

# Why build a static DW Server?

Tuesday, April 07, 2015 9:41 AM

Users often need stable underlying data

- Changed data is removed as a variable during query project
- Assumes the underlying data set is large enough to provide statistical power

Two ways to achieve this:

- 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

# Static DW Server Goals

Tuesday, April 07, 2015 9:39 AM

## **Provide a standalone server with DW Data**

- This server should appear to end users with the same database names and view names as the source data warehouse server.
- They know from the server name that the data is static.

## **Store static FY2014 data**

- Determine a cutoff date (such as December 31, 2014)
- Do not allow updating of data after that date.

## **Support current users**

- Users in their work databases should be able to connect and query from as usual

## **Support project databases**

- Users can request this server for a project database

## **Convert tables to Clustered Columnstore Indexes (CCI)**

- Make large table scans more efficient
- Reduce disk space footprint and implement better compression

## **Reclaim data storage on the SAN**

- Give saved space back to the Storage Team for other uses

# Implementing (Case Study)

Tuesday, April 07, 2015 2:40 PM

## **Provide a standalone server with DW Data**

- Upgraded the current production database server to a new server and SAN
- Removed user databases and logins
- Continued to update the old server nightly

## **Store static FY2014 data**

- Determined a cutoff date
- Stopped nightly ETL processes at that point
- Took FULL database backups.
- Set the databases read\_only.

## **Support current users**

- Left the user work database in place.

## **Convert tables to Clustered Columnstore Indexes (CCI)**

- Converted over a series of weeks
- 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

# 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
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 the LoadPartition procedure, INSERT SELECT per partition, in parallel
3. Rename the CI tables to \_old, rename the CCI tables to original name
4. Refresh user views
5. Close all OPEN filegroups
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 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](#)

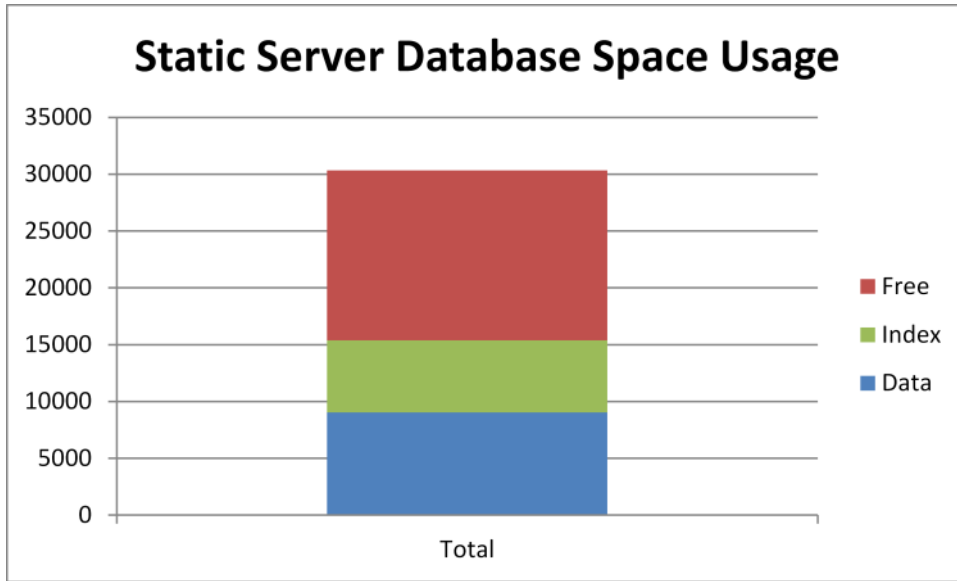
# Open Issues

Wednesday, April 08, 2015 1:10 PM

1. CCI Loading
  - a. Initial load of partitioned CCI tables:
    - i. Partition by partition? (Yes, in clustered order)
2. CREATE STATISTICS on CCI tables:
  - a. One statistics on every column, or one statistic for each column? (Each column)
  - b. Job to recreate statistics with each server restart? (Only required once)
  - c. Properties of CCI statistics (no\_recompute, etc.)

