH(NOLOCK) ON [SerialDetailLine].[OrderDetail_id] = @OrderDetailID INNER JOIN [BlindType] WITH(NOLOCK) ON [OrderDetail].[blindtype_id] = [BlindType].[id] INNER JOIN [Blind] WITH(NOLOCK) ON [BlindType].[blind_id] = [Blind].[id] INNER JOIN [Fabric] WITH(NOLOCK) ON [OrderDetail].[fabric_id] = [Fabric].[id] WHERE [SerialDetailLine].[OrderDetail_id] = @OrderDetailID AND [Blind].[code] <> 32 AND [Fabric].[is_carriage] <> 1	225.025	1.251.253.652	5.560	888.118.024.029	3.946.752
INSERT INTO [EDIMappingCustomer] (OuterCustCode, OuterCustName) SELECT DISTINCT [EDISenderinformation].[CompAccCode], [EDISenderinformation].[CompanyName] FROM [EDISenderinformation] WHERE [EDISenderinformation].[IsOnline] != 1 -- we don't nedd mapp online customer AND [EDISenderinformation].[CompAccCode]+' '+[EDISenderinformation].[CompanyName] NOT IN (SELECT DISTINCT [EDIMappingCustomer].[OuterCustCode]+' '+ [EDIMappingCustomer].[OuterCustName] FROM [EDIMappingCustomer])	201.984	171.322.601	848	462.983.383.857	2.292.178
*select * from [dbo].[PowerBI_MainScreen_K3_Line1]*	*40.385*	*369.543.231*	*9.150*	*194.477.688.859*	*4.815.592*
*select * from [dbo].[PowerBI_MainScreen_K3_Line3]*	*40.370*	*367.912.468*	*9.113*	*194.241.444.448*	*4.811.529*
*select * from [dbo].[PowerBI_MainScreen_K3_Line2]*	*40.267*	*366.668.115*	*9.105*	*193.668.709.684*	*4.809.613*
*select * from [dbo].[PowerBI_MainScreen_K3_Line4]*	*40.519*	*368.226.868*	*9.087*	*193.181.208.626*	*4.767.669*

Query sorted by Total reads	Count	Tot time	Avg time	Tot reads	Avg reads
*select * from [dbo].[PowerBI_MainScreen_K3_Line5]*	*40.305*	*366.123.654*	*9.083*	*191.815.213.376*	*4.759.092*
SELECT @CountSerialDetailLine = COUNT([SerialDetailLine].[ID]) FROM [SerialDetailLine] WITH(NOLOCK) INNER JOIN [OrderDetail] WITH(NOLOCK) ON [SerialDetailLine].[OrderDetail_id] = @OrderDetailID INNER JOIN [BlindType] WITH(NOLOCK) ON [OrderDetail].[blindtype_id] = [BlindType].[id] INNER JOIN [Blind] WITH(NOLOCK) ON [BlindType].[blind_id] = [Blind].[id] INNER JOIN [Fabric] WITH(NOLOCK) ON [OrderDetail].[fabric_id] = [Fabric].[id] WHERE [SerialDetailLine].[OrderDetail_id] = @OrderDetailID AND [Blind].[code] <> 32 AND [Fabric].[is_carriage] <> 1	224.283	763.120.926	3.402	97.685.467.661	435.545
SELECT TOP 1 @IsExists = 1, @SalesOrderID = [EDIOrderHeader].[SalesOrderID] FROM [EDISenderinformation] INNER JOIN [EDIOrderHeader] ON [EDISenderinformation].[ID] = [EDIOrderHeader].[ParentID] WHERE ([EDIOrderHeader].[PurchaseOrderNumber] = @ID) AND ([EDISenderinformation].[CompAccCode] = @CompAccCode) ORDER BY [EDIOrderHeader].[SalesOrderID] DESC	17.068	57.488.613	3.368	83.765.083.499	4.907.726
update [fabric] set ExpectedDeliveryDate = (select [Delivery Date] from decoraerp.dbo.DECORA_Next_Delivery_Date where [No_]=code collate cyrillic_general_CI_AS) where is_enable=0 and display_stock_message=1	31	82.532.623	2.662.342	53.550.703.433	1.727.442.046
UPDATE [EDIOrderDetails] SET [EDIOrderDetails].[Imported] = 0 WHERE [EDIOrderDetails].[ID] IN (SELECT DISTINCT [EDIOrderDetails].[ID] FROM [EDIOrderHeader] INNER JOIN [EDISenderinformation] ON [EDISenderinformation].[ID]=[EDIOrderHeader].[ParentID] AND [EDISenderinformation].[CompAccCode]=@CustCode ..... OR [MDC].[InnerID] IS NULL))	187.756	477.644.116	2.543	20.633.279.329	109.894
UPDATE [EDIOrderHeader] SET [EDIOrderHeader].[Imported] = 0 WHERE [EDIOrderHeader].[ID] IN (SELECT DISTINCT [EDIOrderHeader].[ID] FROM [EDIOrderHeader] INNER JOIN [EDISenderinformation] ON [EDISenderinformation].[ID]=[EDIOrderHeader].[ParentID] AND [EDISenderinformation].[CompAccCode]=@CustCode AND [EDISenderinformation].[CompanyName]=@CustName INNER JOIN [EDIOrderDetails] ON [EDIOrderDetails].[ParentID] = [EDIOrderHeader].[ID] WHERE [EDIOrderDetails].[Imported] = 0) AND [EDIOrderHeader].[SalesOrderID] IS NULL	180.601	17.735.932	98	19.903.491.132	110.206
SELECT [vOrder].[id], [vOrder].[IsArchived], CASE WHEN [vOrder].[type] = dbo.Order_CN_Type_Order() OR [vOrder].[type] = dbo.Order_CN_Type_ArchiveOrder() THEN [vOrder].[order_id] WHEN [vOrder].[type] = dbo.Order_CN_Type_Quotations() OR [vOrder].[type] = dbo.Order_CN_Type_ArchiveQuotations()	26.851	181.678.993	6.766	19.556.023.374	728.316
SELECT DISTINCT [EDIOptionList].[ChoiceCode], [EDIOptionList].[Choice], [dbo].MappingOption() AS [TypeData], [EDIMappingCustomer].[InnerCustID], -- [EDIOrderDetails].[BlindTypeCode] INTO #tmpMO FROM [EDISenderinformation] INNER JOIN [EDIOrderHeader] ON [EDIOrderHeader].[ParentID] = [EDISenderinformation].[ID] ..... AND LEN(LTRIM(ISNULL([EDIOptionList].[ChoiceCode], ''))) > 0	168.973	173.138.852	1.024	17.391.019.602	102.921
SELECT [Fabric].[fieldname_id] AS [FNID] INTO #TempEDIOnlineCheckOptionsList FROM [EDIOrderHeader] INNER JOIN [EDIOrderDetails] ON [EDIOrderHeader].[ID] = [EDIOrderDetails].[ParentID] INNER JOIN [EDIOptionList] ON [EDIOptionList].[ParentID] = [EDIOrderDetails].[ID] INNER JOIN [EDISenderinformation] ON [EDISenderinformation].[ID] = [EDIOrderHeader].[ParentID] ..... AND [EDISenderinformation].[CompanyName] = @CustomerName	146.571	14.651.257	99	15.957.933.394	108.875

