SQL Server Storage: The Terabyte Level

Brent Ozar, Microsoft Certified Master, MVP
Consultant & Trainer, SQLskills.com

BrentOzar.com/go/san

SAN Storage Best Practices for SQL Server

Using SQL Server on SAN storage doesn't always guarantee fast performance. Before the SQL Server installation starts, you need to get the SAN configuration right. Here's my posts on how to get things started right.

How to Configure Your SAN Storage for SQL Server

Database administrators usually see the SAN as a black box. We ask the SAN administrators for a few arrays, and they just ask us what size we need. Configuring storage for SQL Server is more complex than other applications, but thankfully we've got a lot of help. Your SAN vendor has already put a lot of work into documenting how to set up arrays on your SAN controller specifically for SQL Server. Here's the most common vendor document repositories:

About Brent

Race Facts

- 333 miles
- 375 boats invited
- 33 DNFs

Typical Terabyte Troubles

- *Restores*
- Backups
- DBCCs
- Querying

- Bad Ideas for:
  - Schema
  - Indexes (CL/NCL)
  - Partitioning
  - Storage

SQL CRUISE

What Do We Restore Most Often?

1. The whole server
2. One whole database
3. One whole table
4. A few records
5. A schema change (like an SP or view)

The 1TB DBA's Worst Enemy

Picture Source http://www.flickr.com/photos/slworking/2825225419/

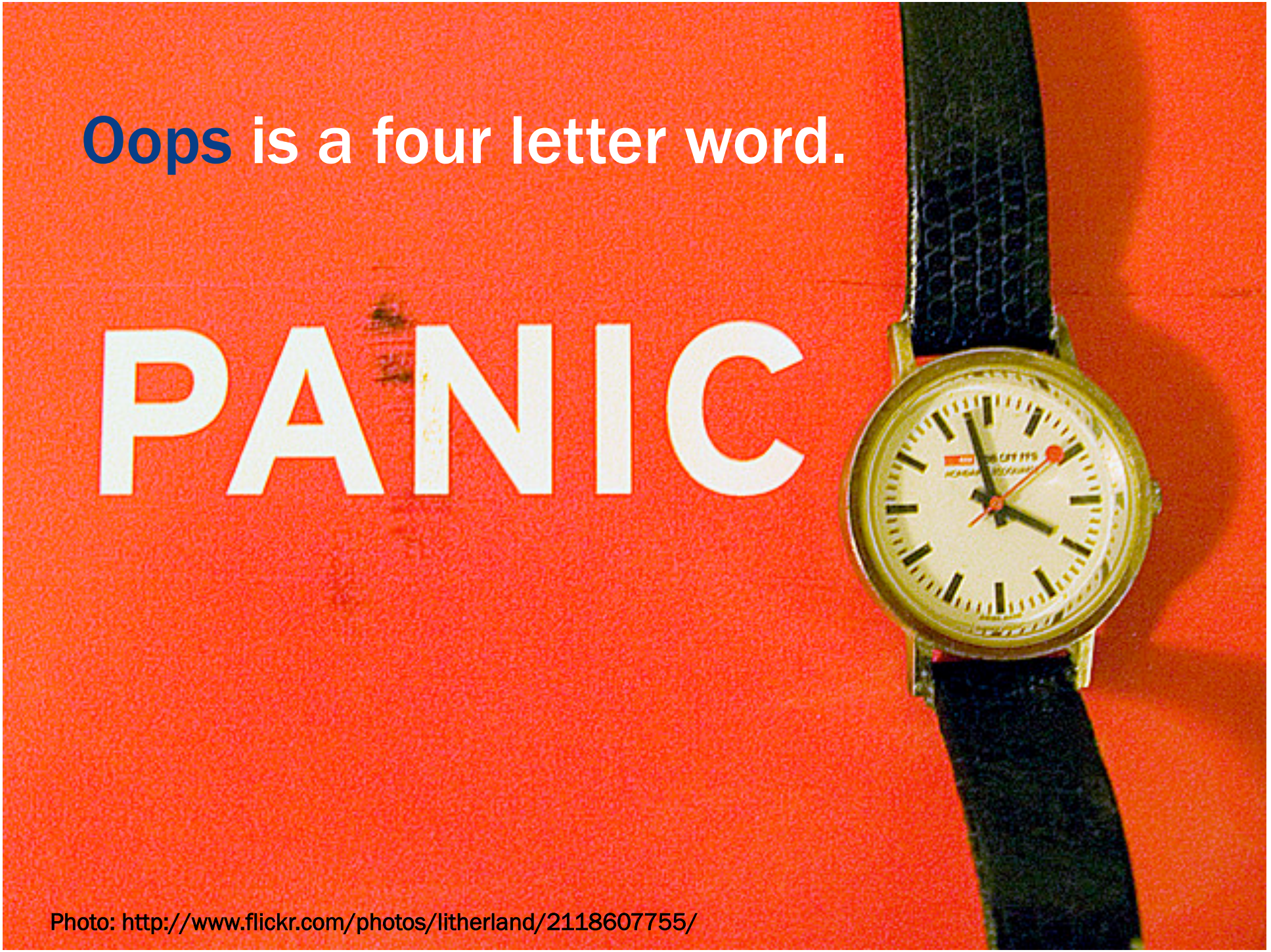


Photo: http://www.flickr.com/photos/litherland/2118607755/

The “Before You Touch That” Backup

- Snapshot: instant backup
- Near-zero time of database size
- Near-zero space required
- Very easy to restore the whole thing
- Somewhat easy to restore objects

Database Snapshots: CONTROL-Z

- SQL 2005 & newer
- Enterprise Edition
- Looks like a DB
- Great for ad-hoc safety plans

SQL CRUISE

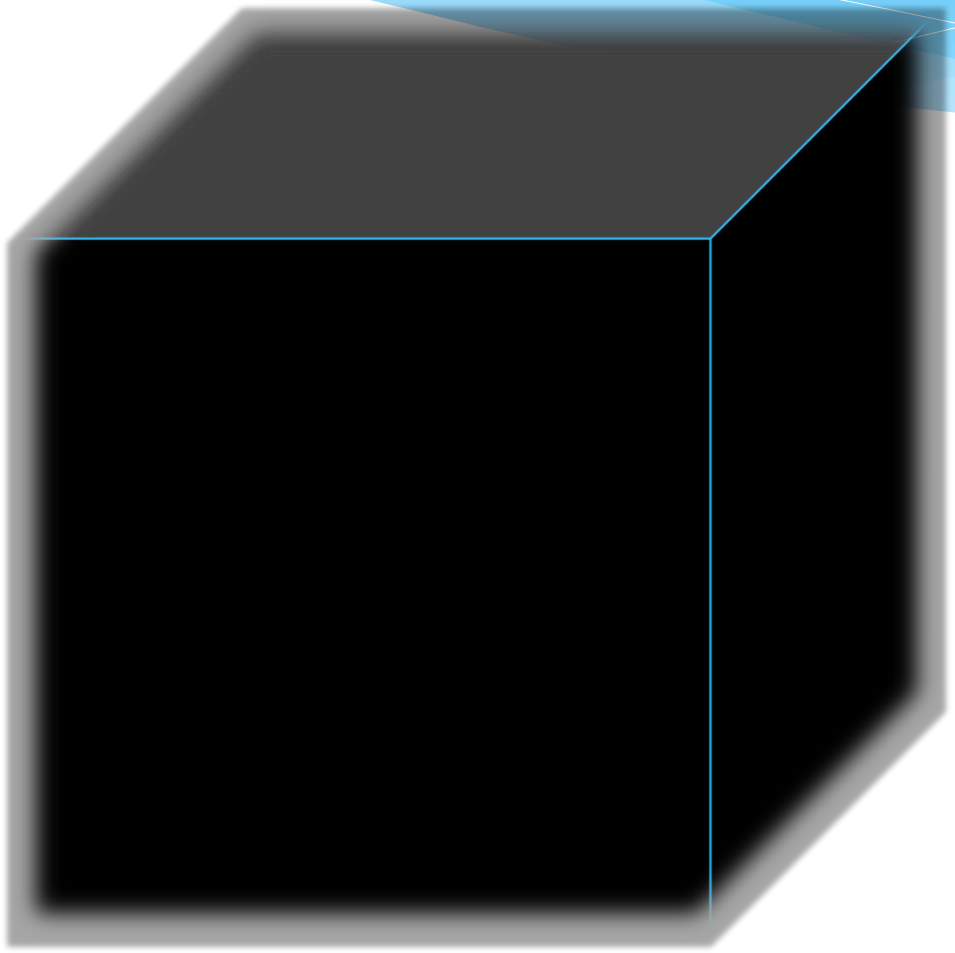
Photo: http://www.flickr.com/photos/croma/557971148/

Database Snapshot Pros & Cons

- Ad-hoc backups before app upgrades
- Let users query DB during ETL loads
- Presto, locking problems disappear
- Automated switching is messy
- Snapshots stop some operations
- Watch that disk space
- Still live on this database server

SQL CRUISE

SAN Snapshot Backups



SQL CRUISE

SAN Snapshots

- Not related to SQL Server at all
- Does require SQL Server awareness
- Involves doing a SQL Server backup
- Back up any size database in seconds
- Restore any whole database to any server in seconds*

SQL CRUISE

* Well, Not Really.

