Experimenting with Burrows-Wheeler Compression

Juha Kärkkäinen

University of Helsinki

(Work done mostly as Visiting Scientist at Google Zürich)

3rd Workshop on Compression, Text and Algorithms

Melbourne, Australia, 13 November 2008

Experimental Burrows-Wheeler Compressor

Goals for the project

- Fast BW transform with large block sizes on repetitive texts
- Platform for experimenting with different techniques for
  - BW transform
  - compressing BWT
- Study the effect of block size on compression
- Compressor with high compression and good speed

Burrows-Wheeler Transform

Reverse and add sentinels

BANANA ⟶ #ANANAB#

Split into prefix and suffix at every position
Sort by suffix and take the last symbols of prefix

#	0	ANANAB#
#A	1	NANAB#
#AN	2	ANAB#
#ANA	3	NAB#
#ANAN	4	AB#
#ANANA	5	B#
#ANANAB	6	#

⟶

#ANANA	B	6	#
#ANA	N	4	AB#
#A	N	2	ANAB#
	#	0	ANANAB#
#ANAN	A	5	B#
#AN	A	3	NAB#
#	A	1	NANAB#

BWT = BNN#AAA

Example

```
sprang up, mounted their horses, and gallo
self had attempted the ascent.  It was alm
  He then observed that the grass partly h
a sound and seized the bird’s two feet wit
the price, he paid the man in gold, who, s
ed."  "Good," said the czar.  "If you have
t your word," said the hunter.  He then be
 "Very well," said the hunter.  "’You will
if they can," said the czar. The hunter wa
ome nankeen," said the second.  The younge
if he could behold the top of the mountain
e Unlucky was told that an enormous army o
tiful skies."  And the apple began to roll
es and riders! And this had been the end o
```

Example

All characters following “`th`” in a 16 KiB block of English.

```
oreeereoeeieeeeaooeeeeeaereeeeeeeeeeeeereee
eeeeeeeaaeeaeeeeeeeeeaeeeeeeeeaeieeeeeeeeer
eeeeeeeeeeeeeeeeeeeeeeeeaeeieeeeeeaaieeeeee
eeeeeeeeeeeeeeeeeeeeeeeeeeaeieeeeeeeeeeeeee
eeeeeeeeeeeeeeaeeeeeeeeeeeeeeeeeeeeereeeeee
eeeeeieaeeeeieeeeaeeeeeeeeeieeeeeeeeeeeeiee
eeeeeeioaaeeaoereeeeeeeeeeaaeaaeeeeieeeeeee
ieeeeeeeeaeeeeeeaeeeeeereeeaeeeeeieeeeeeeei
ieee. e  eeeeiiiiii e          ,  i   o
   oo e  eiiiiee,er  ,  ,     ,  . iii
```

Burrows Wheeler Compression

1. Divide text into blocks (if necessary)
2. Compute BWT for each block
3. Compress the BWT with an entropy compressor

- BWT brings characters with similar context together.
- Easy to compress using simple local models
  - Run-length encoding
  - Move-to-front encoding

Compressing Distant Repeats

- Many compression algorithms need a compression model with a “long memory”.
- BW compression survives with “short memory” entropy compressor.
- BW compression needs BW transform for large blocks.
  - `bzip2` blocksize is only 900 KB

Fast BWT for Large Blocks

Computing BWT is demanding when

- blocksize is large
- text contains lots of repeats, i.e., is highly compressible
  - `bzip2` performance suffers

Combination of techniques

- Optimized induced copying
- Tuned multikey quicksort
- Difference cover sampling → $\mathcal{O}(n \log n)$ worst case

- Inverse BWT modified for large blocks

Entropy Compressor for BWT

- Inspired by *bbb* compressor by Matt Mahoney

1. Run-length encoding

   aaabbbbb... → (a,3)(b,5)...

2. Bit encoding: (8 bit code, Elias gamma code)

   (a,3)(b,5)... →
   (01100001,101)(01100010,11001)

3. Determine a probability for each bit
   - Complex adaptive model

4. Arithmetic coding

Predicting Bits

- Each bit has a context
- Character bit context depends on
  - position of bit in the byte
  - preceding bits in the byte
  - last few preceding *distinct* characters (MTF)
    - Are the preceding bits same?
    - If yes, the bit in this position
- Run length code bit context depends on
  - bit position
  - associated character (first bit)
  - some preceding bits

