

Title Page

Friday, June 26, 2015
9:13 PM

Building a Static Data Warehouse Server

Ron Talmage
SolidQ
rtalmage@solidq.com



Why build a static DW?

Tuesday, April 07, 2015

9:41 AM

Some Users need stable underlying data

- Assuming the underlying data set is large enough to provide statistical power
- Unchanging data provides a stable base for refining analytical queries

Options:

- Copy a snapshot of the data to each user database (impractical)
- Provide a server with static data
 - A new server instance is created at the end of each year
 - Performance is not expected to match production

Goals of our static DW

Tuesday, April 07, 2015

9:39 AM

Provide a standalone server with DW Data

Store static FY2014 data

Support current users' work database

Support user's project databases

Convert tables to Clustered Columnstore Indexes (CCI)

Simplify file management

Reclaim data storage on the SAN

What we did

Tuesday, April 07, 2015

2:40 PM

Provide a standalone server with DW Data

- Upgraded the current production database server side-by-side
- Removed user databases and logins on the legacy server
- Continued to incrementally update the old server nightly

Store static FY2014 data

- Determined a cutoff date for the fiscal year
- Stopped nightly ETL processes at that point
- Took FULL database backups.
- Set the databases read_only temporarily.

Support current users

- Left the user work database in place.

Convert tables to Clustered Columnstore Indexes (CCI)

- Converted over a series of weeks

Simplify file management

- Gradually consolidated files and shifted file locations database by database

Reclaim data storage on the SAN

- DW databases will be fixed in size, and free space mostly reclaimed
- Final reduction: 50.2TB to 13.7TB = 36.5TB reclaimed

File and FG Problems

Friday, June 26, 2015

9:35 PM

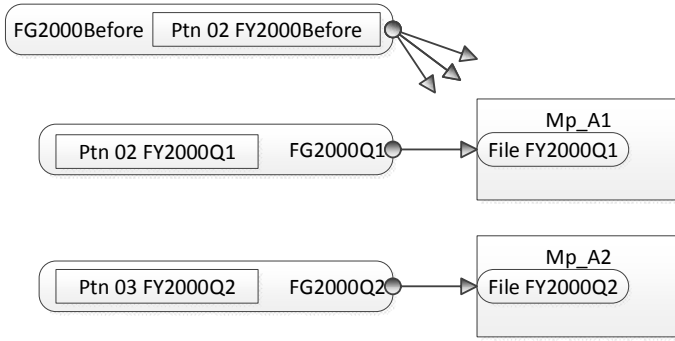
Legacy Partitioning Design

In each fact database:

- **Partition by Qtr**
- **One partition for each filegroup**
- **One file per filegroup**
- **One year: 4 partitions, 4 filegroups, 4 files**
- **16 years: 64 partitions, 64 filegroups, 64 files**
- **Placed on 8 mount points**

File/FG issues 1

Wednesday, June 24, 2015
10:18 PM

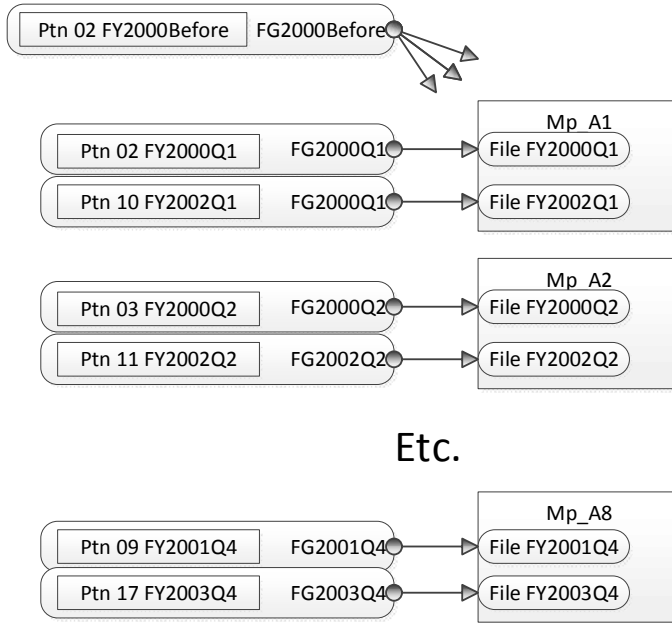


Etc.



