Compression Performance Evaluation Report

0. Purposes:

- To evaluate bzip2 compression on real NASA data by comparing the performance with gzip compression
- To check the possibility of integrating bzip2 to HDF5

1. What's bzip2?

Burrow-wheeler block-sorting text compression [1] + Huffman coding [2] lossless compression package

1) Utility:

- Compression: bzip2 original_name
- Decompression: bunzip2 or bzip2 –d original_name.bz2
- error detection: 32-bit CRC, only tell you something is wrong
- Depression level 1-9 (like gzip: from fast to best)

2) Library:

- Interfaces similar to zlib library, including low-level, high-level and utility functions
- Low-level interface: thread-safe

3) Features according to the author:

- Files containing long runs of repeated symbols may compress more slowly
- May perform best on machines with very large caches
- Handle errors gracefully

4) Misc:

- Not use autoconf

5) Future work according to the author:

- In the library interface, one parameter called "working factor" should be adjusted by the library automatically instead by the user application. The author may get rid of this parameter by making changes in the library in the future.

For more information, check the bzip2 web page at [3].

2. Users' point of view of bzip2

According to one user's email:

- Compression time is slower than gzip
- 5% better compression ratio than gzip

3. Some definitions in this report

1) ***Compression ratio***: The ratio of the compressed file size or array size to the original file size or array size.

2) ***Encoding time of the utility***: Elapsed time counted from starting the process of compressing the file until the end of this process

3) ***Decoding time of the utility***: Elapsed time counted from starting the process of decompressing the file until the end of this process

4) ***Encoding time of the library***: Difference of the elapsed time between writing an HDF5 dataset with compression and without compression

5) ***Decoding time of the library***: Difference of the elapsed time between reading an HDF5 dataset with compression and without compression

Note:

- "time" utility is used to calculate encoding time and decoding time of the utility
- gettimeofday is used to calculate encoding and decoding time of the library

4. Data

1) NASA data with HDF4 to HDF5 converter utility

We are using semi-real NASA data to do performance analyses. Since the current NASA EOS data are all stored in HDF4 format, to do the performance analyses in HDF5, we use NCSA H4toH5 converter utility [4] to convert all NASA data from HDF4 format to HDF5 format. With the rough comparison of file size between the converted HDF5 files and the original HDF4 files, we find the converted HDF5 files are reliable.

2) Detailed data information

Based on about 30 real data samples, we choose 10 samples. These data are SSMI, CERES, TOMS, TRIM, MODIS, MISR, ASTER and LANDSAT products. The file size and main data types of the file are listed in the table 1. Since each file may include several arrays and meta data. So we only list the percentage of data types of those arrays that are over 1% of the whole file size. If several arrays share the same data type, we add those array sizes up and then calculate the overall percentage.

Table 1: File information of the experiment

File name	**File Size (Unit: MB)**	**Data type**
SSMI	2.02737	Unsigned 8-bit big-endian integer (97.8%)
TOMS	6.093707	Unsigned 16-bit big-endian integer (97.4%)
TRIM	13.69254	Unsigned 16-bit big-endian integer (59.6%) IEEE 32-bit float (35.6%) Struct (4.4%)(see note [1])
CERES1	22.77592	IEEE 32-bit float (98.0%) 32-bit big-endian integer (1.2%)
MISR	70.01059	Unsigned 16-bit big-endian integer (80.3%) IEEE 32-bit float (19.4%)
CERES2	72.66951	IEEE 32-bit float (98.5%) 32-bit big-endian integer (1.2%)
ASTER2	74.94336	Unsigned 16-bit big-endian integer (93.2%) Unsigned 8-bit big-endian integer (l6.7%)
ASTER1	118.6585	Unsigned 8-bit big-endian integer (91.3%) Unsigned 16-bit big-endian integer (4.8%)
MODIS1	262.343	Unsigned 16-bit big-endian integer (56.0%) Unsigned 8-bit big-endian integer (36.0%) IEEE 32-bit float (8.0%)
LANDSAT	561.8911	Unsigned 8-bit big-endian integer (99.98%)

Note 1: The Struct is the mixing of float, short and char.

5. Utility performance analysis result

- Platform independent (SGI O2K, windows 2000, Linux 2.2.18, solaris 2.7). We find stronger similarities among all four platforms. For elapsed encoding and decoding times: Windows is the best and SGI O2K is the worst. Compression ratio is exactly the same (should be! Even one byte should not be wrong). In the following, only use charts from linux running to show typical results.

- Bzip2 can always give a better compression ratio, from 0.1% to almost 20%.

- Bzip2 is almost always taking longer for decoding and encoding the data, especially the decoding time is much longer for all data samples.

- Compression ratio is better with the increasing of compression level for gzip under all cases. However, the improvement is small.

- Compression ratio is generally better with the increasing of compression level for bzip2. However, for some cases (see Appendix), the compression ratio becomes worse.

