CONFIO is now part of solarwinds

Checklist for SQL Server 2012 Upgrade

Tips to Ensure the Best Outcome for Your Upgrade

By Thomas LaRock, Technical Evangelist and Head Geek

Confio Software
4772 Walnut Street, Suite 100
Boulder, CO 80301

www.confio.com

Introduction

While you can certainly click “Next, Next, Finish” and consider your SQL Server 2012 upgrade to be complete, the truth is that the process for most upgrades is often considerably more complex. A proper upgrade process involves detailed research, planning and execution.

Failing to prepare a proper upgrade process for your database server is likely to result in your end users seeing diminished performance after the upgrade is complete. Since your goal is to increase performance and stability as a result of the upgrade process you can understand that your users are likely to be upset if things were to get worse!

It can be a daunting task to put together everything you need in a pre-upgrade checklist. We’ve compiled the top eleven items that you should include in any checklist you put together for migrating your database server to SQL Server 2012. Including these items is likely to help you avoid many potential upgrade issues.

Failing to use the SQL Server 2012 Upgrade Advisor

The SQL 2012 Upgrade Advisor (UA) is just that: an advisor. It doesn't fix everything, it merely advises you on what actions you should take when upgrading to SQL Server 2012. The actions the UA recommends will come in two forms: actions to be done prior to a migration, and actions to be completed post-migration. The UA is really good at finding what I call the "stub-your-big-toe" things that need fixing prior to a migration. But it’s not foolproof, and it will not identify every last detail.

Not reviewing the "breaking changes" section in Microsoft’s Books Online

Did you know that Microsoft publishes a list of breaking changes for each version of SQL Server in its Books Online (BOL) section? You should review these lists to the point that they are familiar to you. You don't have to memorize them all, just be familiar with them so that if something odd happens you can think to yourself, “Hey, is this odd behavior listed in the breaking changes section of the BOL?” I would like to believe that the UA will alert you to many of these breaking changes, but the UA is not as dynamic as the BOL. The BOL may have an entry or two that doesn't make it into the UA checklist, and that is why you should review this section.

Skipping the "behavioral changes" section in Microsoft’s Books Online

Similar to the breaking changes, the behavioral changes are definitely worth reviewing, and they are also things that the UA is likely to never report back to you because they aren't things that will break, but merely things that could break.

© 2013 Confio Software

Forgetting to execute DBCC CHECKDB WITH DATA_PURITY

One of your post-migration or upgrade tasks should be to run the following statement:

```
DBCC CHECKDB WITH DATA_PURITY;
```

This statement will check your data for values that are no longer valid for the column datatype. For databases created prior to SQL Server 2005 (and you know they are still out there), this step is rather important to take. For databases created in SQL Server 2005 and later, the DATA_PURITY check is supposed to be done automatically with a regular CHECKDB.

But what about a database that was created in SQL Server 2000, migrated (poorly) to a SQL Server 2008 instance, and left in the SQL Server 2000 (80) backward compatibility mode? What about that little feller? Do you want to assume that the DATA_PURITY check has been getting done? Here's a thought: just go run it yourself anyway. That way you know it is getting done.

Ignoring the execute DBCC UPDATEUSAGE command

While not as critical as the DATA_PURITY command noted above, this one still has a place in any migration or upgrade process:

```
DBCC UPDATEUSAGE(db_name);
```

This command will help fix any page count inaccuracies that are resulting in the sp_spaceused stored procedure returning wrong results. And much like the DATA_PURITY check, this command is also recommended for databases that were created prior to SQL Server 2005. For databases created in SQL Server 2005 and later, you should only run this command if you feel you are getting inaccurate results from sp_spaceused, and you should note that for very large tables this command could take a long time to execute.

Not updating statistics

```
This one is not to be skipped and is simply a MUST for any
migration or upgrade checklist:

USE db_name;

GO

EXEC sp_updatestats;
```


© 2013 Confio Software

This command will update the statistics for all the tables in your database. It issues the UPDATE STATISTICS command, which warrants mentioning because you may want to use that command with the FULLSCAN option. I'm the type of person that would rather be safe than sorry and therefore would end up running something like this:

```
USE db_name;

GO

EXEC sp_MSforeachtable @command1='UPDATE STATISTICS ? WITH
FULLSCAN';
```

Bottom line here: don't forget to update the statistics.