File/FG issues 2

Friday, June 26, 2015
9:24 PM



File/FG issues 3

Friday, June 26, 2015
9:28 PM

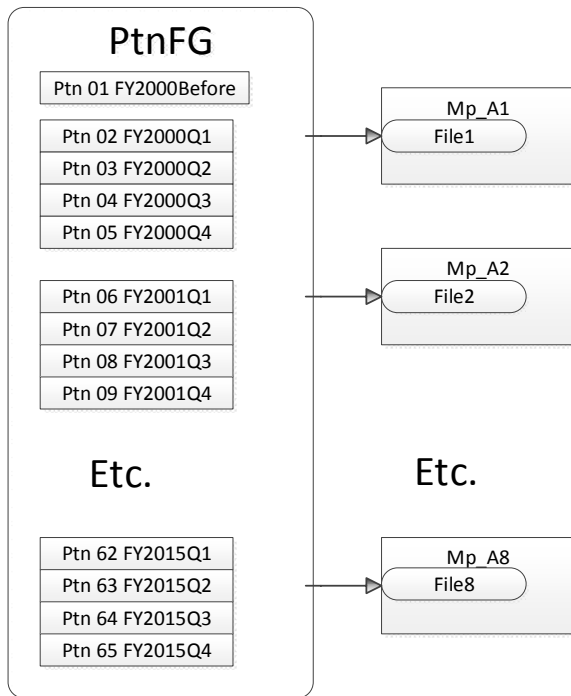
All your partition are belong to us

	Mp1	Mp2	Mp3	Mp4	Mp5	Mp6	Mp7	Mp8
DefFG	fDB10def1	fDB10def2	fDB10def3	fDB10def4	fDB10def5	fDB10def6	fDB10def7	fDB10def8
Current	fDB10F00Q1 fDB10F02Q1 fDB10F04Q1 fDB10F06Q1 fDB10F08Q1 fDB10F10Q1 fDB10F12Q1 fDB10F14Q1	fDB10F00Q2 fDB10F02Q2 fDB10F04Q2 fDB10F06Q2 fDB10F08Q2 fDB10F10Q2 fDB10F12Q2 fDB10F14Q2	fDB10F00Q3 fDB10F02Q3 fDB10F04Q3 fDB10F06Q3 fDB10F08Q3 fDB10F10Q3 fDB10F12Q3 fDB10F12Q3	fDB10F00Q4 fDB10F02Q4 fDB10F04Q4 fDB10F06Q4 fDB10F08Q4 fDB10F10Q4 fDB10F12Q4 fDB10F12Q4	fDB10F01Q1 fDB10F03Q1 fDB10F05Q1 fDB10F07Q1 fDB10F09Q1 fDB10F11Q1 fDB10F13Q1 fDB10F15Q1	fDB10F01Q2 fDB10F03Q2 fDB10F05Q2 fDB10F07Q2 fDB10F09Q2 fDB10F11Q2 fDB10F13Q2 fDB10F15Q2	fDB10F01Q3 fDB10F03Q3 fDB10F05Q3 fDB10F07Q3 fDB10F09Q3 fDB10F11Q3 fDB10F13Q3 fDB10F15Q3	fDB10F01Q4 fDB10F03Q4 fDB10F05Q4 fDB10F07Q4 fDB10F09Q4 fDB10F11Q4 fDB10F13Q4 fDB10F15Q4
New	fDB10F16Q1	fDB10F16Q2	fDB10F12Q3	fDB10F12Q4	fDB10F17Q1	fDB10F17Q2	fDB10F17Q3	fDB10F17Q4

- ▶ **Total files:**
 - ▶ 16 years * 4 files/year = 64 files + 8 Before = 72
 - ▶ DB10, 12, 14, 15, 16 = 72 files * 5 databases = 360 files

