|  |
| --- |
|  |
| SQL Server Database Troubleshooting Guide |
|  |

|  |
| --- |
| Prepared by: Database Engineering  Version: 0.2  Date: July 2018 |

Table of Contents

[Version History 3](#_Toc520563457)

[Introduction 3](#_Toc520563458)

[Audience 3](#_Toc520563459)

[Purpose 3](#_Toc520563460)

[Outline of SQL Server Performance Testing 4](#_Toc520563461)

[SQL Server Configurations and Best Practices 4](#_Toc520563462)

[CPU 6](#_Toc520563463)

[MAXDOP 6](#_Toc520563464)

[Cost Threshold for Parallelism 7](#_Toc520563465)

[Processor and I/O Affinity 7](#_Toc520563466)

[Memory 8](#_Toc520563467)

[Disk 9](#_Toc520563468)

[Instance Level Configurations 10](#_Toc520563469)

[Maximum Worker Threads 10](#_Toc520563470)

[Boost SQL Priority 11](#_Toc520563471)

[Maximum Number of Concurrent Connections 11](#_Toc520563472)

[Allow Remote Connections to the Instance 12](#_Toc520563473)

[Remote Query Timeout 12](#_Toc520563474)

[Query Wait 13](#_Toc520563475)

[Index Maintenance 13](#_Toc520563476)

[Database Level Configurations 13](#_Toc520563477)

[Auto Close / Auto Shrink 13](#_Toc520563478)

[Database file growth sizes 14](#_Toc520563479)

[Tempdb 15](#_Toc520563480)

[Windows Server Configurations and Considerations 15](#_Toc520563481)

[General Configurations 15](#_Toc520563482)

[Power Settings 16](#_Toc520563483)

[Page File 18](#_Toc520563484)

[Anti-Virus 18](#_Toc520563485)

[Instant File Initialization 18](#_Toc520563486)

[VMware Configurations and Best Practices 20](#_Toc520563487)

[General Configurations 20](#_Toc520563488)

[CPU Ready 21](#_Toc520563489)

[Memory Ballooning 21](#_Toc520563490)

[Disk 24](#_Toc520563491)

[Troubleshooting Performance Issues with the Database 24](#_Toc520563492)

[Checklist for Performance Troubleshooting 24](#_Toc520563493)

[Checklist – General 24](#_Toc520563494)

[Checklist – Server is down or not responding 25](#_Toc520563495)

[Checklist – Server is going to go down in a few minutes if we do not fix the problem 25](#_Toc520563496)

[Checklist – Server is in bad shape but will survive even if we don’t do anything 26](#_Toc520563497)

[Checklist – Some parts of DatAdvantage are unusually slow 26](#_Toc520563498)

[Checklist – A user report or a particular job is running slow 27](#_Toc520563499)

[Event Logs – Application and System 27](#_Toc520563500)

[SQL Server Error Log 27](#_Toc520563501)

[Stored procedures 27](#_Toc520563502)

[spWhoVaronis 27](#_Toc520563503)

[spDiag 29](#_Toc520563504)

[Notifications 29](#_Toc520563505)

[PerfMon 30](#_Toc520563506)

[DMVs and queries 30](#_Toc520563507)

[SQL Profiler 32](#_Toc520563508)

[Extended Events 32](#_Toc520563509)

[Capacity Planning – Sizing Script 32](#_Toc520563510)

[Appendix 32](#_Toc520563511)

[References 32](#_Toc520563512)

**SQL Server Database Performance Guide**

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comment** |
| 0.1 | 20180725 | Jeff Mlakar | Initial Draft |
| 0.2 | 20180728 | Jeff Mlakar | Added troubleshooting checklists and to its below headers |

# Introduction

## Audience

This document is intended for support and customers looking to optimize SQL Server to work with DatAdvantage. There is an assumption of technical proficiency with SQL Server database administration.

## Purpose

The main purpose of this document is to suggest optimal configurations for SQL Server to run DatAdvantage. Additionally, we will discuss troubleshooting techniques for diagnosing performance issues.

# Outline of SQL Server Performance Testing

There are multiple components to strong server health. A good starting place is to understand the best practices of configurations. From there we can make small adjustments to reach optimal server performance. Since every environment is different, it becomes a question of “it depends” on many variables. Nonetheless, we can mitigate many performance problems by simply setting up the environment in a good way.

We will discuss the following topics for best practices and configuration:

* SQL Server
* Windows Server
* VMware

Various troubleshooting techniques will be illustrated as well:

* Log Analysis
* Stored procedures
* PerfMon
* DMVs and queries
* SQL Profiler
* Extended Events

SQL Server Configurations and Best Practices

Many default configurations in SQL Server are sub-optimal. Let us go through many common ones:

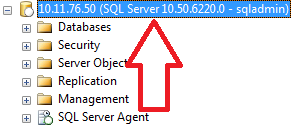
* CPU
* Memory
* Disk
* Instance level configurations
* Database level configurations
* Index Maintenance

Terminology

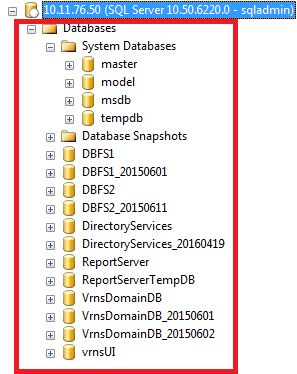
Microsoft has a glossary of SQL Server concepts and terminology:

<http://social.technet.microsoft.com/wiki/contents/articles/1145.sql-server-glossary.aspx>

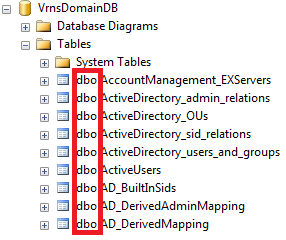
**Instance** – a copy of SQL Server running on a computer. This database server software runs as a service on the OS and manages the database system resources, disk files, etc.



**Database** – the object in the SQL instance that holds the collection of tables, views, stored procedures, etc.



**Schema** – the namespace of a collection of objects inside of a database



**SSMS -** SQL Server Management Studio. The default SQL Server management / development environment. It is the client to the database engine build upon a shell of Visual Studio.

## CPU

Below are the configurations and best practices regarding CPU in SQL Server. There are 3 main areas to check:

* MAXDOP
* Cost Threshold for Parallelism
* Processor and I/O Affinity

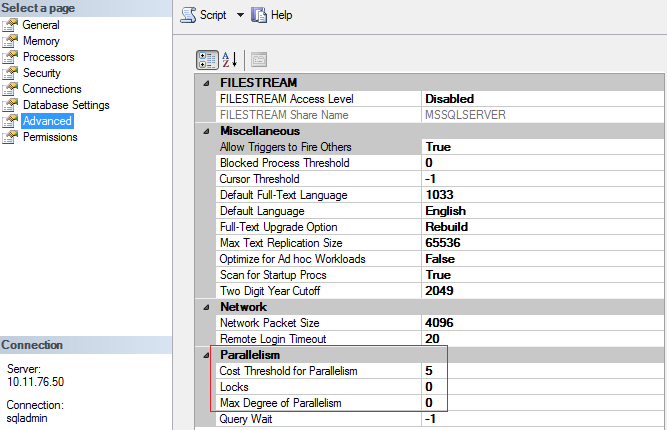
### MAXDOP

The Microsoft SQL Server max degree of parallelism (MAXDOP) configuration option controls the number of processors that are used for the execution of a query in a parallel plan. This option determines the computing and thread resources that are used for the query plan operators that perform the work in parallel.

