Converting GRIB to netCDF-4

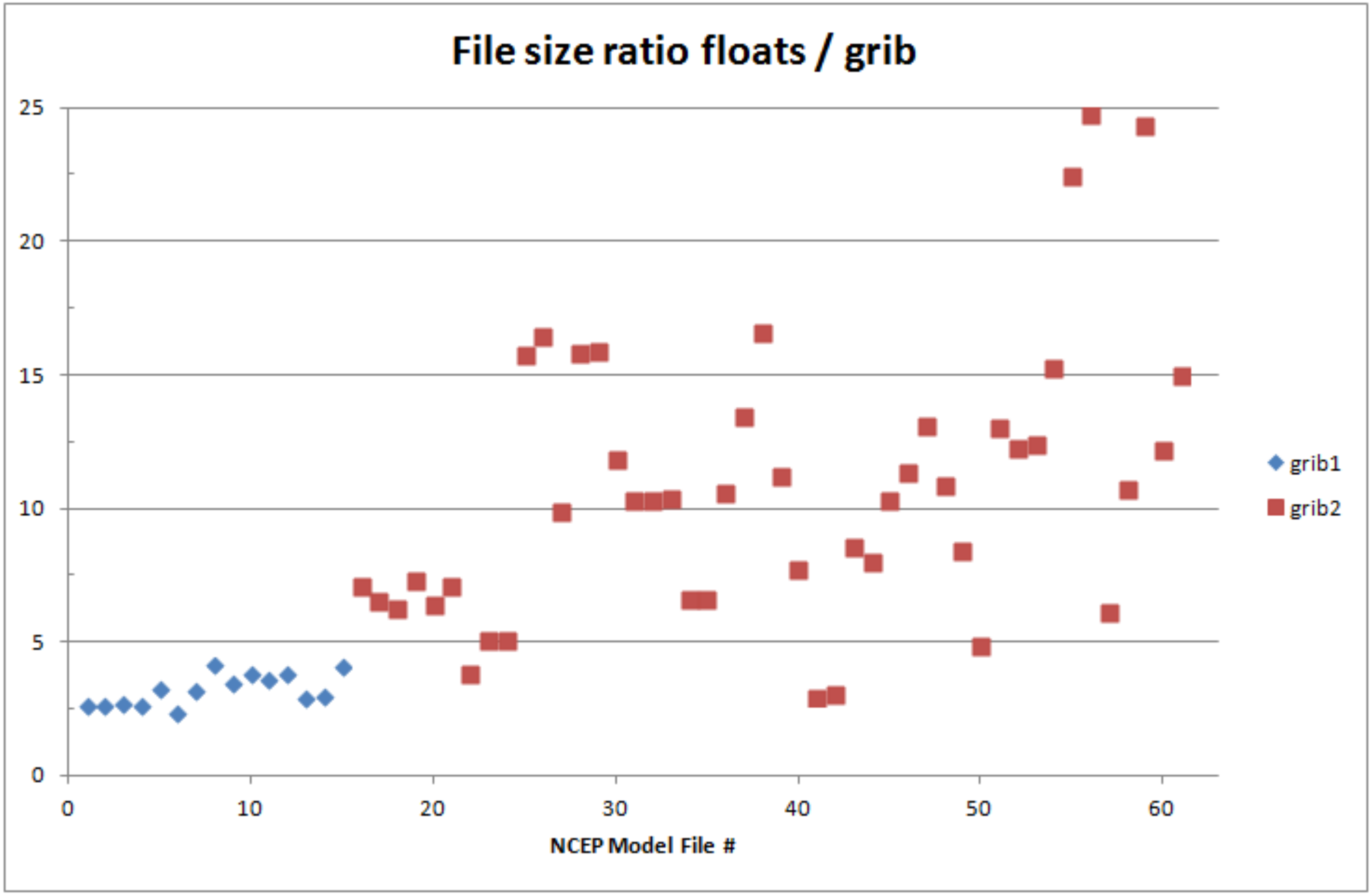
Compression studies

John Caron, UCAR/Unidata

Sep 25, 2014

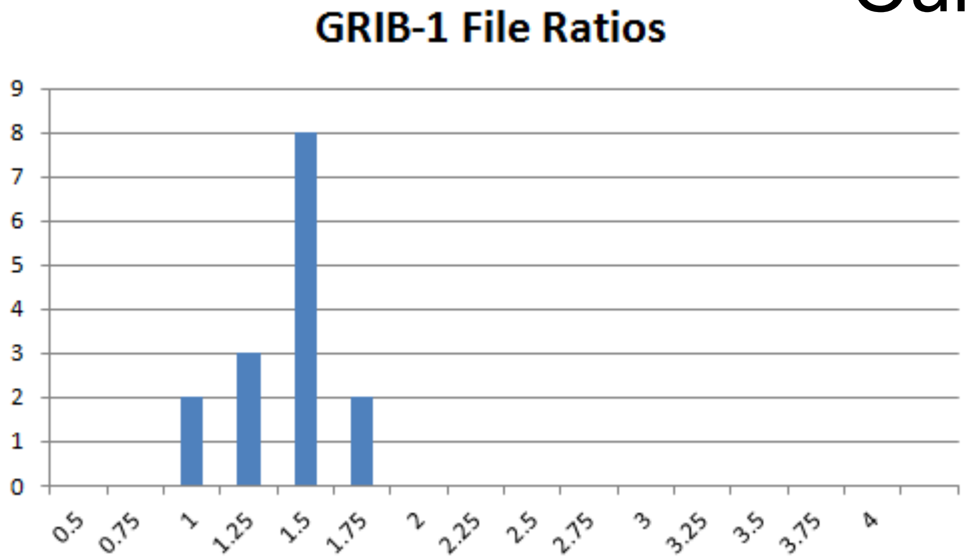
GRIB floating point compression

- GRIB uses lossy compression of floating point data
  - store integers; f = i*scale + offset
  - Bounds the absolute precision : abs(f_{org}-f) <= scale/2
- GRIB-1 uses bit-packing
- GRIB-2 uses JPEG-2000 wavelet compression
- GRIB has excellent compression
  - On our test NCEP data, GRIB is 2.5-25x smaller than uncompressed single precision floating point, eg netCDF-3
  - Recent NCEP model runs (15 Grib-1, 46 Grib-2, 26 Gbytes)
- Can netCDF-4 get close to this?
  - JPEG-2000 considered patent encumbered (?)
  - What about other compression?