Solution: single FG

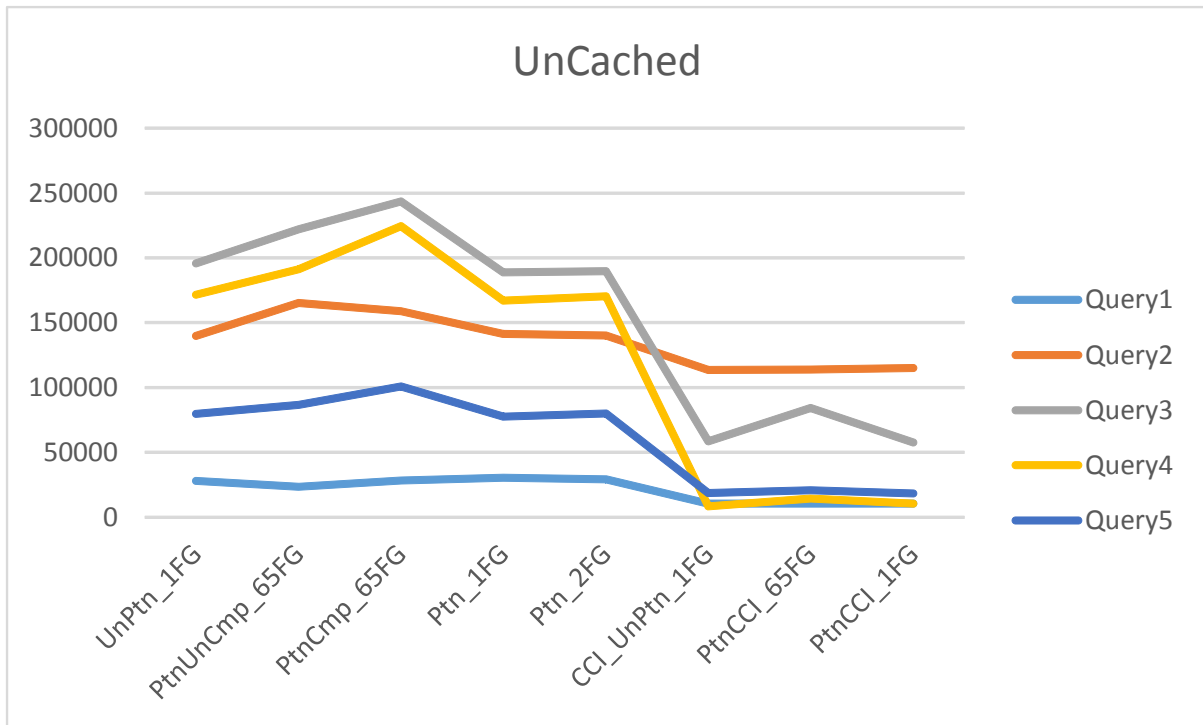
Friday, June 26, 2015
9:22 PM



Testing - Uncached

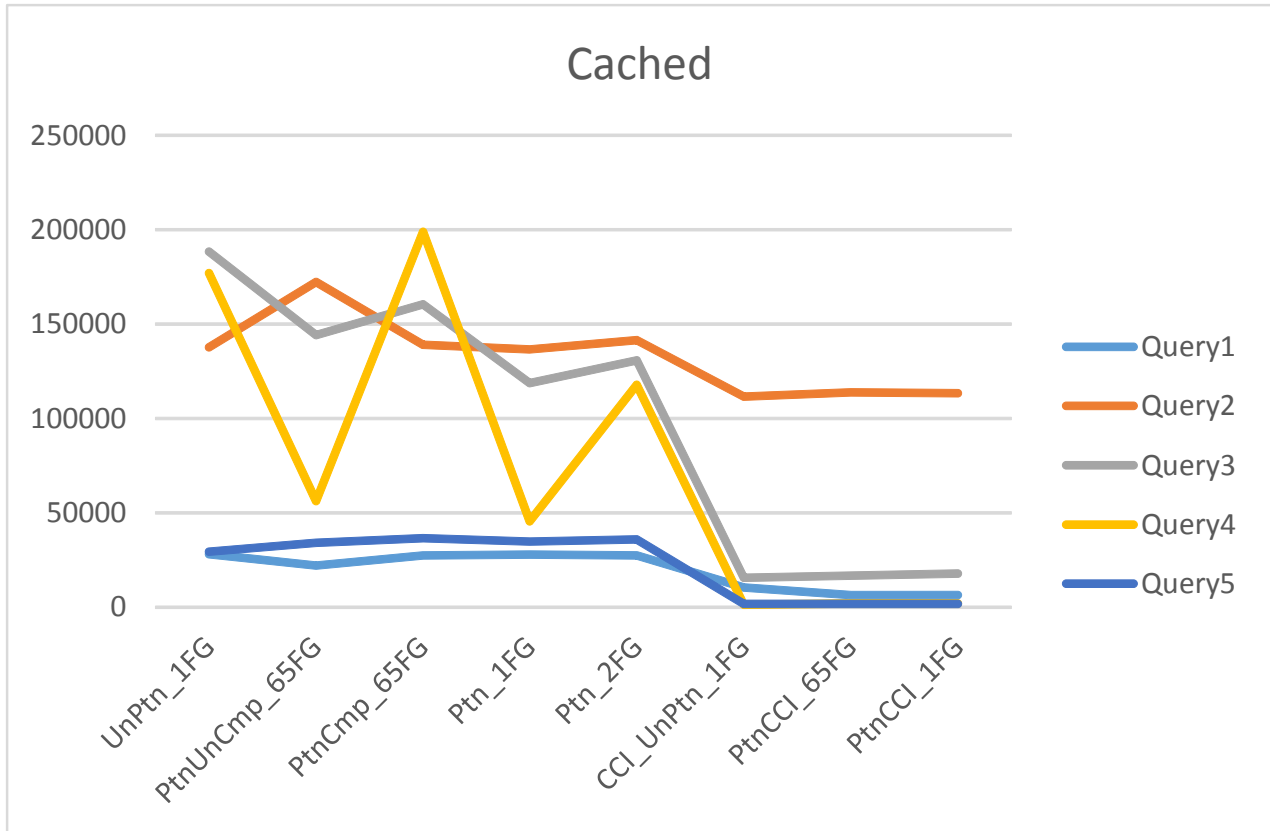
Friday, June 26, 2015

9:36 PM



Testing - Cached

Friday, June 26, 2015
9:36 PM



Conversion Steps

Tuesday, April 07, 2015

9:40 AM

[Prepare DB and Remove NC Indexes](#)