* The default is 0 i.e. use all cores
* For < 8 logical cores, assign the value to be the number of logical cores
* For >= 8 logical cores, assign the value to be 8
* There are exceptions but this is a good starting place

<https://support.microsoft.com/en-us/kb/2806535>

In SSMS, right click on Instance -> Properties -> Advanced -> options are under Parallelism section



### Cost Threshold for Parallelism

The cost threshold for parallelism option specifies the threshold at which Microsoft SQL Server creates and runs parallel plans for queries. SQL Server creates and runs a parallel plan for a query only when the estimated cost to run a serial plan for the same query is higher than the value set in cost threshold for parallelism. The cost refers to an estimated elapsed time in seconds required to run the serial plan on a specific hardware configuration.

* The default is 5
* Suggest changing in conjunction with MAXDOP to a value of 50

In SSMS, right click on Instance -> Properties -> Advanced -> options are under Parallelism section

### Processor and I/O Affinity

Leave this option alone. This controls the CPU processors that SQL will use. We typically want SQL Server to use all of the CPU cores presented to it by the operating system.

<https://blogs.msdn.microsoft.com/psssql/2010/11/19/how-it-works-io-affinity-mask-should-i-use-it/>

There are rare circumstances where this may be changed such as multiple SQL instances on the same host. We always prefer a dedicated host system to run SQL Server.

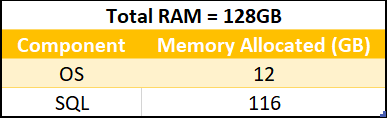
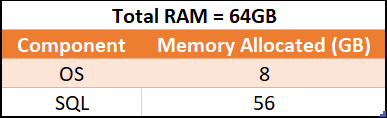
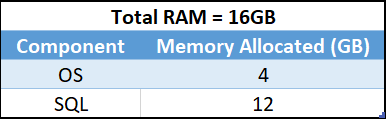
## Memory

We can and should control how much memory SQL Server is allowed to consume. All RDBMS are memory hungry and SQL Server is no different in this regard from Oracle, Sybase, or any other RDBMS vendor. It is important to leave enough memory for the operating system (OS) along with whatever other applications are running on the same host.

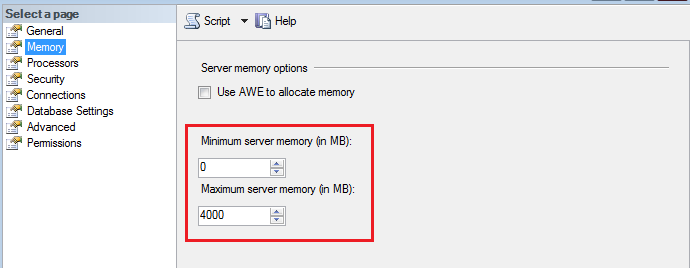
**NOTE:** Ideally, we want a host dedicated to SQL Server – this means the database is all that runs on it besides the OS

* Minimum Server Memory
  + Set the minimum server memory to 0
* Maximum Server Memory
  + The default memory allocated to a SQL Instance is 0 for min server memory and 2147483647 for max server memory (meaning all). This is a poor default value and we need to change it.
  + Set the maximum server memory depending on what else is running on the machine
    - The OS needs memory to function. If it does not have enough i.e. SQL Server has too high a memory grant, then the OS will start to take memory from SQL Server as needed resulting in performance degradation.
      * A good rule is to allow 4GB for the first 16GB of memory and add 1 GB for each additional 16GB.
      * This means for a 64GB server running nothing besides SQL Server and the OS we would leave about 8GB for the OS and assign the rest to SQL Server.
    - SSRS, depending on the environment, should have no less than 4GB of memory
    - If other applications are running on the machine then leave an appropriate amount of memory for them to run as well – ex. DatAdvantage GUI, MC, etc.
    - Assign the rest to SQL Server in the max memory option

Examples of maximum memory assignment to SQL Server – assume only the OS and SQL Server are on the host:



Right click on Instance -> Properties -> Memory -> options are under Parallelism section



**NOTE:** CLR Concerns - Starting with SQL Server 2012, CLR allocations are also included in memory limits that are controlled by max server memory (MB) and min server memory (MB).

* This means that our CLR objects and assemblies in SQL will have their memory managed by the SQL Instance and **not** the OS.
* Therefore err on the side of giving SQL Server more memory for SQL Server 2012+ and err on the side of giving the OS more memory for versions of SQL Server <= 2008 R2

<https://support.microsoft.com/en-us/kb/2663912>

## Disk

Placing both data and log files on the same device can cause contention for that device and result in poor performance. Placing the files on separate drives allows the I/O activity to occur at the same time for both the data and log files.

Strive to separate database files onto different LUNs / physical disks:

* Data files (mdf, ndf)
* Transaction Log files (ldf)
* Tempdb files
* OS / SQL binaries
* System databases
* Backups
* Traces

At a minimum, the data files and transaction log files should be separated. Once this is done, the next part to improve is separating the tempdb files. Significant performance degradation can occur by way of disk contention to these files.

Example disk separation from VMware VM:



<https://technet.microsoft.com/en-us/library/cc966534.aspx>

**NOTE:** If SSDs are in use then place either tempdb or the transaction logs on the fastest disk.

## Instance Level Configurations

Various instance level configurations can affect performance. At this level, they affect all of the databases on the instance.

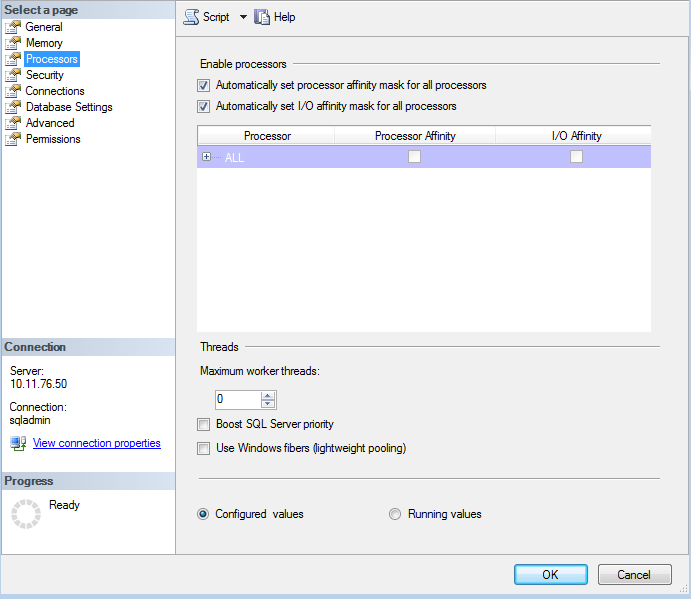
### Maximum Worker Threads

Maximum Worker Threads isthe number of worker threads available to run SQL processes, forms the connection pool for client connections.

* Default is 0
* Leave as 0 otherwise it the database engine can reject worker threads coming from our applications e.g. DatAdvantage
* If need to change then use the table in the below MS KB link for the number of threads based on the number of CPUs

In SSMS, right click on Instance -> Properties -> Processors

<https://msdn.microsoft.com/en-us/library/ms190219(v=sql.110).aspx>



### Boost SQL Priority

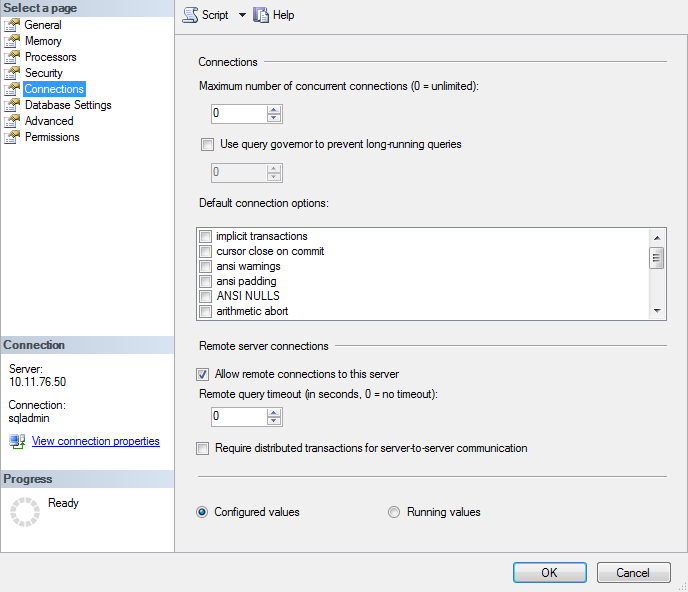
* Default is unchecked
* Leave this unchecked at all times!