Query sorted by Total reads	Count	Tot time	Avg time	Tot reads	Avg reads
INSERT INTO [EDIMappingData] (OuterCode, OuterDescription, TypeData, CustomerID) SELECT DISTINCT ISNULL([EDIOrderHeader].[OrderCarriageType], 'Standard'), ISNULL([EDIOrderHeader].[OrderCarriageType], 'Standard'), [dbo].MappingCarriageType(), [EDIMappingCustomer].[InnerCustID] FROM [EDISenderinformation] ..... AND [EDIMappingCustomer].[OuterCustName]=@CustName WHERE [EDIMappingData].[TypeData] = [dbo].MappingCarriageType())	148.105	8.960.413	60	11.913.032.010	80.436
SELECT CASE WHEN [Order].[type] = dbo.Order_CN_Type_Order() OR [Order].[type] = dbo.Order_CN_Type_ArchiveOrder() THEN [Order].[order_id] WHEN [Order].[type] = dbo.Order_CN_Type_Quotations() OR [Order].[type] = dbo.Order_CN_Type_ArchiveQuotations() THEN [Order].[quatation_id]	15.736	79.617.714	5.059	11.784.457.000	748.885
SELECT DISTINCT [Fabric].[id], [Fabric].[code], CASE WHEN @IsSelectFabricByStockCode = 0 THEN CAST(ISNULL([dbo].[fnGetFabricSubDescription]([Fabric].[id], @BlindType_ID), [Fabric].[description]) AS NVARCHAR(100)) ELSE LTRIM(ISNULL([dbo].fnGetFabricSubCode([Fabric].[id], @BlindType_ID), [Fabric].[code])) ..... AND ([Fabric].[ID] = @FabricID OR @FabricID = 0) ORDER BY [Name]	76.862	113.990.848	1.483	10.207.971.319	132.809
SELECT DISTINCT [EDIOrderDetails].[FabricCode], [EDIOrderDetails].[FabricDescription], [dbo].MappingFabric() AS [TypeData], [EDIMappingCustomer].[InnerCustID], -- [EDIOrderDetails].[BlindTypeCode] INTO #tmpMF FROM [EDISenderinformation] INNER JOIN [EDIOrderHeader] ON [EDIOrderHeader].[ParentID] = [EDISenderinformation].[ID] ..... AND LEN(LTRIM(ISNULL([EDIOrderDetails].[FabricCode], ''))) > 0	154.925	7.538.337	48	9.640.253.259	62.225
INSERT INTO [EDIMappingData] (OuterCode, OuterDescription, TypeData, CustomerID) SELECT DISTINCT [EDIOrderDetails].[BlindTypeCode], [EDIOrderDetails].[BlindTypeDescription], [dbo].MappingBlindType(), [EDIMappingCustomer].[InnerCustID] FROM [EDISenderinformation] INNER JOIN [EDIOrderHeader] ON [EDIOrderHeader].[ParentID] = [EDISenderinformation].[ID]..... AND LEN(LTRIM(ISNULL([EDIOrderDetails].[BlindTypeCode], ''))) > 0	151.857	7.287.160	47	9.637.181.579	63.462