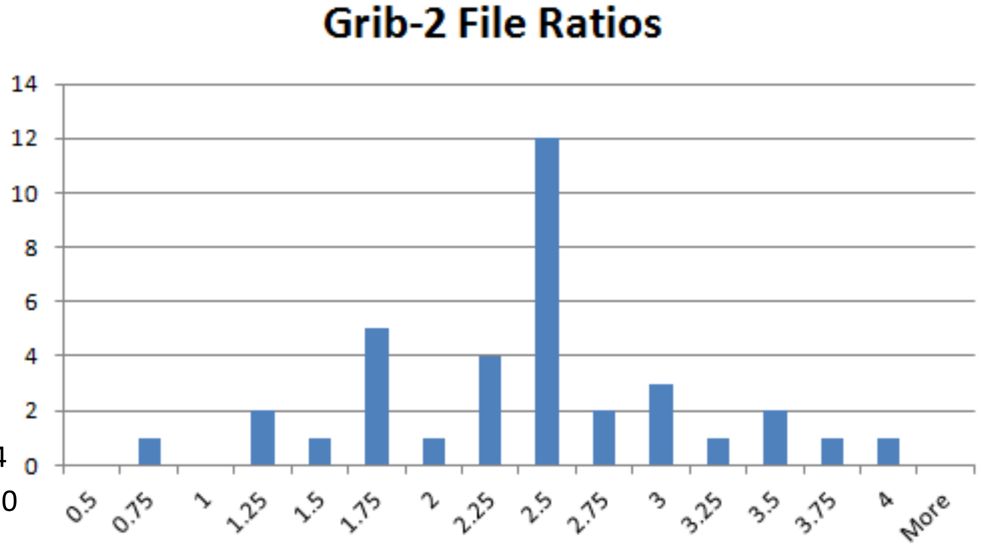
average = 8.9
stdev = 5.5

Current netCDF-4 (deflate) ratio netCDF4 / GRIB



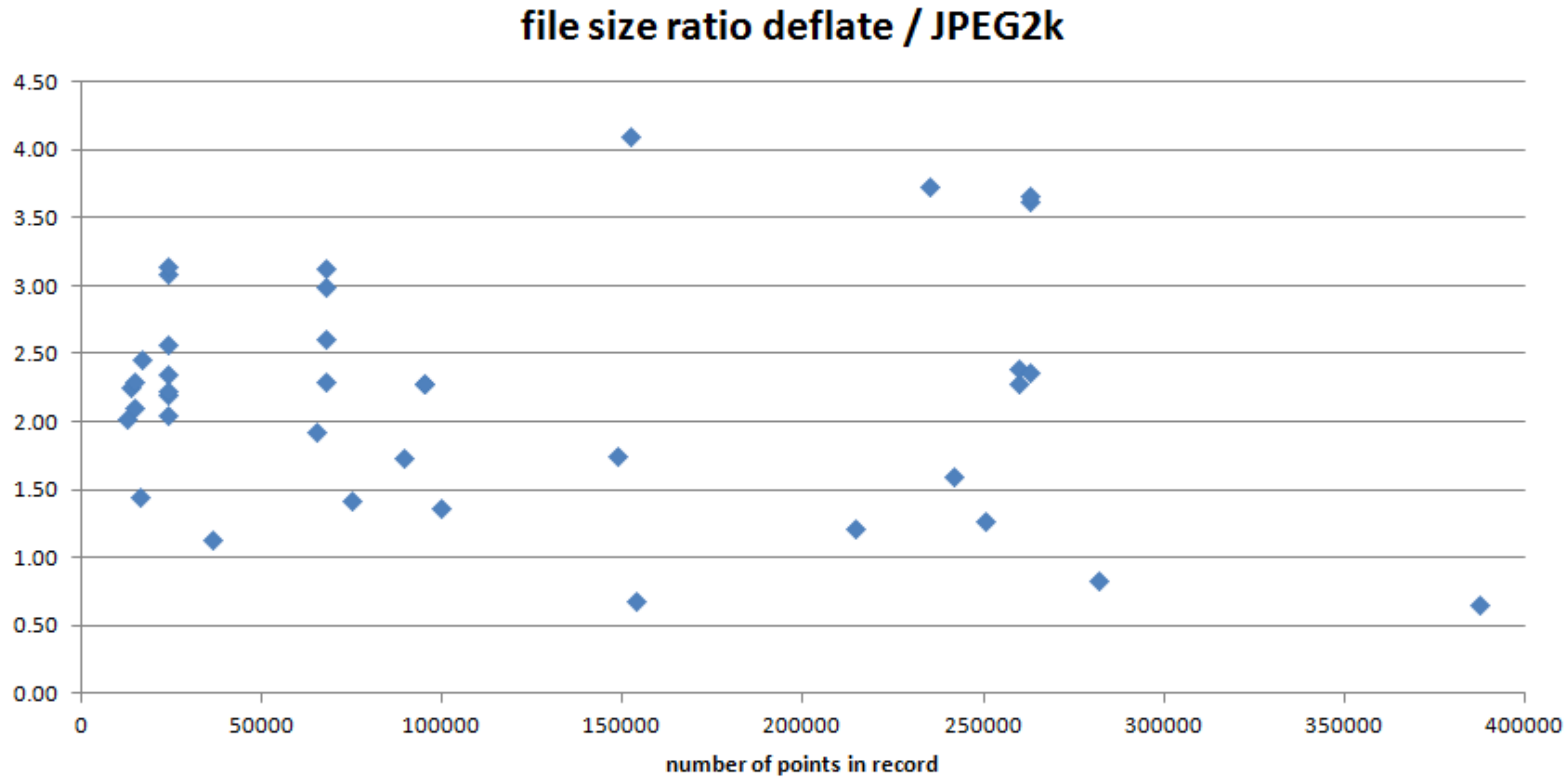
Weighted average ratio = 1.32

Total # grib1 records = 24,933



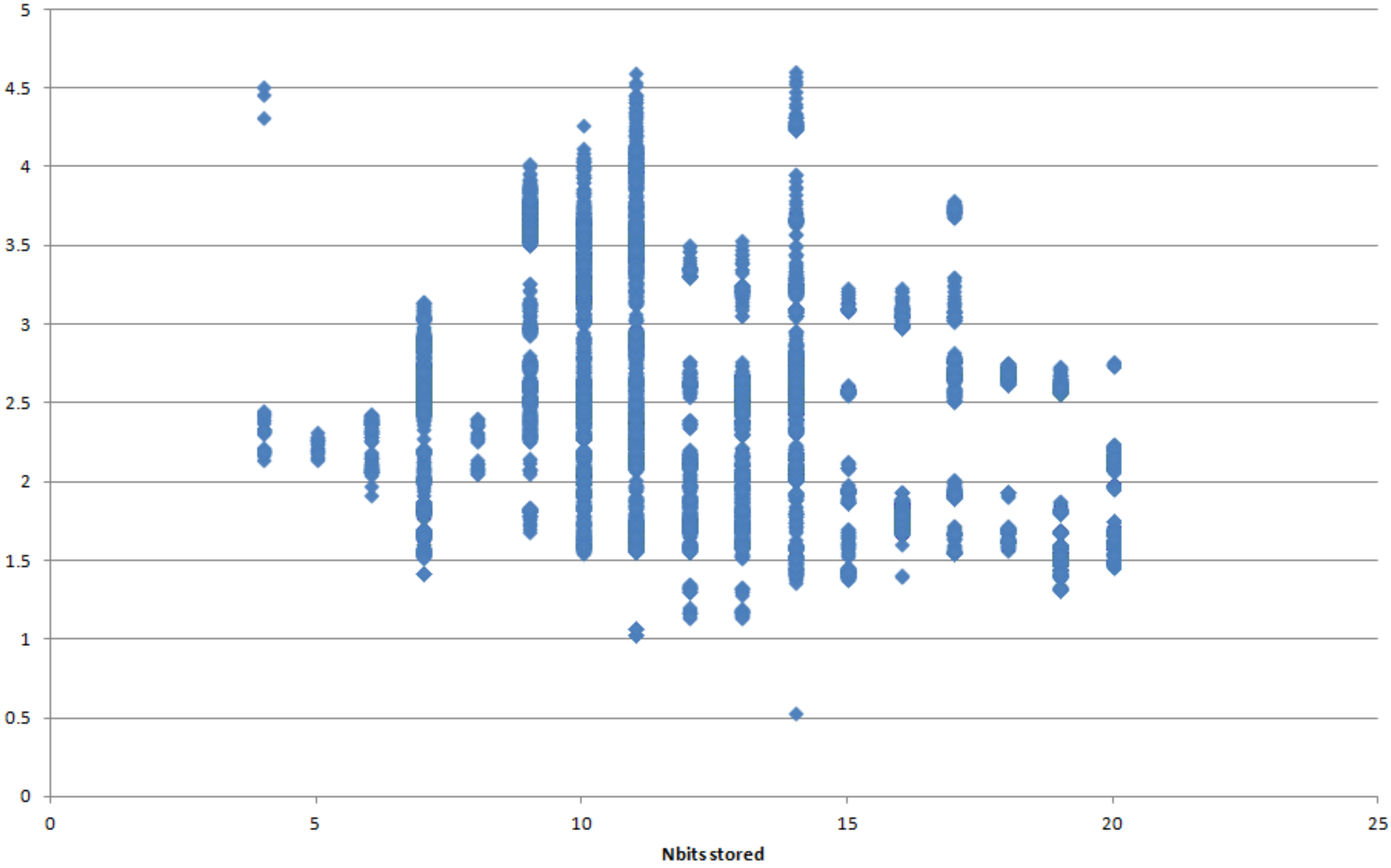
Weighted average ratio = 2.24

Total # grib2 records = 375,470



file size ratio deflate / JPEG2k
4.50
4.00
3.50
3.00
2.50
2.00
1.50
1.00
0.50
0.00
0
50000
100000
150000
200000
250000
300000
350000
400000
number of points in record

GFS_Global_0p5deg
ratio deflate / JPEG2k



Other possibilites

Other compression algorithms

- bzip2
- LZMA (7zip)