In SSMS, right click on Instance -> Properties -> Processors

### Maximum Number of Concurrent Connections

* Default is 0 (unlimited)
* Leave this setting alone!

In SSMS, right click on Instance -> Properties -> Connections



### Allow Remote Connections to the Instance

* Default is checked
* Leave this setting checked

In SSMS, right click on Instance -> Properties -> Connections

Changing this option will not adversely affect DatAdvantage.

### Remote Query Timeout

* Default is 0 (no timeout)
* Leave this as 0

In SSMS, right click on Instance -> Properties -> Connections

### Query Wait

* Default is -1
* Leave this setting alone

In SSMS, right click on Instance -> Properties -> Advanced

## Index Maintenance

Although not a configuration, index maintenance with DatAdvantage needs to be examined. Running index maintenance jobs that address fragmentation in indexes can cause MC jobs to deadlock and fail. It will also hinder overall performance.

At most index maintenance should be performed during a scheduled maintenance window or on a weekly or monthly basis. Understand that our jobs run large batches all 24 hours of the day and rebuilding or reorganizing an index will cause it to fail. Consider doing this during planned downtime.

## Database Level Configurations

The various database level configurations affect the database in scope. Here we review some:

* Auto close
* Auto shrink
* Database file growth sizes
* Tempdb

### Auto Close / Auto Shrink

The auto close option turns the database off when the last user process finishes. There are no circumstances where we would want this to be on. Microsoft advises against it.

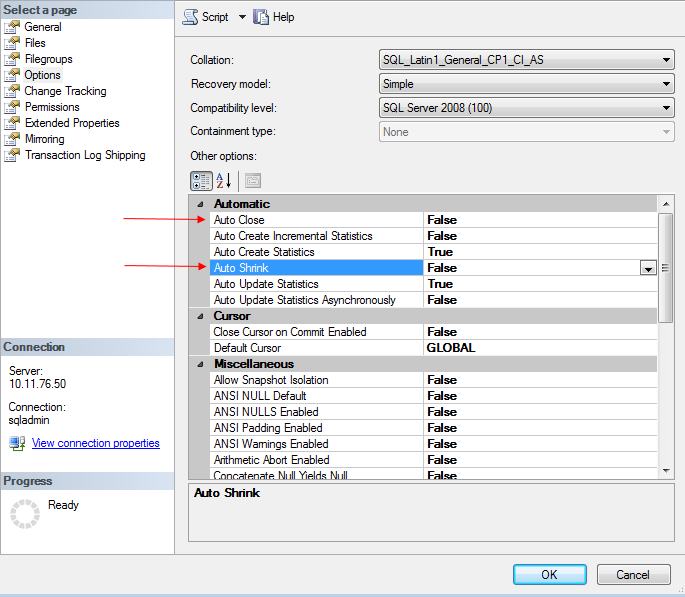
The auto shrink options shrinks the database periodically. Unplanned shrinks may cause MC jobs to fail and cause system stability and performance issues.

* Make sure the Auto Close option is false
* Make sure the Auto Shrink option is false

**MS Best Practice: Considerations for the "autogrow" and "autoshrink" settings in SQL Server**

Read this: <http://support.microsoft.com/kb/315512>

In SSMS, right click on the Database -> Properties -> Options



### Database file growth sizes

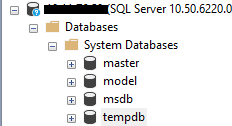
Change default growth sizes of database files to a reasonable amount concerning the database. Remember, growing the database files is one of the most expensive operations SQL Server can make. Therefore, follow these basic guidelines:

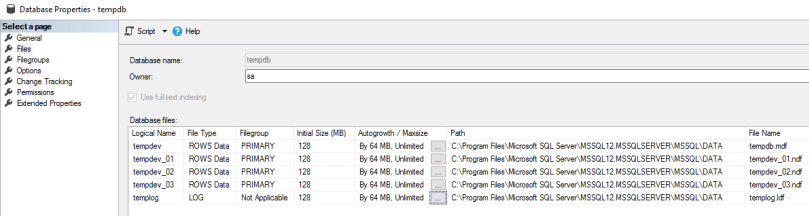
* Prefer to increase file sizes by a specified amount in MB rather than a percentage.
* Adjust the growth to for files to at least 128 MB or some other reasonable growth size.
* Set a maximum size that is not unlimited. Better to receive an error to increase disk space than crash the server by having a full disk.
* Keep autogrowth on as a stopgap for capacity planning. Ideally, we would size the database files so they did not have to grow at all. The initial sizes can be set according to known capacity planning values.

### Tempdb

The tempdb database is shared by all databases on the SQL Instance. It has a lot to do with performance and can be a bottleneck. We want tempdb to have the following best practice guidelines:

* Each tempdb data file should be the same initial size
* Autogrowth to tempdb files should be an explicit value in MB instead of a percentage. Choose a reasonable value based on their workload. Ex. 64MB, 128MB, etc.
* The number of tempdb files should be 1 per processor core up to 8. At that point, the performance should be monitored and if more tempdb files are needed they should be added in sets of 4.
* Ideally, the tempdb files are sized up to the max they will need and never have to autogrow.





**NOTE:** remember that tempdb is rebuild by SQL Server after each service restart/reboot. We will not see any changes to the configuration of tempdb until this happens.

In SQL Server versions < 2016, we can safely enable 2 trace flags to improve performance of tempdb:

* 1117
* 1118

The default in SQL Server 2016+ is to include these trace flags.

Windows Server Configurations and Considerations

## General Configurations

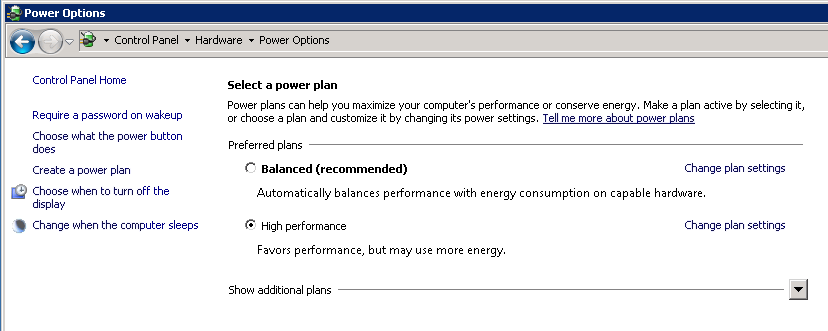
There are various general configurations to consider optimizing on any Windows Server hosting a SQL Server instance.

## Power Settings

The default power setting in Windows Server is “Balanced”. This is not acceptable for a machine hosting SQL Server as this will throttle system resources globally to all applications and significantly hinder SQL Server performance.