- May be groups of databases
- May involve attaching more drives
- May require the SAN admin's help
- May not involve the transaction log

SQL CRUISE

SAN Snapshot Benefits

- Backup speed nearly instantaneous
- Tape work offloaded to SAN controller
- Can keep many backups online
- Easy to run DBCCs on live backups
- Refresh dev/QA servers instantly

SQL CRUISE

SAN Snapshot Drawbacks

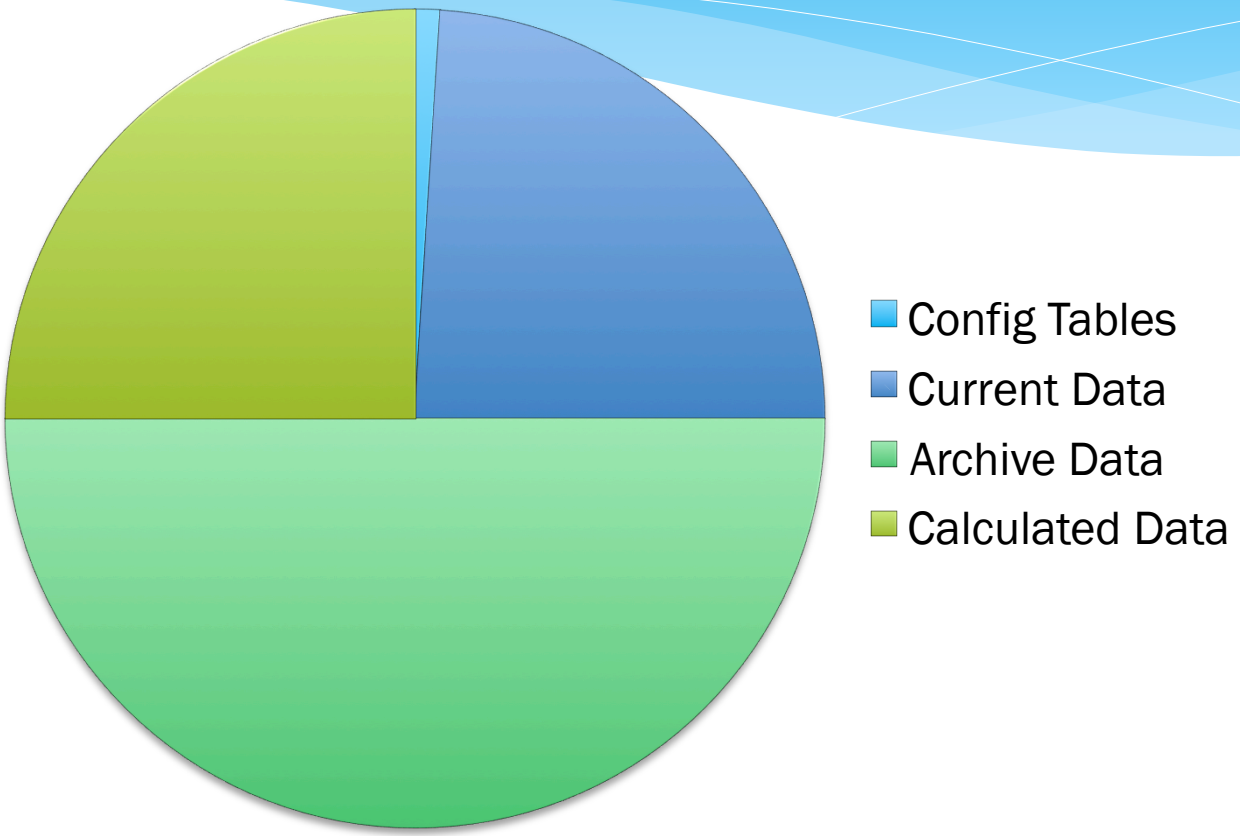
- Can be very expensive
- Presenting snapshots to dev/QA servers can slow down production
- Doesn't help log shipping

SQL CRUISE

- SQL Server Database Snapshots:
  - Before “oops” moments
- SAN Snapshots:
  - Refresh QA, dev
  - Run DBCC there
  - Test version upgrades, new versions