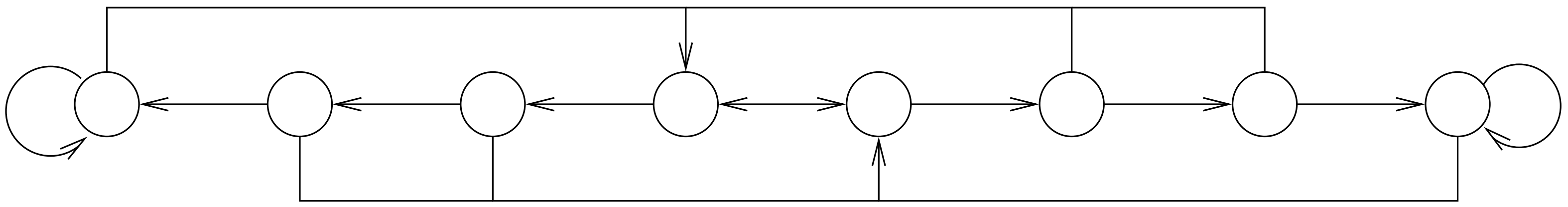
Mapping context to probability

Stationary model

- Each context has its own stored probability
- Small adjustment with each bit

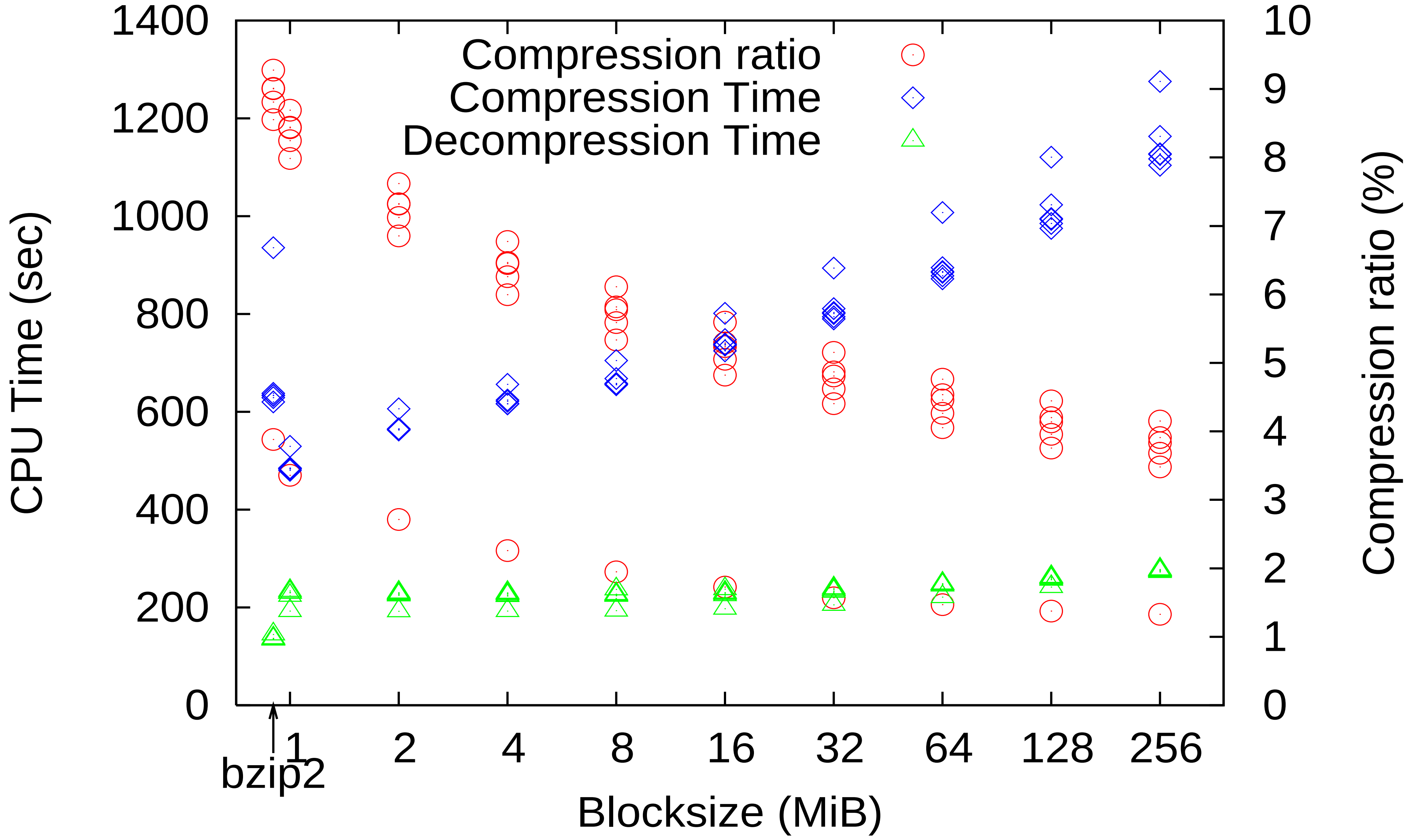
Non-stationary model

- Each bit causes a state transition in an automaton.