Always set the power option to “High Performance”. This can be found via:

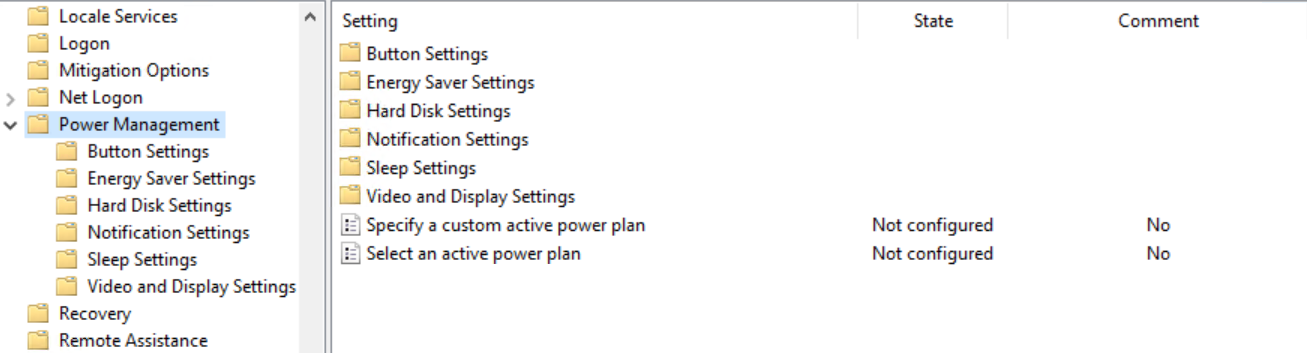
Control Panel 🡪 Hardware 🡪 Power Options



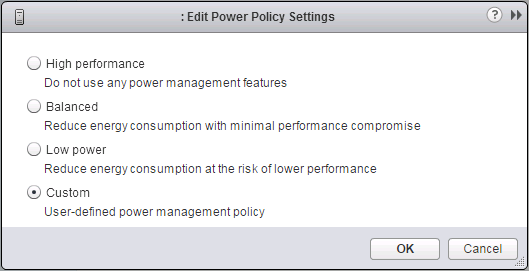
**NOTE:** there are NO circumstances under which SQL Server will perform well with a balanced power setting. It must be set to high performance

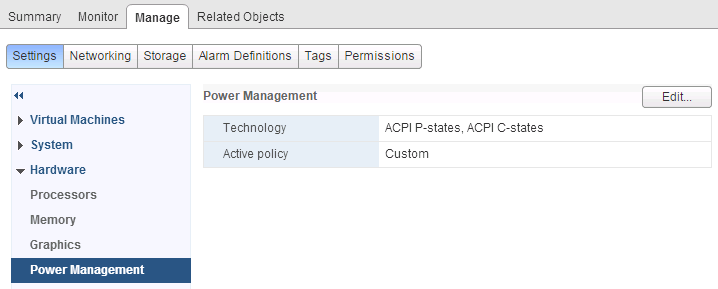
Power options can be tricky because they can be set in a few different places:

* BIOS level power setting – if running on an HP ProLiant or Dell PowerEdge server it can be configured in the bios. Other vendors likely have this option too.
* Group policy level power setting – this is in GP and can be set as well to override user power setting configurations that are set in Windows Server



* vSphere power setting – it can be set in vSphere as well





## Page File

Page files are a special kind of file that Windows uses as a temporary workspace for storing modified pages from disk that are still in use by a process. It is used to hold data that is in the process of being swapped in and out of physical memory to allow for a larger virtual memory set.

Large page files deserve their own disk just like the SQL data file, transaction log, and tempdb files.

PerfMon can be used to track usage and determine problems. Use the following performance counters:

* Memory: Committed Bytes – number of bytes of virtual memory that has been committed
* Memory: Commit Limit – number of bytes of virtual memory which can be committed without having to extend the paging files
* Paging File: % Usage - % of the paging file committed
* Paging File: % Usage Peak – highest % of the paging file committed

Some links about this:

* [Best Practices for Page File and Minimum Drive Size for OS Partition on Windows Server](https://social.technet.microsoft.com/wiki/contents/articles/13383.best-practices-for-page-file-and-minimum-drive-size-for-os-partition-on-windows-servers.aspx)
* [What is the Page File for anyway](https://blogs.technet.microsoft.com/askperf/2007/12/14/what-is-the-page-file-for-anyway/)
* [How to Determine the Appropriate Page File Size for 64-bit Versions of Windows](https://support.microsoft.com/en-us/kb/2860880)
* [Page File – The Definitive Guide](https://blogs.technet.microsoft.com/motiba/2015/10/15/page-file-the-definitive-guide/)

Lots of page file utilization is an indicator of memory pressure i.e. there is not enough memory therefore it spills to disk. This results in poor performance.

## Anti-Virus

If Anti-Virus software is running on the SQL host machine there should be exclusions for the following types of files

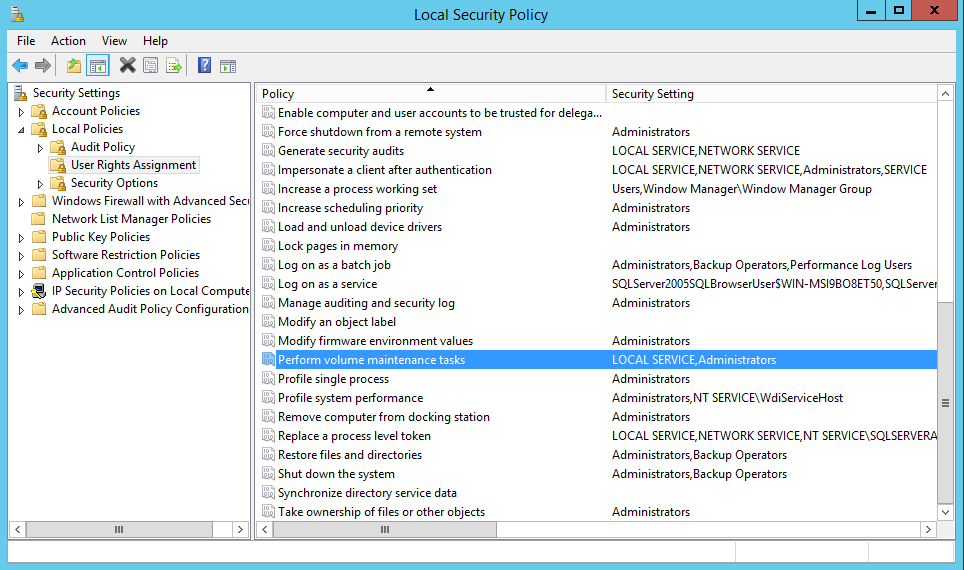
* MDF – these file extensions are associated with SQL Server database files
* LDF – these file extensions are associated with SQL Server transaction log files
* BAK – these file extensions are associated with SQL Server backup files
* TRN – these file extensions are associated with SQL Server trace files
* All SQL Server binaries and directories

AV scans of database files can result in severe performance degradation.

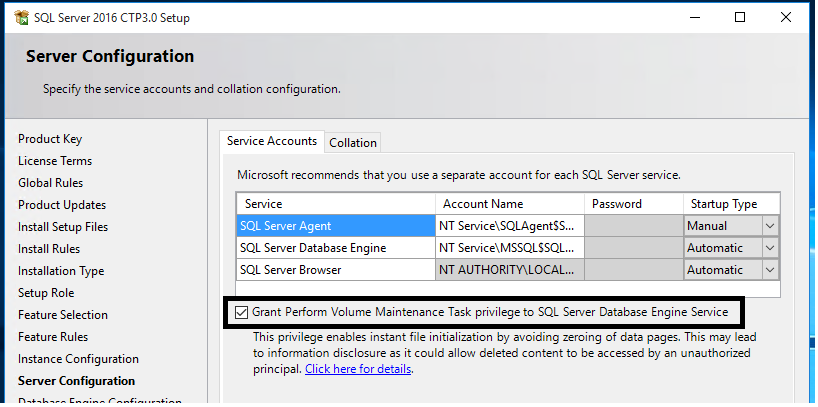
## Instant File Initialization

IFI is an NTFS feature that was added many years ago. When a database file needs to grow, SQL Server must ask the OS for more disk space. Part of this process is that the OS must allocate then zero out the disk sectors. The file is not usable until this is done. IFI skips the zero out process and allows writes right away.