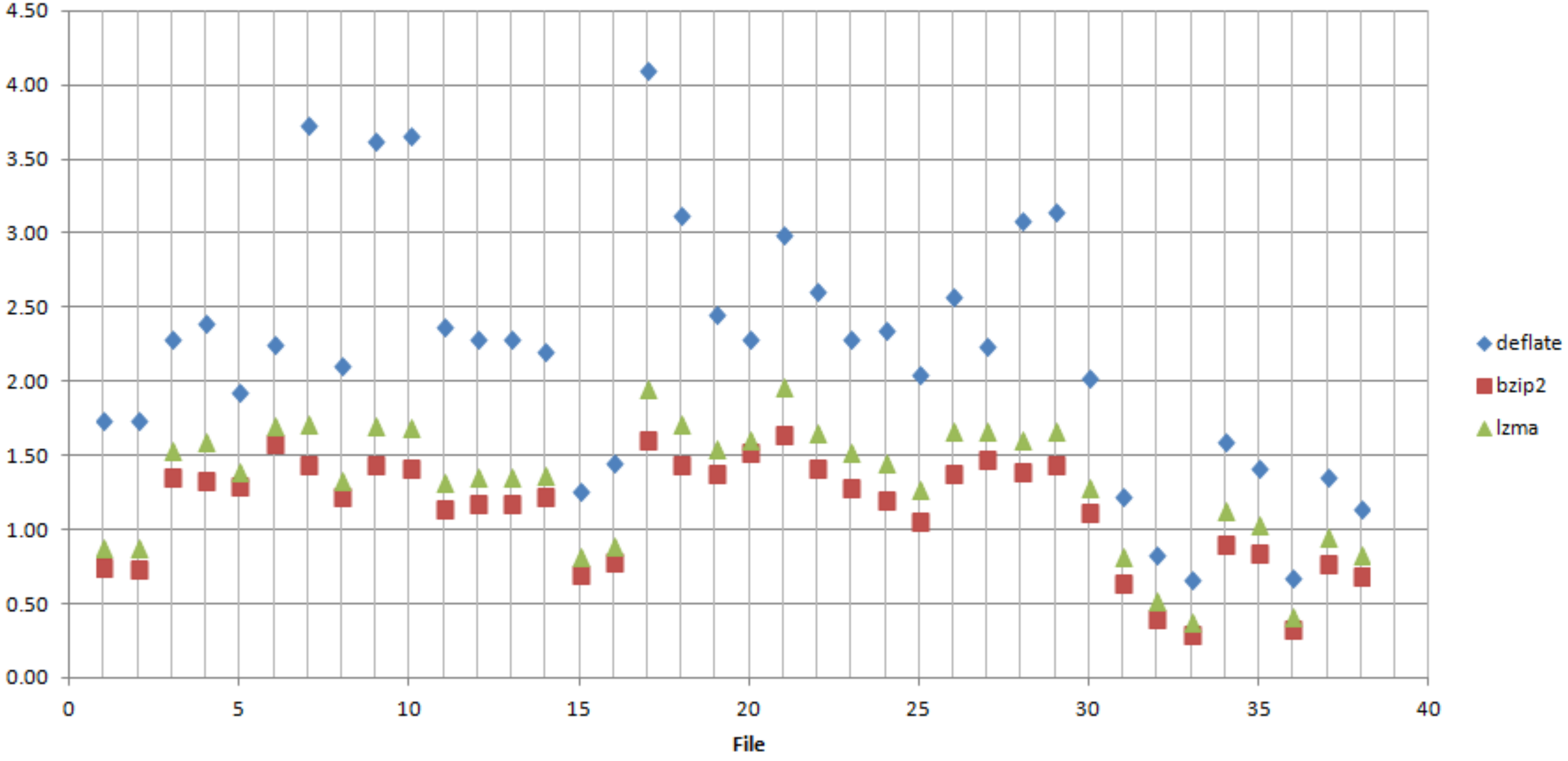
Lossy compression techniques

- bit shaving (set low order bits to 0)
- scale/offset (turn floats into ints)