- Decoding time is not sensitive to different compression levels for both gzip and bzip2, which should behave like this according to the theory

- Encoding time is worse with the increasing of compression level for both compression packages. Encoding time is more sensitive for gzip than for bzip2. In fact, gzip level 9 encoding time for MODIS file is even longer than bzip2 level 9 for MODIS file.

- Both bzip2 and gzip are not impressive for floating point compression. Bzip2 gains little for floating point data compression ratio, however, it takes much longer decoding time.
- For ASTER2, both bzip2 and gzip compress the file very well, with the compression ratio around 0.11. However the encoding time for both packages is longer. We check the data value of several arrays of the file and find many 0 value in the file. That may be the reason that gives both packages good compression ratio and longer encoding time.

The following six figures will show comparisons of compression ratio, encoding time and decoding time in detail. The first three figures show the compression ratio, encoding time and decoding time for the whole 10 EOS files with level 1, 6 and 9 of bzip2 and gzip; the second three figures show those for the first five EOS files in table 1 for better resolution.

Figure 1: Compression ratio comparision between gzip and bzip2 when compressing all ten files(linux 2.2.18)

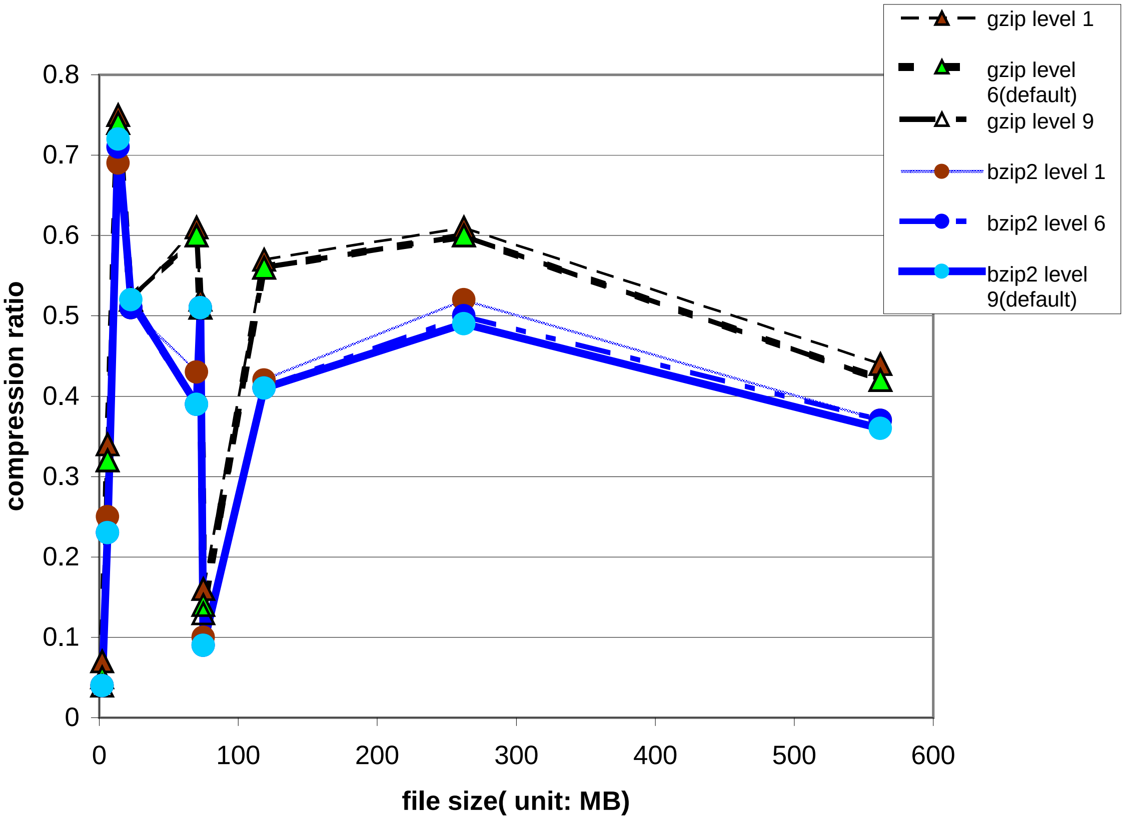


Figure 2: Encoding time comparision between gzip and bzip2 when compressing all ten files (linux 2.2.18)