IFI can be set in the local security policy.



Starting in SQL Server 2016 IFI is an option during installation.



VMware Configurations and Best Practices

There are various general configurations to consider optimizing on any Windows Server hosting a SQL Server instance.

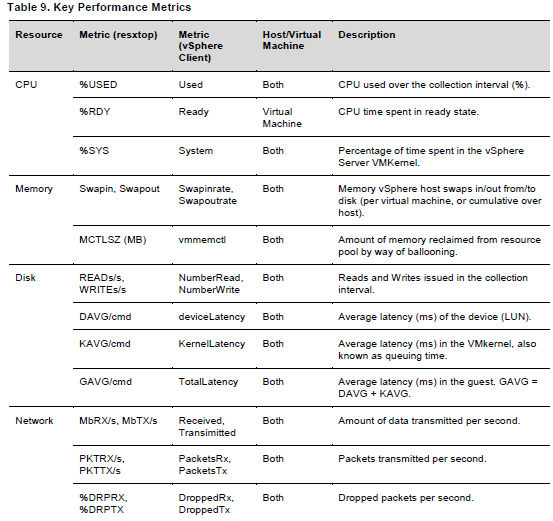
SQL Server on VMware Best Practices Guide

<http://www.vmware.com/files/pdf/solutions/SQL_Server_on_VMware-Best_Practices_Guide.pdf>

## General Configurations

Basic ESXi troubleshooting performance issues can be started by following these guidelines:

* [Troubleshooting ESX/ESXi Virtual Machine Performance Issues](https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2001003)
* [Tips for Configuring Microsoft SQL Server in a Virtual Environment](https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1002951)



## CPU Ready

Overcommitting the VM Host CPU to Guest VMs can cause more trouble than benefit. The hypervisor must keep track of CPUs and context switch between them across all guest VMs. Try to “Right-Size” the guest machines rather than over commit.

***Suggested a CPU Ready % of under 5%.***

The command “esxtop” can be run from the ESX host to get general statistics about the VM host.

For DatAdvantage we like to see waits in CPU Ready to be below 10,000ms. A range of 5000-8000ms should be as high as they get.

Reservations on CPU are **not** required for DatAdvantage.

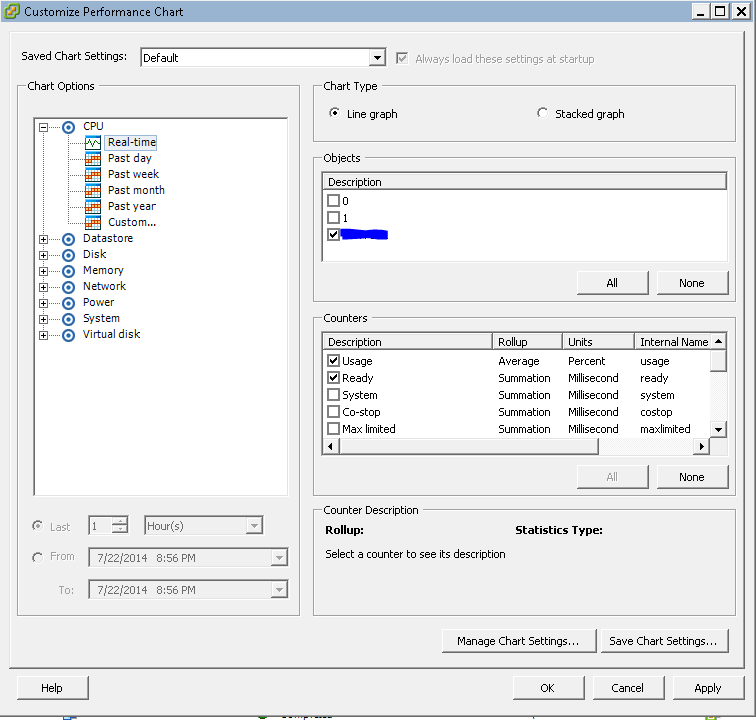
* [Converting Between CPU Summation and CPU % Ready Values](https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2002181)
* [Determining if Multiple Virtual CPUs are Causing Performance Issues](https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1005362)

