gzip & DEFLATE

File format si algoritm de compresie lossless general

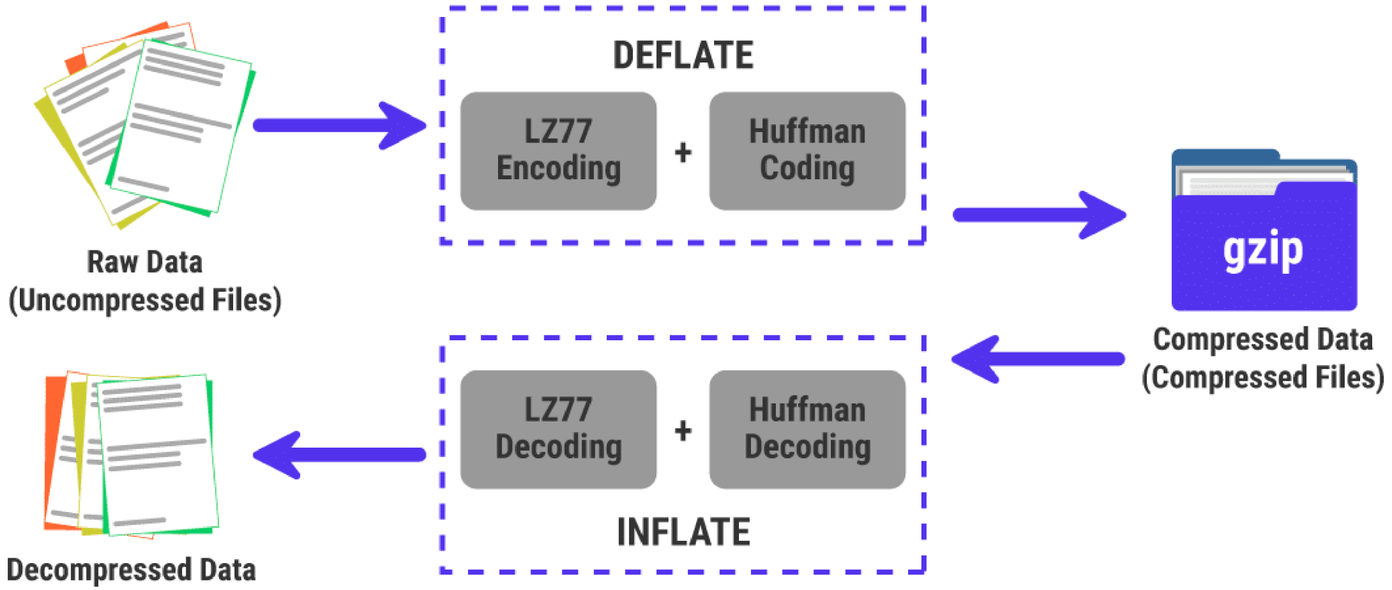
- Compresia lossless asigura reconstruirea in forma origniala a datelor fara nici o pierdere de informatii
- Folosit pe scara larga pentru a reduce dimensiunea fisierelor pentru o stocare eficienta si o transmisie rapida.
- Utilizat in special in compresia end-to-end HTTP pentru a reduce dimensiunea resurselor paginilor web in traficul dintre client si server.
    - Reduce timpul de incarcare a paginilor
    - Reduce cantitatea de date transferate pentru optimizari de bandwidth
- Ofera un compression-ratio destul de bun insa nu este cel mai performant
- Este rapid atat la pasul de compress cat si la decompress
- In cel mai rău caz creste marimea fisierului cu foarte putin
- Implementare open-source

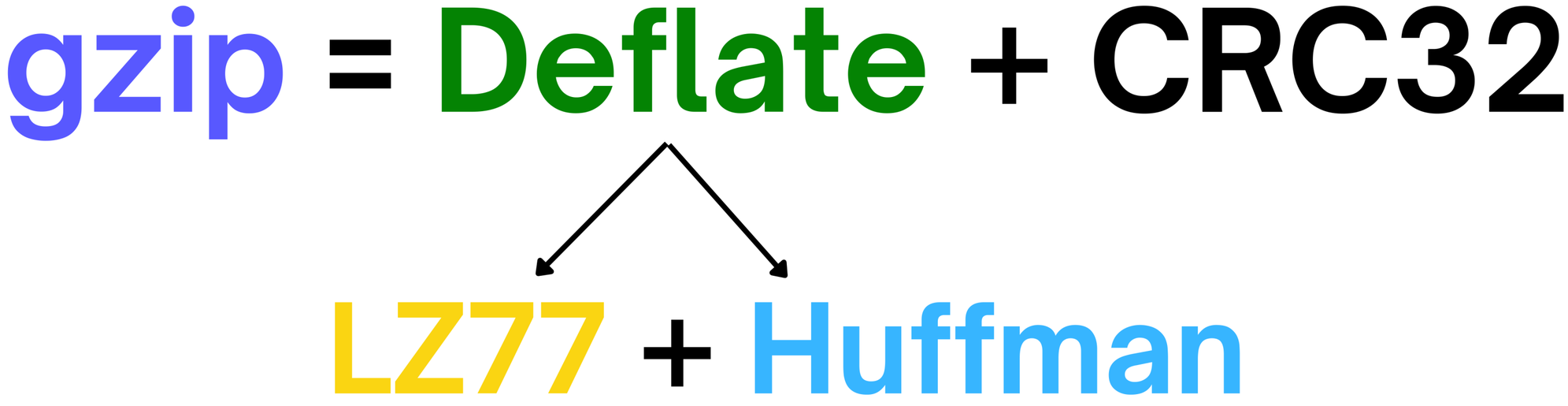
DEFLATE

- **Algoritm** open source
- Ofera **compresie lossless**
- Bazat pe combinarea **LZ77** si **Huffman coding**

gzip

- **File format** si **utilitar** ce foloseste deflate
- Adauga error detection codes prin **CRC**




gzip = Deflate + CRC32
LZ77 + Huffman

```
SoA ❯ ~  gzip -l test-gzip.gz
         compressed        uncompressed  ratio uncompressed_name
               1469                3203  55.0% test-gzip
SoA ❯ ~
```

