

The Top 10 Tips for SQL Server Performance & Resiliency



Chris Shaw

Senior DBA and Microsoft MVP
XTIVIA Inc.

A Publication of
XTIVIA

TABLE OF CONTENTS

About the Author	04
Introduction	05
Tip #1 CHECK YOUR BACKUPS	06
A. Backup SQL Servers	07
B. Testing the Restores	07
C. Take Backups Offsite.....	08
D. Securing Your Backup.....	08
E. Backup Everything	09
Tip #2 CHECK YOUR SECURITY	10
A. Not Everyone Should Be a SA.....	11
B. Not Everyone Should Be a DBO.....	11
C. Securing Your Backups.....	12
D. Linked Servers	12
E. BUILTIN\Administrators.....	12
F. No Policy Management for SQL Server Accounts.....	13
Tip #3 MAINTAIN YOUR SQL SERVER	14
A. Fragmented Indexes	15
B. Stale Statistics.....	15
C. Consistency Check.....	16
D. Monitoring	16
E. Database Mail, Alerts and Operations Not Configured.....	17

Tip #4 HAVE A BASELINE	18
A. Performance Baseline	19
B. Configuration Baseline	20
Tip #5 SET A MAXIMUM MEMORY	21
A. Default Configuration	22
B. Memory Issues	22
C. Memory Configuration	22
Tip #6 KEEP A CHANGE HISTORY	24
A. Changing Interfaces	25
B. Change Process	25
C. Rollback Script	26
Tip #7 DEVELOP A DISASTER RECOVERY PLAN ...	27
A. Cost	28
B. The Basics	28
C. Pay Attention to the Details	28
D. Test	29
Tip #8 OPTIMIZE TEMPDB DATABASE	30
A. About TempDB	31
B. Optimization.....	31
Tip #9 MANUALLY SHRINK YOUR DATABASE	32
A. Differences in Database Administrators	33
B. Problems with Automatically Shrinking	33
C. Status of Indexes	33
D. Shrinking a File	34
Tip #10 STORAGE	35
A. Changes in Storage	36
B. Storage Issue Indicators	36
C. Additional Tools	36
Learn More	37

About the Author

Chris started his database career in 1993. He began by working with Lotus during his time in the Marine Corps. Following the Marines Chris continued working with databases for companies such as Wells Fargo, Pulte Mortgage and Yellow Pages Inc. He later consulted with insurance companies including Anthem Blue Shield and Admini Quest. Chris has been writing and speaking about SQL Servers for over 15 years at events such as SQL Connections, Pass and SSWUG Ultimate conferences. Chris was the Conference Director for SSWUG in 2008. Chris first received the Microsoft MVP award in 200 and just received his fifth MVP award in April of 2014. He is the founding member of the Colorado Springs SQL Server User Group. Chris has many featured articles on the SSWUG website, Chris was a contributing author for SQL Server 2005 Bible, SQL Server MVP Deep Dives, Volume 2 and most recently SQL Server 2012 Best Practices.

Chris joined the XTIVIA database team in July of 2012, as he enjoys “real-life” server support and troubleshooting.

Chris was born at the Air Force Academy in Colorado Springs. Following in his father’s military footsteps, Chris joined the Marines in

1990. After finishing his contract he relocated back to Colorado Springs. Recently, Chris has moved with his wife to Texas.

Chris’ DBA/SQL specialties include: SQL Server Administration, HA and DR with SQL Server, SQL Server 6.5, SQL Server 7.0, SQL Server 2000, SQL Server 2005, SQL Server 2008, SQL Server 2008 R2, SQL Server 2012, SQL Lite Speed, ErWin, Vendor Relations, and SQL Sentry.

“

As much as I enjoy teaching and speaking at events, I want to know my knowledge is based on experience, and not theory. I like to help an organization determine what the obstacles are, and help remove them so they can accomplish their mission.