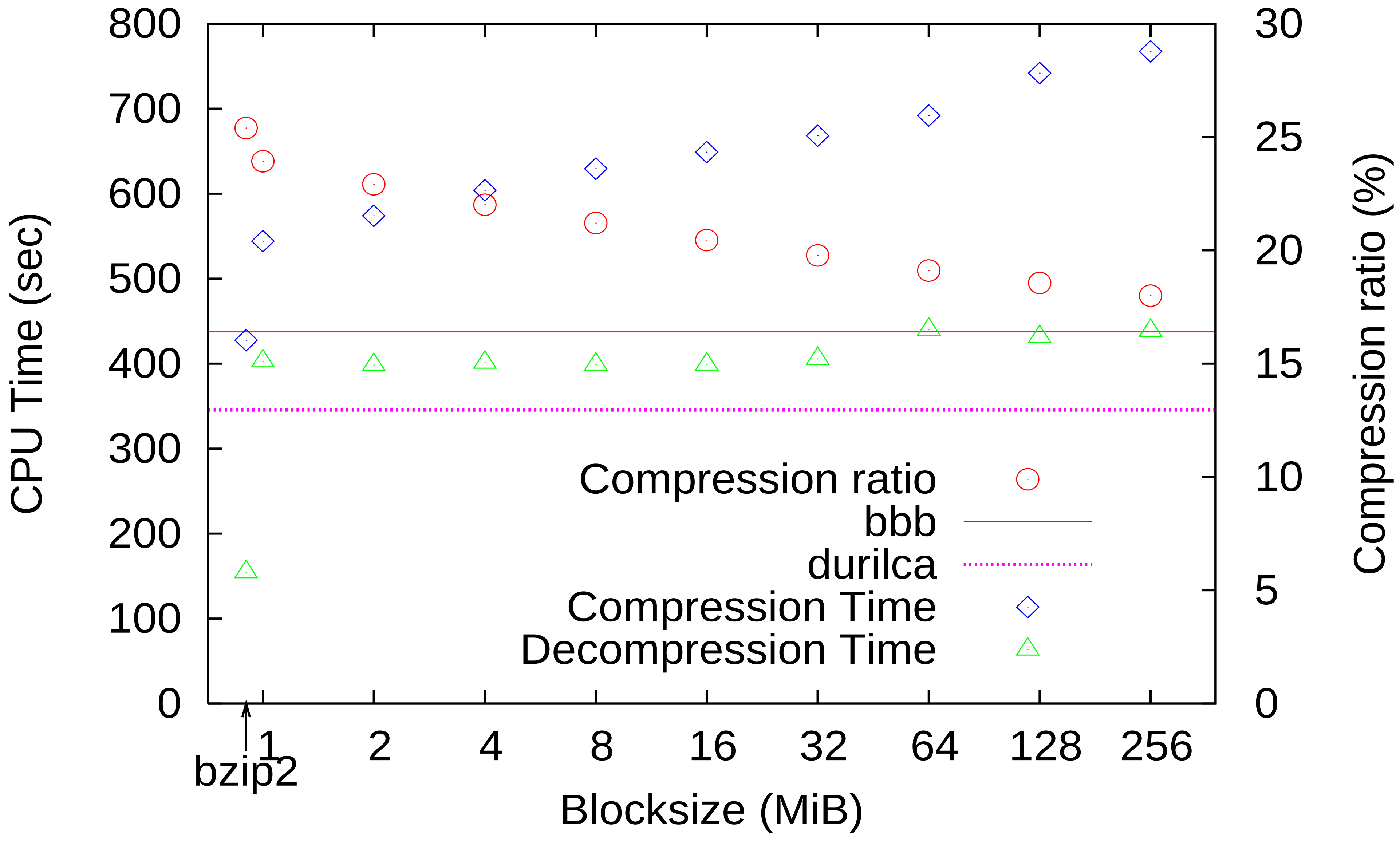


- Each state has a slowly adapting probability.
  - Neighbour states adjusted too.

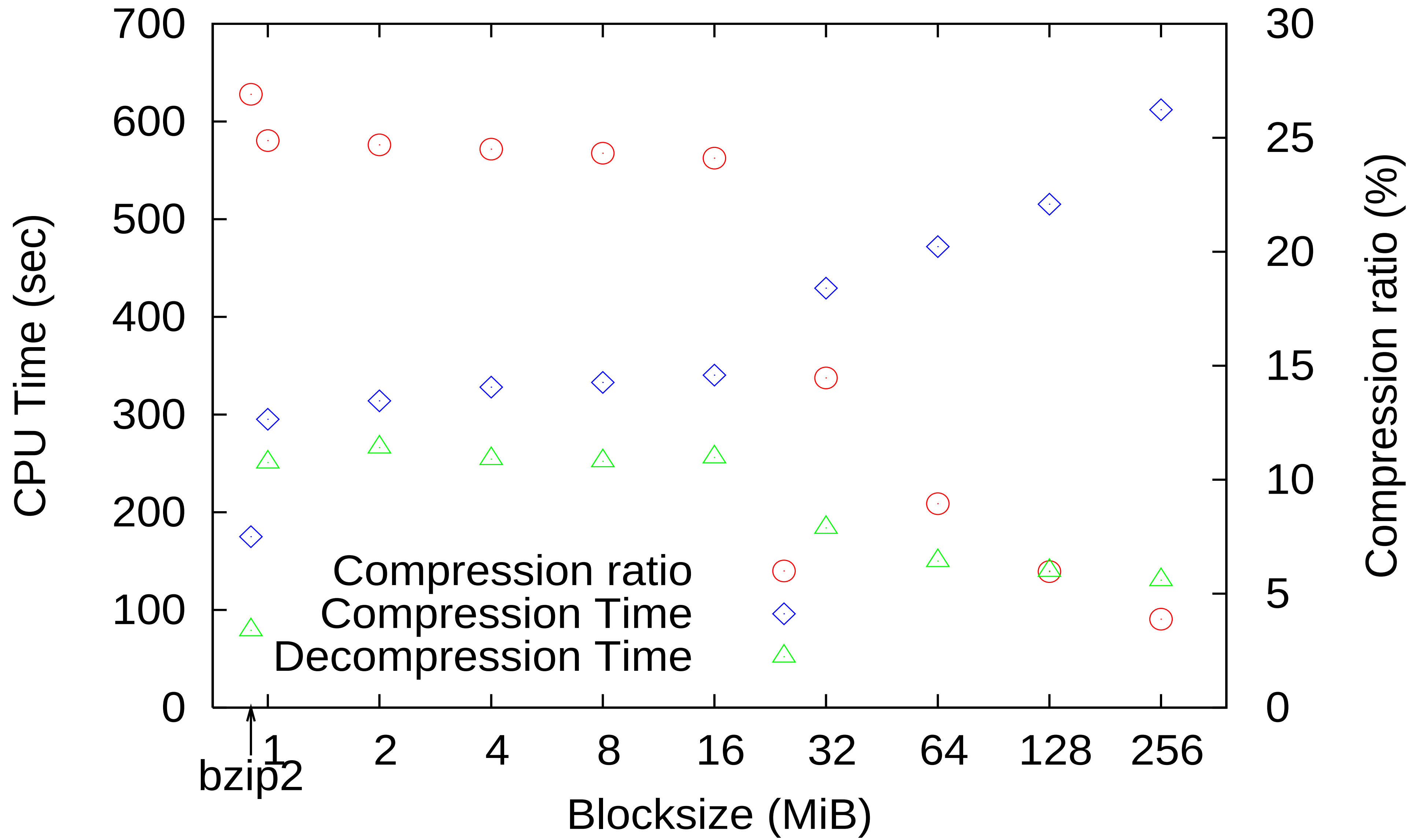
Wikipedia HTML tar-archive (6 × 1 GiB)