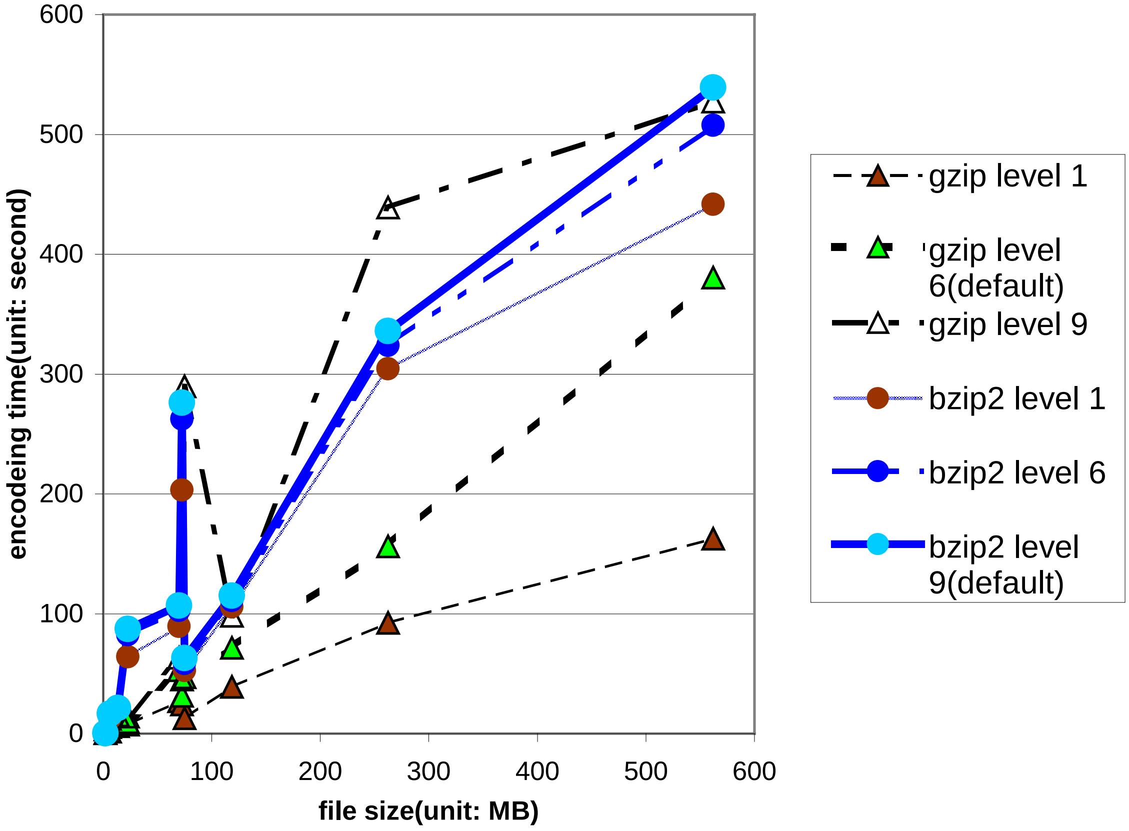


Figure 3: Decoding time comparision between gzip and bzip2 with all ten files(Linux 2.2.18)

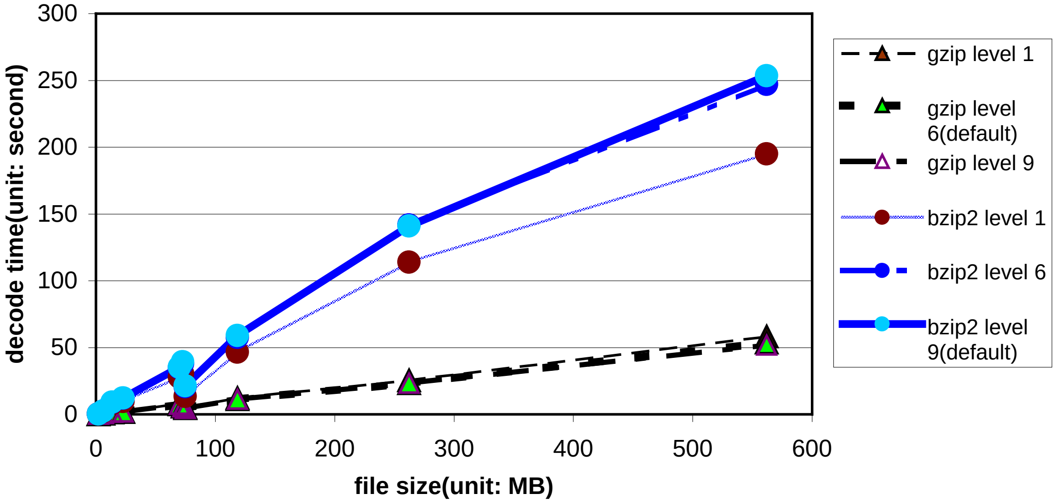


Figure 4: Compression ratio comparision between gzip and bzip2(linux 2.2.18: the first five files)

