How to Provision & Monitor TempDB

8 p1

Agenda

How big does it need to be?

Where should we put it?

How many data and log files do we need?

Should we enable other settings like autogrowth or trace flags?

On an existing server, what do we change?

What should we monitor?

8 p2

How big?

8 p3

It stores all kinds of stuff:

And it's used for all kinds of stuff:

- Version store
- Temp tables, table variables
- Joins, aggregations
- Cursors
- Sorting, hash matches, spools

And it's nearly impossible to know which ones we'll use, and how much.

8 p4

Example: the version store

How long will your longest transaction be?

How much data will change
during that transaction?

Not just in your one session, but:

- In other transactions
- In other databases, too

And will you have any outliers, like
the lock-machine-and-leave scenario?

8 p5

It's impossible to guess this stuff.

If you're building a new server, you just can't know.

My guideline: 25% of the total data size
on the entire server.

Yes, even on 10TB data warehouses:
after all, we do big data loads & reloads.

8 p6

There are exceptions.

Client scenario, simplified:

- The server has 10TB of data
- It's 10 databases, each 1TB
- Each 1TB database has a single 800GB table (that's ten 800GB tables in total)
- They rebuild indexes in parallel, with sort_in_tempdb = on

They needed at least 8TB TempDB space.

8 p7

Another example

- 20TB database
- Crazy fast SAN, no point of sort_in_tempdb
- In a sync Availability Group (we couldn't rebuild indexes)
- No transactions, triggers, version store (all nolock)
- 100GB TempDB would be fine

8 p8

Where do we put it?

8 p9

Solid state storage.

In this day and age, nothing latency-sensitive should be on magnetic hard drives.

8 p10

If you're on a bare metal server...

If you're on dedicated hardware (not a VM),
I strongly recommend using local SSD for TempDB.

It's cheap: under $500/TB, much cheaper than SAN.

It's fast: sub-millisecond latency.

It keeps your storage network free from TempDB traffic, letting it focus on valuable traffic to real databases.

It doesn't need to persist: it's okay if the contents don't fail over when the server restarts.

8 p11

If you're in a VM in the cloud...

If you're in a cloud VM like AWS EC2 or Azure VMs, use ephemeral storage.

That's the cloud term for local SSD.

It does disappear when the VM dies, but that's just like local SSD on a bare metal machine.

It's WAY faster than shared storage in the cloud.

8 p12

If you're in a conventional VM...

If you're hosting your VMs in VMware or Hyper-V, you'll probably have to put TempDB on the SAN.

Your VM admins won't want to use local SSD.

If they use local SSD, they won't be able to fail a VM around to different servers.

This is just the price you pay for virtualization's higher availability.

8 p13

If you're in a VM, continued

Should TempDB be on its own dedicated volume?
(Remember: this point is irrelevant for bare metal or cloud VMs because in those cases, you'd be using a pair of mirrored local SSDs anyway.)

Can your sysadmins disable SAN replication on the TempDB volume? If so, use a separate volume.

Can they set different caching settings on the TempDB volume? If so, check with your storage vendor's best practices.

8 p14

If it's on a dedicated volume...

And if that volume has pretty limited space, I've used an emergency vent valve file.

Create a 1MB file on a much larger volume

Enable autogrowth in a big increment

Set up alerts for when that file grows (and it's a real emergency)

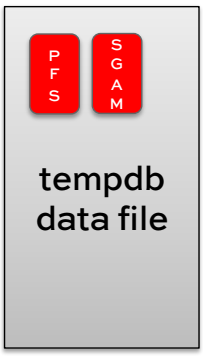
Proportional fill means we won't use it a lot

8 p15

How many data files do we need?

8 p16

Normal databases have 1 data file.



And this file will have one of each of those special PFS and SGAM pages.

(It gets more pages as its size grows.)

This is fine for regular databases, but...

8 p17

TempDB is different.

	New objects created	Objects dropped
System databases	Almost never	Almost never
User databases	Every now and then	Every now and then
TempDB	CONSTANTLY	CONSTANTLY

8 p18

This causes a problem.