[Add new CCI tables](#)

[Load CCI tables](#)

[Remove objects and shrink files](#)

[Set file locations and properties](#)

Prepare DB and Remove NC Indexes

Tuesday, April 07, 2015

9:43 AM

1. Back up the database
2. Find and drop irrelevant tables
3. Get partitioned table information
4. Script out and refresh any users views for these tables
5. Build new user views placing computed columns expression into view columns,
6. Test the user views
7. Remove NC indexes from the partitioned and non-partitioned tables
8. Back up the database

[Back](#)

Add new CCI tables

Tuesday, April 07, 2015

9:43 AM

1. Add the new PtnFG filegroup, with 8 files
2. Add new partition functions and partition schemes for PtnFG
3. Create the initial CCI tables using SELECT INTO, with no index set, on Default FG.
4. Find and remove all computed columns from the new CCI tables
 1. Improvement: leave the columns static!
5. Add a clustered index to the CCI tables: place the Ptn tables in the PtnFG scheme, rest in DefFG
6. Generate and create the CCI indexes with DROP_EXISTING

[Back](#)

Load CCI tables

Tuesday, April 07, 2015

9:43 AM

1. Load the non-Ptn CCI tables with INSERT SELECT
2. Load the Ptn CCI tables using a stored procedure, INSERT SELECT per partition
 1. Improvement: Stage and use SWITCH
3. Rename the CI tables to _old, rename the CCI tables to original name
4. Refresh user views
5. Close all OPEN rowroups
6. Create statistics on each CCI table
7. Back up the database

[Back](#)

Remove objects and shrink files

Tuesday, April 07, 2015

9:44 AM

1. Back up the database
2. Drop the old tables
3. Shrink the files on the old Ptn filegroups with TRUNCATE_ONLY
4. Remove files from old Ptn filegroup
5. Remove old partition schemes
6. Remove old partition functions
7. Remove old filegroups
8. Shrink Default filegroups

[Back](#)

Set file locations and properties

Tuesday, April 07, 2015

9:44 AM

1. Move the Ptn and DefFG files to desired locations
2. Set the Ptn and DefFG filegroups to read only
3. Do not set the database to read only
4. Back up the database

[Back](#)

Open Issues

Wednesday, April 08, 2015

1:10 PM

1. Computed columns

- a. Just create and populate uncomputed columns, the data is static!

2. CCI Loading

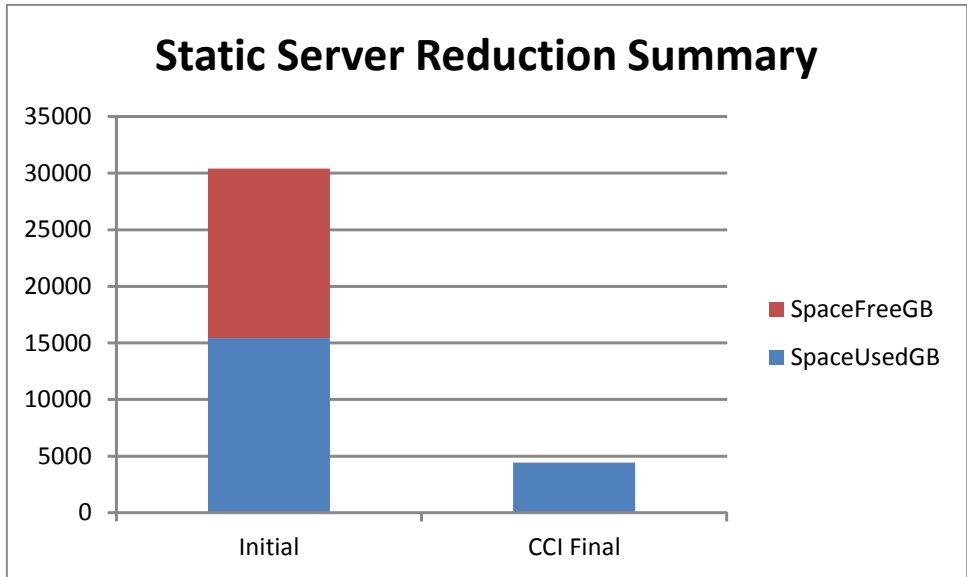
- a. Change initial load to use partition switching
- b. Load a CCI-enabled staging table with a given partition's data
- c. Apply the table constraint
- d. SWITCH the partition