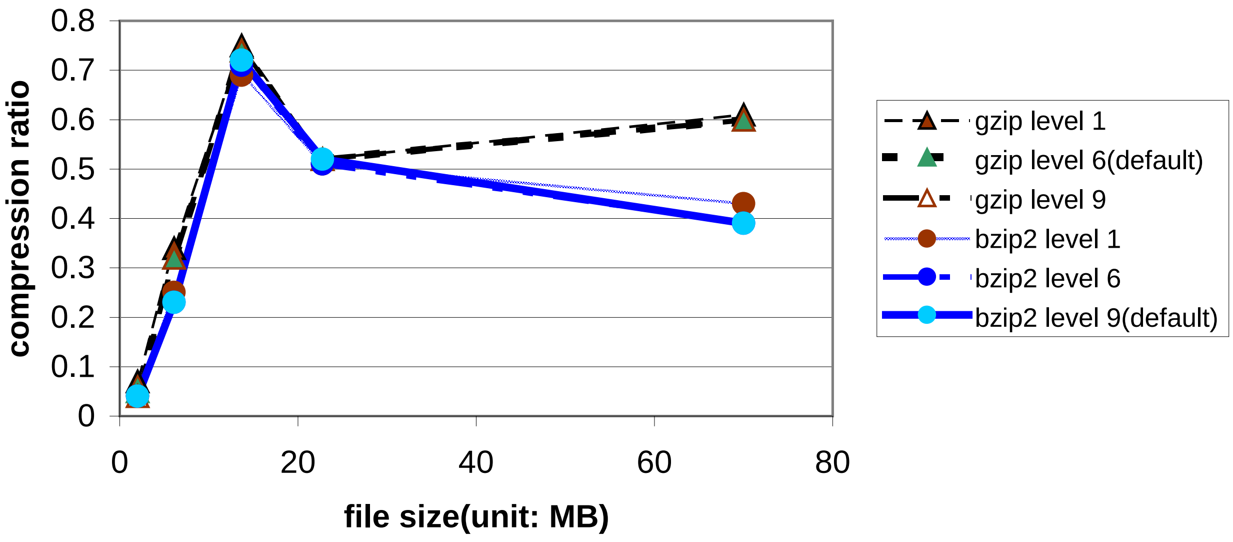


Figure 5: Encoding time comparision between gzip and bzip2 (Linux 2.2.18: the first five files)

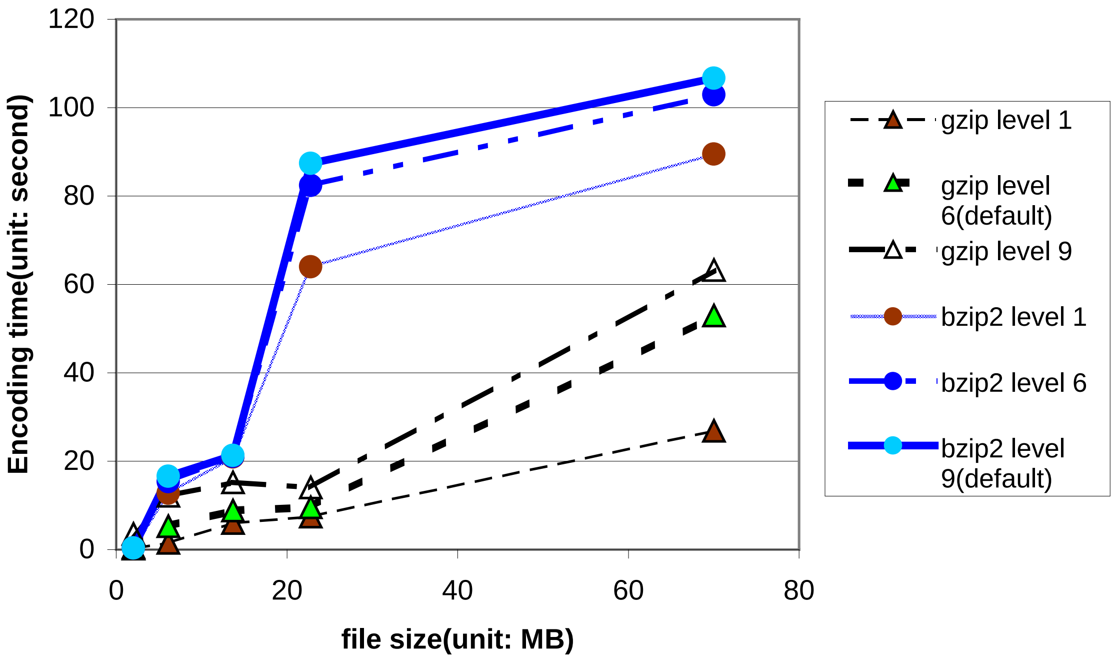
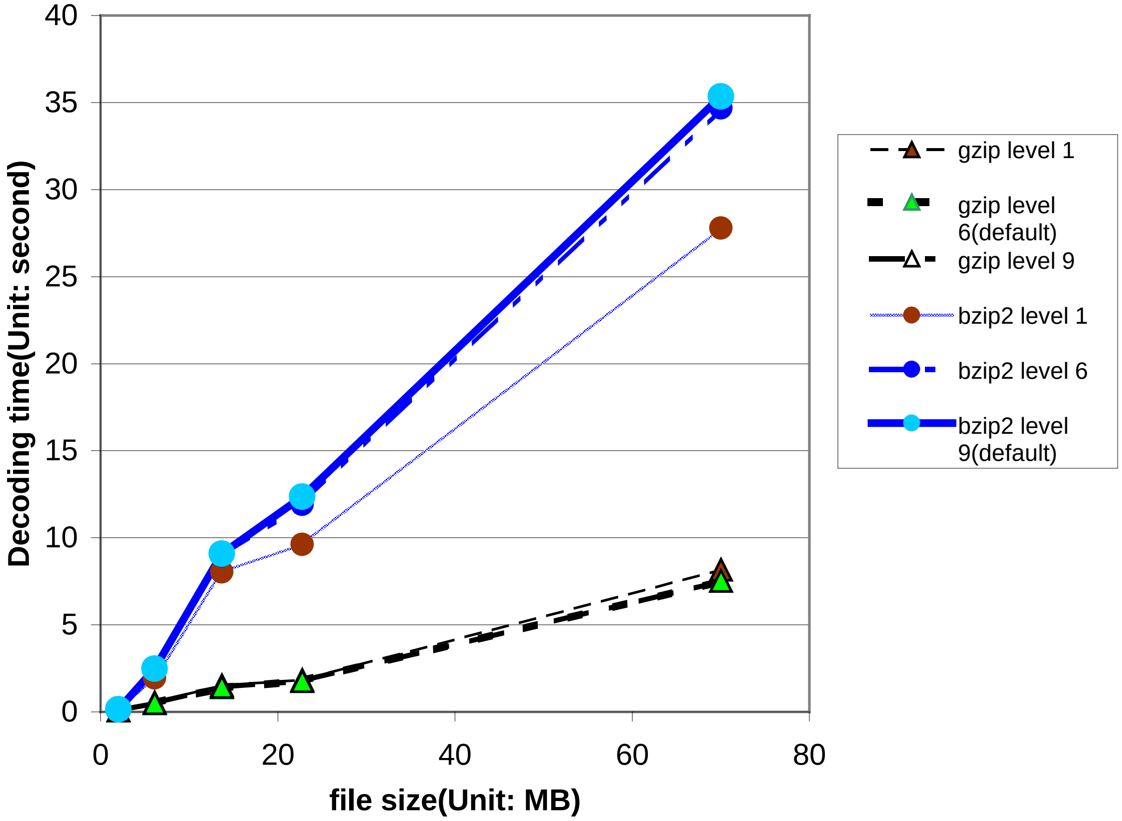