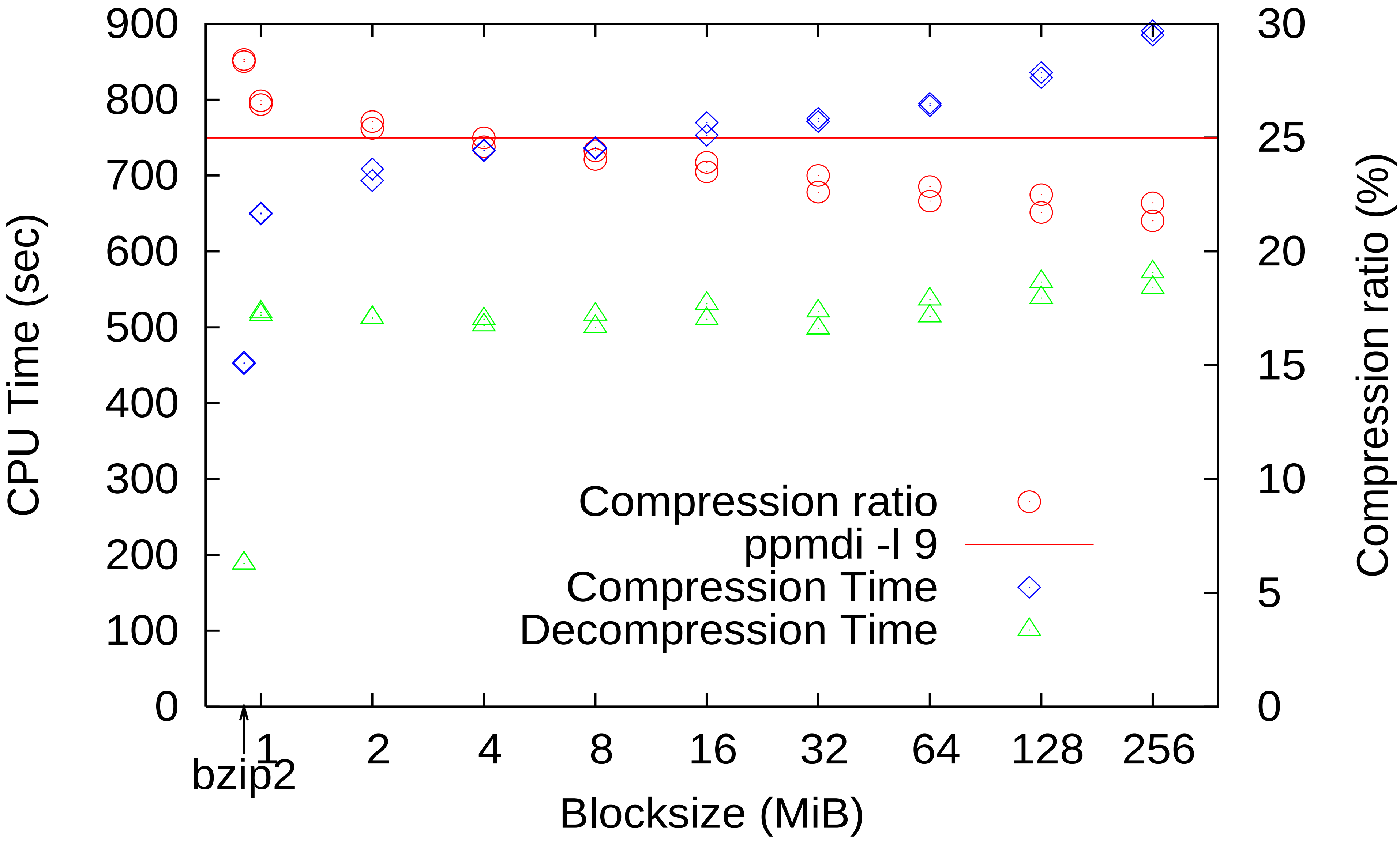
Wikipedia XML from LTCB (1 GB)



