

# Compilation of SQL Server TempDB IO Best Practices



Cindy Gross 20 Nov 2009 9:07 AM

2

It is important to optimize TempDB for good performance. In particular, I am focusing on how to allocate files.

TempDB is a unique database in several ways. The ones most relevant to this discussion are:

- It is often one of the busiest databases on an instance. This means the performance of TempDB is critical to your instance's overall performance.
- It is recreated as a copy of model each time SQL Server starts, taking all the properties of model except for the location, number, and size of its data and log files.
- TempDB has a very high rate of create/drop object activity. This means the system metadata related to object creation/deletion is heavily used.
- Slightly different logging and latching behavior.

General recommendations:

- Pre-size TempDB appropriately. Leave autogrow on with instant file initialization enabled, but try to configure the database so that it never hits an autogrow event. Make sure the autogrow growth increment is appropriate.
- Follow general IO recommendations for fast IO.
- If your TempDB experiences metadata contention (waitresource = 2:1:1 or 2:1:3), you should split out your data onto multiple files. Generally you will want somewhere between 1/4 and 1 file per physical core. If you don't want to wait to see if any metadata contention occurs you may want to start out with around 1/4 to 1/2 the number of data files as CPUs up to about 8 files. If you think you might need more than 8 files we should do some testing first to see what the impact is. For example, if you have 8 physical CPUs you may want to start with 2-4 data files and monitor for metadata contention.
- All TempDB data files should be of equal size.
- As with any database, your TempDB performance may improve if you spread it out over multiple drives. This only helps if each drive or mount point is truly a separate IO path. Whether each TempDB will have a measurable improvement from using multiple drives depends on the specific system.
- In general you only need one log file. If you need to have multiple log files because you don't have enough disk space on one drive that is fine, but there is no direct benefit from having the log on multiple files or drives.
- On SQL Server 2000 and more rarely on SQL Server 2005 or later you may want to enable trace flag - T1118.
- Avoid shrinking TempDB (or any database) files unless you are very certain you will never need the space again.

References:

- Working with tempdb in SQL Server 2005 <http://technet.microsoft.com/en-us/library/cc966545.aspx>
  - "Divide tempdb into multiple data files of equal size. These multiple files don't necessarily be on different disks/spindles unless you are also encountering I/O bottlenecks as well. The general recommendation is to have one file per CPU because only one thread is active per CPU at one time."
  - "Having too many files increases the cost of file switching, requires more IAM pages, and increases the manageability overhead."
- How many files should a database have? - Part 1: OLAP workloads <http://sqlcat.com/technicalnotes/archive/2008/03/07/How-many-files-should-a-database-have-part-1-olap-workloads.aspx>
  - If you have too many files you can end up with smaller IO block sizes and decreased performance under extremely heavy load.
  - If you have too few files you can end up with decreased performance to GAM/SGAM contention (generally the problem you see in TempDB) or PFS contention (extremely heavy inserts).
  - The more files you have per database the longer it takes to do database recovery (bringing a database online, such as during SQL Server startup). This can become a problem with hundreds of files.
- SQL Server Urban Legends Discussed <http://blogs.msdn.com/psssql/archive/2007/02/21/sql-server-urban-legends-discussed.aspx>
  - "SQL Server uses asynchronous I/O allowing any worker to issue an I/O requests regardless of the number and size of the database files or what scheduler is involved."
  - "Tempdb is the database with the highest level of create and drop actions and under high stress the allocation pages, syscolumns and sysobjects can become bottlenecks. SQL Server 2005 reduces contention with the 'cached temp table' feature and allocation contention skip ahead actions."
- Concurrency enhancements for the tempdb database <http://support.microsoft.com/kb/328551>
  - Note that this was originally written for SQL Server 2000 (the applies to section only lists 2000) and there are some tweaks/considerations for later versions that are not covered completely in this article. For example, -T1118 is not only much less necessary on SQL Server 2005+, it can in some cases cause problems.
- FIX: Blocking and performance problems may occur when you enable trace flag 1118 in SQL Server 2005 if the temporary table creation workload is high <http://support.microsoft.com/default.aspx?scid=kb;EN-US;936185>
  - If you have SP2 based CU2 or later you will not see the problems described in this article. Also, on SP2 based CU2 or higher you are much less likely to even need -T1118 on a heavily used TempDB.
  - "This hotfix significantly reduces the need to force uniform allocations by using trace flag 1118. If you apply the fix and are still encountering TEMPDB contention, consider also turning on trace flag 1118."
- Misconceptions around TF 1118 <http://sqlskills.com/BLOGS/PAUL/post/Misconceptions-around-TF-1118.aspx>
  - "turn on TF1118, which makes the first 8 data pages in the temp table come from a dedicated extent"
  - "Instead of a 1-1 mapping between processor cores and tempdb data files (\*IF\* there's latch contention), now you don't need so many - so the recommendation from the SQL team is the number of data files should be 1/4 to 1/2 the number of processor cores (again, only \*IF\* you have latch contention). The SQL CAT team has also found that in 2005 and 2008, there's usually no gain from having more than 8 tempdb data files, even for systems with larger numbers of processor cores. Warning: generalization - your mileage may vary - don't post a comment saying this is wrong because your system benefits from 12 data files. It's a generalization, to which there are always exceptions."
- Storage Top 10 Best Practices <http://sqlcat.com/top10lists/archive/2007/11/21/storage-top-10-best-practices.aspx>

- "Make sure to move TEMPDB to adequate storage and pre-size after installing SQL Server. "
- "Performance may benefit if TEMPDB is placed on RAID 1+0 (dependent on TEMPDB usage). "
- "This is especially true for TEMPDB where the recommendation is 1 data file per CPU. "
- "Dual core counts as 2 CPUs; logical procs (hyperthreading) do not. "
- "Data files should be of equal size – SQL Server uses a proportional fill algorithm that favors allocations in files with more free space.
- "Pre-size data and log files. "
- "Do not rely on AUTOGROW, instead manage the growth of these files manually. You may leave AUTOGROW ON for safety reasons, but you should proactively manage the growth of the data files. "

Optimizing tempdb Performance <http://msdn.microsoft.com/en-us/library/ms175527.aspx>

## Comments

---



[Rob Firth](#) 24 Nov 2009 6:18 AM

I've seen a webcast where it was shown that 1 datafile for tempdb is much more performant for query sorting operations than many tempdb files because the data is written to sequentially. Many tempdb files are better for creation and dropping objects.



[sstange](#) 24 Nov 2009 6:47 AM

I thought an background article on the purpose of those system databases would be helpful for others looking at this entry.

<http://technet.microsoft.com/en-us/library/ms178028.aspx>

Thanks for the article.