Figure 6: Decoding time comparision between gzip and bzip2 (Linux 2.2.18: the first five files)



6. Performance comparison with compression library calls

1) Working procedure

- A user-provided bzip2 filter is integrated with HDF5 library to make the performance comparison between bzip2 and gzip.
- Based on utility performance comparison, we selected three arrays with different datatype. They represent arrays with float, 16-bit integer and 8-bit integer individually.
- We calculate compression ratio, encoding time of the library and decoding time of the library.

2) Tables and charts

The following tables show performance results of the three arrays. To make the comparison and consistent checking between compression libraries and utilities, The compression ratio, encoding and decoding time of the three corresponding files are also included afterwards.

i) Unsigned 8-bit integer

Data source: ASTER (Advanced Spaceborne Thermal Emission and Reflection Radiometer) data on Terra
File name: ast1.h5
File size: 124,422,472 byte
Array size: 22,908,000 byte
Data type: unsigned char
Array dimensions: 4600 * 4980

Table 2: compression information for one array of ASTER1

	Compression ratio	**Encoding time (second)**	**Decoding time (second)**
Bzip2 level 1	0.441	21.6	15.87
Bzip2 level 6	0.428	22.2	21.3
Bzip2 level 9 (default)	0.426	22.7	22.24
Gzip level 1	0.593	9.43	1.95
Gzip level 6 (default)	0.586	14.86	1.86
Gzip level 9	0.585	18.08	1.86

ii) Unsigned 16-bit integer

Data source: ASTER (Advanced Spaceborne Thermal Emission and Reflection Radiometer) data on Terra
File name: ast2.h5
File size: 78,583,805 byte
Array size: 10,458,000 byte
Data type: unsigned short
Array dimensions: 2100 * 2490

Table 3: compression information for one array of ASTER2

	Compression ratio	**Encoding time (second)**	**Decoding time (second)**
Bzip2 level 1	0.1171	9.27	5.08
Bzip2 level 6	0.1141	9.76	7.44
Bzip2 level 9 (default)	0.1136	10.43	8.01
Gzip level 1	0.2014	2.03	0.53
Gzip level 6 (default)	0.1656	7.9	0.44
Gzip level 9	0.1647	48.09	0.44

iii) 32-bit float

Data source: CERES(Clouds and the Earth's Radiant Energy System)
File size: 76,199,508 byte
Array size: 5,359.200 byte
File name: ceres2.h5
Data type: float
Array dimensions: 2030*660

Table 4: compression information for one array of CERES2

	Compression ratio	**Encoding time (second)**	**Decoding time (second)**
Bzip2 level 1	0.4570	19.61	4.14
Bzip2 level 6	0.4545	25.23	5.08
Bzip2 level 9 (default)	0.4519	26.47	5.42
Gzip level 1	0.4801	1.95	0.36
Gzip level 6 (default)	0.4703	2.6	0.34
Gzip level 9	0.4700	3.38	0.34