Chris Shaw

”



Introduction

I have been a consultant for a number of years so I lost track of how many different servers I have worked on a long time ago.

However, today as sit down and think, it occurred to me that so many of the countless hours I have spent working on correcting issues could have been solved by not making some of the common most mistakes I see.

My goal is to help you avoid the same issues, by highlighting each of the common mistakes that I have seen and sharing some of the resolutions associated with those errors. This list is by no means all-inclusive.

Unlike other top 10 lists that start at number 10 and count down in dramatic fashion, we are going to start at the number one position. Drum roll please...

Chris Shaw



1

Check Your Backups



A. Backup SQL Servers

I am not sure if companies don't realize that SQL Server should be backed up independently of the operating system or if they just assume that the servers don't need to be backed up at all. Granted most of the servers that I see that have no backup plan at all are smaller companies, and I am seeing this less and less each day. What I do see most often is backups that don't match the company's data protection needs.

When looking at how to backup your SQL Server you need to start with the question of how much data can you lose. Once that question has been answered it can be used to drive your backup strategy. There are other questions you will need to address such as where to store the backups, how long do you keep the backups around and how fast do you need your system back up and running if you are having to restore the data.

B. Testing the Restores

As crazy as it sounds, getting a backup of your server is not the last step in this backup task. There is only one way to make sure that your backups are going to work the way that you want it to, and the time to find out that the strategy is flawed is not when you need to restore your database. There are many aspects to restoring your database that are often forgotten.

Consider this, if your backup is stored at an offsite facility, how long will it take to get that backup back on site where a DBA can start restoring the database. What are the steps required to retrieve this backup from the offsite location? Then, when that backup file is back on site with your server, who needs to be there to move a backup from one medium to another. This doesn't only impact organizations that have tape media, if you are storing your backups in the cloud how long will it take to get that backup from the cloud and bring it back to your servers, and what happens if the event that triggered you searching for backups has also impacted your internet connection?

Before you need to restore your database, test it. With the technology that we have access to today it is simple to turn up a VM so you can test restore your backups. To take it one step further this should be part of your monthly maintenance. I work with some companies that know how important those backups are and they restore every backup they take to make sure it is a solid backup.

C. Take Backups Offsite

I hate to make a point by using a scare tactic, but the reality of the situation is that disasters happen. There are floods, fires, and earthquakes to just name a few. I live in Colorado and over the course of a couple summers we saw 2 major fires that destroyed many homes and businesses. We saw the loss of a restaurant that has been a spotlight of our community for over 40 years, we saw major shipping companies that shut down, but because of solid disaster recovery plans they were back up and delivering packages within 8 hours and they didn't even have their trucks.

There are so many technologies that can assist you with your backup strategies, you can store your backup in the cloud, you can send it over the internet to a data center in another state. No matter the solution, if your data is critical to your operations your backups should be stored in a location separated from your SQL Servers.

Every layer of your backup strategy needs to consider the protection of your backup. This includes not storing your backups on the same storage where your data file is located. As solid as SAN technology is, there is always the potential of losing your storage. Backups that are stored next to the data itself can be a costly mistake.

D. Securing Your Backups

If you watch the news you may have noticed a number of organizations that have had their data compromised. The larger the organization the more difficult the public relations become after such a compromise. The last thing any DBA wants to be part of is a letter to customers letting them know the data they are responsible for has been compromised.

Protect your database, encrypt them with a password and manage the permissions on your backup locations like they contain your personal social security number and your debt card PIN code.

E. Backing Up Everything

SQL Server today can have many dependencies outside the databases. CLR code allows the database to use code that is not stored in the database, and certificates are used to encrypt the data. If those are not backed up with the system, restores can become a serious problem.

Make sure everything is backed up that your server would need to continue operation. This is another good reason for a complete test restore.