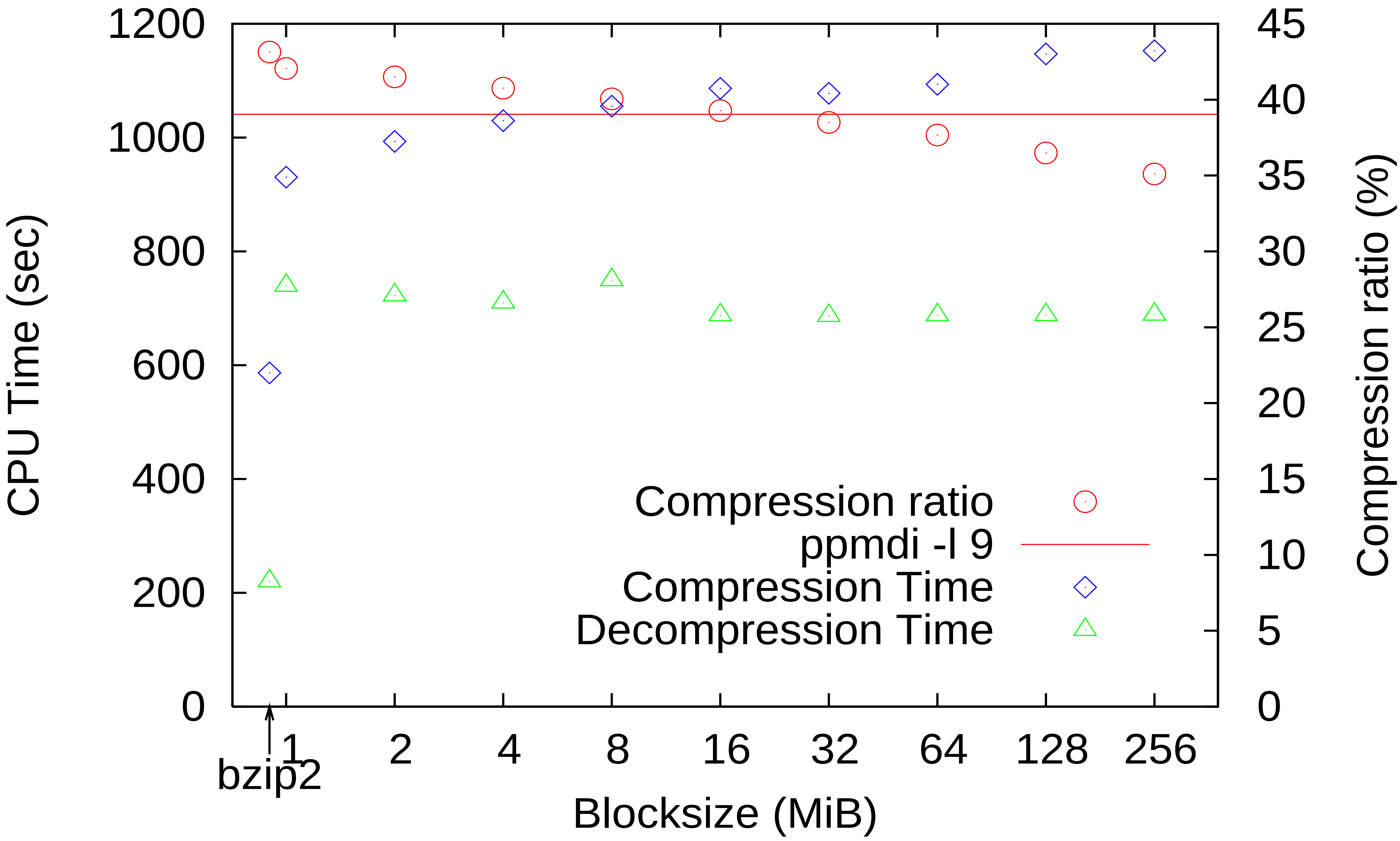
25 mutated copies of DNA (25 × 16 MB)