Figure 7: Compression ratio comparision between gzip and bzip2 with CERES and ASTER files (linux 2.2.18)

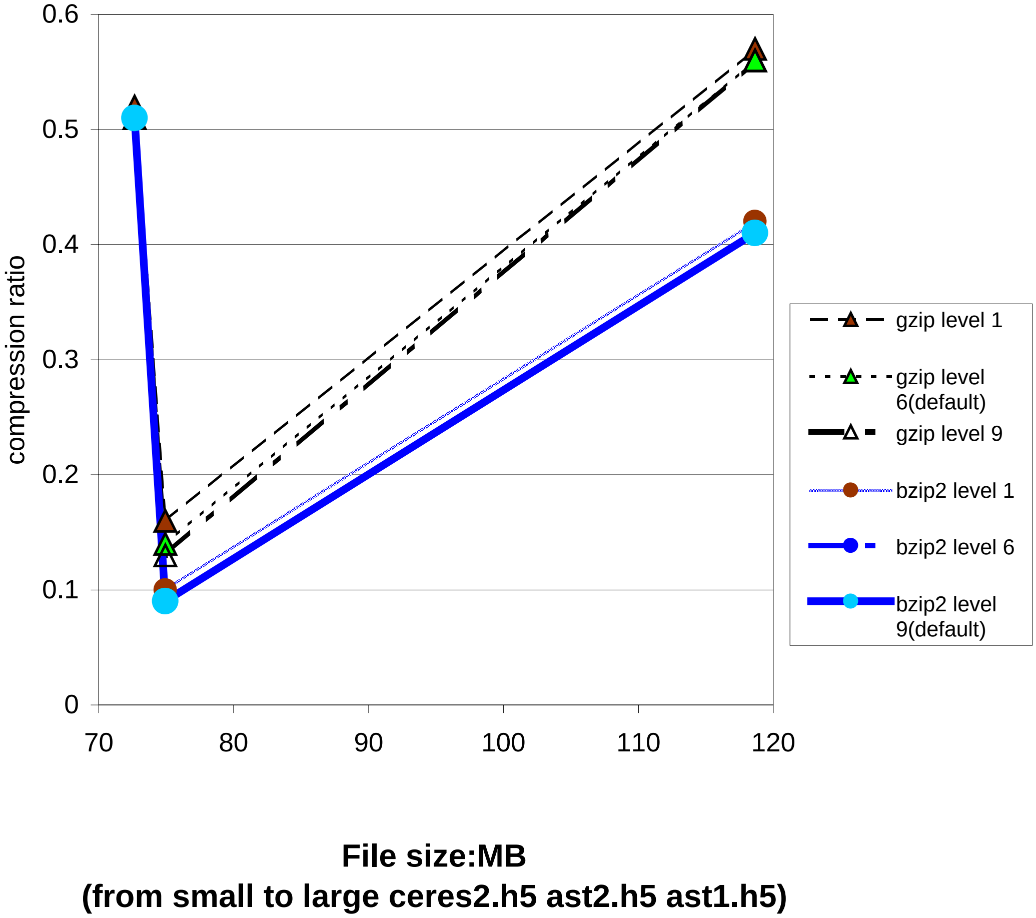


Figure 8: Encoding time comparision between gzip and bzip2 with CERES and ASTER files(linux 2.2.18)