Maybe the primary question you need to answer is; *how important is your data?* What is the impact of not being able to restore your database, can the data be recovered from other means? What is the impact on your backup strategy to your business? Once you have a good solid answer to how important your backups are, look at your existing strategy and determine if your backup strategy matches your backup plan.

2

Check Your Security



A. Not everyone should be a SA

The SA name is short for Systems Administrator. This account has permissions to everything inside SQL Server. When a user requests SA, they may be referring to a couple things. They may be asking for the password of the SA account or they may want you to assign their login to the Systems Administrator role. In my opinion, there are very few people who need this level of access to a SQL Server. Usually, these people have a job title that says, "Database Administrator". Keep in mind: the more people that have this access level to your SQL Server, the more risk you take when it comes to how many people can make really big mistakes.

To be secure, make sure the SA account has a complex password. I recommend setting the SA password to a complex password and disabling the account to be a bit more secure. The administrator's role should have their own accounts. Treat the SA account as hidden back door into your system and only use it when nothing else works.

B. Not Everyone Should Be a DBO

In like fashion to SA access, not everyone needs DBO access to a database. DBO is short for database owner. Accounts with this access have permissions to everything internal on the database. This means they can delete users, tables, stored procedures and even data. Since SQL Server cannot make a determination between DELETE statements done on accident and on purpose, it becomes imperative that someone with DBO or SA access knows what they are doing.

I often see people treat security levels as a negotiation between two parties, almost like what you would experience when purchasing a car. The requestor asks for 'SA' level access and the DBA might respond by haggling them down to Data_Reader access, only to have the requestor push for something in the middle, like DBO access. As funny as this sounds, some DBA's agree to it. The theory being that DBO is more restrictive than 'SA', so logically, security is intact. The end result in this hypothetical bargain is 'DBO access', which many people don't need. Check the accounts and find out what is required to complete the work tasks. Don't give employees any more room to break things than they need.

C. Securing Your Backups

The larger the organization, the more difficult public relations become after a data breach. The last thing any DBA wants to be part of is a letter to customers letting them know the data they are responsible for has been compromised.

Protect your databases, encrypt them with a password and manage the permissions on your backup locations like they contain your personal social security number. The protection of your database backups doesn't stop there, make sure the offsite locations is secure as well.

D. Linked Servers

Linked servers can add tremendously to the ease of retrieving data from your SQL Servers. Often, people get pretty excited when they learn how easy it is to use and configure. However, a big mistake that I often see has to do with the security configuration of the link. When the linked server is configured, settings that should be set to protect the data on the server being linked to. I often see configurations that will allow a specific named user that may have Data_Reader on the primary server connect to the linked server using the SA account.

E. BUILTIN\Administrators

In older versions of SQL Server, if you had an administrator account on the windows machine such as local admin access, you would by default inherently have SA access to the SQL Server. This can be very helpful in small shops where a Windows Admin may also be a SQL Server Admin. The issue is sometimes the Windows Admin shouldn't have that level of access to the data or configuration of SQL Server.

F. No Policy Management for SQL Server Accounts

SQL Server authenticated accounts, or accounts that reside only in SQL Server have the option to have the policy enforced by the Windows Policy. Sometimes these are configured this way because people don't want to change the password for SQL Server, or have to worry about having a strong enough password. This might be related to the theory that if the server is behind a firewall it is protected enough. It is a good idea to have your passwords in SQL Server follow the same policy that your windows accounts follow.

Wrap-up

This is just a short list of some of the common security mishaps I have encountered. There are others you should be aware of, like updating your service packs, staying on top of when people leave your company or auditing your group memberships. When you look at your database and you believe you have secured it enough, take another look. Consider creating a table called My Personal Info, and in that table insert your credit card, your debit card (don't forget the P.I.N), your social security number and your banking information. Once that is all inserted there, how comfortable do you feel? Your data deserves to be secure and people every day are relying on you the data owner to keep it secure.