Query sorted by Total time	Count	Tot time	Avg time	Tot reads	Avg reads
SELECT @CountProcessTransaction = COUNT([ProcessTransaction].[ID]) FROM [SerialDetailLine] WITH(NOLOCK) INNER JOIN [ProcessTransaction] WITH(NOLOCK) ON [SerialDetailLine].[id] = [ProcessTransaction].[Serial_id] INNER JOIN [OrderDetail] WITH(NOLOCK) ON [SerialDetailLine].[OrderDetail_id] = @OrderDetailID ..... AND [Fabric].[is_carriage] <> 1	224.954	1.250.704.740	5.559	887.841.116.525	3.946.767
SELECT @CountSerialDetailLine = COUNT([SerialDetailLine].[ID]) FROM [SerialDetailLine] WITH(NOLOCK) INNER JOIN [OrderDetail] WITH(NOLOCK) ON [SerialDetailLine].[OrderDetail_id] = @OrderDetailID INNER JOIN [BlindType] WITH(NOLOCK) ON [OrderDetail].[blindtype_id] = [BlindType].[id] INNER JOIN [Blind] WITH(NOLOCK) ON [BlindType].[blind_id] = [Blind].[id] INNER JOIN [Fabric] WITH(NOLOCK) ON [OrderDetail].[fabric_id] = [Fabric].[id] WHERE [SerialDetailLine].[OrderDetail_id] = @OrderDetailID AND [Blind].[code] <> 32 AND [Fabric].[is_carriage] <> 1	224.227	762.838.134	3.402	97.660.971.441	435.545
UPDATE [EDIOrderDetails] SET [EDIOrderDetails].[Imported] = 0 WHERE [EDIOrderDetails].[ID] IN (SELECT DISTINCT [EDIOrderDetails].[ID] FROM [EDIOrderHeader] INNER JOIN [EDISenderinformation] ON [EDISenderinformation].[ID]=[EDIOrderHeader].[ParentID] AND [EDISenderinformation].[CompAccCode]=@CustCode ..... OR [MDC].[InnerID] IS NULL))	188.733	478.336.529	2.534	20.634.083.878	109.329

Query sorted by Total time	Count	Tot time	Avg time	Tot reads	Avg reads
select * from [dbo].[PowerBI_MainScreen_K3_Line1]	40.383	369.490.493	9.149	194.465.279.566	4.815.523
select * from [dbo].[PowerBI_MainScreen_K3_Line4]	40.517	368.194.239	9.087	193.168.799.345	4.767.598
select * from [dbo].[PowerBI_MainScreen_K3_Line3]	40.366	367.907.193	9.114	194.240.427.776	4.811.981
select * from [dbo].[PowerBI_MainScreen_K3_Line2]	40.264	366.662.952	9.106	193.668.082.434	4.809.956
select * from [dbo].[PowerBI_MainScreen_K3_Line5]	40.301	366.083.401	9.083	191.802.137.429	4.759.240
SELECT [vOrder].[id], [vOrder].[IsArchived], CASE WHEN [vOrder].[type] = dbo.Order_CN_Type_Order() OR [vOrder].[type] = dbo.Order_CN_Type_ArchiveOrder() THEN [vOrder].[order_id] WHEN [vOrder].[type] = dbo.Order_CN_Type_Quotations()	26.841	181.528.498	6.763	19.546.043.917	728.215
SELECT DISTINCT [EDIOptionList].[ChoiceCode], [EDIOptionList].[Choice], [dbo].MappingOption() AS [TypeData], [EDIMappingCustomer].[InnerCustID], -- [EDIOrderDetails].[BlindTypeCode] INTO #tmpMO FROM [EDISenderinformation] INNER JOIN [EDIOrderHeader] ON [EDIOrderHeader].[ParentID] = [EDISenderinformation].[ID] ..... AND LEN(LTRIM(ISNULL([EDIOptionList].[ChoiceCode], ''))) > 0	160.659	173.082.720	1.077	17.375.330.710	108.150
INSERT INTO [EDIMappingCustomer] (OuterCustCode, OuterCustName) SELECT DISTINCT [EDISenderinformation].[CompAccCode], [EDISenderinformation].[CompanyName] FROM [EDISenderinformation] WHERE [EDISenderinformation].[IsOnline] != 1 ..... FROM [EDIMappingCustomer])	201.967	171.311.861	848	462.945.131.946	2.292.182
SELECT SDL.id, SDL."OrderDetail_id", SDL.MadeDate, SDL.WIPDate, ML.Date as ScheduledDate, SDL.[DetailStatusID], SDL.[WIPOperatorID], ML.ManLocationID, SDL.[DespatchOperatorID], SDL.[WIPUserID], Sc.[ManufacturingTime] as ScheduledTime, ... and (PT."ProcessType_id"<> 121 and PT."ProcessType_id" <> 57 and PT."ProcessType_id" <> 9 and PT."ProcessType_id" <> 63 and PT."ProcessType_id" <> 57)	9.136	153.615.422	16.814	2.421.945.962	265.099
SELECT [Order].[id], [Order].[dat_order], [Order].[user_id], [Order].[orderstatus_id], [Order].[proforma_id], [Order].[is_hold], [Order].[order_id], [Order].[is_hold_cutomer], [Order].[is_remake], [Order].[is_contract], [Order].[is_qualitycontrol] ,.....	26.874	137.497.129	5.116	7.984.115.447	297.094
SELECT DISTINCT [Fabric].[id], [Fabric].[code], CASE WHEN @IsSelectFabricByStockCode = 0 THEN CAST(ISNULL([dbo].[fnGetFabricSubDescription]([Fabric].[id], @BlindType_ID), [Fabric].[description]) AS NVARCHAR(100)) ELSE LTRIM(ISNULL([dbo].fnGetFabricSubCode([Fabric].[id], @BlindType_ID), [Fabric].[code])) ..... OR @FabricID = 0) ORDER BY [Name]	77.026	114.051.964	1.480	10.215.359.121	132.622
IF (@Operation = 3 --Update AND EXISTS (SELECT ID FROM [TransactionLogForOnline] WHERE [TableName] = @TableName AND [TableRowID] = @TableRowID AND [Operation] = 3))	17.218.418	83.807.271	4	1.995.929.045	115
update [fabric] set ExpectedDeliveryDate = (select [Delivery Date] from decoraerp.dbo.DECORA_Next_Delivery_Date where [No_]=code collate cyrillic_general_CI_AS) where is_enable=0 and display_stock_message=1	31	82.532.623	2.662.342	53.550.703.433	1.727.442.046
SELECT CASE WHEN [Order].[type] = dbo.Order_CN_Type_Order() OR [Order].[type] = dbo.Order_CN_Type_ArchiveOrder() THEN [Order].[order_id] WHEN [Order].[type] = dbo.Order_CN_Type_Quotations() OR [Order].[type] = dbo.Order_CN_Type_ArchiveQuotations() THEN [Order].[quatation_id]	15.736	79.599.207	5.058	11.784.409.047	748.882
INSERT INTO #tmpFL EXEC Fabric_GetDropDownListForBlindType @BlindTypeID, @CustomerID	30.508	58.984.770	1.933	5.830.463.975	191.112