3. CREATE STATISTICS on CCI tables:

- a. One statistics on every column, or one statistic for each column? (Each column)
- b. Select only most likely filtered columns
- c. Requires a job to recreate statistics with each server restart

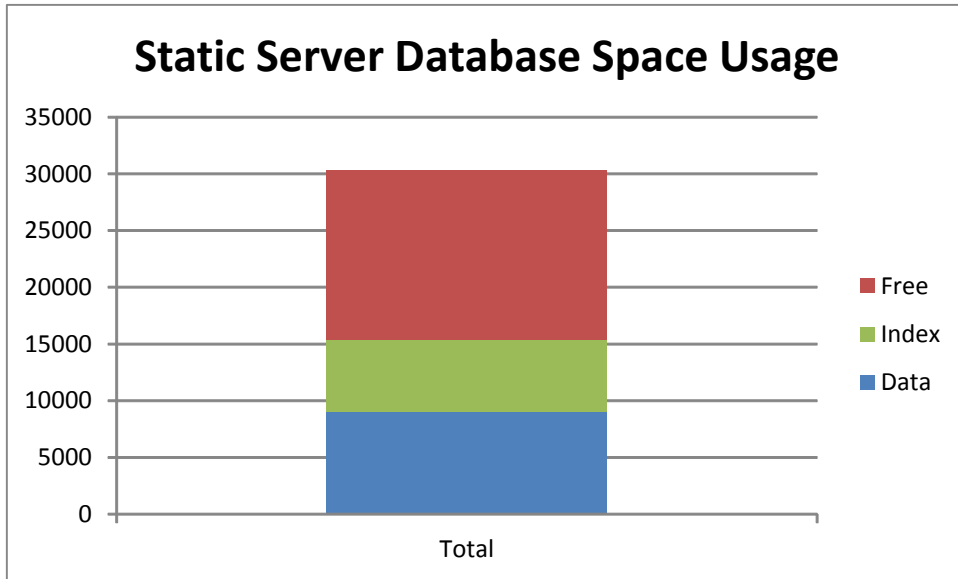
Final Reduction

Tuesday, April 07, 2015
3:30 PM