Failing to refresh your views using sp_refreshview

Believe it or not, every now and then someone will build a view that spans into another database on the same instance. And, in what may be a complete surprise to many, sometimes these views will go across a linked server as well. The point here is that your view may not be contained in just your database. In what could be the most dramatic twist of all, sometimes these views are created using a SELECT * syntax.

What are the odds that you could have such code in your shop? It happens. And when you have bad code on top of views that go to other databases (or views of views of views of whatever else some sadistic person built) you are going to want to use sp_refreshview to refresh those views.

So, if you are migrating a database in your environment to a new server, then it would be a good idea to refresh your views using sp_refreshview. Most of the time, this won't do anything for you. But there is that one chance where it will dramatically improve performance and your customer will be happy as a result. It's like flossing: it doesn't take much effort, and the end result is usually worth that little effort.

Forgetting to take backups

You're a DBA. Backups are in your DNA. You should have taken one prior to the start of any migration, and you had better take one right before you turn that database over to your end users. Also, you should save any output from the seven items listed above, as it could prove helpful should something go awry later. (Bonus tip: make sure your backups are good!)

Not upgrading your hardware

SQL Server 2012 does not support AWE. That means if you still have a 32-bit operating system and are currently using AWE in order to see beyond 4GB of RAM, if you upgrade to SQL 2012 you will be limited to only* that 4GB of physical RAM.

It's time for you to move to a server and operating system that was released within the past ten years. If you are expecting to have more than 4GB of RAM, then you need to use the 64-bit version of SQL 2012.

© 2013 Confio Software

Not knowing the right path

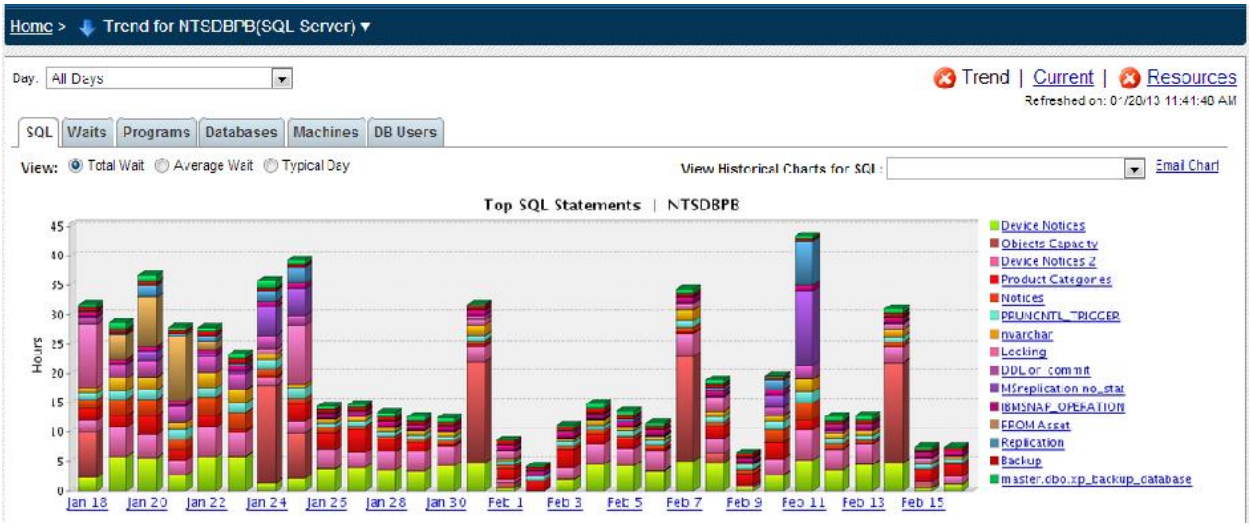
For those folks running SQL Server 2000 instances, you are not going to be able to upgrade directly to SQL Server 2012 without first upgrading to an intermediary version. You have two options to choose from when going from SQL Server 2000. The first option is to do an upgrade in place from SQL Server 2000 to SQL Server 2005, 2008, or 2008 R2. The second option is to do a backup (or even detach) your SQL Server 2000 database and restore/attach to an instance running SQL Server 2005, 2008, or 2008 R2. At that point you will be able to complete the upgrade to SQL Server 2012.

Establish a baseline for normal database workload, resource usage and response times before the upgrade.

If you have an initial baseline for normal workload resource usage and response times prior to an upgrade, then you can objectively assess performance before and after, and you can proactively assess any problems introduced by the upgrade. In fact, this is critical to do for any change event. Tools like the Distributed Replay Utility allow for you to capture a live workload that can be replayed against your upgraded test server. Running a continuous database performance monitoring tool such as SolarWinds Database Performance Analyzer (DPA) against both servers will allow for you to easily compare and contrast performance for your key business critical systems.