Query sorted by Total time	Count	Tot time	Avg time	Tot reads	Avg reads
SELECT DISTINCT [Fabric].[id], CAST([Fabric].[description] AS NVARCHAR(150)) AS [Name], CAST(0 AS INT) AS [Sequence] FROM [Fabric] INNER JOIN [Fabric_BlindType] ON [Fabric_BlindType].[fabric_id] = [Fabric].[id] AND [Fabric_BlindType].[blindtype_id] = @BlindTypeID INNER JOIN ..... AND dbo.fnIsOptionAllowed([Fabric].[id], @CustomerID, @BlindTypeID) = 1 ORDER BY [Name]	211.529	58.042.151	274	5.170.295.535	24.442
SELECT TOP 1 @IsExists = 1, @SalesOrderID = [EDIOrderHeader].[SalesOrderID] FROM [EDISenderinformation] INNER JOIN [EDIOrderHeader] ON [EDISenderinformation].[ID] = [EDIOrderHeader].[ParentID] WHERE ([EDIOrderHeader].[PurchaseOrderNumber] = @ID) AND ([EDISenderinformation].[CompAccCode] = @CompAccCode) ORDER BY [EDIOrderHeader].[SalesOrderID] DESC	17.098	57.634.869	3.370	84.048.770.556	4.915.707

Call to Action

The project plan for stabilizing and improving performance for the DECORA BLIND SYSTEMS LTD systems exists of different phases and will be executed on both hardware and software level. Future performance can only be expected when all different aspects of this project plan are executed.

Hardware

Virtual server

DECORA BLIND SYSTEMS has opted to build the SQL Server on a virtual infrastructure, in this case based on VMware.
This is a good solution for small to middle-sized environments when set up in the correct way. Due to the abstraction layer of the virtual infrastructure and the system load present from other virtual servers not all performance counters can be measured in the guest OS. An important factor is for example the amount of guest systems on one physical host, which can lead to over-commitment of the present physical resources. For SQL Server environments in a virtual infrastructure it is advised to have as little over-commitment as possible. A second possible impact is the usage of thin provisioning on the storage, leading to assigning data volume only as it is being consumed.
As these and other performance KPI's are not visible from inside the guest OS, this report focusses on the measured performance from inside SQL Server to confirm if these are sufficient.

CPU Usage

During the monitoring period CPU usage for the SQL Server services ranges from 30% to 50% with peaks up to 80%. At the same time other processes (non SQL) also peak.

Memory allocation

With a database used size of approximately 116GB of data and 134GB of target server memory (Total Machine Memory = 145GBs) there is about 31% of the total database used size cached by the buffer manager (approximately 36GB). The cache analysis indicates sufficient memory is available for the Blindata database.

Storage system setup & performance

Write response time (ms) for TempDB database data files is not good. This is rather consistent as can be seen from paragraph "Disk response times & IO requests" and also in the "Write response time (ms) per hour" graph.

Blindata database

Blindada database data files:

- 1 data file in the Primary file group located in drive E.

Navision database log file:

- 1 log file located in drive F.

Tempdb

Tempdb database data files:

- 6 data files located in drive E. File growth for all data files is set to 200MB.

Tempdb database log file:

- 1 log file located in drive F.

Power plan

Power Plan setting is “High Performance”.