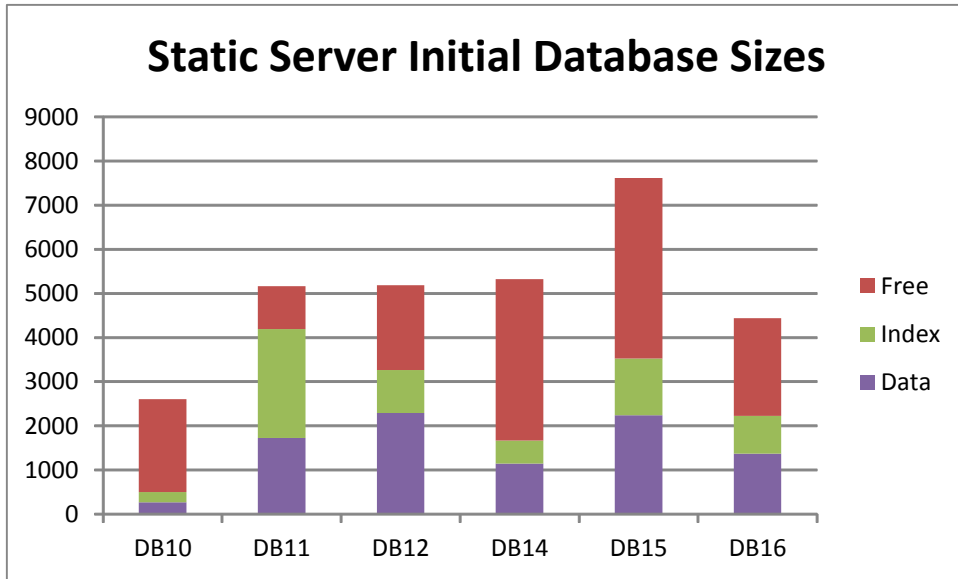
Initial Size Summary

Tuesday, April 07, 2015
9:40 AM



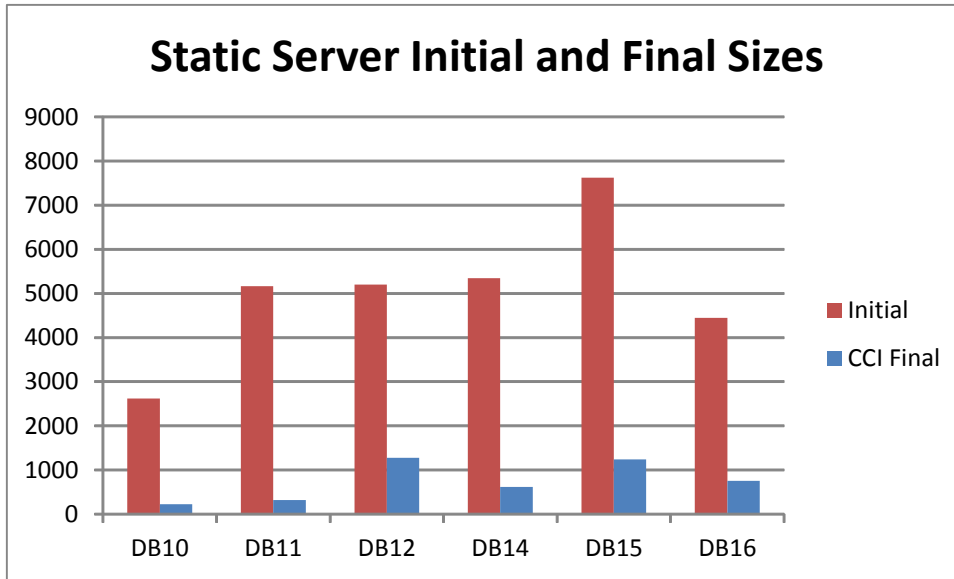
Summary per database

Tuesday, April 07, 2015
3:29 PM



Initial and Final per database

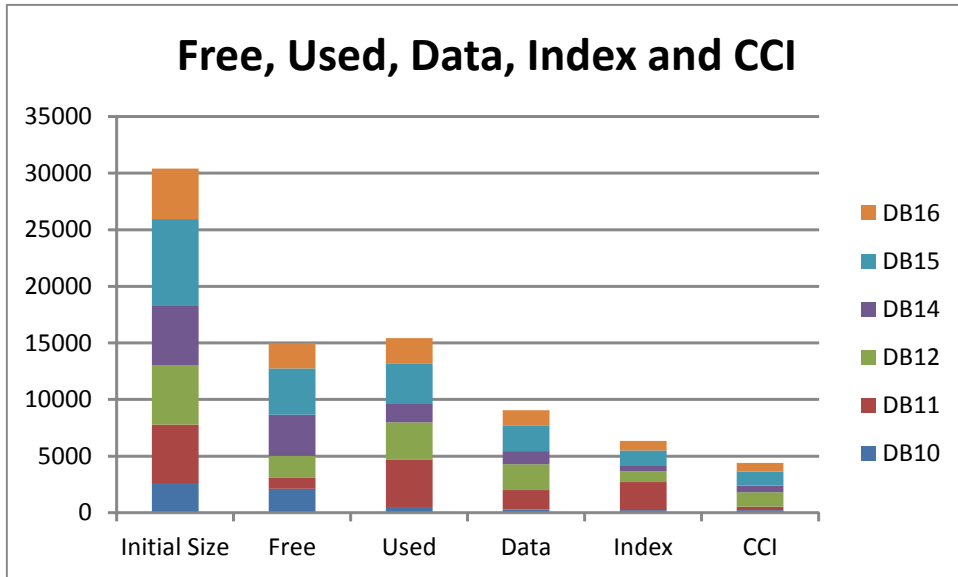
Tuesday, April 07, 2015
3:34 PM



By free, used, etc.