3

Maintain Your SQL Server



A. Fragmented Indexes

Indexes in SQL Server are very similar to indexes in a book. The primary difference between indexes in a book and in your database is the indexes contained within the database are much easier to rebuild when new information is inserted. Imagine you had a book with 100 pages of text. If the book were indexed, finding specific information wouldn't be too difficult and shouldn't take much time. Now add 50 pages of new text in the middle of the book. Finding specific text after the 50 pages has been added would be difficult, all the page numbers would be off and the new text wouldn't be indexed.

To solve this problem with SQL Server, we have the option to rebuild the index or reorganize an index. This should be completed on a regular basis. This ensures data location accuracy, which eases the work load on SQL server. This allows SQL Server to find the specific data it needs quickly and efficiently.

B. Stale Statistics

Sometimes, people have difficulty understanding SQL Server Statistics. Think of it this way: If you managed a sports team and it was your job to find new talent, would you want to see how much potential the talent has? You may want to know how strong they are or how fast they can run. Trying to pick up a new quarterback without that information would be difficult. What if the player's statistics you were looking at were 5 years old? Maybe the statistics were from when they were in the best shape of their career and now they have gone a little past their prime. You may hire a player based on bad information if you didn't know the statistics were out of date, they may not be that star performer you were looking for.

SQL Server keeps statistics for you as well. This information keeps your SQL Server aware of the information in the database and helps it decide on what the fastest way to retrieve data is. However, there are times when this information needs to be refreshed. Keeping your statistics up to date is going to keep your SQL Server aware of what shape your data is in.

C. Consistency Check

Isn't it nice when your car starts to warn you before something really bad is going to happen? Sometimes it can be something as simple as change the oil in the engine or changing the battery.

SQL Server has warning signs as well; DBCC commands or database consistency checks should be executed to catch when something may be amiss in your data. This doesn't mean that it can catch everything, but this layer of protection can save you some serious pain down the road.

D. Monitoring

Children that are being watched at a child care center or by parents are a lot less likely to get themselves into trouble. SQL Server is no different. A monitored SQL Server is less likely to have critical non-recoverable errors than servers that are not monitored. For example, pretend you have a database that requires 100 gigs of space for a backup file. If that storage falls below having 100 gigs of space available, your SQL Server backup is going to fail. What would happen if you lost the storage due to a hardware failure and now you need to restore from a backup? You could lose a tremendous amount of data and time, assuming you are able to recover it at all.

Now consider the same situation with a monitored SQL Server. A monitored SQL Server should inform you that you had a backup failure, and if it is scheduled via a Job in SQL Server, you can receive an alert from the SQL Server itself. In addition, there would be entries in the error log that you would see as well. A monitored SQL Server is going to give you an opportunity to correct the backup issue, before you need to restore the database.

It's getting easier to complete a basic install of SQL Server with each and every version. It takes very little time to have a database up and running. It is so simple that many organizations install their SQL Server and forget about it – at least until there becomes a problem with it. Most of the time when I am asked to perform an emergency fix or a restore of a database, it is being done on a server that isn't being monitored.

E. Database Mail, Alerts and Operations Not Configured

As mentioned in the monitoring section of this post SQL Server can be configured to send emails in a number of situations; if a job fails or errors are raised with specific severities. These emails are sent to operators that are configured within SQL Server. So when a backup fails, SQL Server can send you an email, allowing you to fix the issue.

When my son started to learn how to drive a car, we spent a number of hours talking about the maintenance. We reviewed how he should keep sand in his trunk to help get traction on the ice, how his oil needs to be changed and the tires need to be rotated, all to ensure his safety while on the road. A well maintained SQL Server can give your organization a stable and well performing database.

4

Have a Baseline



A. Performance Baseline

I am often reviewing systems after they have started to exhibit performance issues. My goal with these type of engagements is to return the server to normal or better than normal performance as quickly as I can. Without a performance baseline of how the system runs under normal circumstances, leaves whoever is troubleshooting the performance issues without a valuable tool.