Perform volume maintenance tasks

This is set to the “DECORA\sqladmin” account and to Administrators
Instant Initialization is a new feature as of SQL Server 2005 that is based on an NTFS feature that is available as of Windows 2003 Server. It's a feature that's seemingly simple; it allows file allocation requests to skip zero initialization on creation. As a result, file allocation requests can occur instantly – no matter what the file size.
SQL Server can leverage this feature for DATA file requests only; the transaction log must be zero initialized because of its circular nature. To benefit from this feature the SQL Server service account must have been granted the Windows permission – "Perform Volume Maintenance Tasks".

Anti-Virus software

Webroot SecureAnywhere virus scanner was found installed on the server. We were not able to check the configuration of the antivirus and we could not find if the database file extensions (mdf,ldf,ndf) or database file locations are excluded from the scan.

SQL Server

SQL Version & Service Packs

SQL Server is 2016 SP1, CU9 Standard Edition. We would advise to upgrade to the latest service pack; currently SP2.

Maximum Degree of Parallelism

This is currently set at 8. 16 processors are exposed to SQL.

Memory Dumps

2 memory dumps have occurred, last dump was on May 21, 2018.

TempDB

TempDB data files are located on the same drive with Blindata's data file, drive E.
Tempdb should have its own dedicated drive.

Log fragmentation

The log file is currently fragmented with 137 VLFs created. If the fragments of the log file increase this can lead to serious performance degradations. The default log file is created with only few VLFs. Auto-grows on the log files combined with possible shrinking of the file have increased the amount of VLF's considerably.
We recommend working with two log files to remedy the issue and prevent this from re-occurring. The default log file needs to be pre-sized to handle normal work load, for example 4 GB. Auto grow needs to be disabled for this file. A second log file is created with a starting size of 1GB and auto grow configured at 1GB with unlimited size instead of the default 10% increments.

With this strategy it is possible to remove the second log file, create a log backup (if FULL recovery is selected) and re-create this file again to resolve internal fragmentation easily if the file has become too large for normal operations due to for example a bulk import/export operation. With just one log file the cleanup procedure is much harder to execute.

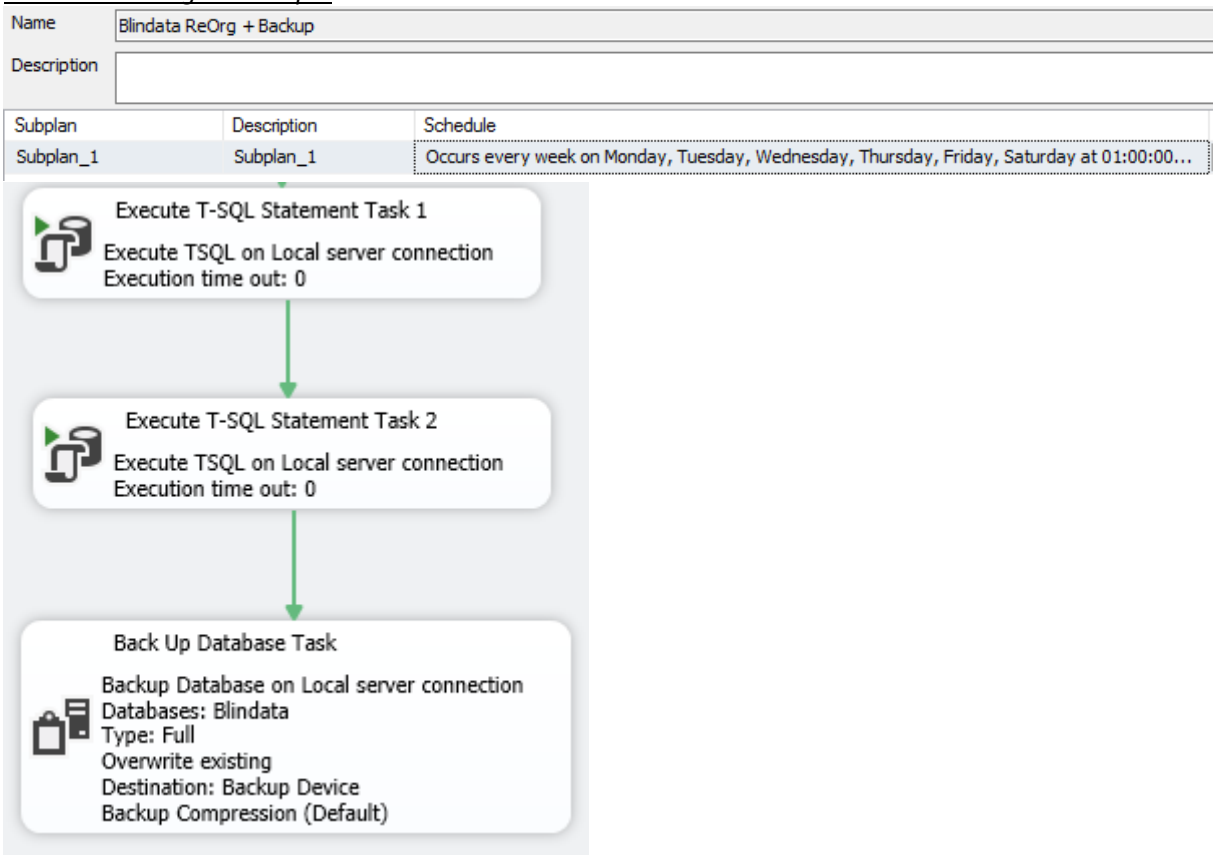
Maintenance plan

Auto Create and Update statistics are disabled for the Blindada database.
SQLPerform's Maintenance Plans are scheduled to run:

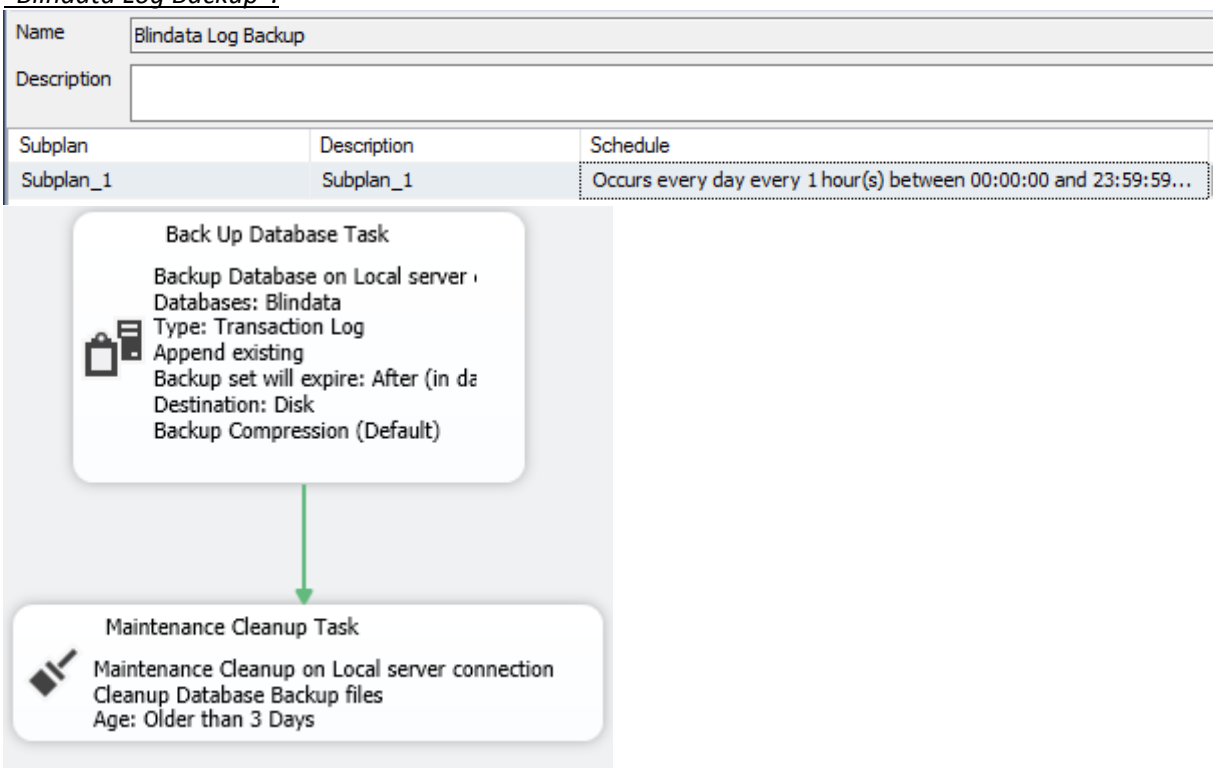
- Index Optimize Maintenance Plan, occurs every Sunday after the completion of the DecoraERP MP
- Check Integrity Maintenance Plan, occurs every Saturday at 03:00AM

We have also found the following Maintenance Plans:

"Blindata ReOrg + Backup":



"Blindata Log Backup":



Backups check of database Blindata

Last 7 days backups since 2018-07-25 00:00 listed below.

Type	Date
Full	25/07/2018 1:11
Full	25/07/2018 6:30
Full	25/07/2018 21:01
Full	26/07/2018 1:11
Full	26/07/2018 6:30
Full	26/07/2018 21:01
Full	27/07/2018 1:12
Full	27/07/2018 6:30
Full	27/07/2018 21:01
Full	28/07/2018 1:10
Full	28/07/2018 6:30
Full	28/07/2018 21:01
Full	29/07/2018 6:30
Full	29/07/2018 21:00
Full	30/07/2018 1:10
Full	30/07/2018 6:30
Full	30/07/2018 21:00
Full	30/07/2018 22:02
Full	31/07/2018 1:11
Full	31/07/2018 6:30
Full	31/07/2018 21:01
Full	1/08/2018 1:11
Full	1/08/2018 6:30

Log backups are created every hour, last 24 hours log backups since 2018-07-31 16:19 listed below.

Type	Date
Log	31/07/2018 17:00
Log	31/07/2018 18:00
Log	31/07/2018 19:00
Log	31/07/2018 20:00
Log	31/07/2018 21:00
Log	31/07/2018 22:00
Log	31/07/2018 23:00
Log	1/08/2018 0:00
Log	1/08/2018 1:00
Log	1/08/2018 2:00
Log	1/08/2018 3:00
Log	1/08/2018 4:00
Log	1/08/2018 5:00
Log	1/08/2018 6:00
Log	1/08/2018 7:00
Log	1/08/2018 8:00
Log	1/08/2018 9:00
Log	1/08/2018 10:00
Log	1/08/2018 11:00
Log	1/08/2018 12:00
Log	1/08/2018 13:00
Log	1/08/2018 14:00
Log	1/08/2018 15:00
Log	1/08/2018 16:00

Application

Indexes

Some tables have a high amount of indexes created. With each new or removed row and all data updates contained in the index these will also need to be changed, increasing the bookings process.