SQL CRUISE

Our Sample Database



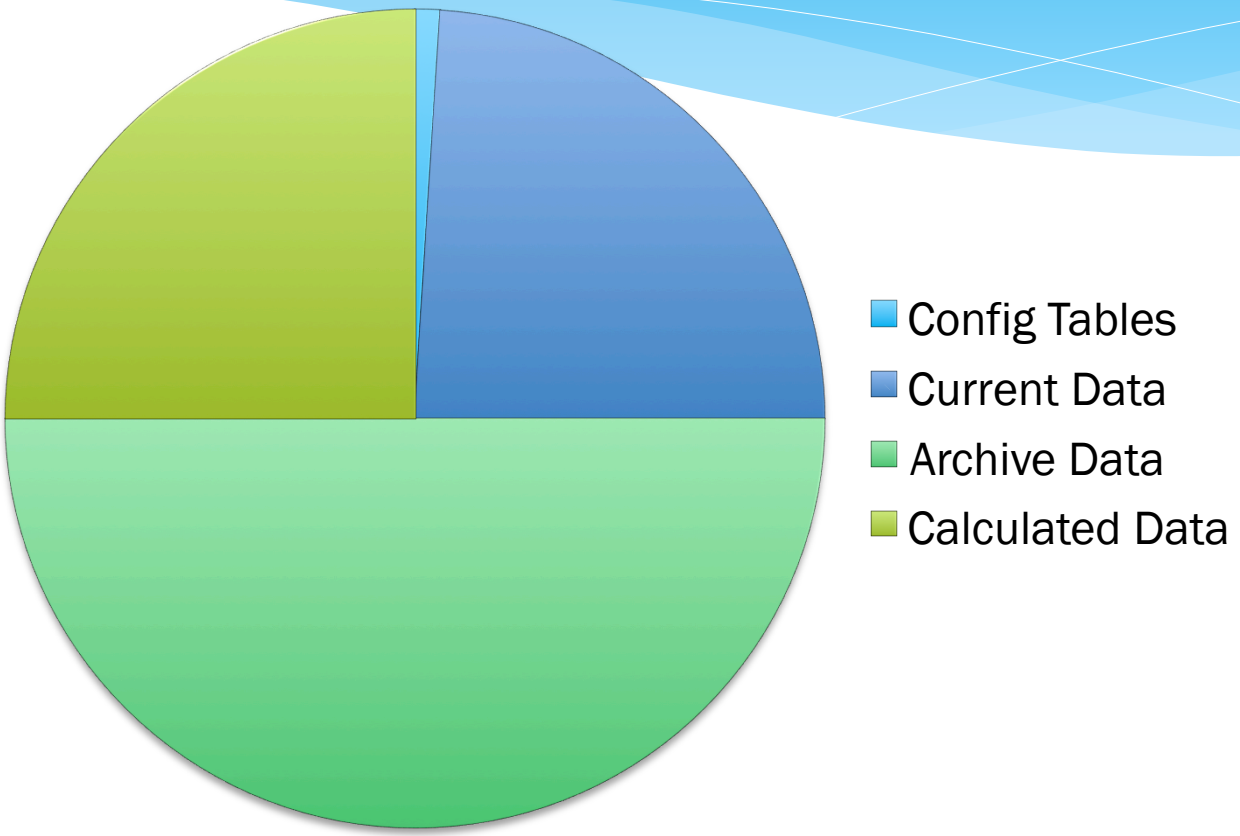
SQL CRUISE

Online Piecemeal Restore

- Restore PRIMARY filegroup first
- Database is instantly online
- Restore other filegroups individually
- Tables come online with each FG
- Demo: BrentOzar.com/go/piecemeal

SQL CRUISE

Our Sample Database



SQL CRUISE

Designing for Restores

- Keep PRIMARY filegroup very small
- Stop new PRIMARY objects (incl. indexes) using DDL triggers
- Design filegroups for recovery needs
- Keep related schema, objects together (including indexes)

SQL CRUISE

The Easy Button for Restores

- Instant File Initialization
- SQL Server 2005+, all editions
- Instantly grows/creates data files
- Doesn't work with log files
- All gain, no pain

SQL CRUISE

SQL skills.com

Kimberly L. Tripp

Improving my SQL skills through your questions!

Instant Initialization - What, Why and How?

Instant Initialization is a new feature of SQL Server 2005 that is based on an NTFS feature that was added to Windows XP (and therefore is also available in Windows 2003 Server). It's a feature that's seemingly simple; it allows file allocation requests to skip zero initialization on creation. As a result, file allocation requests can occur instantly - no matter what the file size. You might wonder why this is interesting or why this make a difference? Most file allocation requests are small requests, with small incremental changes (like .doc files, .xls files, etc.) but database files can be rather large. In fact, they should be rather large as pre-allocation of a reasonable file size is a best practice to reduce file fragmentation. Additionally, autogrowth causes performance delays (more so in 2000 than 2005) but it's generally something that you want to avoid when possible. As as result, database creation times can take minutes to hours to days, depending on file allocation request. But - it's not just for database creation. ALL file requests can leverage this feature: file creation for a new database, adding a file to an existing database, manually or automatically growing a file and (IMO - the best) restoring a database where the file (or files) being restored does not already exist. The reason I think the last feature is the best is that it can reduce downtime if a database is damaged and allow you to get back up and running more quickly. This is especially important for databases that cannot leverage partial database availability, which is an Enterprise Engine feature. So, to give you some motivation, here is a test that I performed just to have some interesting and comparable numbers.

