A New and Better Way to Compress FITS Binary Tables

W. D. Pence, NASA/GSFC
R. Seaman, NOAO
R. L. White, STScI

Summary: The size and number of FITS binary tables that are generated by modern astronomical observatories is growing rapidly, for example, in the form of photon event lists and in mega- and giga-sized object catalogs. The disk space and network bandwidth needed to archive and download these tables can be significantly reduced by using the best available compression techniques. We propose a new FITS binary table compression method that is modeled after the FITS tiled-image compression convention that has been in use for the past decade. Preliminary tests show that this new table compression technique generally works much better than the commonly used method of simply gzipping the whole FITS file. On average, it results in 1.6 times more compression and saves an additional 20% in disk space.

A. How does this new compression method work?

1. Divide the input binary table into tiles, each containing the same number of rows (except perhaps the last tile). Tables smaller than ~10 MB may be compressed as a single tile.

2. Transpose the rows and columns in each tile. Compressing the table column-by-column rather than row-by-row almost always results in higher compression.

3. Compress each column with a suitable algorithm that may be optimized for the particular type of data in that column.

4. Store the compressed bytes in the corresponding column of the output compressed table (as a variable length array).

5. Copy all the header keywords from the input to the compressed output table. (Keywords remain uncompressed).

B. What compression algorithms are supported?

In principle any compression algorithm can be supported. So far, we have evaluated the following algorithms:

Gzip – This is the default algorithm that works reasonably well on all types of table columns.

Shuffled GZIP – Numeric columns (integer or floating point) can often be more highly compressed if the bytes in the values are sorted into decreasing order of significance before compressing them with gzip. For example, the bytes in 2 integer*4 values would be shuffled as:
A1 A2 A3 A4 B1 B2 B3 B4 –> A1 B1 A2 B2 A3 B3 A4 B4

Rice – For integer type columns. Our tests show that Rice is particularly effective on integer*4 “J” columns.

Bzip2 – preliminary tests show that bzip2 is ~10 times slower than gzip, while only producing a modest gain in compression, hence bzip2 does not appear suitable as a general purpose algorithm.

C. What are the advantages over just gzipping the whole FITS file?

1. This new method generally produces higher compression. Preliminary tests (see figures) on a sample of FITS tables shows that on average it produces 1.6 times higher compression and recovers 20% more disk space (relative to the original file size).

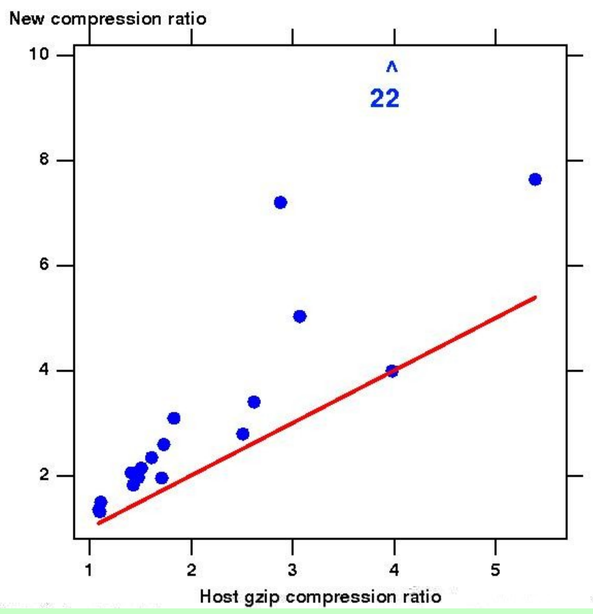
2. The compressed binary table is itself a valid FITS binary table and thus may be directly manipulated by general FITS file utility programs.

3. The header keywords remain uncompressed for rapid read and write access.

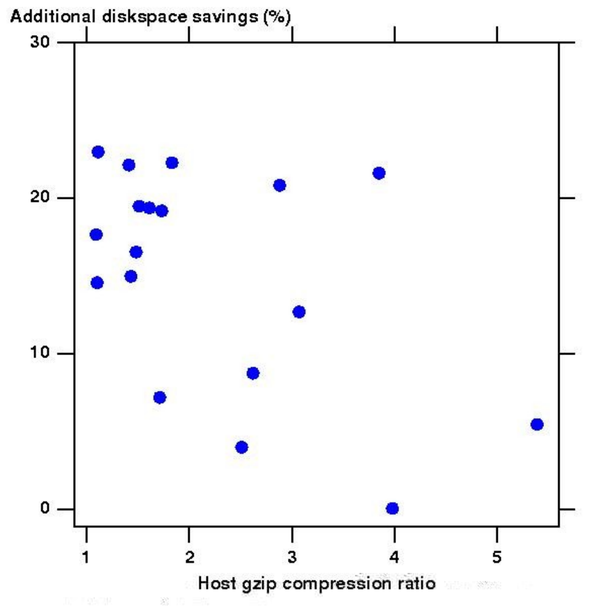
4. Each table in a multiple-HDU FITS files is compressed separately, so the whole FITS file does not need to be uncompressed to view a single table.

5. Compressing the table as a series of tiles in principle allows rapid access to a given row in the table, without having to uncompress the whole table.

6. By re-compressing the compressed binary table multiple times, infinite compression can be achieved.☺



Comparison of the compression ratio when externally compressing the whole file with gzip (X axis) versus our new compression method (Y axis). On average, the new ratio is 1.6 times larger than when just using gzip to compress the file.



Amount of addition disk space that would be saved using this new compression method instead of simply gzipping the file, expressed as a percentage of the original file size.

D. Other considerations

Currently, this compression method only applies to binary tables and not to FITS ASCII tables.

Since the minimum possible size of a compressed FITS table is 2880 bytes (1 FITS block), this compression method is not useful for binary tables that are smaller than this.

Since the header keywords remain uncompressed (which is generally a good thing!) this compression method will be less effective in cases where the size of the FITS table is dominated by the header keywords.

E. Future development

Further testing needs to be done before making this new compressed FITS table format available for general use. Most importantly, we need to verify and optimize this compression method using a much larger sample of FITS binary tables.

We need your help! Please let us know about other suitable FITS tables from your own project that we can use for testing.

Additional Resources

Tiled-image compression convention:
http://fits.gsfc.nasa.gov/registry/tilecompression.html

Tiled-table compression convention:
http://fits.gsfc.nasa.gov/tiletable.pdf

Fpack and funpack FITS image compression tools:
http://heasarc.gsfc.nasa.gov/fitsio/fpack/