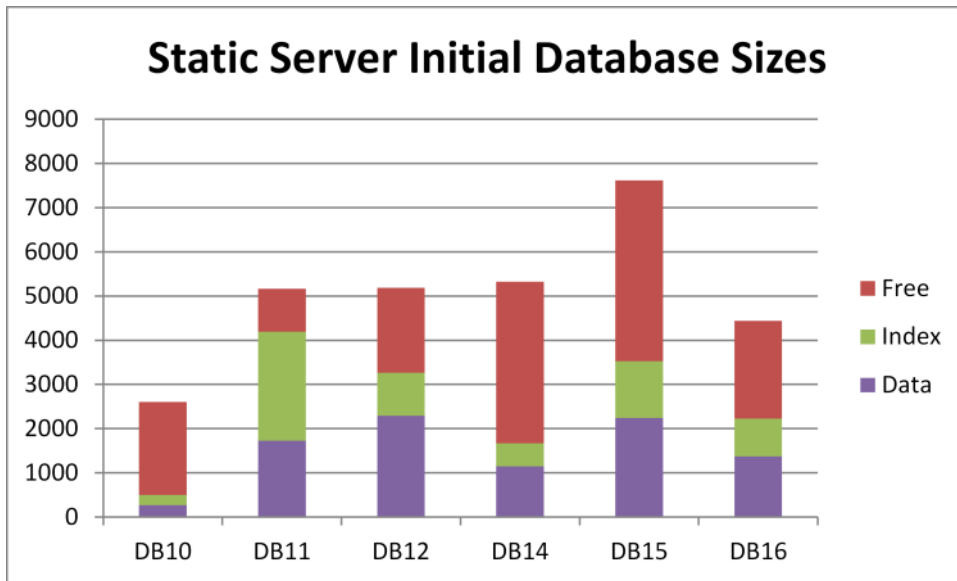
# Initial Size Summary

Tuesday, April 07, 2015 9:40 AM



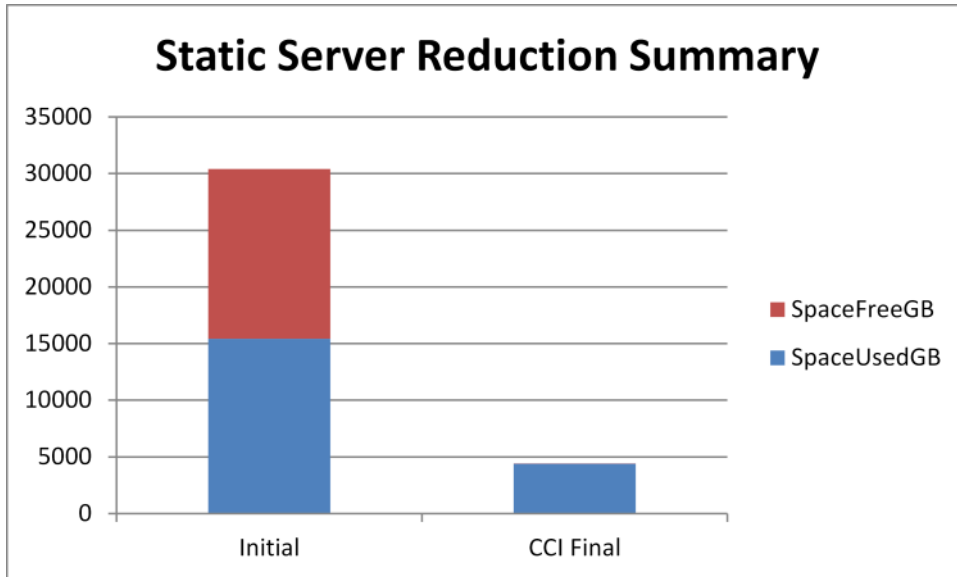
# Summary per database

Tuesday, April 07, 2015 3:29 PM



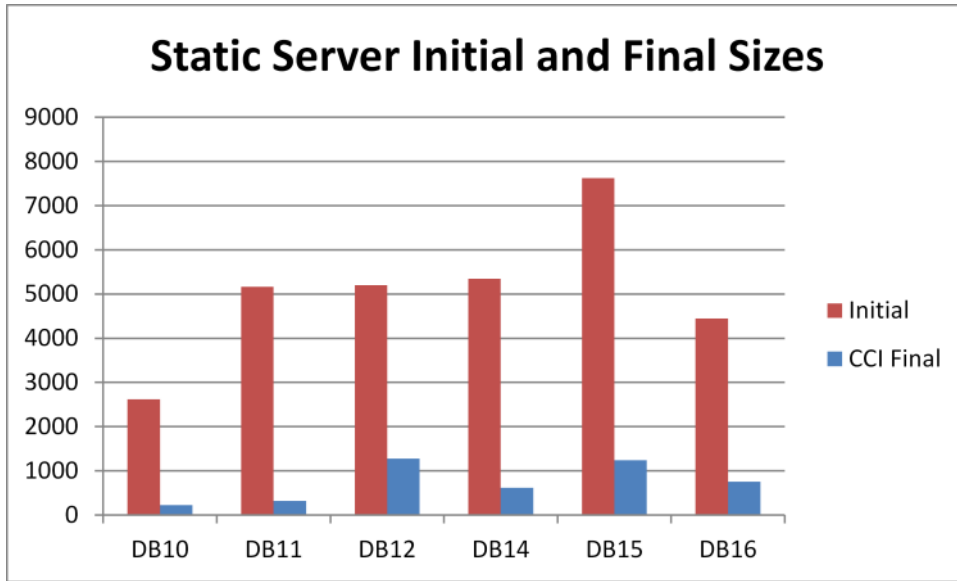