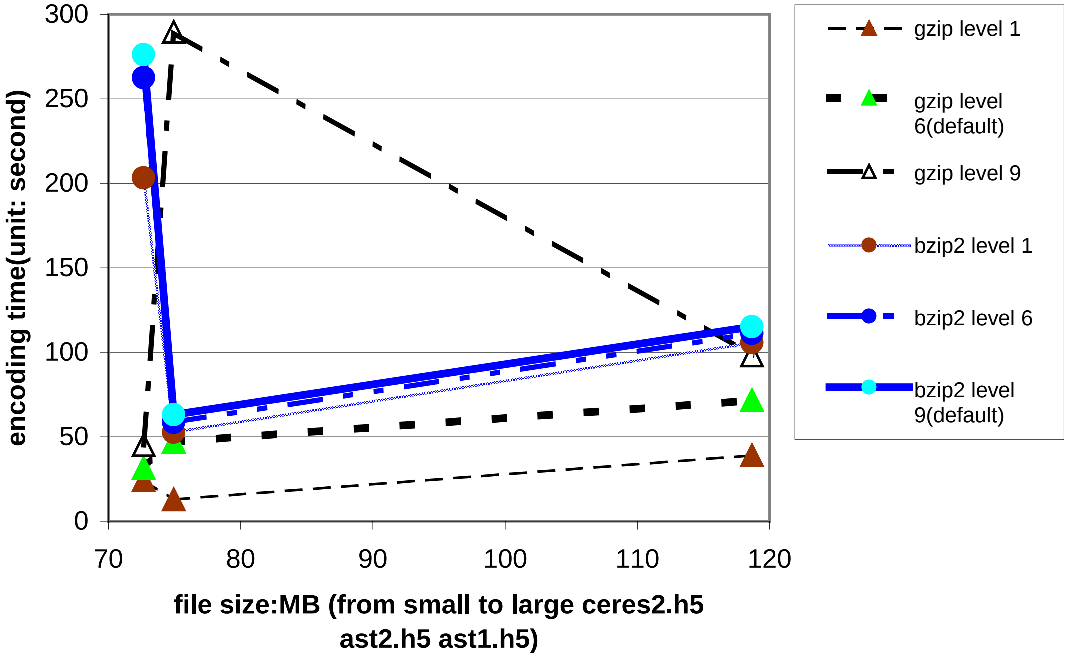
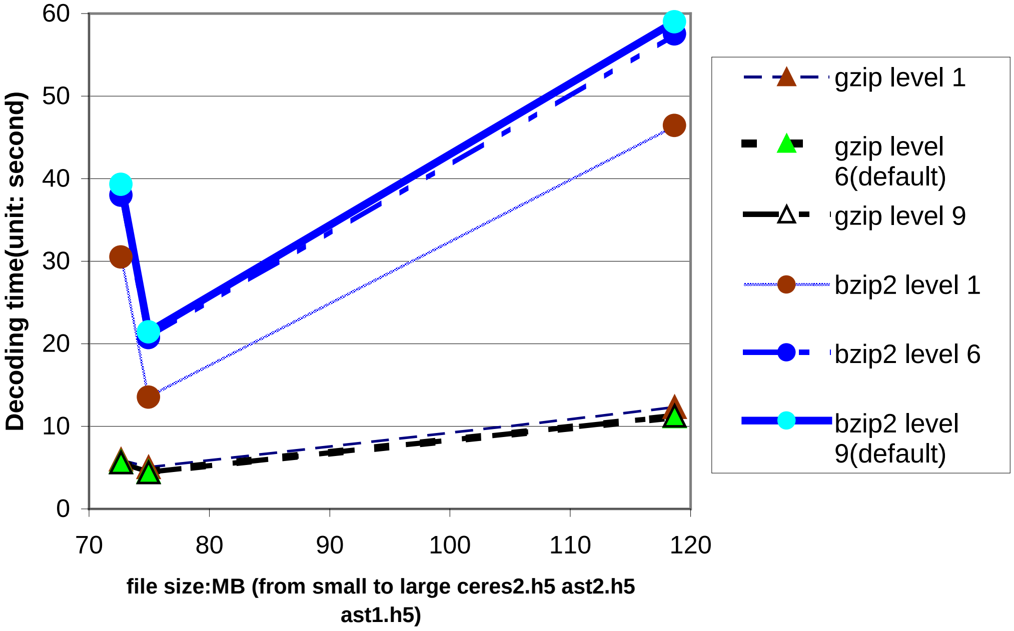


Figure 9: Decoding time comparision between gzip and bzip2 with CERES and ASTER files(Linux 2.2.18)



3) Library analysis results:

i. With the respect of compression ratio, library analyzing results are consistent well with utility analyzing results.
ii. With the respect of encoding time and decoding time, library analysis results are consistent with utility analyzing results qualitatively .
iii. Overhead to call the compression library inside HDF5 library is endurable.

Tables that show the comparison of size/second between library and utility are as follows. We only compare the default level of bzip2 and gzip.

Relative efficiency =
((Library compression size/second)/library compression ratio)/
((Utility compression size/second)/utility compression ratio)

According to the table, the relative efficiencies of both libraries are above 80% except for aster2.h5. More reliable results should be obtained by counting encoding time and compression ratio of every array of the file.

Table 5: relative encoding time comparison between library and utility for ceres data
Data source: ceres2.h5

	Size (byte)	**Compression ratio**	**Encoding time (second)**	**Encoding Size/second (byte/s)**	**Relative efficiency**
Bzip2 Library (L9)	5359200	0.4519	26.47	202463.2	0.8279
Bzip2 Utility (L9)	76199508	0.51	276.09	275995.2	1
Gzip library (L6)	5359200	0.4703	2.6	2061231	0.9228
Gzip Utility (L6)	76199508	0.51	31.46	2422108	1

Table 6: relative encoding time comparison between library and utility for ASTER data I
Data source: ast2.h5

	Size (byte)	**Compression ratio**	**Encoding time (Second)**	**Encoding Size/second (byte/s)**	**Relative efficiency**
Bzip2 Library (L9)	10458000	0.1136	10.43	1002685	0.6378
Bzip2 Utility (L9)	78583805	0.09	63.09	1245583	1
Gzip library (L6)	10458000	0.1656	7.9	1323797	0.6694
Gzip Utility(L6)	78583805	0.14	47	1671996	1

Table 7: relative encoding time comparison between library and utility for ASTER data II
Data source: ast1.h5

