

By Robert Bogue

## Takeaway

For many enterprises, the performance and efficiency of database transactions is vital to the overall operation. This article offers the information technology professionals and administrators responsible for database performance several speed-enhancing tips for Microsoft's SQL Server.

---

## Table of Contents

<b>BEFORE QUERIES .....</b>	<b>2</b>
MEMORY IS SQL SERVER'S BEST FRIEND .....	2
ENTERPRISE ONLY .....	2
WHY BUY THE ENTERPRISE VERSION? .....	3
OPTIMIZING WHAT YOU'VE GOT .....	3
MAJOR IMPROVEMENTS .....	3
TECHREPUBLIC ADDITIONAL RESOURCES AND VERSION HISTORY .....	4

## Before queries

When looking at SQL performance there's an immediate desire to jump into the queries that the SQL Server is being forced to process and see what can be done to improve them. Unfortunately, it's not always that simple. Sometimes the queries are coming from packaged software that cannot be changed. Other times the queries come from report writing tools and there's just no way to make the queries any better.

It's those times when it's critical to take a look at the server you have and figure out if it's really running as fast as it can.

## ***Memory is SQL Server's best friend***

Microsoft SQL Server is quite adept at using memory when it's given. Even an extra gigabyte of memory can be a great help when confronting a poor performing database. Although SQL Server is generally good about taking the memory it needs, there are cases, such as when you have 4GB of RAM in your system, when SQL Server may not be able to get to the RAM it needs.

Without any changes to the basic configuration of a server with 4GB of RAM, SQL Server will have access to at most 2GB of that RAM. This is because Windows server divides the 4GB memory space into two equal sized chunks: one for the operating system and the other for the applications. The net effect of this is that applications can only use as much as 2GB of RAM.

The solution is to add the /3G switch to the end of the line in the boot.ini file from which the operating system boots. This shifts the memory division so that the applications have 3GB of space and the operating system has 1GB.

This is a great start. It allows SQL server to consume 3GB of space. There is, however, one more step. SQL Server will assume that it's running with only the 2GB of space that it would normally have. As a result it won't allocate more than about 1.7 GB of memory for caching, handles, and other buffers. The remainder of the memory is reserved for the core SQL server executable and the variables it needs internally. In order to get SQL server to utilize the additional RAM it will be necessary to specifically set the amount of memory that you want SQL server to use. This is done from the memory tab of the SQL server properties dialog. This dialog can be displayed by right clicking a server in Enterprise Manager and selecting properties.

## ***Enterprise only***

If you have more memory than 4GB then it's time to move up to SQL Server Enterprise. In addition to the other performance benefits that SQL Server Enterprise offers, which will discuss in a moment, SQL Server Enterprise edition is capable of using more than just the 3GB of memory that SQL Server Standard is capable of using.

This is done through Programmable Address Extensions (PAE) in the operating system and Address Windowing Extensions (AWE) in SQL Server. The first step is to turn on PAE in the Windows Boot.ini file. This is done by first removing the /3G switch and adding /PAE to the line from which Windows boots.

This does slow down memory accesses; however, it's a requirement to get the operating system to be able to see more than 4GB of RAM in the server. If you boot a server with more than 4GB of RAM and the operating system shows that you only have 4GB of RAM this is a sure sign that you're missing the PAE switch, or it wasn't recognized.

The next step is to enable AWE in SQL Server. This is best done from query analyzer. From an open connection run *sp\_configure 'awe enabled', 1*. You'll receive a message about RECONFIGURE—which isn't necessary since you'll have to stop and restart SQL Server to get it to take this configuration change anyway. Once you've restarted the service make sure that you check the error log to make sure that it indicates that AWE was enabled. If it doesn't there's something wrong—like perhaps you're running the standard version of SQL server and not the enterprise version which is required for AWE to work.

One side effect you should expect when you turn AWE on is that SQL server will gobble up all of the memory on the box to improve its performance. This shouldn't be surprising after all you want it to use every resource it can to improve performance—however, I've set off more than a few monitoring alarms when the memory of a recently reconfigured server jumped up to 93 percent. This generally signals a problem to most server operators who expect tons of free memory, so it sometimes takes a bit of explaining as to why this is the right thing to be seeing on a SQL Server box.

## ***Why buy the Enterprise version?***

If you're like most folks the difference between the standard version of SQL server and the enterprise version is a bit of a mystery. Sure there are some features which are enterprise-only features, however, they don't seem to make much difference—or do they?

In some unofficial benchmarks I'll have to say that they do add-up. On a very generously sized server with 8GB of RAM and a fast SAN the performance difference was somewhere in the 4x to 5x range of SQL Server standard installed on the same hardware. Obviously the ability to use more memory made a big difference but the ability to better manage the underlying disk resources helped as well.

The bottom line is that when you hit a performance wall with SQL Server standard, the Enterprise edition may completely change the performance playing field.

## ***Optimizing what you've got***

It's great to think pie in the sky and wait for the new Enterprise edition to land in your lap and for the server to magically have more memory, but what if that isn't an option? Here are some tips that you can use to make your databases perform faster.

- **Use RAID 1+0 instead of RAID5**—RAID 5 is more efficient on disk utilization than RAID1+0 but it comes at a performance hit when reading—the thing done most often in a database. Consider switching to RAID 1+0 if you can afford the space.
- **Defragment your database**—Databases can become fragmented just like disks. Using DBCC INDEXDEFRAG you can reduce the fragmentation in your database. However, this won't be effective for all tables. If your system tables have been fragmented by a large number of table additions and deletions you'll need to use DTS to transfer the database from one file to another. You can't use BACKUP DATABASE and RESTORE database because they will replicate the fragmentation.
- **Defragment your disks**—This tip seems obvious, but is rarely done. Defragment your drives after defragmenting your databases—with SQL server stopped so the defragmenter can take control of the database files.
- **Exclude your database files from Anti-Virus scanning**—It's amazing how many people have their anti-virus programs scanning their database files. This has little or no value and can substantially slow down database performance.
- **Stop all unnecessary services**—If you're not using a service on your SQL Server box, stop it. The processing and memory that it's taking up can be put to better use supporting SQL Server.

## ***Major improvements***

Taking care of the SQL server box won't solve every performance problem; however, the amount of performance that can be gained on most SQL servers with a few straightforward tasks shouldn't be taken lightly. In many cases these gains can exceed 100 percent improvement.

## ***TechRepublic Additional Resources and Version History***

### ***TechRepublic Additional Resources***

#### **SQL Server**

*TechRepublic's free SQL Server newsletter, delivered each Tuesday, contains hands-on tips that will help you become more adept with this powerful relational database management system. [Automatically subscribe today!](#)*

- [Thinking SQL: Set-based logic can improve query performance](#) (Article)
- [Tech Tip: Secure SQL Server transactions](#) (Article)
- [The Advantages of 64-bit to SQL Server 2000](#) (White paper)
- [Overview of Native XML Web Services for Microsoft SQL Server 2005](#) (White paper)
- [Five SQL Server scripts for novice DBAs](#) (Download)
- [Ten great Windows Server hacks](#) (Download)

### ***Version History***

1.0 March 2, 2005