At the basic of level, SQL Server performance can be grouped into 3 categories where a performance issue may lie; hardware, operating system and SQL Server. Consider having a performance monitor counter run that includes some basic counter that could be relevant to each section.

Memory\Available Bytes

Memory\Pages/sec

PhysicalDisk(*)\Avg. Disk sec/Read

PhysicalDisk(*)\Avg. Disk sec/Write

PhysicalDisk(*)\Current Disk Queue Length

PhysicalDisk(*)\Avg. Disk Queue Length

Processor(_Total)\% Processor Time

SQLServer:Access Methods\Full Scans/sec

SQLServer:Access Methods/Index Searches/sec

SQLServer:Buffer Manager\Page life expectancy

SQLServer:General Statistics\User Connections

SQLServer:Latches\Average Latch Wait Time (ms)

SQLServer:Locks(_Total)\Lock Wait Time (ms)

SQLServer:Locks(_Total)\Lock Waits/sec

SQLServer:Locks(_Total)\Average Wait Time (ms)

SQLServer:Memory Manager\Memory Grants Pending

SQLServer:Memory Manager\Target Server Memory (KB)

SQLServer:Memory Manager\Total Server Memory (KB)

SQLServer:Plan Cache(_Total)\Cache Hit Ratio

SQLServer:SQL Statistics\Batch Requests/sec

SQLServer:SQL Statistics\SQL Compilations/sec

SQLServer:SQL Statistics\SQL Re-Compilations/sec

System\Processor Queue Length

System\Context Switches/sec

B. Configuration Baseline

The baseline you have shouldn't stop at the performance monitor. The configuration of your SQL Server with a document explaining some of the non-standard configurations should be done as well.

Recently I was working on a performance problem for a client who has had their SQL Server online for years. The server had been slowing down over time and the client needed the performance to be better. When we were looking at the database configuration, we noticed the compatibility level on the database was set to SQL Server 2000 even though the SQL Server instance was 2008. Everyone who worked in the IT department when the server was configured had moved on a few years prior. The server and a number of settings configured that were not default, some of the settings were not best practice. The only way we could determine if the setting was correct or if it was just something someone overlooked was to make the change back to the default and complete regression testing to see the impact. This added a lot of time and delay to correcting the overall performance issue. A configuration baseline would have saved a lot of time and money. I recommend doing a health check\baseline on your SQL Servers once a year.

5

Set a Maximum Memory



A. Default Configuration

Default configurations are not always the best configuration for your server. Maximum memory allocated to SQL Server is a great example of this. If this setting is left unchanged, there is a chance your SQL Server could be starving the operating system of memory. When SQL Server starves the operating system, the OS cannot provide a stable platform. A struggling OS will have a difficult time retrieving data from disk, swapping processes, and completing standard OS functions to name a few issues.

B. Memory Issues

There are a number of ways to see if the operating system is running into memory issues. One of the fastest ways is to simply look at the task manager and see how much is available. If you review the available memory when the server is not performing well and the amount of available memory is low, there is a good chance the OS is being starved. With tasks like file copies, backups and virus scans, the operating system will look for memory to help complete the process. If SQL Server has that memory reserved, the OS will not have the memory it needs to work efficiently.

C. Memory Configuration

Proper SQL Server memory configuration can take some fine tuning as a number of variables need to be taken into consideration. For example if there are 2 SQL Server instances installed you will need to leave the memory configured for each SQL Server low enough so the Operating System has enough to work with. And don't forget processes like copying large files to different media, such as a file copy for a backup. Finding the balance can often leave DBA's going in circles. When you add virtual servers into the mix the variables increase even more and sometimes the memory configurations go against SQL Server best practices (VMWare recommends min and max memory to be set to be the same). There are many formulas for recommend initial max memory settings.