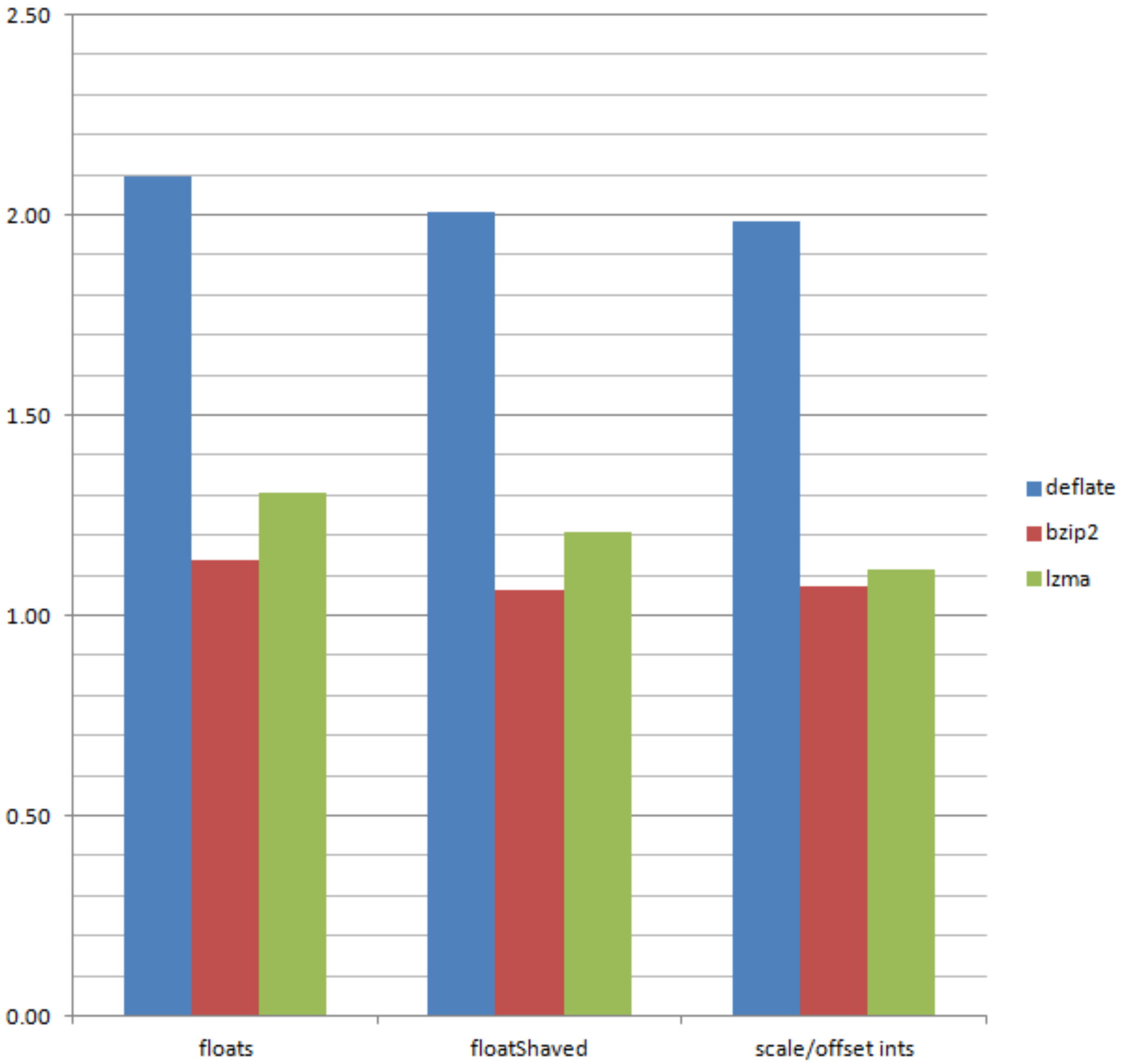
Testing methodology

- all in Java :expect to be good estimate of C library
- read GRIB, use Java compression libraries
    - floats as they are returned from GRIB reader (limited precision)
    - floatShaved: use Nbits from GRIB, set lower bits to 0
    - ints: use exact same integer array as GRIB

File size ratio with GRIB2 JPEG2k On limited precision floats (Java)