The tables with the highest amount of indexes created are “Order“ with 12 indexes and “OrderDetail” with 9 indexes.

There are also other tables with indexes which were not used during the period of measurement: “Order” (3 indexes), “OrderDetail” (1 index), “StockTransaction” (1 index), etc.
The unused indexes of the top 10 tables for this company only lead to close to 54 million unneeded updates on these indexes. Optimizing the overhead on the Top 10 tables will reduce the overall update workload.

The largest tables in number of records are “aOrderDetail_Fabric” with approximately 78 million records and “aStockTransaction” with approximately 75 million records.
In terms of size the largest tables are “EDILog” with a total data size of approximately 38GB and “aStockTransaction” with approximately 21GB. These 2 tables account for 49% of the total reserved database size.

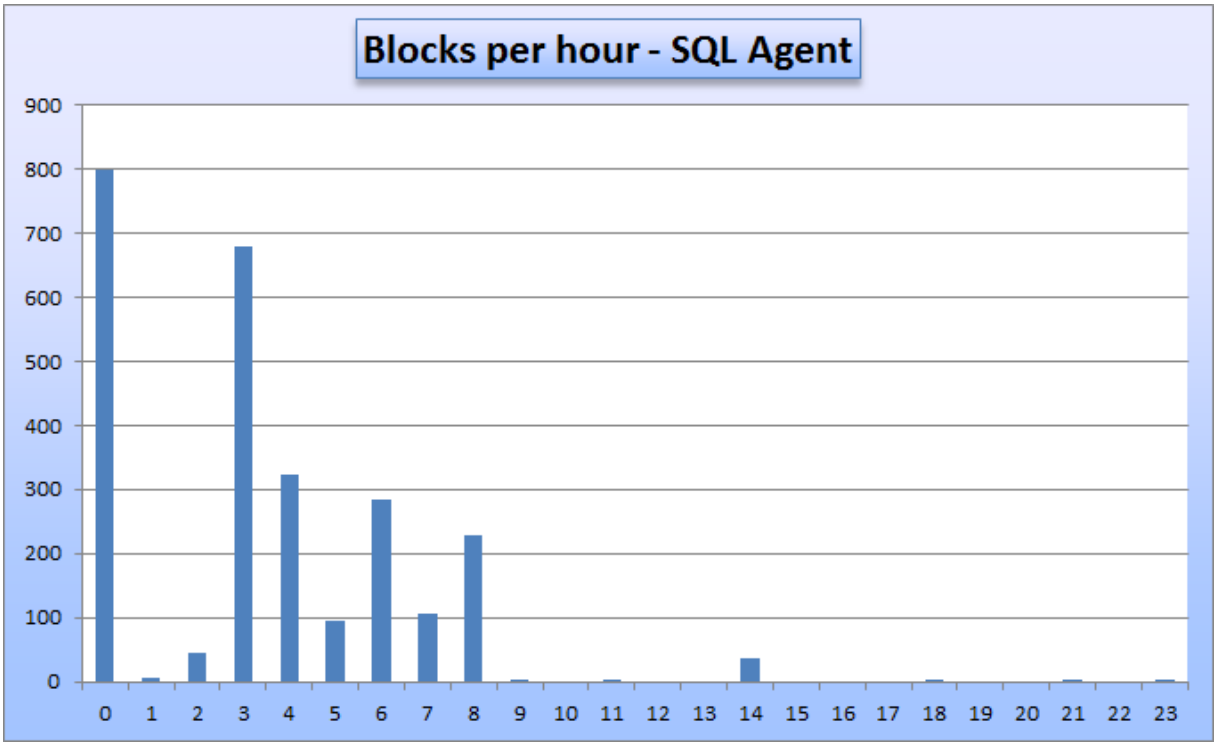
Blocks, deadlocks & timeouts

Blocks:
Quite a few blocks per day up to 617.
Days with peaks in blocks (above 450) during the monitoring period:

- 1/7/2018 with 452 blocks, 156 on table “OrderDetail”, 62 on “aOrderDetail” and 47 on “Address”. Top blocking user “NT SERVICE\SQLSERVERAGENT”.
- 3/7/2018 with 458 blocks, 184 on table “Table_Block”, 125 on “Order” and 57 on “Fabric” . Top blocking user “BLINDATA”.
- 11/7/2018 with 522 blocks, 263 on table “Table_Block”, 121 on “user_activity” and 71 on “Order” . Top blocking user “BLINDATA”.
- 1/8/2018 with 617 blocks, 415 on table “Table_Block”, 90 on “Order” and 49 on “Fabric” . Top blocking user “BLINDATA”.

Top tables in blocks are “Order”, “Table_Block” and “Fabric”. The blocks from these 3 tables account for 74% of the blocks during the monitoring period.
Top blocking accounts are “BLINDATA” and “NT SERVICE\SQLSERVERAGENT”.