SQL Server was never really designed for constant creation/dropping of objects.

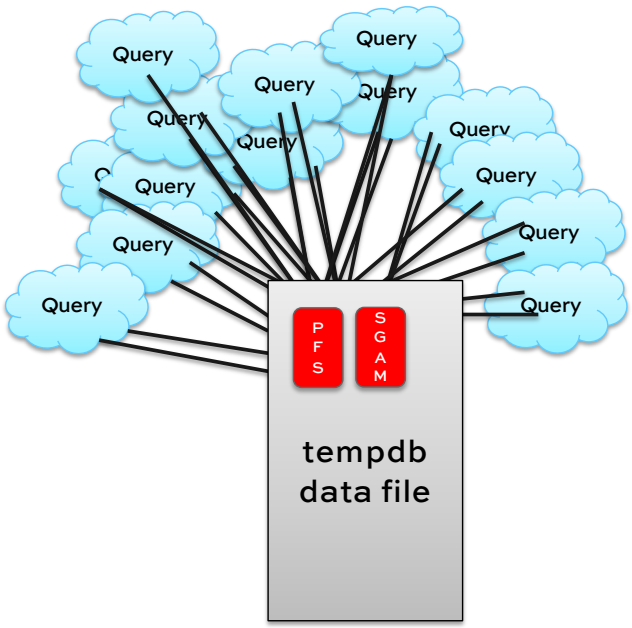
To create or drop an object, SQL Server has to write to special system pages.

- PFS: Page Free Space pages
- SGAM: Shared Global Allocation Map pages

Latches protect these pages in memory.

Think of this as a lightweight lock.

8 p19



Not in TempDB.

8 p20

This isn't a disk storage problem.

It's about contention for a system page.

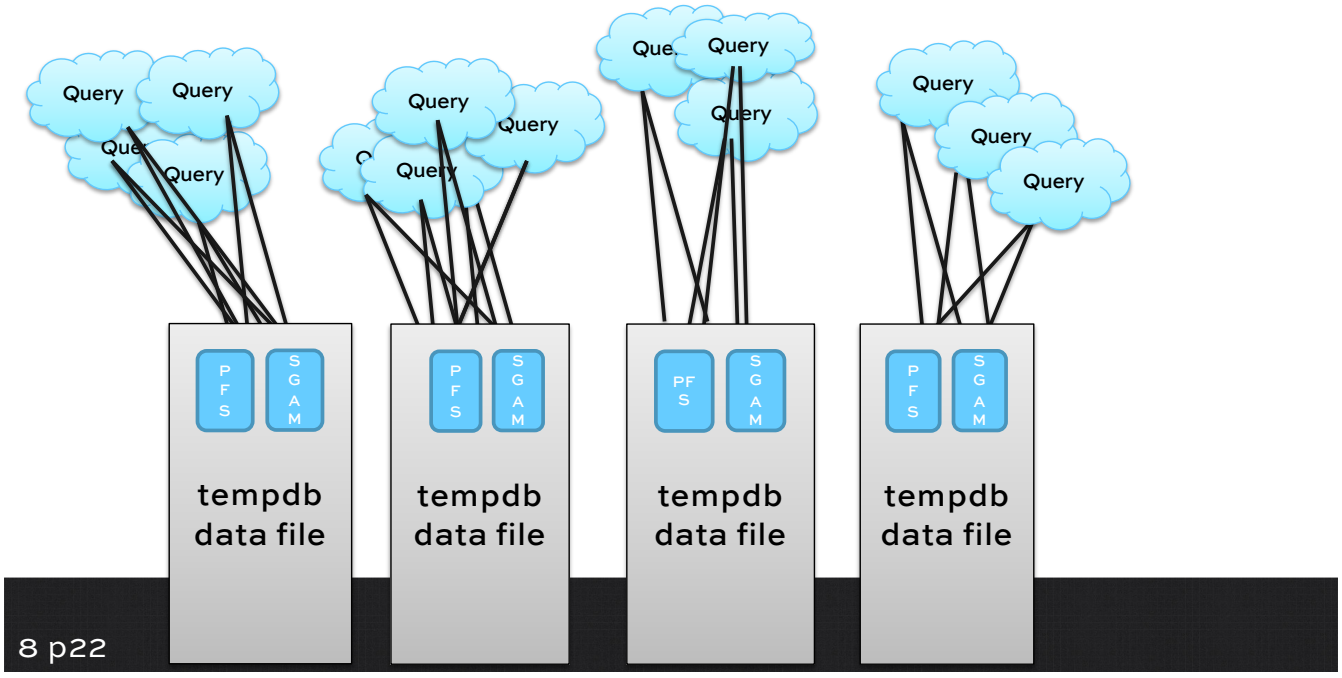
These pages are stored on disk.