Performance Test with Zero Initialization

Hardware: Dell Precision 670 Dual Proc (x64) with Dual Core, 4 GB Memory, RAID 1+0 array w/4-142 GB, 15000rpm disks

CREATE DATABASE with 20 GB Data file = **14:02 minutes**

ALTER DATABASE BY 10 GB = **7:01 minutes**

RESTORE 30 GB DATABASE (EMPTY Backup) = **21:07 minutes**

Typical Terabyte Troubles

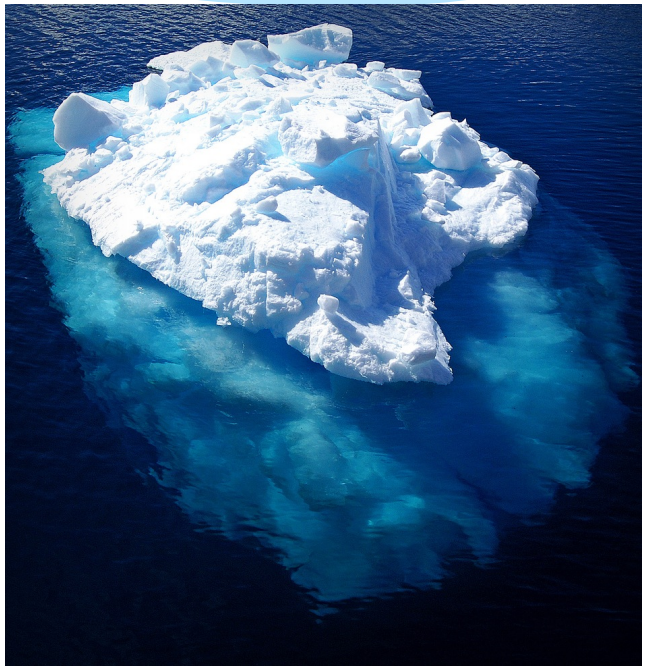
- Restores
- ***Backups***
- DBCCs
- Querying

- Bad Ideas for:
  - Schema
  - Indexes (CL/NCL)
  - Partitioning
  - Storage

SQL CRUISE

Read-Only Filegroups

- Separate frozen data
- Enable compression
- Set 100% fill factor
- Update statistics
- Set FG to read-only
- Redesign backups



SQL CRUISE

Picture Source http://www.flickr.com/photos/guilleavalos/2139208615/

Read-Only Backup Strategy

- Full Backup
- Transaction logs every X minutes
- Filegroup differentials for active FGs