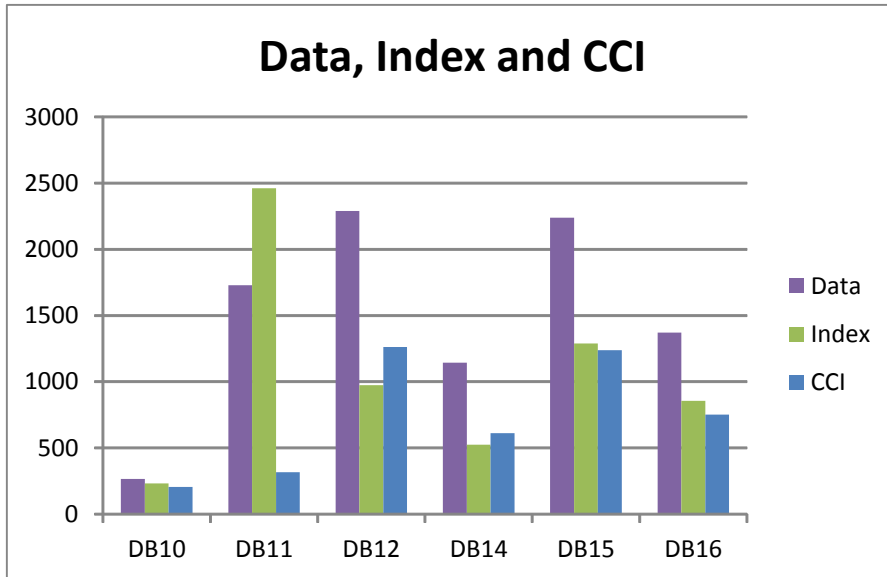
Tuesday, April 07, 2015

3:35 PM



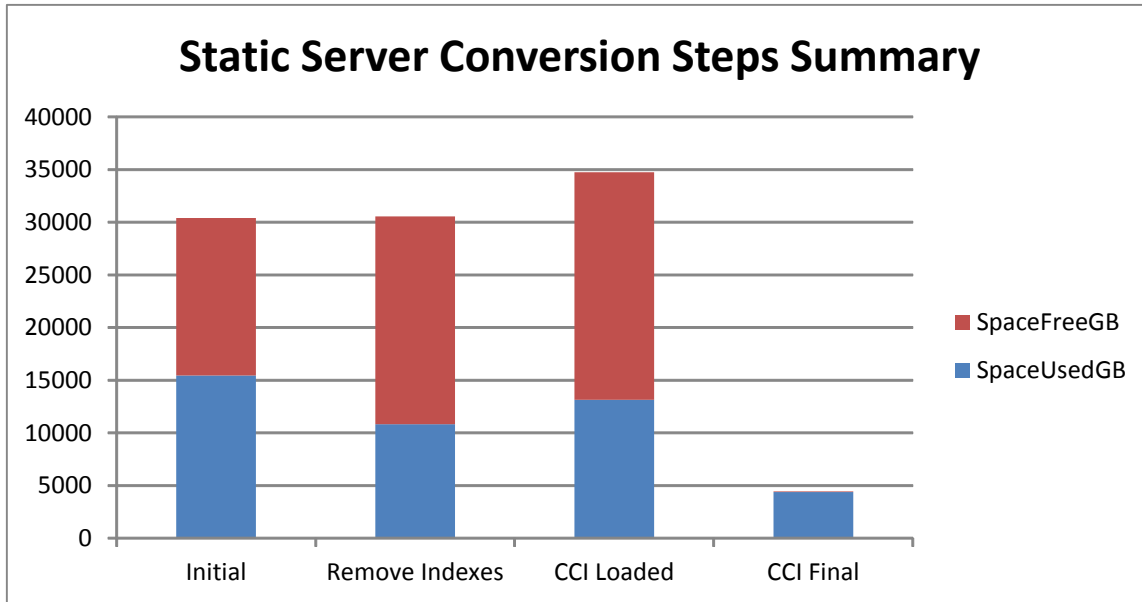
By data, index, etc.

