

## Best Practice use RAID 10

2 x Raid 1 disk (Mirroring) For operating system

2 x Raid 1 disk (Mirroring) Place pagefile here and also can be used for applications and ad hoc Stuff

4 x RAID 1 +0 Striping and mirroring for database files (2 partitions 1 for Data File and 1 for Log File)

SQL Server with say 8GB RAM, and Configure AWE for to use 75% of that, 6GB.

## AWE

The following example shows how to activate AWE and configure a limit E.G. of 1 GB for **min server memory** and 6 GB for **max server**

<http://technet.microsoft.com/en-us/library/ms190673.aspx>

## Using AWE-mapped Memory with Windows Server 2003

SQL Server 2005 supports dynamic allocation of AWE memory on Windows Server 2003. During startup, SQL Server reserves only a small portion of AWE-mapped memory. As additional AWE-mapped memory is required, the operating system dynamically allocates it to SQL Server.

Similarly, if fewer resources are required, SQL Server can return AWE-mapped memory to the operating system for use by other processes or applications. For more information about the **awe enabled** configuration option, see [awe enabled Option](#).

See above link for more info.....

Also included below is some info on SANs, and how to maximize DB performance.

## SANs

Depending on how SANs is setup, there are great for storing and transferring large volumes of Data,

whereas we have found they can cause Gaps in a Write Intensive DB like Orion.

See this Article below on SANs

<http://thwack.com/blogs/geekspeak/archive/2008/05/08/the-fast-and-the-furious-orion-sql-and-sans.aspx>

Also see this on SQL Server Optimization

<http://thwack.com/blogs/geekspeak/archive/2008/03/24/optimizing-orion.aspx>

**Stick with RAID 1+0 or some other RAID that offers higher disk I/O performance.**

## SANs

I believe that the product documentation and the official stance of our tech support team is that we don't recommend running Orion w/NTA with a SAN, and for good reason based upon our overall experience in this area. You see, SANs are great for moving and storing very large amounts of data. In many cases you can actually read and write data more quickly to a high-performance SAN than to locally attached disk. The problem is that with applications like Orion you're not moving large chunks of data; instead, you're moving ginormous amounts of itty bitty pieces of data and most SANs just don't have the ability to handle this number of I/O transactions in the timeframes that applications like this demand. Time and time again we've seen issues where data is getting dropped when trying to write to a high-performance SAN but after moving the data to even a moderately performing local disk array the problem goes away.

For example, I worked with a customer recently that was seeing holes within some of the data sets the he was collecting and was leveraging a SAN to house his SQL database. Additionally, when trying to query the database for these results the queries would sometime time out. We turned on some perfmon counters on the SQL server and we were seeing disk queue lengths (read and write) of 200-300. Microsoft recommends that for SQL Servers with high amounts of I/O the disk queue lengths not exceed twice the number of physical disks (which in this case was 13 if I remember correctly). After moving the database to a local disk array (RAID 1+0), the problems went way...

Raid 5 can become a bottleneck, especially once Netflow, best option is a RAID 10  
**Stick with RAID 1 or 10 or some other RAID that offers higher disk I/O performance.**

Move away from RAID 5 for your SQL disk array. Stick with RAID 1+0 or some other RAID that offers higher disk I/O performance.

<http://thwack.com/blogs/geekspeak/archive/2008/03/24/optimizing-orion.aspx>

<http://thwack.com/blogs/geekspeak/archive/2008/07/21/tips-and-tricks-for-improving-sql-performance.aspx>

Raid 5 vs Raid 1 <http://thwack.com/forums/p/11363/46604.aspx>

We do not recommend:

**1 SANs depending on setup.**

**2 Slow Disks 10K**

**3 RAID 5/6 (Especially with Netflow)**

**4 SQL Servers on Virtual Machines**

**5 SQL Server Installed on Same Server as Orion with NETFLOW.**

## Tips and Tricks for Improving SQL Performance

SQL Server performance is a hot topic these days, especially if you're leveraging your SQL Server for a high performance NMS.

This can become even more critical when you add applications like NetFlow which tend to carry a significant I/O burden.

In some organizations you can rely on the DBA team to own/maintain/optimize the database servers for you.

Unfortunately, for many of us this isn't an option either because we don't have a DBA team or because it's such a political mess trying to work with them.

This causes us to have to implement and maintain our own database servers to support our apps.

The thing is, most of us network engineers don't know diddly about database servers.

So, with that in mind, here are a few tips for optimizing your SQL Server:

### **Head Geek's Top 5 Tips for Improving SQL Performance**

#5 - Add more RAM. Doesn't really matter how much you have, adding more will almost always help.

Be sure that your SQL instance and OS are capable of consuming the additional RAM and if not make it so.

#4 - Just say "no" to RAID 5. It's great for application servers but horrible for database servers where I/O performance is important.

#3 - Place the data and log files (.mdf and .ldf) on separate logical drives and separate channels or controllers.

#2 - Unless your SAN is optimized for high I/O vs. large I/O stick with a locally attached disk array.

#1 - Buy disk controllers with battery backed-up write-back cache. The more the better, but at least 256MB.