But we're not waiting on disk to get 'em.

We have these contention problems
even when the data is stored in RAM.

The fix is a little odd...

8 p21



8 p22

How to configure TempDB

Create 4-8 equally sized data files.

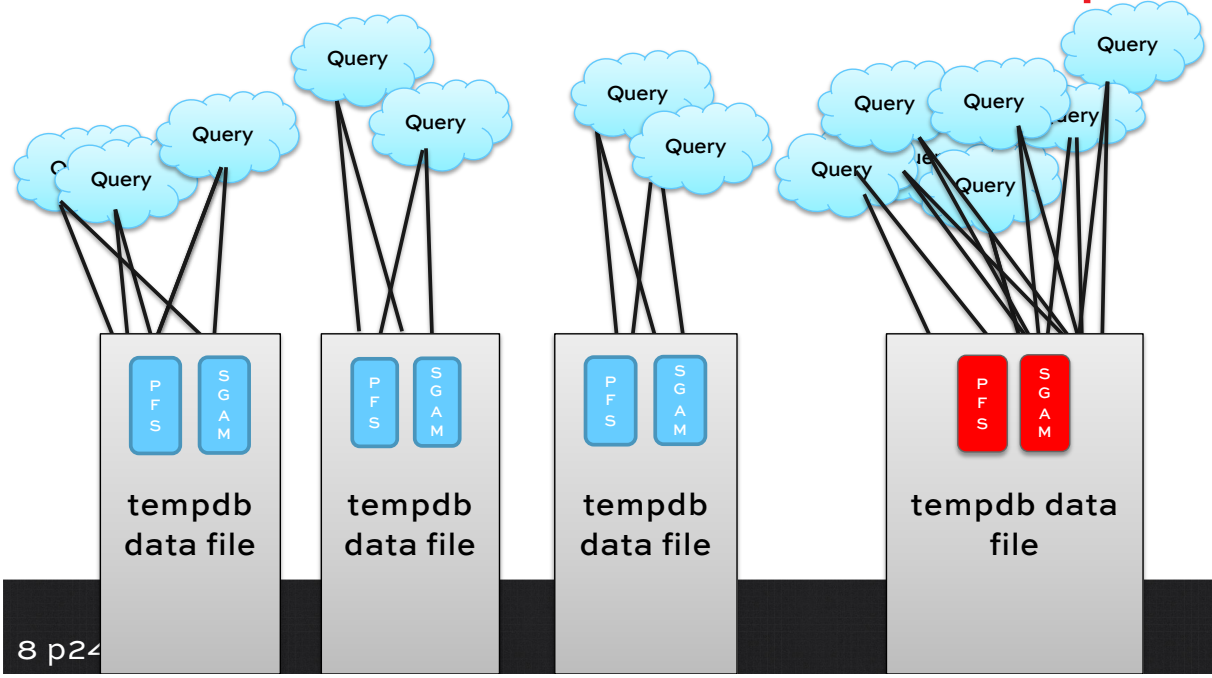
The exact number is less important than just “more.”

If you’re using local (ephemeral) storage:

- Take the total available space
- Divide that by 5
- Create 4 data files with that size
- Create 1 log file with that size

8 p23

Make sure data file sizes are equal.



8 p24

How many log files do we need?

8 p25

Just 1. Log file access is sequential.