SQL CRUISE

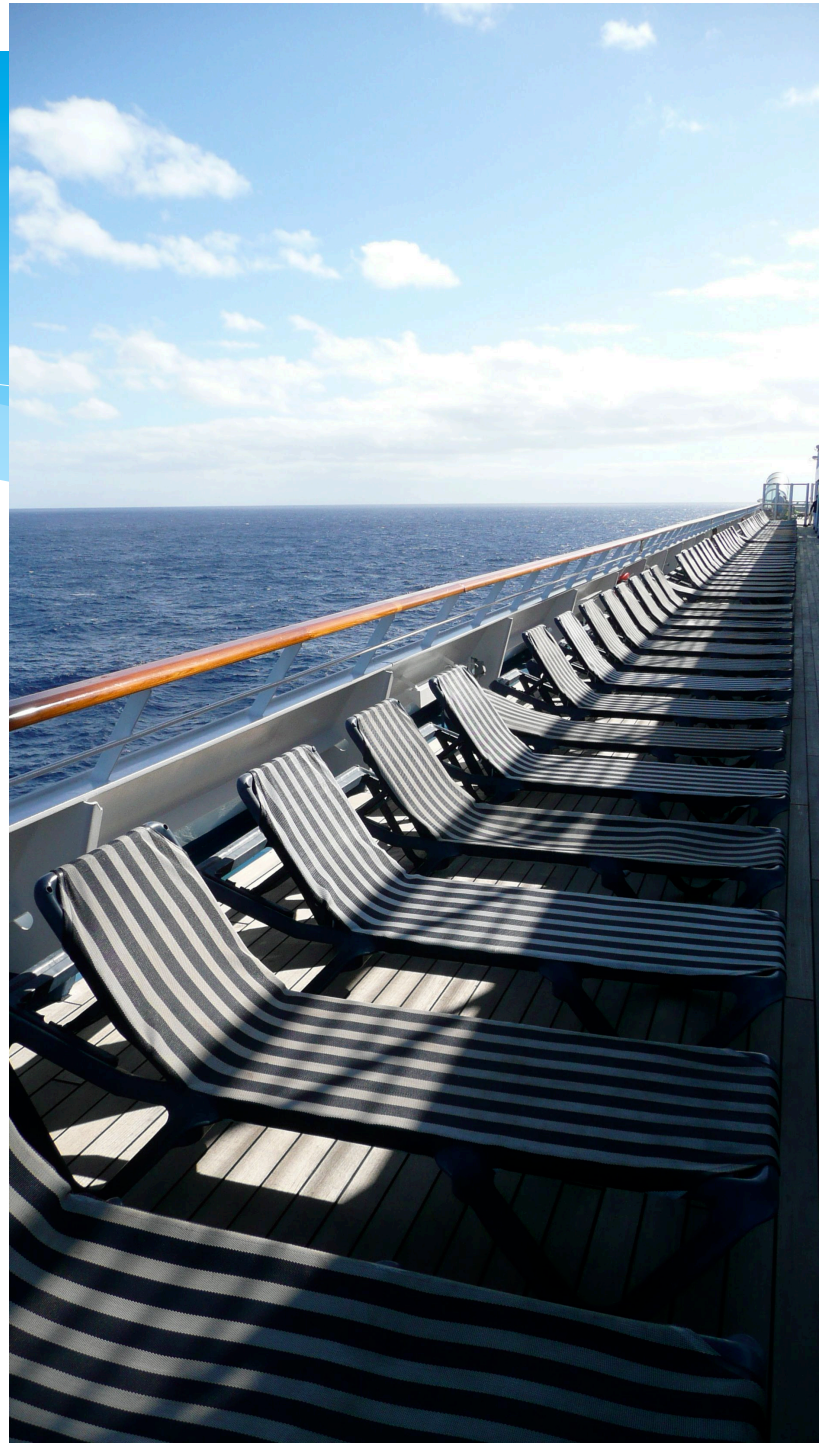
Restore Strategy

- Restore PRIMARY FG from full
- Restore read-only FGs from full
- Restore active FGs from:
  - Old full
  - Most recent diff
  - Transaction logs

SQL CRUISE

Who Does This?

- Not supported in GUI
- Barely supported in third party tools
- Totally inconvenient