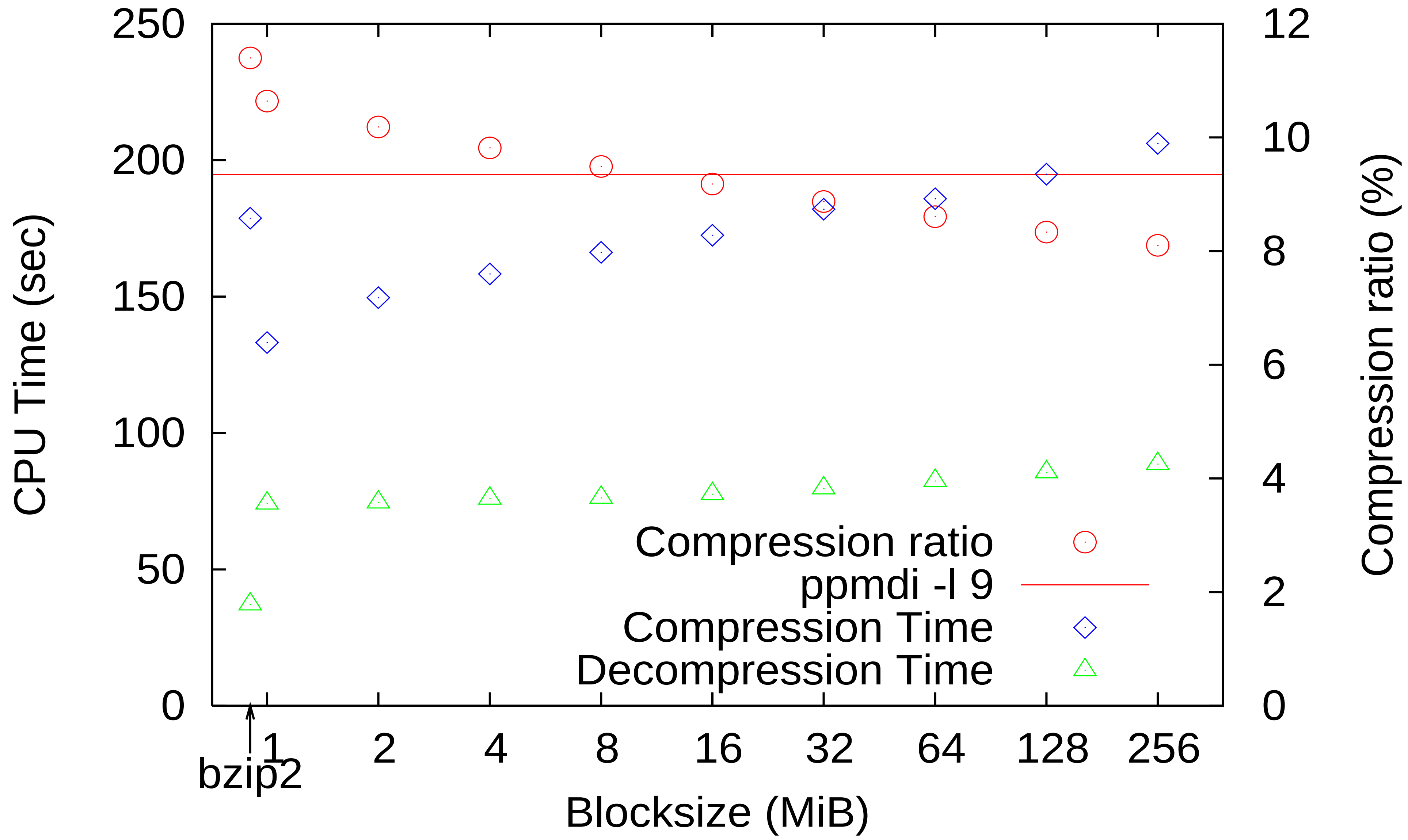
English from Pizza & Chili (2 × 1 GiB)



