Tips for Upgrading From SQL 2008 to 2012 or 2014 Part 2

Previously, I mentioned in part 1 that a line has been drawn bisecting the upgrade path leading from SQL 2008 versions to SQL 2012+. As of SQL 2012, Microsoft has implemented a new licensing scheme and SQL instances are now licensed by core rather that by CPU.

In most cases, if an upgrade plan involves crossing this line, a side-by-side migration is the only viable option. Yes, an in-place migration is possible, but in many scenarios the old hardware will simply not be a good match for the upgraded SQL Server software. To summarize and (greatly) simplify the issue; (a) licensing is per core and existing servers with many cores per socket will cost more to license and (b) licensing is per core and sold in 4 packs – servers with 1 or 2 CPU will pay for 4.

So, while the side-by-side technique is usually more complex than an in-place, it is the better option. On the bright side, once an upgrade project is committed to side-by-side, it opens up a few opportunities.

To complete the side-by-side you will have to move the data from one place to another and that is the perfect time to modify the underlying infrastructure. Now is your chance to version-up or otherwise change the OS, optimize the server hardware, make a move to another platform entirely (virtualization or the cloud), or even encompass the migration within a larger effort to consolidate the environment.

Once the target destination is defined, and ideally it is an improvement from the existing server, the next step is to get the data to the new instance with minimal complications and down time. Let's take a look at options for a side-by-side migration from a physical SQL 2008 instance to a physical SQL 2014 instance.

Moving the Data

Single server vs two-server migrations – A side-by-side migration can be accomplished on a single server. This involves installing a new 2014 instance on the existing 2008 instance server. The second instance must have its own name. (If the existing instance is the default, the new instance must be named and vice-versa). As you know by now, although this would get the job done, and may be ok in some scenarios, we likely will still have the problem of the underlying infrastructure not being well matched to SQL 2012+ licensing. I wanted to point it out here for completeness, the techniques below all involve two servers.

DB mirroring – DB Mirroring is an excellent tool for migrating databases to new instances. One of its big advantages is that it minimizes down-time during the actual migration since it streams transactions over the wire and applies them synchronously (EE only) to the database on the new instance. Once the database mirror is configured, which can be well ahead of the actual migration date, the data has effectively been moved and the migration step just consists of ensuing the DBs are in-synch, failing over the mirror session, and redirecting the

11800 Ridge Pkwy., Suite 125, Broomfield, CO 80021
Phone: 877.722.8247 | Email: info@datavail.com
Website: www.datavail.com

client connection. One caveat – when mirroring up-version, one the failover happens, there is no way to down-version database back to the original (i.e. failback). Enterprise Edition allows synchronous mirroring so it's easy to ensure a mirror database is up to date before the failover. Standard Edition allows only asynchronous mirroring so in that case it's good practice to quiescence the database before failover to ensure an up to date migration target database.

Log Shipping – If database mirroring is not a desirable option, the next best thing maybe log shipping. Conceptually similar to DB Mirroring, rather that streaming transaction log data, log shipping copies and restore entire transaction log backups. Like mirroring it is an automated process and also can be set up well ahead of the migration date. The actual migration step is a manual process since first the log shipping must be stopped, then the last (or last several) transaction log files copied to the new server must be restored. Finally, the client can be redirected to the 2014 instance and database. Migration via log shipping has a longer outage window that mirroring because of the time taken to restore the log files to bring the target database up to date. Like DB Mirroring, it's best to quiescence the original database before migration.

Backup\Restore – One of the initial steps when configuring log shipping or DM Mirroring is a backup of the original database and a restore to the new database on the up-version instance. In fact, this can be a migration technique itself. The down-side is a much longer outage as no access can be allowed as the DBA backs up, copies over, and restores the database. On the other hand, if there is an appropriate maintenance window, this can be great – a simple solution with a minimum of moving parts. Consider striping the backup to break it up into several files and use backup compression if possible to reduce the copy time.

Detach\Attach – Similar to backup and restore, this is another solution that is not too complex: detach using sp_detachdb, copy database datafiles to the new server, and attach using sp_attachdb. Again the drawback is a longer outage as the datafiles are copied from one instance to the other. Since the move involves copying the datafiles (rather than a large single backup file) this may be a good option for VLDBs, but only if the database has been split up into reasonably sized files before the move (though a striped, compressed backup may still be the better solution).

SSIS aka Import\Export Wizard – Lastly, to get the data over to the 2014 instance table by table, the Import\Export Wizard is an option. This might be a viable solution if the data needs some kind of transformation as it is moved since you can save the wizard's work as an SSIS package and make further edits to the package as required.

Server Objects – Note, all of the options above only concern databases. None take server objects into account. Server level objects must also be migrated over – typically as a manual script-out-script-in process – before the migration date. Server level object include: server logins, SQL Agent jobs, linked servers, SSIS packages, etc.

Use the tools

The goal of this post is to point out a few concerns and opportunities around upgrading SQL Server to 2012+ versions from older versions look at a few ways to do it. However, we are only scratching surface of the many things to consider when planning and performing a SQL Server 2012+ version upgrade.

You can get a good look at the bigger picture here where you'll find the excellent document 'SQL Server 2014 Upgrade Technical Guide'. This comprehensive guide takes many more factors into consideration including connection protocols, upgrading other SQL features like SSAS and SSRS, upgrade paths from SQL 2000,

pre-migration testing techniques, and also suggests several tools like the Upgrade Advisor which helps insure the instance or database is prepared for the upgrade, and the Best Practices Analyzer to help ensure the database or instance is in good shape post-migration. If you are planning a migration project take a look at the guide, consider the issues and opportunities, and let us know if we can help.

About the Author: Andy McDermid
SQL Server DBA, Datavail

Andy is a MCITP certified MS SQL DBA who delivers and manages delivery of DBA services to many diverse clients. He enjoys helping his clients by finding and deploying pragmatic and practical solutions for their database issues. Andy is always working to improve and expand his DBA skills and he likes to share the experience via writing.