C. Memory Configuration (continued)

So where do you start? A general rule of thumb is to leave the operating system 20% of the memory. With that being said, I hesitate to leave the operating system anything less than 2 gigs. After making a memory setting adjustment, close attention should be given to monitoring the memory counters in the performance monitor and the DMV's in SQL Server to ensure the operating system has enough memory to perform well without fighting with SQL Server.

6

Keep a Change History



A. Changing Interfaces

Business today moves fast, companies change interfaces to keep content relevant and are continuing to offer new services to their client base. In a perfect world code is tested 10 ways to Sunday and is never released with bugs or design issues, however we don't live in that world. I mean no disrespect to the developers that spend hour upon hours trying to provide the best changes possible. There are just so many variables that making a release perfect every time is just not a reality.

There is an inherent risk with change; software, hardware or configurations including SQL Server.

When a change is being planned no matter how small of a change, these changes should be documented. Chances are most DBA's have heard this before; however, simple changes are often overlooked. Documenting the change is a great start; however, keep in mind that it is critical that the documented changes are easily researched. In my experience with working with SQL Server, most issues with SQL Server can be tracked back to a date when an issue started. Because of this, I recommend storing documents so they can be searched by date.

B. Change Process

Stability increases when you embrace change history and take change management couple steps further. Consider setting up a change process. Implementing policies such as not allowing changes to be made to the SQL Server unless they are being made via scripts. There are some great benefits to only allowing changes via script. Consider a small table change, a few of the benefits that you will see by using a script over the user interface:

- When changes are made via script, and the script has been saved it is easy to see exactly what was executed. When changes are made via the user interface you have to assume you know what the exact change was.
- Changes made via the user interface are susceptible to the "fat finger" otherwise known as a typo. You can review your change with your eyes, but having others review your change is more difficult. If you mistype the name you are now left with having to make a second change to correct your mistake.
- When using a script, you can share what the change is going to look like before the change is made. You can send the new table layout to other teams in the company and they can see exactly the totality of the change.

C. Rollback Script

When making changes it is also important to have a roll back script. This is most likely the most overlooked part of change management. Having a roll back script is not often needed and when everything is rolling out as planned there is no thoughts to having to roll back these changes. When a changes are made and it isn't going as well as it was planned someone may make the decision to cancel the change. A restore can often lead to extended hours of downtime and unexpected impacts to customers. These are the times when the extra effort put into a rollback script prove to be priceless.



Develop a Disaster Recovery Plan



A. Cost

Often when people hear “disaster recovery plan” their first concern is cost. Disaster recovery plans don't have to be expensive, expensive disaster recovery plans come from strict requirements.

About 10 years ago when I started as an independent consultant one of my first clients was contacting me to help build a disaster recovery plan for them. After our initial discussion I learned that some consulting firms had forecasted one hundred thousand dollars for solutions. Many large companies would look at that number as a bargain, however this client's company made less than 50k a year. The data changed about once a year, and if the database was down a week or two it was questioned if anyone would even notice. It was easy to see that the hundred thousand dollar solution was extremely over engineered for this client.

B. The Basics

Don't ignore the basics.

Disaster Recovery Solutions should start with two basic questions, what is the recovery point object and what is the recovery time objective.

- RPO – Recovery Point Objectives – To what point must the database be restored after a disaster? Another way to ask this question would be, how much data can be lost?
- RTO – Recovery Time Objectives – How much time can elapse after the disaster has occurred? Or, how long can your system can be down? Depending on these answers additional questions will arise, however these two questions can help determine what potential solutions will work. SQL Server offers a number of solutions from Transaction Log shipping to AlwaysOn Availability Groups.

C. Pay Attention to the Details

Whenever I visit a datacenter for a client I make sure that I take some time to review how the cages are wired. On more than one occasion I have seen servers with redundant power supplies that have both of the power cords plugged into one circuit.