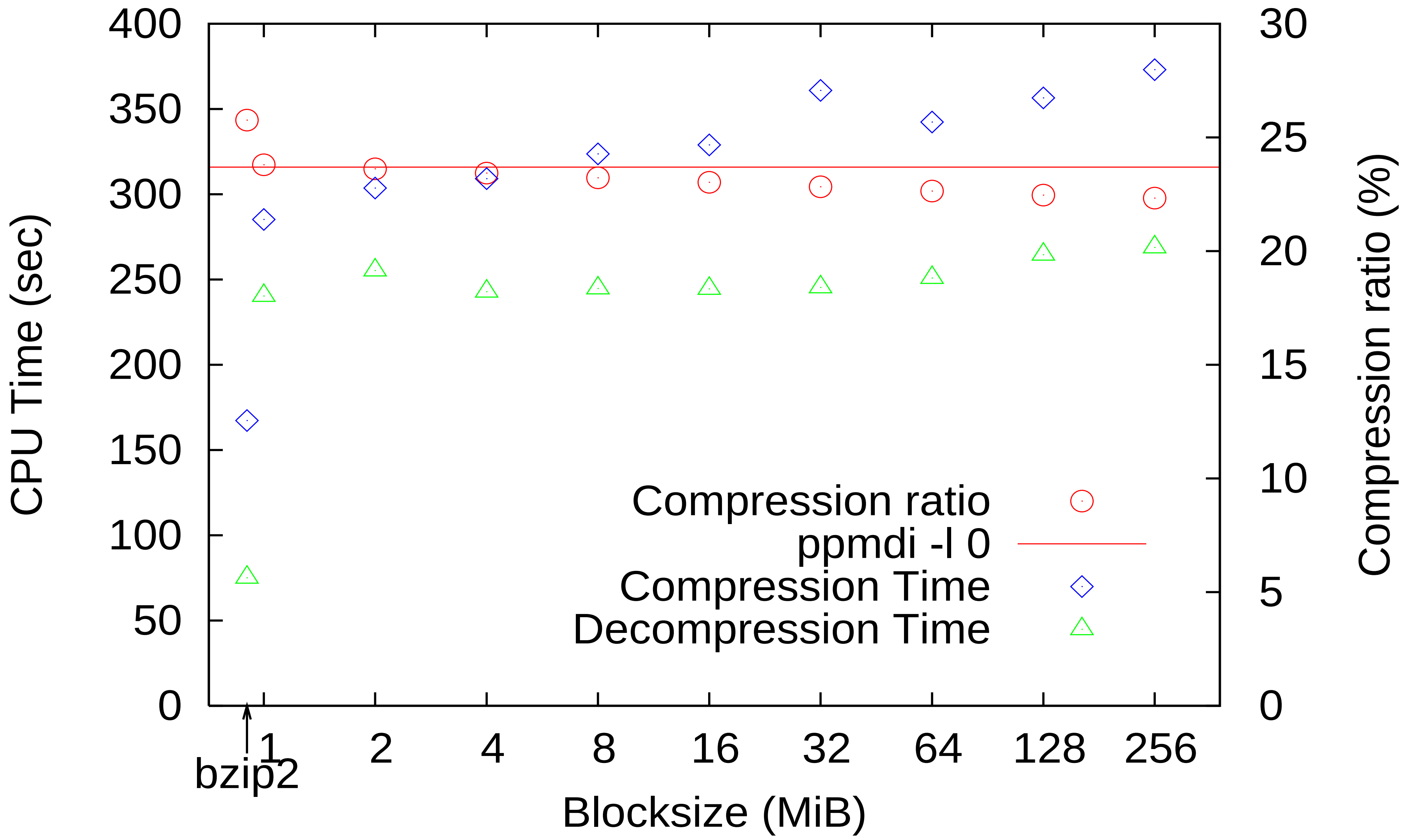
Proteins from Pizza & Chili (1 GiB)



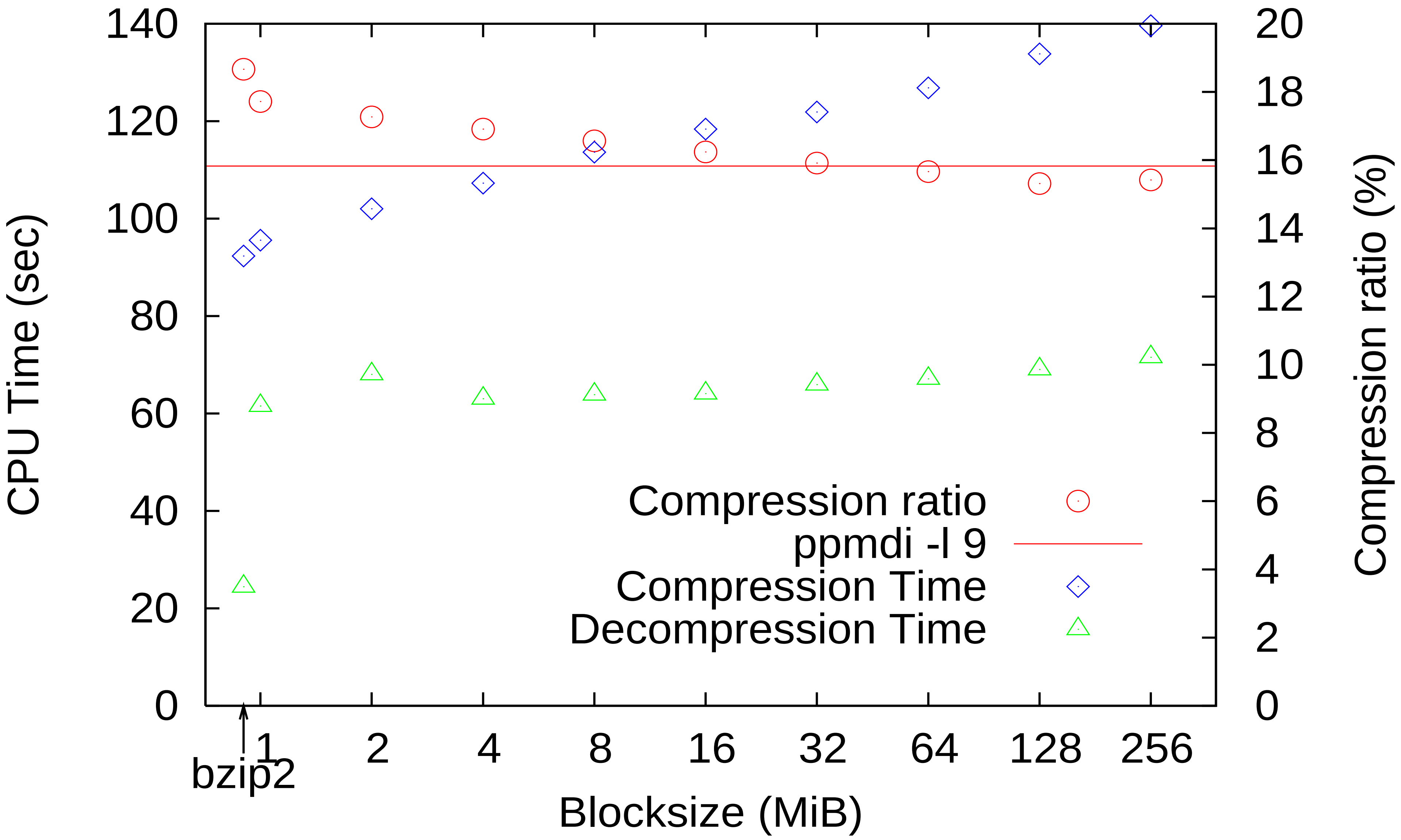
XML from Pizza & Chili (282 MiB)