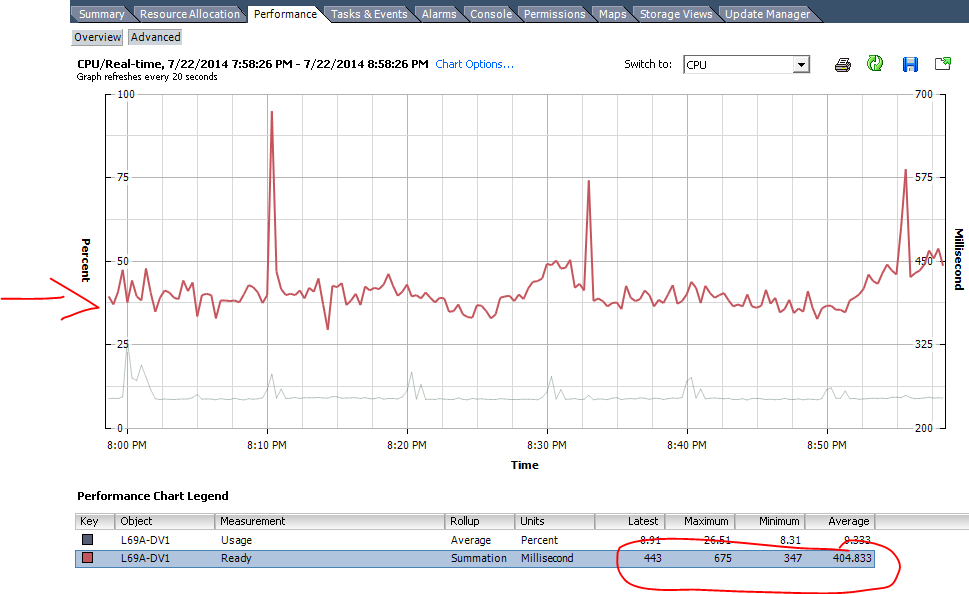
## Memory Ballooning

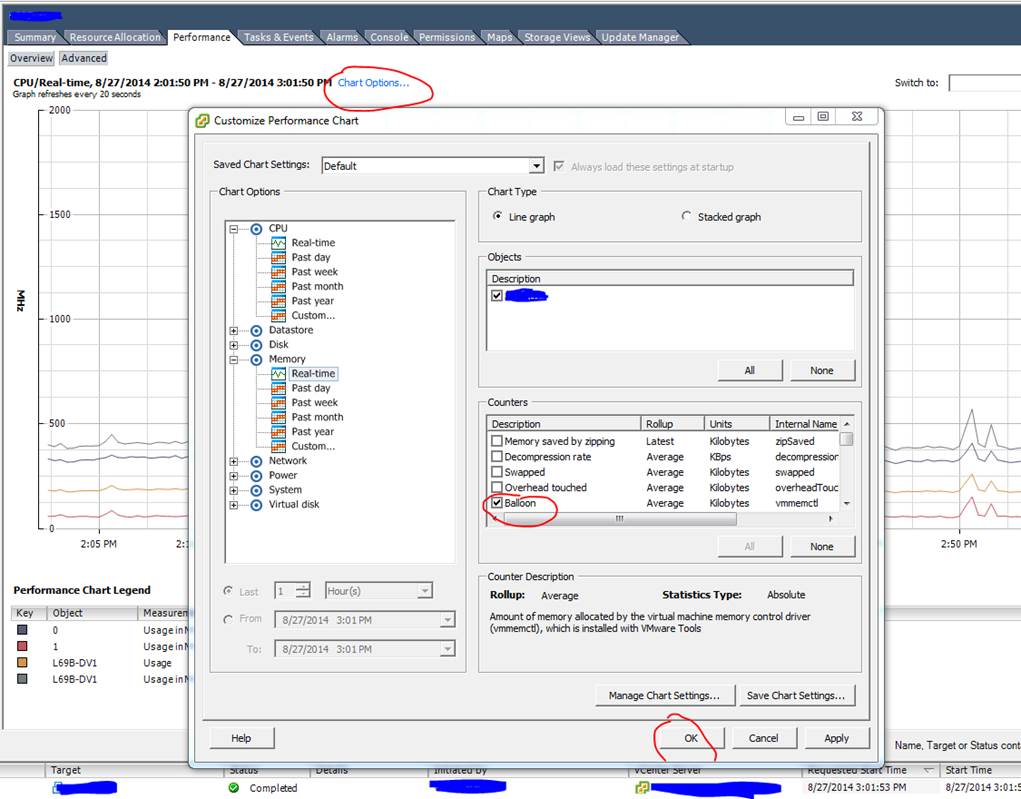
If memory reservations are in place make sure there is enough memory in the lower bound for the guest machine to perform without excessive paging. Also, make sure the VM host is not stressed for memory and the hypervisor does not have to reclaim memory to service other guests.

* If memory is overcommitted then either increase memory to the host OR reduce memory to the guest VMs
* Run esxtop, type "m" for memory, type "f" for fields, select "j" for memory ballooning statistics, look at MCTLSZ value (amount of memory reclaimed by balloon driver)
* Type "f" for field, select letter for Memory Swap Statistics (SWAP STATS), review SWCUR value (current swap usage)

Another way to display this information is through vSphere. Examine the chart option for the guest VM.









Memory reservations can be made but be careful here. We need to make sure there is enough memory in the lower bound for the guest VM to perform without excessive paging. Check that the host VM is not stressed for memory and the hypervisor is not always reclaiming memory from other guest VMs.

## Disk

Utilize Paravirtual SCSI adapters if possible for disks presented to guest VMs.

[Configuring Disks to Use VMware Paravirtual SCSI (PVSCSI) Adapters](https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1010398)

Troubleshooting Performance Issues with the Database

## Checklist for Performance Troubleshooting

In order to properly troubleshoot possible performance issues we need a general approach. This checklist gives a straightforward way to conduct a general assessment of the performance of the database.

Similar to the concept of triage first we need to determine how serious is the problem. There are varying degrees which merit different actions to remedy. An example of problem severity:

1. Server is down or not responding
2. Server is going to go down in a few minutes if we don’t fix the problem
3. Server is in bad shape but will survive even if we don’t do anything
4. Some parts of DatAdvantage are unusually slow
5. A user report is running slow

Essentially, we first need to assess if we are dealing with an outage or a performance issue. Knowing that helps us know how to proceed.

Below are detailed accounts of what to do depending on the severity of the problem as denoted above. Let us start with a general approach safe, but not necessarily enough, for all issues.

### Checklist – General

There is some information that usually tells us the health of a SQL Server database server.

Any health check should include at a minimum:

* Examine the event logs
* Procure and analyze SQL Error Logs – or send to TierIIV for analysis
* Run spWhoVaronis and analyze or send to TierIV for analysis
* Understand currently running MC jobs and long running jobs (as needed)
* Gather data from the InstalledPatches table in VrnsDomainDB
* Collect output from sys.configurations on the main IDU database server
* DA version
* Exact SQL Server build, version, and edition
  + This query can retrieve that info:

|  |
| --- |
| SELECT SERVERPROPERTY('productversion') AS 'ProductVersion', SERVERPROPERTY ('productlevel') AS 'ProductLevel', SERVERPROPERTY ('edition') AS 'Edition'; |

* Check basic performance configurations and assess if optimal as described earlier in this document
  + SQL Server
    - MAXDOP
    - Cost Threshold for Parallelism
    - Processor I/O Affinity
    - Memory allocation to SQL Server (min and max)
      * **NOTE: screenshots of each property tab of the instance shows all this information**
    - Tempdb configuration
      * Under the SQL instance look into the system databases and right click to select properties on tempdb.
      * Capture the filers, initial values, autogrowth, max. Screenshot is good.
  + Windows Server
    - Capture output of System Information either by typing that into Windows Server or “systeminfo” at the command prompt
    - Check that power settings are always High Performance
  + VMware (if applicable)
    - CPU Ready %
    - Memory Ballooning
    - Screenshots of the settings for the VM guest hosting SQL Server

### Checklist – Server is down or not responding

This is a severity 1 event. We need to verify if we can get a client thread open to the database or get the database to respond to some input. A simple list of actions to take here include:

* Can you ping it?
  + Do this in both directions.
  + If we cannot ping the server then engage the customers’ network team to evaluate. No further action can be taken here.
* Can you connect to the database server?
  + Try using SSMS on the database server and another computer.
* If SSMS does not work then try using another client like SQLCMD to establish a connection.

### Checklist – Server is going to go down in a few minutes if we do not fix the problem

This is a high severity incident. You are here if you can successfully perform the steps in the above severity 1 example. Since we can query the database, we need to find out what is running.

* If you are able to form a connection and get the database to respond to your input then start to understand what is currently running on the system.
  + Use spWhoVaronis to capture the processes on the database server
  + Use the following queries to get a list of MC jobs currently running - run on VrnsDomainDB
    - Currently running jobs

|  |
| --- |
| --currently running jobs  select DateDiff(minute,TimeStarted,isnull(TimeFinished,getdate())) as 'ElapsedMin', JobExecutionID, RootID, JobID, [Description], AggregatedStatus, ExecutionState, ExceptionState, TimeStarted, TimeFinished  from JOB\_Executions with(nolock)  where JobExecutionID = RootID and IsCompleted = 0  order by ElapsedMin desc; |

* + - Jobs started yesterday and today

|  |
| --- |
| --jobs started yesterday and today  DECLARE @StartDate DATETIME;  SET @StartDate = CONVERT(CHAR(8),GETDATE() - 1,112)  select DateDiff(minute,TimeStarted,isnull(TimeFinished,getdate())) as 'ElapsedMin', \*  from JOB\_Executions with(nolock)  where JobExecutionID = RootID and TimeStarted >= @StartDate  order by TimeStarted desc; |

* + - Jobs running over the past 30 days

|  |
| --- |
| --jobs over the past 30 days  DECLARE @StartDate DATETIME;  SET @StartDate = CONVERT(CHAR(8),GETDATE() - 30,112)  select DateDiff(minute,TimeStarted,isnull(TimeFinished,getdate())) as 'ElapsedMin', RootID, JobID, [Description], AggregatedStatus, TimeStarted, TimeFinished, IsCompleted  from JOB\_Executions with(nolock)  where JobExecutionID = RootID and TimeStarted >= @StartDate  order by TimeStarted desc; |

* + **NOTE:** any of the above queries can be copied into Excel to analyze.
  + Analyze the output to see what job executions are taking place. Send to TierIV DB Engineering if assistance is required.