# Comparison Summary

Tuesday, April 07, 2015 3:30 PM



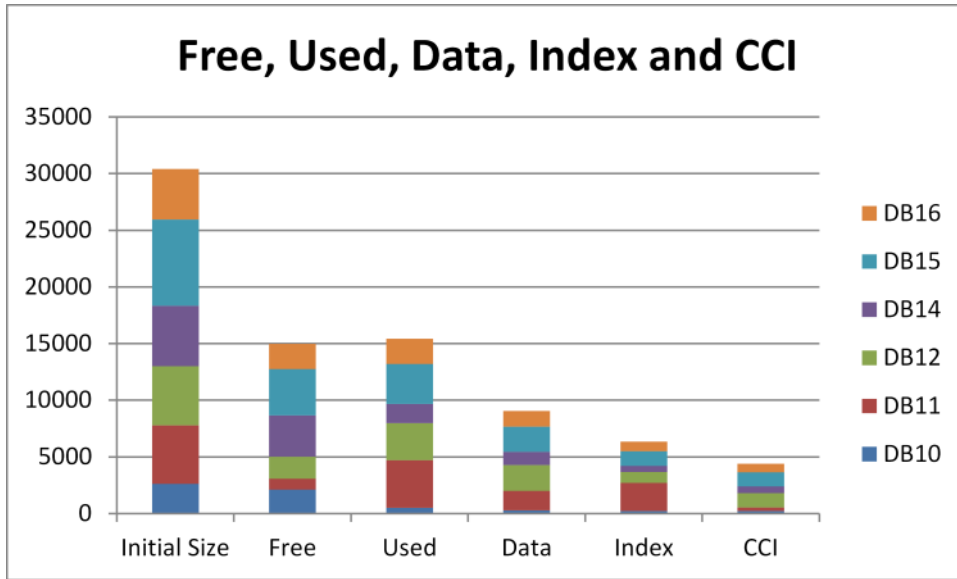
# Per database

Tuesday, April 07, 2015 3:34 PM



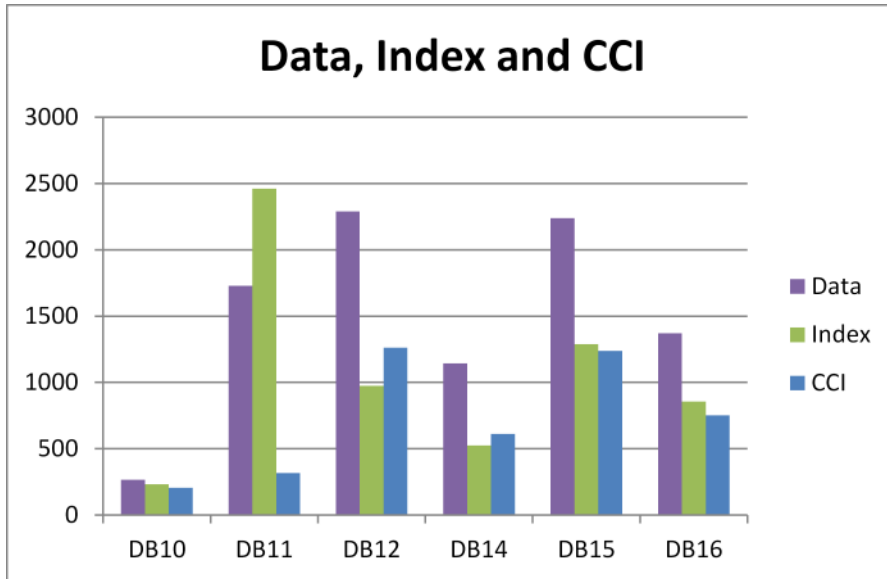
# By free, used, etc.

