#SASGF

VIRTUAL
SAS® GLOBAL FORUM 2021

AMERICAS | MAY 18 – 20   ASIA PACIFIC | MAY 19 – 20   EMEA | MAY 25 – 26

Making SAS Tables Smaller and Faster Without Data Loss

Glen Becker, USAA

Glen Becker has been optimizing SAS data footprints and run times since the early 1980's.   His background is Computer Performance and Capacity Planning, and now serves as an internal consultant helping analysts and modelers save time and space.

Making SAS Tables Smaller and Faster

Overview

- Save space with short numerics
- SAS Compression
- OS-based Compression
- Save space and time with SAS views
- Performance Implications of Short Numerics
  - A surprising technique that can make a SAS table smaller by making some variables larger.
- Zip, gzip, and Zipper

Making SAS Tables Smaller and Faster

Save Space with Short Numerics

- All numeric variables are 8-byte double-precision floating-point in memory
- SAS allows them to be truncated to 3-8 bytes when written to disk. (Mainframe allows 2 bytes)
- Maximum consecutive integer for each length:
  - Distributed: 2 ** (length*8 – 11)
  - Mainframe: 2 ** (length*8 – 8)   or 256 ** (length-1)
  - SAS does NOT warn you about truncation. Be careful!

Making SAS Tables Smaller and Faster

SAS Compression

- Two options: Compress=Yes (Char) or Compress=Binary.
- Compress=Yes changes 4+ identical bytes to 3, like Varchar
- Compress=Binary is for sparse or repetitive numeric data (70% vs 34%), but slower.
- Changing nulls to zeros with Compress=Yes is smaller and faster.
- New VARCHAR type converted to fixed-length on disk.
- See "Compare Compression Types"

Making SAS Tables Smaller and Faster

OS-based Compression

- Transparent compression: set and forget
  - OS compression + SAS compression better than either alone
- Windows: NTFS compression
  - File manager: Right-click, Properties, Advanced, check "Compress contents to save disk space"
  - Command line "Compact" command
  - Your site may use other filesystems that do allow this
- Not yet generally available on Linux

Making SAS Tables Smaller and Faster

Save Space and Time with SAS Views

- What views are:
  - An SQL view is a SELECT statement
  - A Data step view is a program that runs as the view is read
- A simple view can save storing copies of data.
- Improve performance by not writing and re-reading data to/from disk. View does derivations on-the-fly.
- Layers of views are cheap.
- See "View example"

Making SAS Tables Smaller and Faster

Performance Implications of Short Numerics

- SAS physically arranges numeric variables in a different order than their logical order:
  - Length 8 are first, on double word alignment.
  - Length 4 are next, on single word alignment.
  - Lengths 3, and 5-7 are last, in VARNUM order no alignment.
- CPUs are faster with word-aligned storage.
  - Numeric lengths 4 or 8 are fastest. See “Alignment example”
- Zero-loading: best compression if variables most likely to be zero are bunched together. See “Demo zero-loading”

Making SAS Tables Smaller and Faster

Use ZIP Technology to Compress Better

- Zip the physical files with “zip” or “gzip”
  - A SAS table is physically memname.sas7bdat, where memname is the SAS member name in lower case.
  - If the table has an index, there is also memname.sas7bndx
  - If the table has extended attributes, memname.sas7bxat.
  - On zOS, library must be HFS.
  - Problem: to use the data, you must completely unzip it.
- The “Zipper” macro zips SAS data, and builds a SAS data view that reads it on-the-fly. XCMD not req’d.

Thank you!

Glen.Becker@USAA.com