**NOTE: Do not trust Activity Monitor! It lies and often provides information that does not give us the whole truth.**

Since there must be some reason as to why someone thinks the server may go down soon please find out what are the leading thoughts from relevant stakeholders. That will lead your investigation after the basics are covered.

### Checklist – Server is in bad shape but will survive even if we don’t do anything

For this, you can use the same checklist as in the above step.

We need to gather information about the current workload and job history. Additionally, please make sure to gather the Application and System server event logs and SQL Error Logs.

### Checklist – Some parts of DatAdvantage are unusually slow

This is detailed enough of a problem that we can employ other techniques to drill down into the root cause. First, get a clear problem statement: ex. DCF jobs are slow, reports do not render, running out of memory, etc.

If the complaint is against a specific set of jobs then go through the JOB Executions table to learn about run history.

Like every step before – always grab the Application and System server event logs along with the SQL Error Logs for analysis.

Try to reproduce the undesired behavior and capture information about it including logs, screenshots, and other details.

### Checklist – A user report or a particular job is running slow

Here we can dig down deep into the complaint.

If we have reporting issues then make sure to acquire the AdvSearchSessions table.

If we have a slow job then please gather the relevant job executions. Additionally, gather any relevant notifications. Every DatAdvantage associated database has a “notifications” table. Query this to find some history. **Caution – this table can be quite large.**

Like every step before – always grab the Application and System server event logs along with the SQL Error Logs for analysis.

Below in this section there are more details about the techniques described above.

## Event Logs – Application and System

This is a first line approach to learning about system errors. Export these and send for analysis. Filter the log by errors and warnings to start. You may need to include more to gain context of the happenings on the server.

## SQL Server Error Log

Many useful information regarding system health and performance can be found in the SQL Server error log.

The default location:

C:\Program Files\Microsoft SQL Server\MSSQL<version #>.MSSQLSERVER\MSSQL\Log

**NOTE: You can zip up the entire LOG folder, which contains the SQL Error Logs, and send to TierIV DB Engineering.**

## Stored procedures

Let us discuss the relevant stored procedures bundled with DatAdvantage that we can use to diagnose and understand performance issues.

### spWhoVaronis

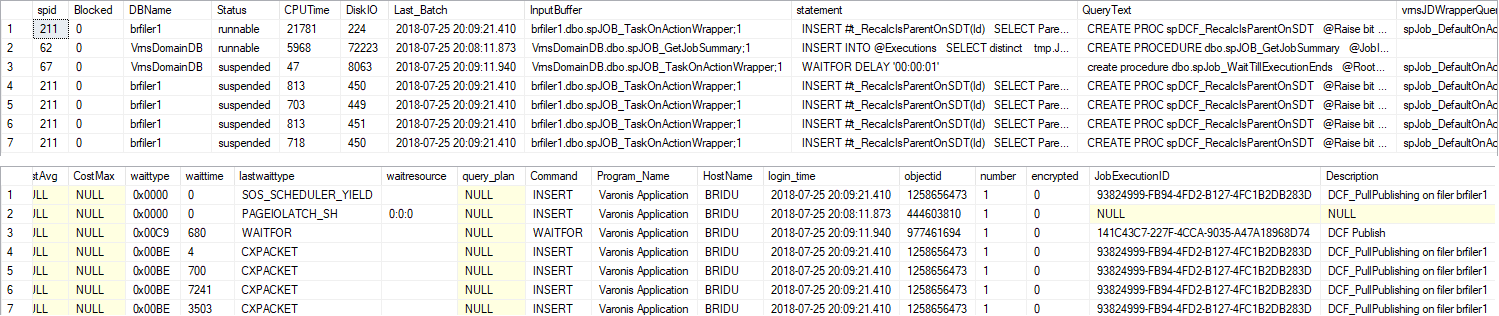
Every release of DatAdvantage comes with a procedure called spWhoVaronis that we wrote internally to help us learn what is running. Run this procedure on the VrnsDomainDB database.

This is a great entry point to troubleshooting and assessing performance. Start here:

**Usage:**

|  |
| --- |
| EXEC dbo.spWhoVaronis running; |

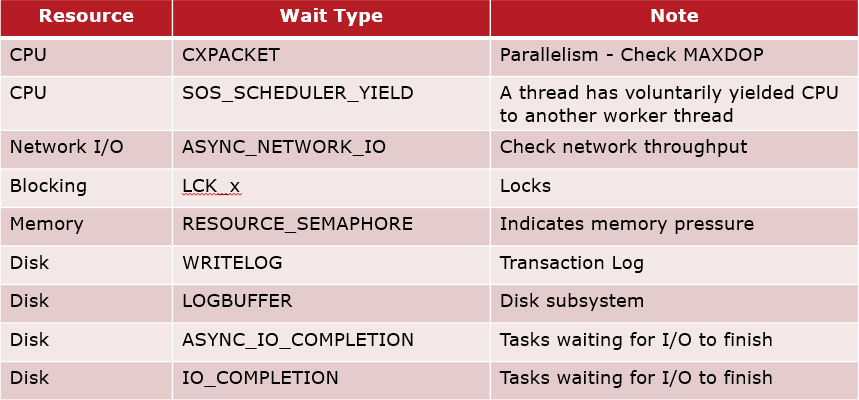
**Output:**



This output tells us some things about query performance.

1. First – it tells us what processes are running in the database. This is our jobs, reports, and any ad-hoc queries being run in SSMS or other client. This is a quick way to gauge activity on the database server.
2. Blocked – this column tells the SPID that is blocking the SPID to the left of the Blocked column. This is useful to observe if there is a lot of blocking. It can explain why a job is not starting or able to finish.
3. DB Name – shows the database where the SPID is executing
4. Status – the status of the query. Running queries are in 1 of 3 states: running, runnable, suspended. Ignore the rest.
5. CPU Time – an indicator of CPU resources being consumed by the SPID
6. Disk IO – an indicator of Disk IO resources being consumed by the SPID
7. Input Buffer / statement / QueryText – these 3 columns store varying amounts of the query text which the SPID is running. This is often helpful debugging information for knowing what exactly the query is currently trying to process or do.
8. Last Wait Type – this indicates the wait type that last occurred on the SPID.

**Chart of Wait Types and their Cause**



1. Program Name – useful to filter what SPID belong to Varonis Application or other users
2. HostName – tells which host the query is running
3. JobExecutionID – this is the unique identifier of the MC job that is running
4. Description – this is the MC job that is running

### spDiag

We have bundled many internal diagnostic procedures to assist troubleshooting processes. The naming convention we use is “spDiag\_” as a prefix.

* spDiag\_ENVI (fka spENVI\_FoundProblems) – this procedure has been around since version 5.8.
  + This list of returned output and meaning is located here: <https://varonis.sharepoint.com/tfs/dev/DBTeamSite/SitePages/KnowledgeBase/ENVI_FoundProblems.aspx>
  + This is a great way to troubleshoot the ENVI table structure of our product features and components.
* Diagnostic Jobs Framework
  + Documentation located here: <https://varonis.sharepoint.com/tfs/dev/DBTeamSite/SitePages/KnowledgeBase/Diagnostics%20-%20Jobs%20framework.aspx>
  + Various jobs that can easily tell us what is running and many useful details about jobs. As of this writing, there are 26 different procedures here to use.
* FileWalk Diagnostic Framework
  + Documentation: <https://varonis.sharepoint.com/tfs/dev/DBTeamSite/SitePages/KnowledgeBase/Diagnostic%20-%20FW.aspx>
  + This is a quick way to assess FileWalk related activities on the system
* Events Diagnostic Framework
  + Documentation: <https://varonis.sharepoint.com/tfs/dev/DBTeamSite/SitePages/KnowledgeBase/Diagnostic%20-%20events.aspx>
  + This is a quick way to assess events related activities on the system
* DCF Diagnostic Framework
  + Documentation: <https://varonis.sharepoint.com/tfs/dev/DBTeamSite/SitePages/KnowledgeBase/Diagnostic%20-%20DCF.aspx>
  + This is a quick way to assess DCF related activities on the system

Try out some of the Diagnostic procedures included in each release. They are intended to make troubleshooting faster and easier.