The blocks from SQL Agent are recorded mostly during the time between midnight and 09:00 AM:



Top blocking query is:
"CREATE PROCEDURE dbo.Order_Calc_Stock_For_Additional @ORDERDETAIL_ID int, @OrderID INT = NULL, @OperatorID INT = 0 AS BEGIN DECLARE @STOCKTRANS_ID int DECLARE @FABRIC_ID int DECLARE Stock_Trans_U_cur INSENSITIVE CURSOR FOR (SELECT [OrderDetail_Fabric].[fabric_id] FROM [OrderDetail_Fabric] WHERE [OrderDetail_Fabric].[orderdetail_id] = @ORDERDETAIL_ID UNION ..... DEALLOCATE Stock_Trans_U_cur END"

Top blocked query is:
CREATE PROCEDURE dbo.Blind_Made @Order_ID INT, @Order_Detail_ID INT AS BEGIN --Declare variabless DECLARE @Fabric_ID INT DECLARE @Quantity REAL DECLARE @IsCutLength BIT DECLARE @ResultGetCutLength INT DECLARE @StockTransactionID INT DECLARE @IsEnable BIT DECLARE @IsDiscontinued BIT DECLARE @Stock INT ....."

Deadlocks:
Just a few deadlocks per day during the monitoring period, with a peak of 107 deadlocks on July 12.
Deadlocks on July 12:

- 107 deadlocks, 63 on table "User" and 40 on "Order", 92 deadlocks on table "BLINDATA".

Top tables in deadlocks are "Warehouse Activity Line", "Warehouse Entry" and "G/L Entry".
Top account in deadlocks is "BLINDATA".

Timeouts:
Only 4 timeouts during the monitoring period. With 3 on table “PowerBI_MainScreen_K3_Section33” and 1 on table “ProcessTransaction”.

Query analysis

The analysis of top queries shows several frequently occurring important queries that can be optimized, with a high average page read count, either by creating a new or modified index.
Here are some of these queries:

```
SELECT      @CountProcessTransaction = COUNT([ProcessTransaction].[ID])
 FROM      [SerialDetailLine] WITH(NOLOCK)
  INNER JOIN [ProcessTransaction] WITH(NOLOCK)
  ON [SerialDetailLine].[id] = [ProcessTransaction].[Serial_id]
  INNER JOIN [OrderDetail] WITH(NOLOCK)
  ON [SerialDetailLine].[OrderDetail_id] = @OrderDetailID
  INNER JOIN [BlindType] WITH(NOLOCK)
  ON [OrderDetail].[blindtype_id] = [BlindType].[id]
  INNER JOIN [Blind] WITH(NOLOCK) ON [BlindType].[blind_id] = [Blind].[id]
  INNER JOIN [Fabric] WITH(NOLOCK) ON [OrderDetail].[fabric_id] = [Fabric].[id]
 WHERE      [SerialDetailLine].[OrderDetail_id] = @OrderDetailID
  AND      [Blind].[code] <> 32
  AND      [Fabric].[is_carriage] <> 1

SELECT      @CountSerialDetailLine = COUNT([SerialDetailLine].[ID])
 FROM      [SerialDetailLine] WITH(NOLOCK)
  INNER JOIN [OrderDetail] WITH(NOLOCK)
  ON [SerialDetailLine].[OrderDetail_id] = @OrderDetailID
  INNER JOIN [BlindType] WITH(NOLOCK)
  ON [OrderDetail].[blindtype_id] = [BlindType].[id]
  INNER JOIN [Blind] WITH(NOLOCK) ON [BlindType].[blind_id] = [Blind].[id]
  INNER JOIN [Fabric] WITH(NOLOCK) ON [OrderDetail].[fabric_id] = [Fabric].[id]
 WHERE      [SerialDetailLine].[OrderDetail_id] = @OrderDetailID
  AND      [Blind].[code] <> 32
  AND      [Fabric].[is_carriage] <> 1

SELECT TOP 1     @IsExists = 1,     @SalesOrderID = [EDIOrderHeader].[SalesOrderID]
 FROM     [EDISenderinformation]
  INNER JOIN [EDIOrderHeader]
  ON [EDISenderinformation].[ID] = [EDIOrderHeader].[ParentID]
 WHERE     ([EDIOrderHeader].[PurchaseOrderNumber] = @ID)
  AND     ([EDISenderinformation].[CompAccCode] = @CompAccCode)
 ORDER BY [EDIOrderHeader].[SalesOrderID] DESC

update [fabric] set ExpectedDeliveryDate =
 (select [Delivery Date] from decoraerp.dbo.DECORA_Next_Delivery_Date
  where [No_]=code collate cyrillic_general_CI_AS)
  where is_enable=0 and display_stock_message=1
```

With many queries taking over 250ms per execution, these are prime examples of queries causing slowness in user response on the system. Reducing the logical pages read will speed up the queries considerably.

Further investigation & further actions to reduce blocking / deadlocks

The top “heavy” statements should be investigated for optimizations.

Also for each of the top tables in blocks we would suggest to do the following:

- Analyze the blocking/blocked statements on these tables for optimizations.
- Remove the overhead by disabling unused indexes and VSIFTs.
- Examine the business Processes involved.