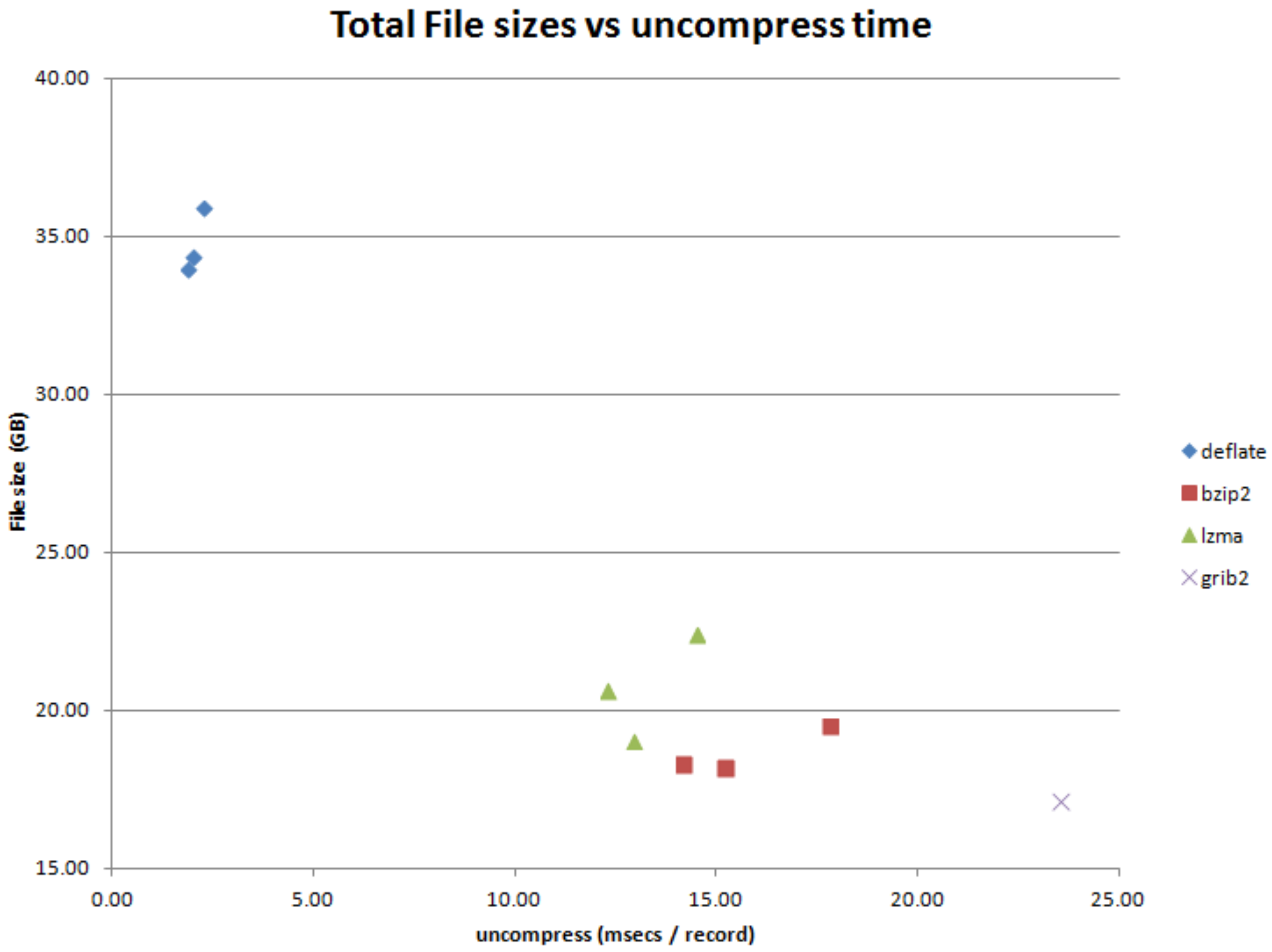


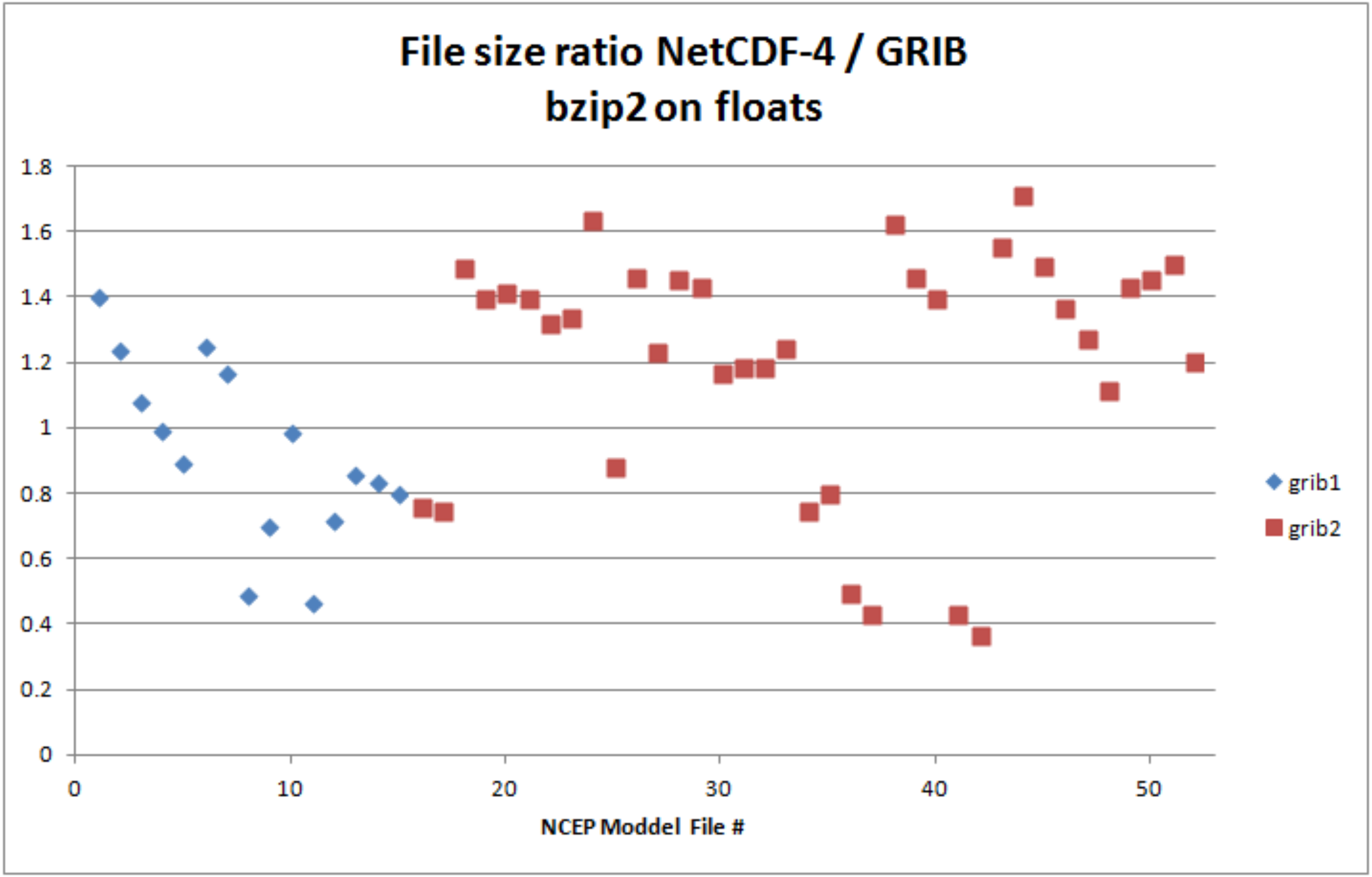
Total file sizes ratio with GRIB2 JPEG2k (Java)



Total File Sizes
Average times (millisecs)

	size (GB)	uncompress	compress
deflate floats	35.90	2.28	14.71
deflate floatShaved	34.38	1.98	13.59
deflate ints	33.98	1.89	11.96
bzip2 floats	19.50	17.80	55.84
bzip2 floatShaved	18.18	15.20	48.86
bzip2 ints	18.32	14.17	43.09
lzma floats	22.40	14.50	473.19
lzma floatShaved	20.64	12.31	454.08
lzma ints*	19.05	12.94	482.02
grib	17.12	23.53	


Total File sizes vs uncompress time
40.00
35.00
30.00
25.00
20.00
15.00
File size (GB)
0.00
5.00
10.00
15.00
20.00
25.00
uncompress (msecs / record)
◆ deflate
■ bzip2
▲ lzma
× grib2



	avg	stdev
total	1.12	0.36
grib1	0.92	0.27
grib2	1.20	0.37

Conclusions

- On NCEP Model GRIB files "limited precision" floats
  - Bzip2 can get to within 20% of GRIB on average
  - Ratios of bzip2/grib vary between .4 and 1.7
- Bzip2 looks like a good candidate to add as a standard compression option in netCDF-4
  - tradeoff files size and un/compress times
- We are considering a "lossy compression" option in netCDF-4 using bit shaving and/or scale/offset
  - expect bzip2 within 10% of GRIB-2 JPEG-2000
- Possible utility to copy GRIB to netCDF-4 and get the exact floating point numbers back
- Other compression options still to explore
  - fpzip, zfp from Peter Lindstrom at LLNL
  - ??