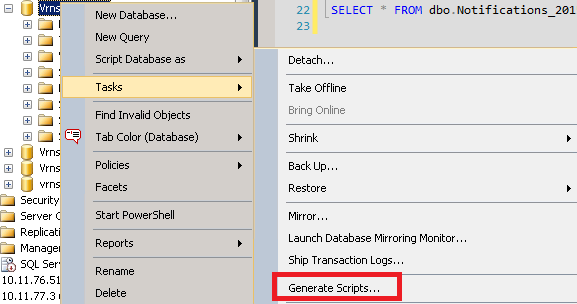
## Notifications

Every database has a notifications table that logs information about activities. Here is how to gather this information for analysis:

1. Create a table to hold the desired data
   1. In this case we are interested in the day of the DTE error – October 1st.
   2. Run this on the domain database:

|  |
| --- |
| SELECT \*  INTO dbo.Notifications\_20151001  FROM dbo.[Notifications] WITH(NOLOCK)  WHERE [timestamp] >= '2015-10-01 00:00:00:000' AND [timestamp] <= '2015-10-01 23:59:59:999'; |

1. Generate a script of the data and schema
   1. Right click on the database VrnsDomainDB 🡪 Tasks 🡪 Generate Scripts…



* 1. Follow the wizard to select the table created above

## PerfMon

Performance Monitor (PerfMon) is a tool that comes with Windows Server that tracks resource utilization on the server.

**NOTE: typically, we want PerfMon to run for at least a 24 hour cycle. It depends on the problem behavior we are tracking. It is not uncommon to run this for 5-7 days.**

Combined with data from VrnsDomainDB, in particular JOB\_Executions, we can discern what is causing high resource utilization during a period of time.

**NOTE: be sure to capture this information both before and after running PerfMon!**

Documentation for instructions and usage is located here:

<https://varonis.sharepoint.com/tfs/dev/DBTeamSite/SitePages/KnowledgeBase/Data%20Collection%20Scripts%20+%20PerfMon.aspx>

## DMVs and queries

This script can gather configuration details of SQL Server:

|  |
| --- |
| WITH recommend AS (  SELECT  cost\_threshold = 50  , physical\_ram\_mb = sm.total\_physical\_memory\_kb/1024  , mem\_max\_limit = (sm.total\_physical\_memory\_kb \* 85)/102400  , si.cpu\_count  , maxdop\_proposed = CASE  WHEN si.cpu\_count <= 4 THEN 0  WHEN si.cpu\_count <= 10 THEN si.cpu\_count - CEILING(si.cpu\_count / 4.0)  ELSE 8  END  FROM master.sys.dm\_os\_windows\_info AS wi WITH (NOLOCK)  CROSS JOIN  master.sys.dm\_os\_sys\_info AS si WITH (NOLOCK)  CROSS JOIN  master.sys.dm\_os\_sys\_memory AS sm WITH (NOLOCK)  )  , recommend\_s AS (  SELECT  cost\_threshold = CAST(r.cost\_threshold AS NVARCHAR(MAX))  , physical\_ram\_mb = CAST(r.physical\_ram\_mb AS NVARCHAR(MAX))  , mem\_max\_limit = CAST(r.mem\_max\_limit AS nvarchar(MAX))  , cpu\_count = CAST(r.cpu\_count AS NVARCHAR(MAX))  , maxdop\_proposed = CAST(r.maxdop\_proposed AS NVARCHAR(MAX))  FROM  recommend r  )  ,conf AS (  SELECT  c.configuration\_id  , c.name  , c.value  , c.value\_in\_use  , CASE  WHEN c.configuration\_id IN (1538) THEN rs.cost\_threshold  WHEN c.configuration\_id IN (1539) THEN rs.maxdop\_proposed  WHEN c.configuration\_id IN (1544) THEN '< ' + rs.mem\_max\_limit  + ' MB (85% of ' + rs.physical\_ram\_mb + ' Physical RAM )'  ELSE ''  END AS Recommended  , CASE  WHEN c.configuration\_id IN (1538) THEN CASE WHEN c.value\_in\_use = r.cost\_threshold THEN 1 ELSE 0 END  WHEN c.configuration\_id IN (1539) THEN CASE WHEN c.value\_in\_use = r.maxdop\_proposed THEN 1 ELSE 0 END  WHEN c.configuration\_id IN (1544) THEN CASE WHEN c.value\_in\_use <= r.mem\_max\_limit THEN 1 ELSE 0 END  ELSE NULL  END AS IsValid  , c.description  , c.is\_advanced  , c.is\_dynamic  , c.minimum  , c.maximum  FROM  master.sys.configurations AS c WITH (NOLOCK)  CROSS JOIN  recommend r  CROSS JOIN  recommend\_s rs  )  SELECT \* FROM conf c  ORDER BY  c.IsValid DESC,  1  OPTION (RECOMPILE);  GO |

The supporting KB is located: <https://varonis.force.com/Central/s/article/WDOC-3616>

## SQL Profiler

TODO

## Extended Events

TODO

## Capacity Planning – Sizing Script

TODO

# Appendix

TODO: Include helpful explanations of pieces in the above document which merit their own explanations here.

# References

<http://social.technet.microsoft.com/wiki/contents/articles/1145.sql-server-glossary.aspx>

<https://support.microsoft.com/en-us/kb/2806535>

<https://blogs.msdn.microsoft.com/psssql/2010/11/19/how-it-works-io-affinity-mask-should-i-use-it/>

<https://support.microsoft.com/en-us/kb/2663912>

<https://www.sqlskills.com/blogs/jonathan/how-much-memory-does-my-sql-server-actually-need/>

<http://support.microsoft.com/kb/315512>

<https://technet.microsoft.com/en-us/library/cc966534.aspx>

<https://msdn.microsoft.com/en-us/library/ms190219(v=sql.110).aspx>

<http://support.microsoft.com/kb/315512>

<https://social.technet.microsoft.com/wiki/contents/articles/13383.best-practices-for-page-file-and-minimum-drive-size-for-os-partition-on-windows-servers.aspx>

<https://blogs.technet.microsoft.com/askperf/2007/12/14/what-is-the-page-file-for-anyway/>

<https://support.microsoft.com/en-us/help/2860880/how-to-determine-the-appropriate-page-file-size-for-64-bit-versions-of>

<https://blogs.technet.microsoft.com/motiba/2015/10/15/page-file-the-definitive-guide/>

<http://www.vmware.com/files/pdf/solutions/SQL_Server_on_VMware-Best_Practices_Guide.pdf>

<https://kb.vmware.com/s/article/2001003>

<https://kb.vmware.com/s/article/1002951>

<https://kb.vmware.com/s/article/2002181>

<https://kb.vmware.com/s/article/1005362>

<https://kb.vmware.com/s/article/1010398>

<https://docs.microsoft.com/en-us/sql/sql-server/install/file-locations-for-default-and-named-instances-of-sql-server>

<https://varonis.sharepoint.com/tfs/dev/DBTeamSite/SitePages/KnowledgeBase/Data%20Collection%20Scripts%20+%20PerfMon.aspx>

<https://varonis.force.com/Central/s/article/WDOC-3616>