Tuesday, April 07, 2015
3:37 PM



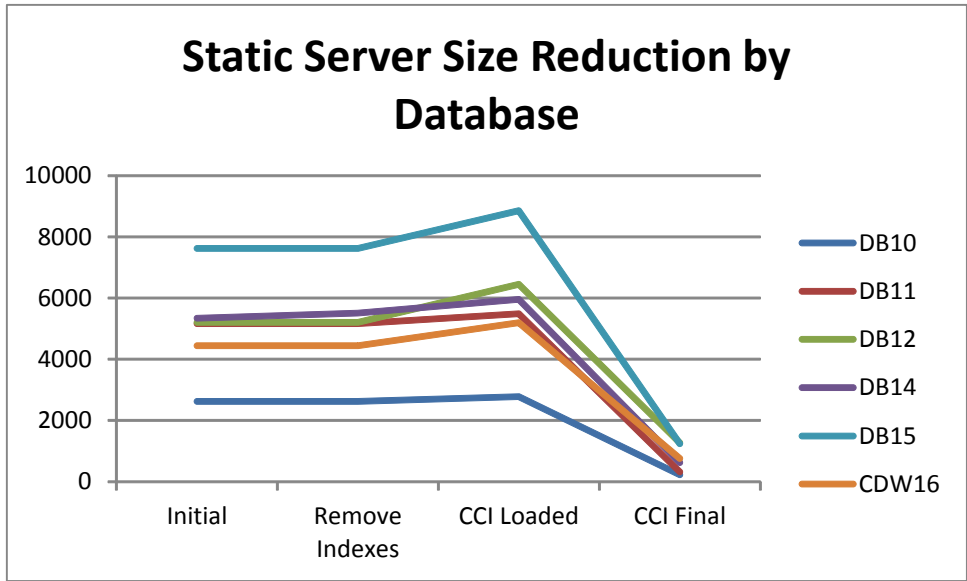
Conversion Steps Summary

Tuesday, April 07, 2015
3:39 PM



For all databases - total size

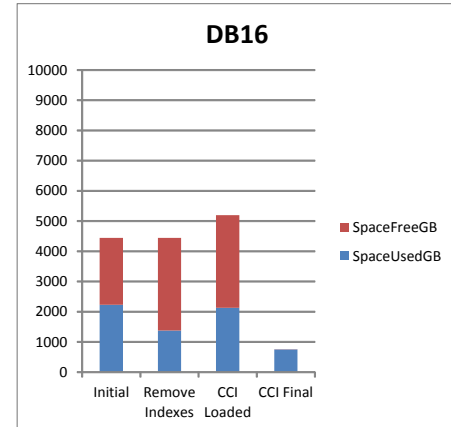
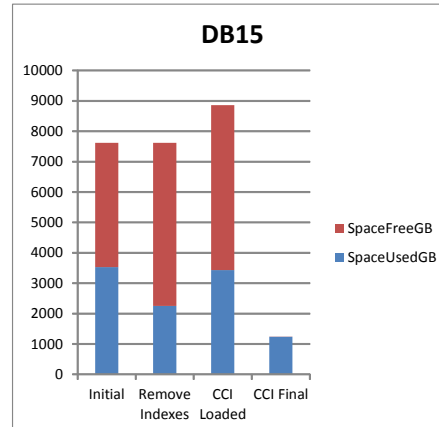
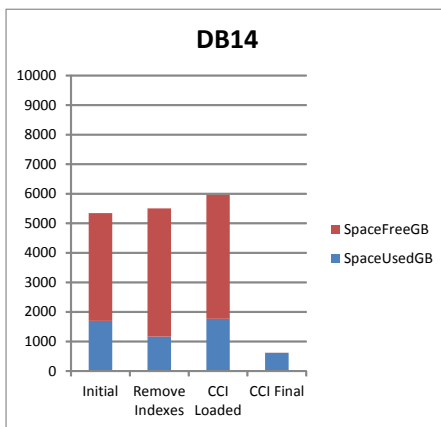
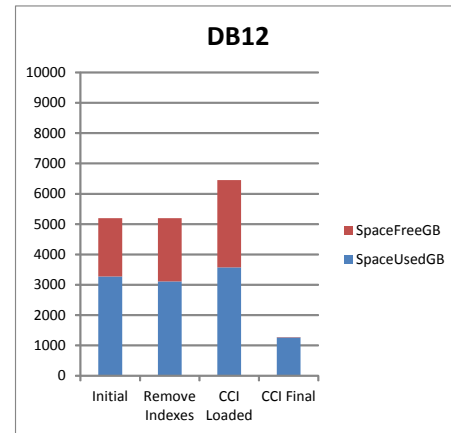
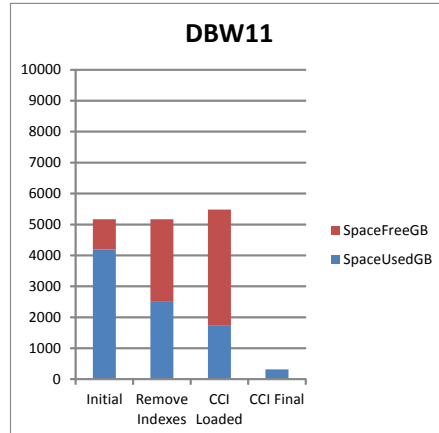
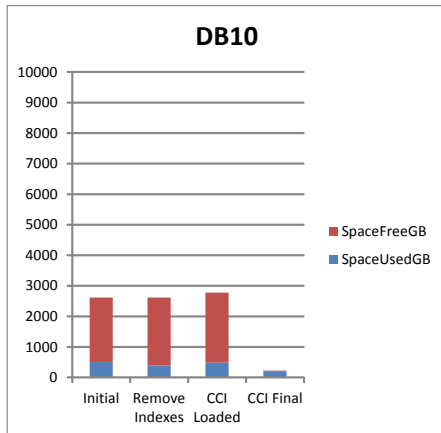
Tuesday, April 07, 2015
3:40 PM



For all databases - free and used

Tuesday, April 07, 2015

3:42 PM



Reclaiming Space Details

Tuesday, April 07, 2015

3:46 PM

		Initial GB	Remaining GB	Reclaimed GB
CCI and LUN consolidation	Backup	2600	2600	0
	Log	3400	1800	1600
	Data	50200	28400	21800
LUN renaming	Data	28400	13700	14700
Total reclaimed				38100

Resources

Wednesday, April 08, 2015

3:02 PM

Clustered Columnstore Indexes:

- [Using Clustered Columnstore Indexes](#)
- [SQL Server Columnstore Index FAQ](#)
- [Data Loading performance considerations with Clustered Columnstore indexes](#)

Blogs:

- Nikoport: <http://www.nikoport.com/columnstore/>

Thanks!

Friday, June 26, 2015
9:13 PM



Thank You!
local PASS Community & Sponsors!



The Conference for SQL Server Professionals.

October 27-30 / Seattle, WA