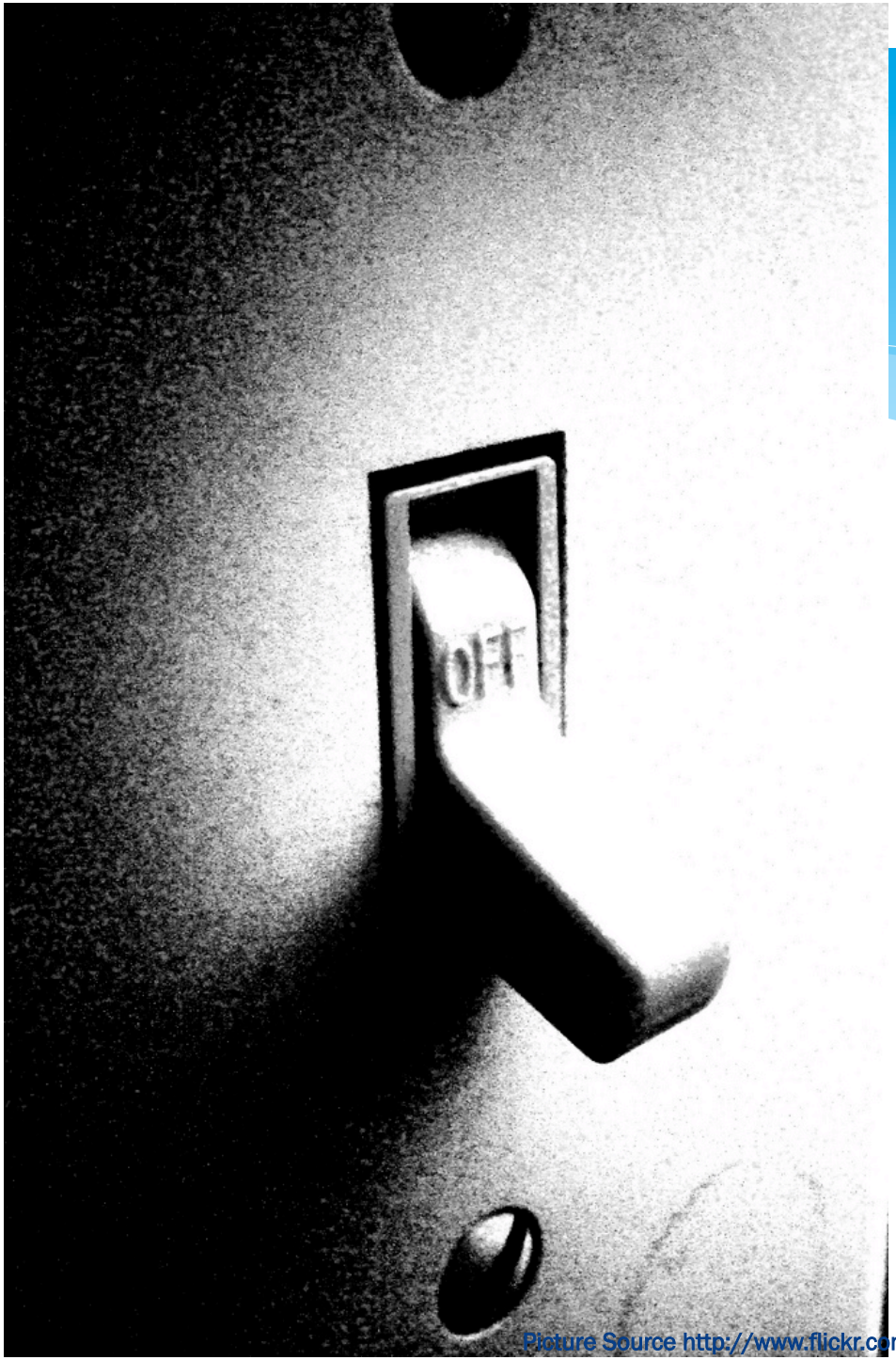


SQL CRUISE

Better Option: Back Up Less*

- Compress the backups
- Compress the data
- Archive data out

SQL CRUISE



Native Backup Compression

- Only one setting
- Compresses everything, every time

Picture Source http://www.flickr.com/photos/20052121@N00/2247588193/

Compress the Data

- 2005+ Enterprise Edition only
- Compress it once and you're done
- Often saves >50%
- Faster SELECT queries
- Bad for write-biased OLTP

SQL CRUISE

Compression Gotchas

- No inheritance
- No built-in alerting
- Must revisit frequently

SQL CRUISE

Archive Data Out of the DB

It may not be your job
to **build** an archiving strategy,
but it is your job to **sell** one.

SQL CRUISE

Typical Terabyte Troubles

- Restores
- Backups
- *DBCCs*
- Querying

- Bad Ideas for:
  - Schema
  - Indexes (CL/NCL)
  - Partitioning
  - Storage

SQL CRUISE

Why do DBCCs?

Picture Source http://www.flickr.com/photos/taniwha/62716776/

The Corruption Timeline

- Tuesday 1AM: Full backup
- Every 15 minutes: t-log backup
- Tuesday 11AM:
  User reports corruption
- What do we restore?

SQL CRUISE

Sad Facts About Corruption

- It's not logged when it happens
- Error 825 – Informational only
- No idea which backups to restore
- Page may have changed at any time
- Bigger databases = more to corrupt

SQL CRUISE

Sad Facts About DBCC

- Time-intensive
- Resource-intensive
- No progress indicator

SQL CRUISE

EZ DBCCs on VLDBs

- SAN snapshots
- 1st thing after refreshing dev/QA
- Break DBCC into small chunks: BrentOzar.com/go/vldbdbcc

SQL CRUISE

Typical Terabyte Troubles

- Restores
- Backups
- DBCCs
- *Querying*

- Bad Ideas for:
  - Schema
  - Indexes (CL/NCL)
  - Partitioning
  - Storage

SQL CRUISE

When Do Statistics Update?

1. When 1% of the data changes
2. When 10% of the data changes
3. When 20% of the data changes
4. When 50% of the data changes
5. It depends

SQL CRUISE

When Do Statistics Update?

1. When 1% of the data changes
2. When 10% of the data changes
3. **When 20% of the data changes**
4. When 50% of the data changes
5. It depends

SQL CRUISE

How Do Statistics Update?

1. By sampling 1% of the data
2. By sampling 10% of the data
3. By sampling 20% of the data
4. By scanning of the data
5. It depends

SQL CRUISE

How Do Statistics Update?