C. Pay Attention to the Details (continued)

This configuration will protect you if one of the power supplies goes bad, however if the circuit goes down the redundant power supply isn't any help.

When executing a disaster recovery plan, ensure all the small details are double checked. If there is a single point of failure in the system, Murphy is going to find it.

D. Test

I can tell you the most common mistake I see, on a regular basis, with Disaster Recovery solutions is the lack of testing. Some testing is better than no testing, but the best testing is testing that mimics actual disasters. If there is a power outage for your servers and you have 5 min. to get everything moved, do you know the steps to complete, before the unlimited power supply loses its charge? What steps must you take if you don't have the 5 minutes? I was working with the chief technology officer for a major education facility and he had another vendor telling him he was safe. Saying he didn't have to worry about it. The contract was for a 15-minute recovery point. We reached out to the vendor and asked them to prove it.

The lesson here, perform regular realistic tests, if they don't work, find out why, and make the needed changes.



Optimize TempDB Database



A. About TempDB

By default when you install SQL Server the TempDB database is not optimized. For SQL Servers that use the TempDB even moderately, I recommend optimizing it for the best performance. The TempDB is the storage area that SQL Server uses to store data for a short periods of time, information that can be found in the TempDB include temporary tables, and data that is being sorted for query results. The data in the TempDB is so temporary that each time the SQL Server service restarts, the TempDB is recreated.

Due to the specific nature of the data in TempDB and the frequency of use it is important to ensure you have the most optimal configuration.

B. Optimization

Optimizing the TempDB is so important that Microsoft is changing how it is installed with SQL Server 2016 to encourage the optimization of Temp DB. How do you optimize TempDB? It's not difficult at all if you follow these few pointers:

Place your Temp DB on that fastest storage you have. Ultimately this storage should be pounding out a latency less than 5 milliseconds. Does your server have access to SSD storage? If so that is a great place for the TempDB.

There are a great number of studies that have been done to determine the ideal number of files you should split your Temp DB over. With my experience I tend to create one temp DB file for each processor core on the server, however I don't do this until I find there is some contention in the TempDB.

Grow your TempDB to the size you need it. Your TempDB is going to be recreated each time your service is restarted, so if your default database size is smaller than the normal operational size you are going to have to go through some grow events. Speaking of growth events, it is better to have controlled growth rather than a number of little growth events, so we recommend reviewing the auto growth size.

9

Manually Shrink Your Database



A. Differences in Database Administrators

Often you will see IT professionals approaching their tasks from different angles. Consider if you were a Systems Admin and you knew you needed some additional storage on a server you might send a request to the storage admin requesting an additional 50 gigs, or whatever amount you need. As a Database Professional, you would be wise to not only include the size of storage that you need but also the performance specifications that you require. As a DBA, we need to understand that SQL Server management may not always translate well to other types of systems management. Now granted this should be no surprise, it is understood we do not approach all things the same way, but where this comes into play is the understanding we all have different backgrounds. We became DBA's from different career paths.

B. Problems with Automatically Shrinking

If you are new to being a Database Administrator or the Primary focus of your job is not to be a DBA you may see the benefits of shrinking a database automatically. If the database shrinks by itself, it might be considered self-management; however, there is a problem when doing this.

When you shrink a data file SQL Server goes in and recovers all the unused pages, during the process it is giving that space back to the OS so the space can be used somewhere else. The downstream effect of this is going to be the fact your indexes are going to become fragmented. This can be demonstrated in a simple test.

C. Status of Indexes

I have a database in my lab based on the Chicago Crime Stats. I have been doing a lot of testing in the database with an automated indexing script, that has me inserting a deleting a large number of rows at different times. Over time this database has become rather large for my small lab, it is time to shrink it down to a more manageable size. The first thing done is to check what the status of my indexes is.

This is a simple query that will return all the indexes in the database with its fragmentation level.