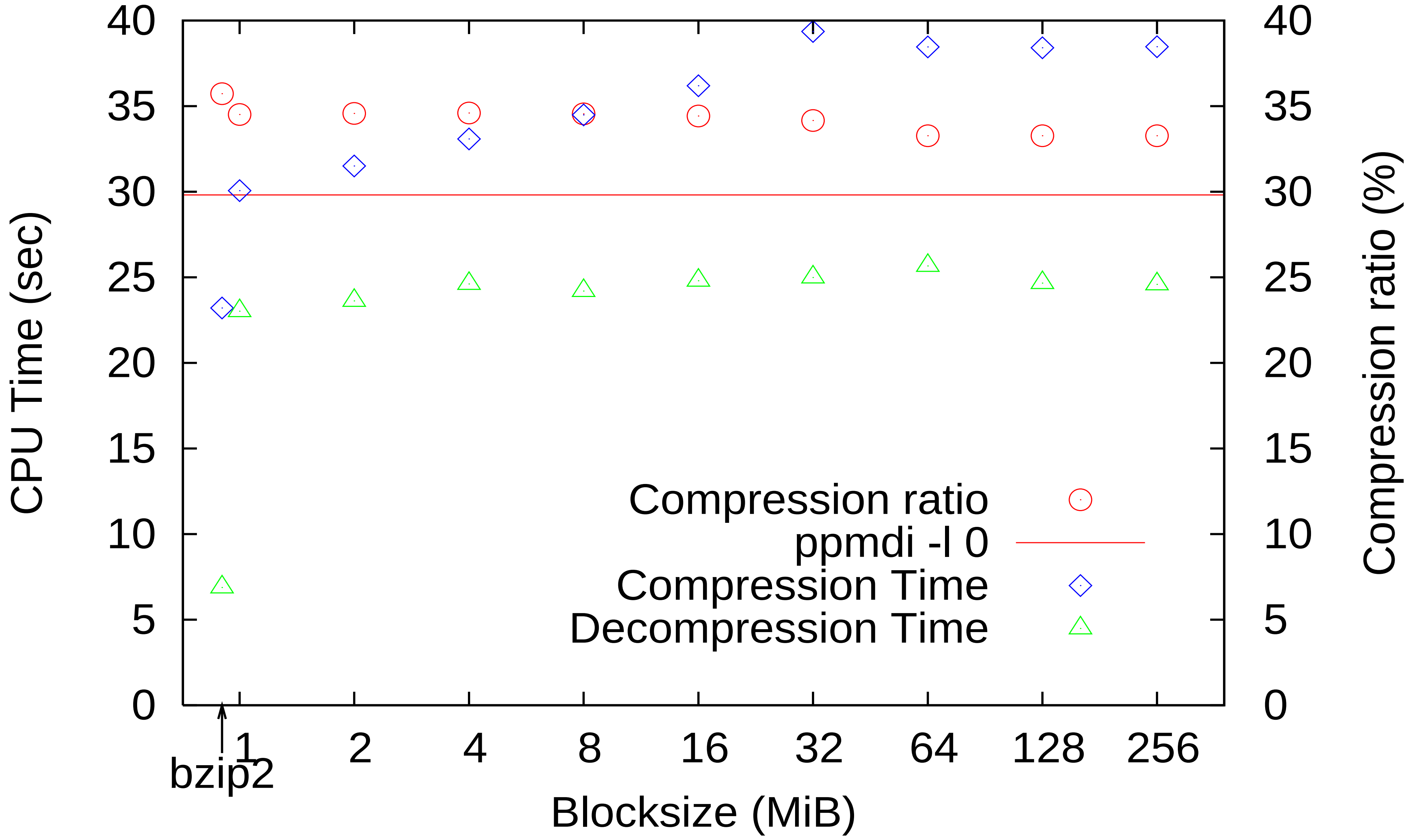
DNA from Pizza & Chili (385 MiB)



Sources from Pizza & Chili (201 MiB)



Pitches from Pizza & Chili (53 MiB)



Things to do

- Computing BWT
  - Faster
  - Use less space → larger blocks
- Faster entropy (de)compression
  - Reduce bits: Huffman?
  - From bits to larger units
- Better compression?
- Pre-BWT compression: LZ?
- Compressed self-indexes