1. By sampling 1% of the data
2. By sampling 10% of the data
3. By sampling 20% of the data
4. By scanning of the data
5. It depends

SQL CRUISE

How Statistics Update

- When an index is rebuilt:
  statistics updated with full scan
- When you update statistics:
  uses specified sampling rate
  (or whatever SQL decides)

SQL CRUISE

Updating Stats In a Perfect World

- Set a weekly maintenance window (not outage, just slowdown)
- Integrate stats update with ETL
- Track which indexes you rebuild
- Only touch stats that weren't touched

SQL CRUISE

SQLfool.com

SQL FOOL

ADVENTURES IN SQL TUNING – A BLOG FOR THE REST OF US

HOME | ABOUT THE AUTHOR | SQL SPEAK | SUMMIT RESOURCES

22
APR/10
33

Index Defrag Script, v4.0

In my blog post, "Index Defrag Script Updates - Beta Testers Needed", I stated "I'll hopefully have the new version online in just a few days." That was dated January 26th. I had every intention of following through with it, too, but something came up:

Typical Terabyte Troubles

- Restores
- Backups
- DBCCs
- Querying

- ***Bad Ideas for:***
  - Schema
  - Indexes (CL/NCL)
  - Partitioning
  - Storage

SQL CRUISE

Eureka!

"Let's separate
data and indexes
into filegroups!

SQL CRUISE

Picture Source http://www.flickr.com/photos/20052121@N00/2247588193/

My personal preference for configuring initial filegroups for any system is:

PRIMARY (This can't be changed) on minimal I/O path (not much I/O needed)

DATA on a separate I/O path; DATA gets the DEFAULT filegroup property

INDEX on a separate I/O path

On PRIMARY I place no additional files other than the .MDF file (which has to be in the PRIMARY filegroup), and no additional data goes into the .MDF file. It stays very small and isn't used for user data.

On DATA I place all of my clustered indexes or (if I have any) heaps. Since this is also the Default filegroup, if I accidentally forget the "ON [DATA]" clause at the end of my object creation statements, they'll go here instead of the PRIMARY filegroup (the reason for that is for a later post).

On the INDEX filegroup I place all of my non-clustered indexes.

This of course assumes that DATA and INDEX are both stored on separate I/O paths. As long as they are, I/O resources spent retrieving data from tables won't interfere with I/O searching through indexes. I/O resources spent performing updates, inserts and deletes will be split between the tables and the indexes. Will you see double the performance? Of course not. But you should see a measurable and visible increase in performance vs storing everything on the PRIMARY filegroup.

- **For heavily accessed tables, place these tables in one filegroup and place the table's indexes in a different filegroup on different physical disk arrays.** This will improve performance, because separate threads will be created to access the tables and indexes.

For VLDB, tables and their related indexes should be separated onto separate files and physical disks for optimum performance running queries, but not separate filegroups. If they are in separate filegroups, then you cannot back up and restore them as a single unit. [7.0, 2000, 2005] *Updated 10-02-2006*

You-Reek-A - May Not Be Able To:

- Identify and control IO paths
- Control parallelism
- Back up & restore units of data
- Use SAN snapshots
- Set one filegroup to read-only

SQL CRUISE

Eureka!

"Let's partition the data and SQL Server queries will be faster!

SQL CRUISE

You-Reek-A – It Doesn't Work

- Partition elimination rarely works
- Have to know your queries intimately
- Bottom line: great for management, archiving, sliding loads – but not faster queries

SQL CRUISE

Eureka!

"We're low on space - let's add a big empty file instead of growing these."

File 1 – 95% used

File 2 – 95% used

File 3 – 95% used

SQL CRUISE

Picture Source http://www.flickr.com/photos/lifeisaprayer/2282011834/

Eureka!

Guess where the inserts go?

SQL CRUISE

Picture Source http://www.flickr.com/photos/lifeisaprayer/2282011834/

The Better Way

- Start with 4 files per filegroup
- Put all on different IO paths (maybe)
- Add new filegroups, move objects to it
- Rebuild all indexes after adding files
- Extremely important for 1GB iSCSI

SQL CRUISE

Typical Terabyte Troubles

- Restores
- Backups
- DBCCs
- Querying

- Bad Ideas for:
  - Schema
  - Indexes (CL/NCL)
  - Partitioning
  - Storage

SQL CRUISE

BrentOzar.com/go/san

SAN Storage Best Practices for SQL Server

Using SQL Server on SAN storage doesn't always guarantee fast performance. Before the SQL Server installation starts, you need to get the SAN configuration right. Here's my posts on how to get things started right.

How to Configure Your SAN Storage for SQL Server

Database administrators usually see the SAN as a black box. We ask the SAN administrators for a few arrays, and they just ask us what size we need. Configuring storage for SQL Server is more complex than other applications, but thankfully we've got a lot of help. Your SAN vendor has already put a lot of work into documenting how to set up arrays on your SAN controller specifically for SQL Server. Here's the most common vendor document repositories:


Fear begins
where
control ends.
Beth Leonard, circumnavigator