	Size (byte)	**Compression ratio**	**Encoding time(second)**	**Encoding Size/second (byte/s)**	**Relative efficiency**
Bzip2 Library (L9)	22908000	0.426	22.7	1009163	0.8986
Bzip2 Utility (L9)	124422472	0.41	115.12	1080807	1
Gzip library (L6)	22908000	0.586	14.86	1541588	0.8473
GzipUtility (L6)	124422472	0.56	71.56	1738715	1

7. Concluding remarks and suggestions

According to the analyses with very limited samples, we find

- Bzip2 is always better than gzip in compression ratio.
- Bzip2 is always taking longer processing time than gzip, especially for decoding time.
- Neither compression packages is especially good for floating point data.
- Compression ratio gains little improvement with the increasing of compression levels for both compression packages.
- Encoding time is worse with the increasing of compression level for both compression packages. Encoding time is more sensitive for gzip than for bzip2.
- Overall library analysis is consistent with utility analysis.
- Integrating bzip2 to HDF5 is not hard but maintenance effort cannot be ignored.

Suggestions:

- Don’t use bzip2 for floating point data if you don’t have to.
- If you care about compression ratio more than anything else, you may consider using bzip2.
- If you care about decoding time more than anything else, you may choose to use gzip.

8. What’s left?

- Configuration integration with HDF5 tests and tools
- Add proper comments to bzip2 filter that a user provided
- Tests on other platforms
- Implementation of bzip2 filter at user’s applications

9. Reference:

1. Michael Burrows and D.J. Wheeler, 1994. “A block-sorting lossless data compression algorithm,” Digital SRC Research Report 124. ftp://ftp.digital.com/pub/DEC/SRC/research-reports/SRC-124.ps

2. Huffman, D.A., 1952. “A method for the construction of minimum redundancy codes,” Proceedings of the IRE, Volume 40, Number 9, pages 1098-1101.

3. bzip2 URL: http://sources.redhat.com/bzip2/

4. h4toh5 utility URL: http://hdf.ncsa.uiuc.edu/h4toh5/

Appendix: Tables of compression ratio, encoding time and decoding time for all ten NASA files

CR: Compression Ratio ET: Encoding Time (unit: second) L1: Level 1
FS: File Size (unit: MB) DT: Decoding Time (unit: second) L6: Level6
FN: File Name L9: Level 9

Table 8: gzip compression ratio, encoding time and decoding time for all ten NASA files with the compression level 1,6,9 at linux 2.2.18

FN	FS	CR L1	ET L1	DT L1	CR L6	ET L6	DT L6	CR L9	ET L9	DT L9
SSMI	2.02	0.07	0.22	0.09	0.05	0.54	0.08	0.04	3.58	0.08
TOMS	6.09	0.34	1.52	0.56	0.32	5.28	0.5	0.32	12.18	0.5
TRIM	13.69	0.75	6.02	1.53	0.74	8.87	1.4	0.74	15.22	1.44
CERES1	22.78	0.52	7.48	1.84	0.52	9.53	1.76	0.52	14.06	1.76
MISR	70.01	0.61	26.85	8.17	0.6	52.97	7.54	0.6	63.24	7.53
CERES2	72.67	0.52	24.25	5.96	0.51	31.46	5.69	0.51	45.17	5.63
ASTER2	74.94	0.16	12.87	5.03	0.14	47	4.48	0.13	289.6	4.41
ASTER1	118.66	0.57	38.98	12.32	0.56	71.56	11.21	0.56	98.14	11.26
MODIS1	262.34	0.61	92.4	25.5	0.6	156.14	23.35	0.6	439.07	23.38
LANDSAT	561.89	0.44	162.65	58.28	0.42	380.55	53.64	0.42	527.33	51.72

Table 6: bzip2 compression ratio, encoding time and decoding time for all ten NASA files with the compression level 1,6,9 at linux 2.2.18

FN	FS	CR L1	ET L1	DT L1	CR L6	ET L6	DT L6	CR L9	ET L9	DT L9
SSMI	2.02	0.04	0.39	0.15	0.04	0.37	0.15	0.04	0.36	0.15
TOMS	6.09	0.25	12.78	1.95	0.23	15.35	2.51	0.23	16.56	2.49
TRIM	13.69	0.69	20.95	8.03	0.71	21.06	8.97	0.72	21.28	9.08
CERES1	22.78	0.51	63.99	9.62	0.51	82.39	11.91	0.52	87.36	12.35
MISR	70.01	0.43	89.49	27.79	0.39	102.92	34.68	0.39	106.69	35.34
CERES2	72.67	0.51	203.2	30.52	0.51	262.5	37.99	0.51	276.09	39.32
ASTER2	74.94	0.1	52.58	13.52	0.09	58.6	20.74	0.09	63.09	21.42
ASTER1	118.66	0.42	105.77	46.44	0.41	111.19	57.53	0.41	115.12	58.98
MODIS1	262.34	0.52	304.48	113.99	0.5	323.96	141.82	0.49	335.88	141.09
LANDSAT	561.89	0.37	441.53	195.08	0.37	507.59	247.04	0.36	539.1	253.44