SQL Server starts at the beginning of the log,

Logs transactions,

And when it gets to the end of the log file,

If space at the beginning is available for reuse, it'll loop back and reuse that,

Otherwise it grows the log file out.

8 p26

Should we enable autogrowth?

8 p27

If you're on bare metal or cloud...

Then you used a dedicated local SSD volume for TempDB anyway.

It's dedicated to TempDB.

Just grow the files out to fill the space.

You're done: you'll never have to worry about monitoring for growths. Growing is done.

You just have to worry about it filling up.

8 p28

If you're on a VM, sharing space...

If you decided to use a single volume for both user databases & log files, plus TempDB, it gets harder.

You'll probably want to:

- Configure the TempDB data files to total up to 25% of the size of the server's data, but
- Still leave autogrowth enabled in case someone does something terrible – try not to go down.

8 p29

What if we run out of space?

8 p30

You will. It's a matter of time.

Someone's gonna do something bad.

You just need to prepare by:

- Monitoring what percentage of TempDB has been used, and
- Send alerts when it's high enough that you need to investigate, and
- Leave yourself time to take action

8 p31

Should we enable trace flags?

8 p32

For SQL Server 2014 & prior, yes:

Trace flag 1117: grows all data files in a filegroup equally (but also applies to user databases).

Trace flag 1118: doesn't use mixed extents in TempDB, alleviates some of the pressure.

To set up trace flags to run at startup, use SQL Server Configuration Manager: https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/scm-services-configure-server-startup-options

But 2016 & subsequent versions don't need these.

8 p33

SQL Server 2016 & later

1117: replaced by AUTOGROW_ALL_FILES option of ALTER DATABASE. (Default = on for TempDB, but not user dbs.)

1118: replaced by MIXED_PAGE_ALLOCATION ON argument for ALTER DATABASE SET. (Default = on for all system databases, but off for user dbs.)

8 p34

Should we change an existing server?

8 p35

One extreme: tiny TempDB

Someone's been shrinking files

TempDB's size is much smaller than what I'd expect, like total DBs are 1TB, and TempDB 10GB (1%)

They were shrinking because we didn't have enough available drive space

Solution: get more drive space before you change anything

8 p36

Other extreme: giant single file

No one was managing TempDB

We have just one data file, and it's big

Solution:

- Note its data file size
- Shrink it back down to zero
- Grow it back out to ¼ of the prior size
- Add 3 more data files, all that same size

8 p37

Anything else: leave it.

Don't stress out over:

- The exact number of files
- The exact size of data & log files

As long as:

- Data files are all equally sized
- Monitoring doesn't indicate a TempDB bottleneck

8 p38

Monitoring tempdb

8 p39

What we need to monitor

Does it have enough capacity?
If not, what's using the space?

Are the pages in memory fast enough?
If not, do we need more files, and why?

Are the pages on disk fast enough?

I demoed it with scripts, but the tool you use to save the data may vary.

8 p40

Common monitoring tools

Quest Spotlight

Idera SQL DM

Red Gate SQL Monitor

Solarwinds DPA &
SentryOne SQL Sentry

8 p41

How to use them

Review the list of TempDB things to track

Review the tool's thresholds

Configure email alerting in a way
that gives you time to react,
but minimizes false alarms

Don't set up rules to folder-ize emails

8 p42

Recap

8 p43

TempDB consumers we covered

Version store, triggers

Temp tables, table variables, in-memory tables

Execution plan spills

Cursors, index builds, AG stats

8 p44

Where to store data temporarily

	Temp Tables	Table Variables	In-Memory Table Variables
Has statistics	Yes	No (until 2019)	No
Statistics can get reused across sessions	Yes	No (until 2019)	No
GAM/SGAM/PFS contention at scale	High	Medium	Low

It's a lot – but you can do this!

For questions, leave comments on the relevant module.

For private help after the class, email Help@BrentOzar.com with:

- A note that you were in this class
- sp_Blitz @CheckServerInfo = 1
- sp_BlitzFirst @SinceStartup= 1

Thanks, and I hope you had a great time!

8 p47