C. Status of Indexes (continued)

```
SELECT db_name() as [database],  
Object_Name(ps.object_id) as [table],  
i.name as Index_Name,  
round(avg_fragmentation_in_percent, 0) as Frag  
FROM sys.dm_db_index_physical_stats(db_id(), null, null, NULL, NULL) ps  
Join sys.indexes i on ps.Object_ID = i.Object_ID and ps.index_id = i.index_id
```

The results look like this:

database	table	Index_Name	Frag
1 ChicagoCrimeStats	Crimes_Frag	IDX_Crimes_Frag_XCOORD_Clustered	0

More or less the indexes are looking good; there is not a lot of fragmentation except in the one table (that is a discussion for later topics). What happens if I shrink the whole database, to include not only the log but also the data file as well?

Use the following T-SQL:

```
DBCC ShrinkDatabase ([ChicagoCrimeStats])
```

Rerunning the index fragmentation script, I now receive these results:

database	table	Index_Name	Frag
1 ChicagoCrimeStats	Crimes_Frag	IDX_Crimes_Frag_XCOORD_Clustered	100

If I have queries that use the `IDX_Crimes_Frag_XCOORD_Clustered` index, there is a real good chance the performance on that query is going to degrade.

D. Shrinking a File

There are times when you may need to shrink a file, some considerations could be after a large delete of records or maybe you archived much of the data out of the database. These sort of operations remove data leaving your databases with a lot of free space. This free space can be reclaimed by using the `DBCC Shrinkfile` or `DBCC Shrinkdatabase` T-SQL commands, however be aware you should re-index after those statements are run.

It is not a bad thing to shrink a database as long as you do it in a controlled manor with proper maintenance afterwards.

10

Storage



A. Changes in Storage

Storage is by far the number one cause of performance issues I have experienced, and over the last 20 years of my career, the changes with storage have been huge. When I first started as a professional, we were using internal spinning disks and then quickly changed up to internal RAID arrays. Five or so years later, we moved to external RAID cages. Eventually the road took us to SAN and NAS storage and most recently SSD. With all these changes, it is easy to see why we have administrators who focus on nothing but storage. If you are fortunate enough to have a Storage Administrator, do you as a DBA still need to understand what the storage is doing?

B. Storage Issue Indicators

How can you identify if you are having some sort of performance bottleneck? There are a number of indicators that can provide you with the evidence your database is having a storage issue. A common indicator used for a number of years is the storage latency. Storage latency information is collected by using the Performance Monitor in Windows. Add the counters Average Disk/sec Read and Average Disk/sec Write. The storage latency can be monitored in real time or the data can be recorded by starting a data collection.

According to Microsoft Best Practices, the latency on the disk the log file resides should be less than five milliseconds, and the data file latency should be less than 20 milliseconds. In my experience, I have seen log file latency climb as high as 10 millisecond and sometimes a little higher during spikes without any end user impact. In addition take location note of the TempDB database as we talked about in Tip 8. You will want to ensure you are keeping the TempDB on the fastest storage you can.

C. Additional Tools

There are additional tools you may want to consider using to dig deeper into the performance of your storage systems such as SQLIO and SQLIOSIM both from Microsoft. A couple popular third party tools include IOMeter and CrystalDiskMark.

Thanks for reading the

Top 10 Tips for SQL Server Performance & Resiliency

The Top 10 Tips for SQL Server Performance & Resiliency e-Book

- 1. Check Your Backups*
- 2. Check Your Security*
- 3. Maintain Your SQL Server*
- 4. Have A Baseline*
- 5. Set A Maximum Memory*
- 6. Keep a Change History*
- 7. Develop a Disaster Recovery Plan*
- 8. Optimize TempDB Database*
- 9. Manually Shrink Your Database*
- 10. Storage*

**Need a SQL Server Assessment?
Call 1-888-685-3101 ext. 2
Or visit vitural-dba.com for more info**