Tuesday, April 07, 2015 3:35 PM



# By data, index, etc.

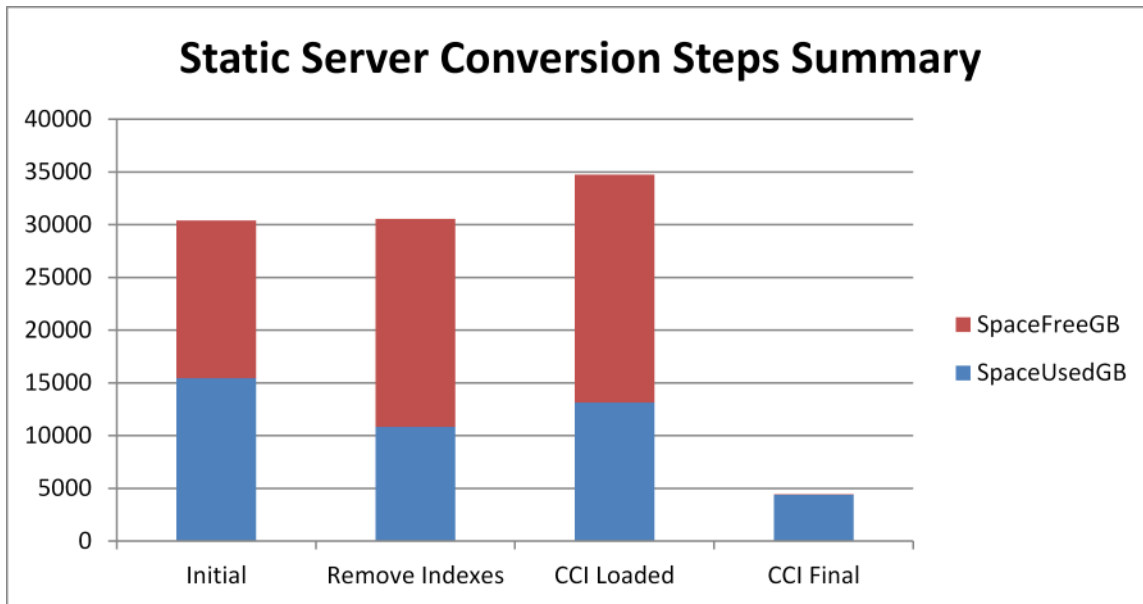
Tuesday, April 07, 2015 3:37 PM





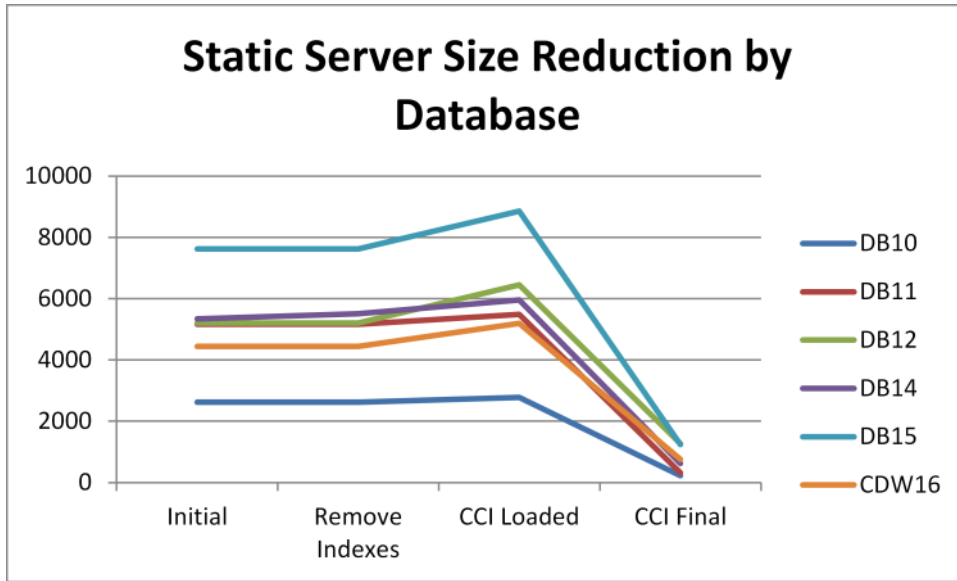
# Conversion Steps Summary

Tuesday, April 07, 2015 3:39 PM



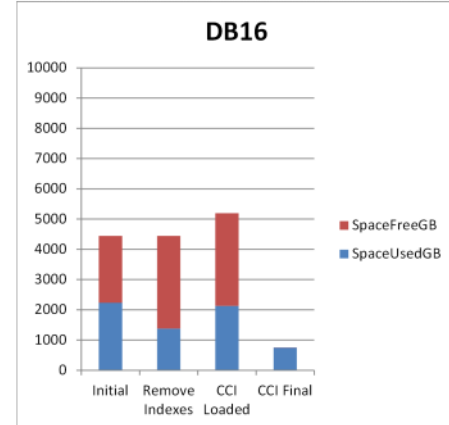
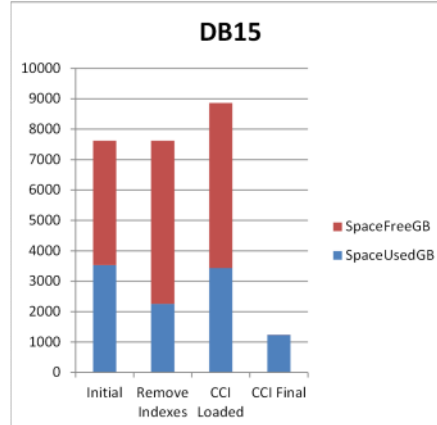
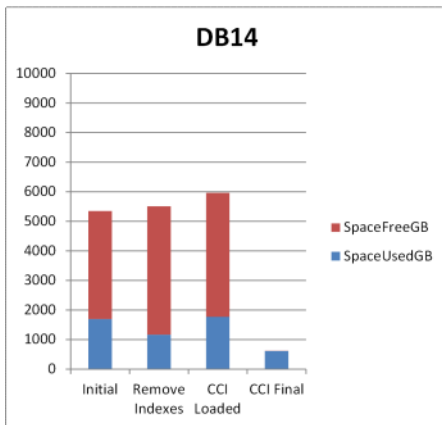
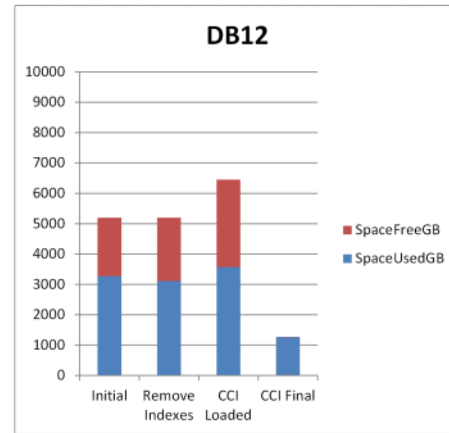
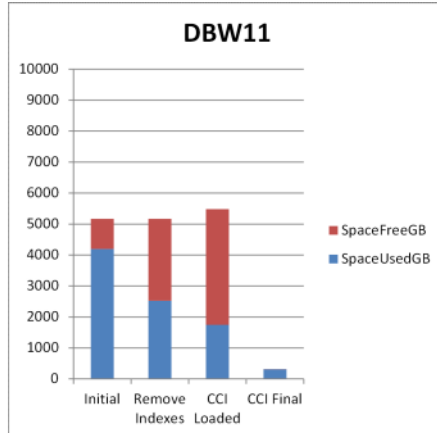
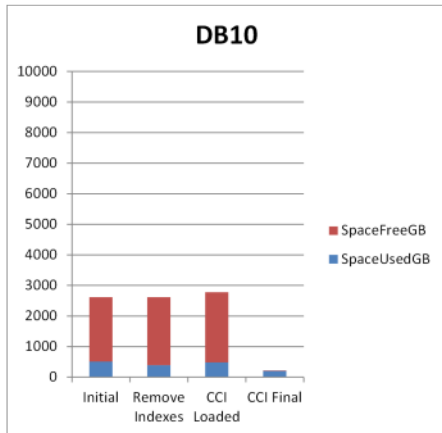
# For all databases - total size

Tuesday, April 07, 2015 3:40 PM



# For all databases - free and used

Tuesday, April 07, 2015 3:42 PM



# Reclaiming Space

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/>

## Supporting utilities:

- [sp\\_WholsActive](#) (Adam Machanic)
- `sp_spaceused2`
  - Uses `sp_spaceused` and other system views to obtain table data
  - Used to capture table sizes and properties at each step of the conversion process
- `sp_spaceused3`
  - Returns log and data information
- `LoadByPartition`
  - Loads each partition into a CCI table
  - Created for each database
  - Handles potential identity columns