NASA EOS DATA COMPRESSION WITH HDF5 SCALEOFFSET FILTER

Muqun Yang, Fang Guo, Xiaowen Wu, Robert E. McGrath, Quincey Koziol

National Center for Supercomputing Applications
University of Illinois, Urbana-Champaign

What is the ScaleOffset filter?

- ScaleOffset compression performs a scale and/or offset operation on each data value and truncates the resulting value to a minimum number of bits before storing it. The datatype is either integer or floating-point.
- Offset in Scale-Offset compression means the minimum value of a set of data values.
- If a fill value is defined for the dataset, the filter will ignore it when finding the minimum value.

How does ScaleOffset work?

An example for Integer Type

Dataset: 32-bit integer array
Maximum is 7065; Minimum is 2970
The "span" of dataset values
= Max-Min+1 = 4076

Case 1: No fill value is defined.
Minimum number of bits per element to store
= ceiling(log2(span)) = 12

Case2: Fill value is defined in this array.
Minimum number of bits per element to store
= ceiling(log2(span+1)) = 13

Compression:
1. Subtract each element from minimum value.
2. Pack all data with minimum number of bits.

Decompression:
1. Unpack all data.
2. Add each element to minimum value.

Outcome:
It can save about 60% disk space for this case.

An example for Floating-point Type

D-scaling factor: the decimal precision to keep for the filter output
Floating-point data: {104.561, 99.459, 100.545, 105.644}
D-scaling factor: 2

Preparation for Compression

Step 1: Determine the minimum value = 99.459
Step 2: Subtract the minimum value from each data element
Modified data: {5.102, 0, 1.086, 6.185}
Step 3: Scale the data by multiplying 10 ^ D-scaling factor = 100
Modified data: {510.2, 0, 108.6, 618.5}
Step 4: Round the data to integer
Modified data: {510 , 0, 109, 619}

Compression and Decompression: using ScaleOffset for integer

Restoration after decompression

Step 1: Divide each value by 10^ D-scaling factor
Step 2: Add the offset 99.459
The floating point data {104.559, 99.459, 100.549, 105.649}

Outcome:
Produces lossy compression.
Compression ratio will depend on D-scaling factor.

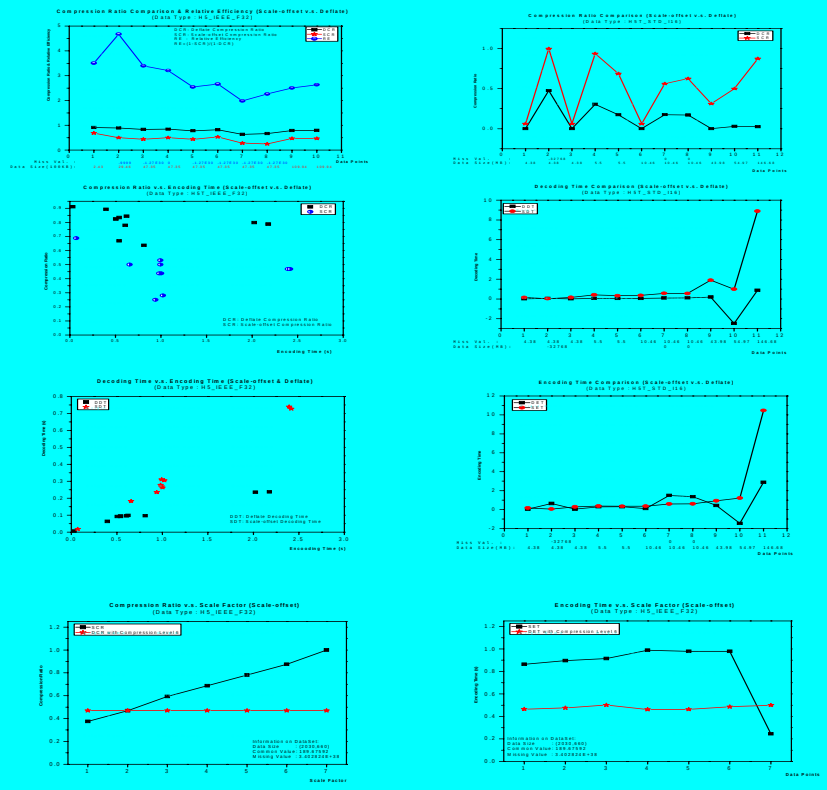
Why use ScaleOffset filter?

- Internal HDF5 filter
- Easy to understand
  - Integer: lossless
  - Floating-point: GRIB lossy compression
- Easy to control floating compression
  - D-scaling factor
- Easy to estimate the outcome

Limitations

- Compressed floating-point data range is limited by the size of corresponding unsigned integer type.
- Long double is not supported.

EOS Data Compression Performance Comparison ScaleOffset vs. Deflate Filter



Conclusions

- ScaleOffset Filter can gain better compression ratio for floating-point data.
- ScaleOffset Filter can use more encoding time and decoding time.
- Knowledge of data characteristics benefits the usage of ScaleOffset filter.

This work was funded by the NASA Earth Science Technology Office under NASA award AIST-02-0071 and UCAR Subaward No S03-38820. Other support was provided based upon the Cooperative Agreement with the National Aeronautics and Space Administration (NASA) under NASA grant NNG05GC60A. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NASA.

For more information or documentation, contact hdfhelp@ncsa.uiuc.edu