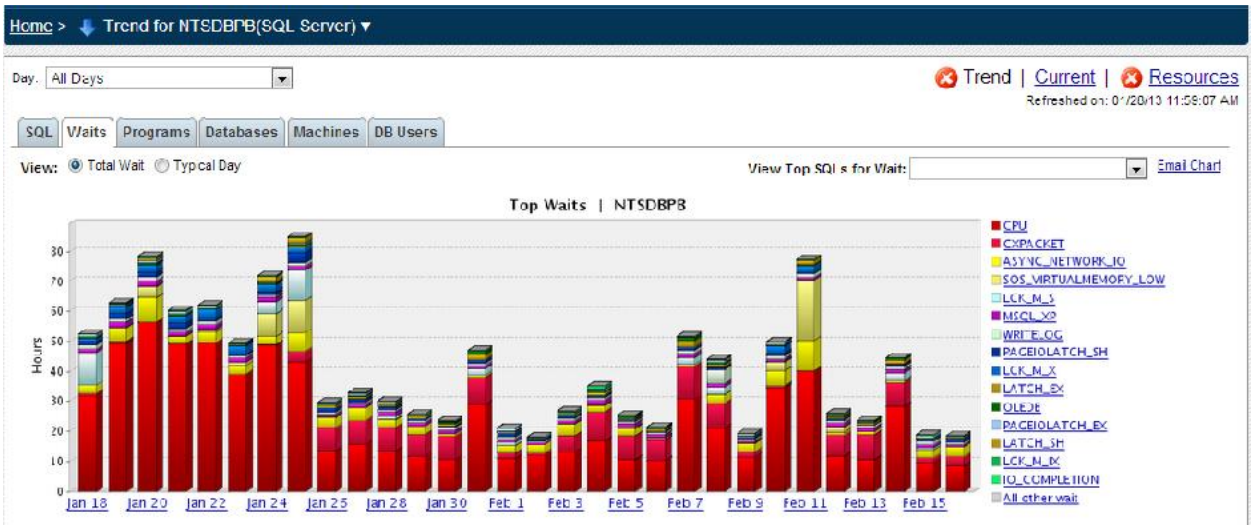
DPA will allow for you to easily sort and filter through your performance data to help ensure your upgraded system is performing as expected. DPA's agentless architecture and easy to understand visual charts make it an ideal tool for sharing performance data across IT teams, from developers to architects to QA to DBAs and even managers, which can be helpful when preparing for an upgrade.

Want to see the overall trend analysis for all the SQL statements running against your new server? That's easy, simply go to the home screen for the database instance:

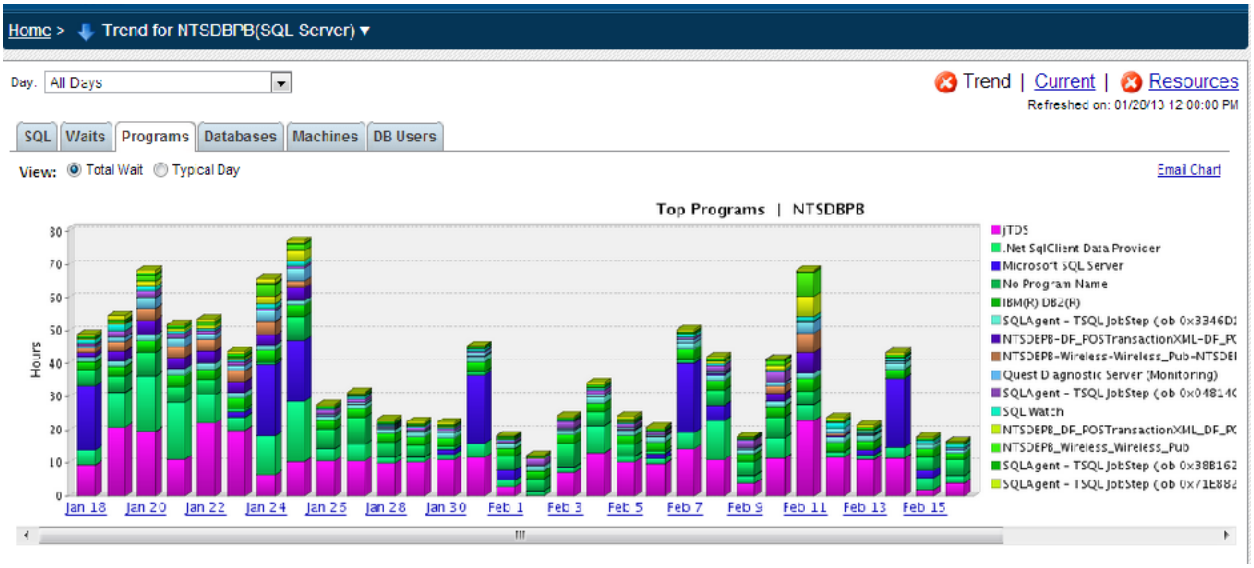


If you prefer looking at specific wait types to analyze the overall response time for the upgraded instance, just click on the 'Wait' tab and you see a similar screen:

© 2013 Confio Software

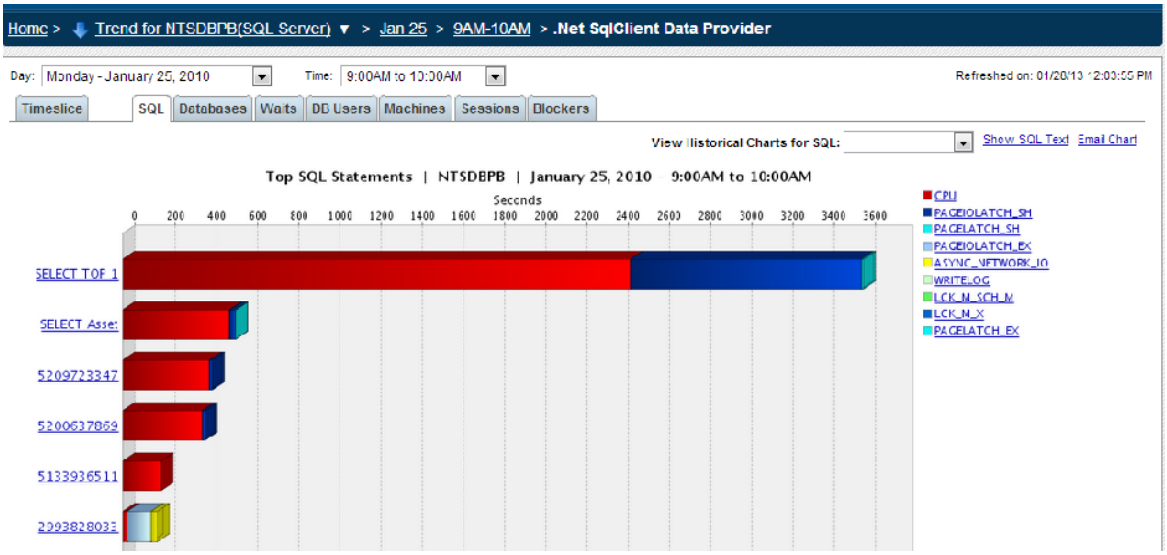


If you want to focus on the specific business critical application you can do that by going to the 'programs' tab:



You can then drill into a specific day, then hour, and then select the program name. You will then see a list of specific SQL statements that were executed by that program:

© 2013 Confio Software



In the upper right there is the ability to quickly view the historical chart for any SQL statement:

© 2013 Confio Software

At this point you can compare these results to the results on the original production server and verify if performance is acceptable before completing the upgrade process by switching your database connections to point to the new server.

It's all about the preparation

Upgrades are a necessary part of any development lifecycle. The chances of having a successful upgrade increases along with the amount of planning and preparation you invest in building a proper upgrade process. If you are planning to upgrade to SQL 2012, you can use this paper as a guide to developing your own checklist.

Incorporating DPA into your upgrade process will allow for all stakeholders in the project to easily interpret database performance metrics at every step. Offering insight to database performance during the upgrade process will reduce surprises and help ensure a smooth cutover when you transition to the new server.

About Confio Software

Confio Software, now a part of the SolarWinds family, builds award-winning database performance analysis tools for DBAs and developers. SolarWinds Database Performance Analyzer (formerly Confio Ignite) improves the productivity and efficiency of IT organizations. By resolving problems faster, speeding development cycles, and squeezing more performance out of expensive database systems, Database Performance Analyzer makes DBA and development teams more productive and valuable to the organization. Customers worldwide use our products to improve database performance on Oracle, SQL Server, Sybase and DB2 on physical and virtual machines.

For more information, please visit: http://www.confio.com/performance/sql-server/ignite/

© 2013 Confio Software