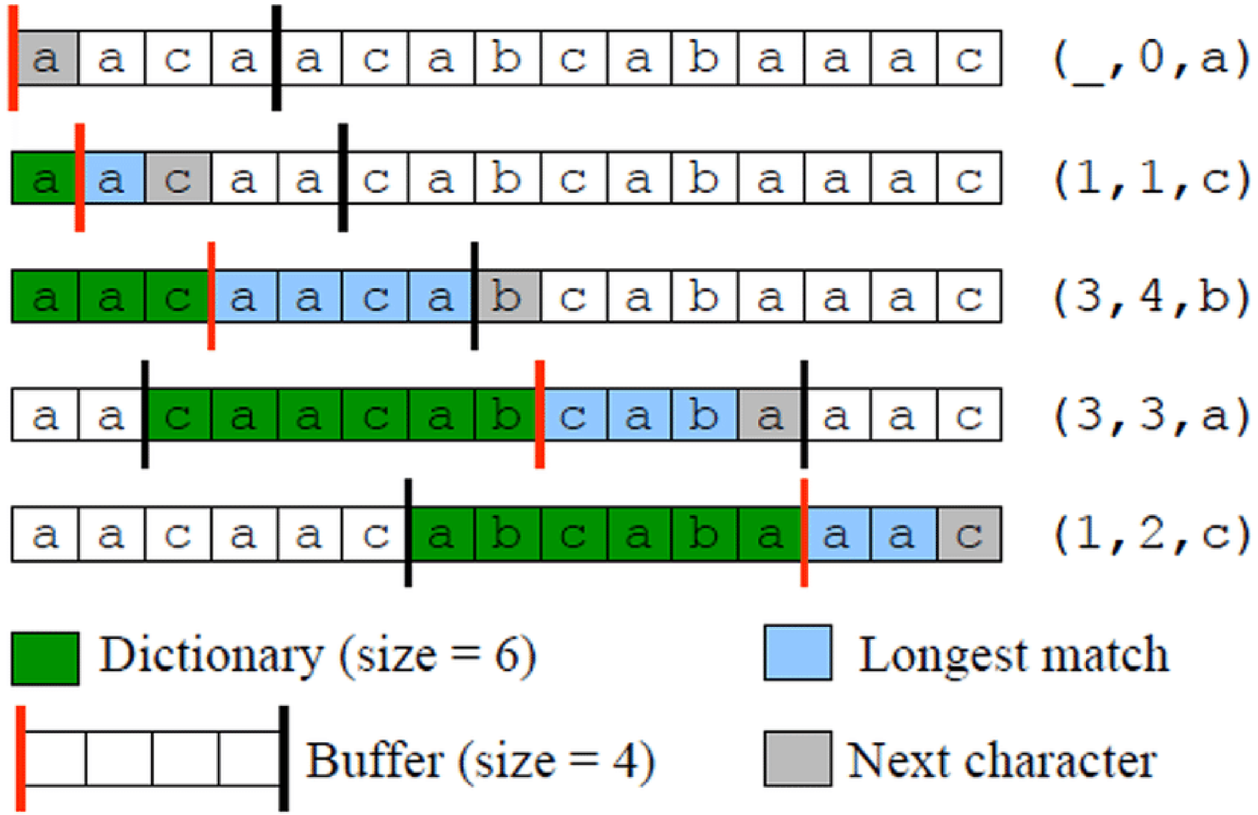
```
00000000: 1f8b 0808 a105 a965 0003 7465 7374 2d67  .......e..test-g
00000010: 7a69 7000 ed56 4b6e 1c37 10dd f729 e855  zip..VKn.7...).U
00000020: 8060 5a17 f022 1026 0e22 c088 8cf8 9375  .`Z..".&."......u
00000030: 0dbb 7a44 834d 36d8 e420 d0ca c0ec bdc9  ..zD.M6.. ......
00000040: c68a 912c 7c80 5c42 b398 73f8 2479 8f6c  ...,|.\B..s.$y.l
          . . .
00000580: 4de4 3fca 0c2d 532f d029 5e28 c08d c698  M.?..-S/.)^(....
00000590: 9918 42ce b8f9 bb27 5862 7501 3328 e832  ..B....'Xbu.3(.2
000005a0: 93f7 ed1d 89c5 65ae 464f c6e3 b560 55d5  ......e.FO...`U.
000005b0: 17bf ebfe 05d6 ea31 ad83 0c00 00         .......1.....
```

```
1f 8b       - magic number
08          - compression method
08          - file flags
a1 05 a9 65 - 32bit timestamp
00          - compression flags
03          - operating system ID
```

DEFLATE compressed data

CRC-32 checksum

LZ77 Lempel Ziv 1977

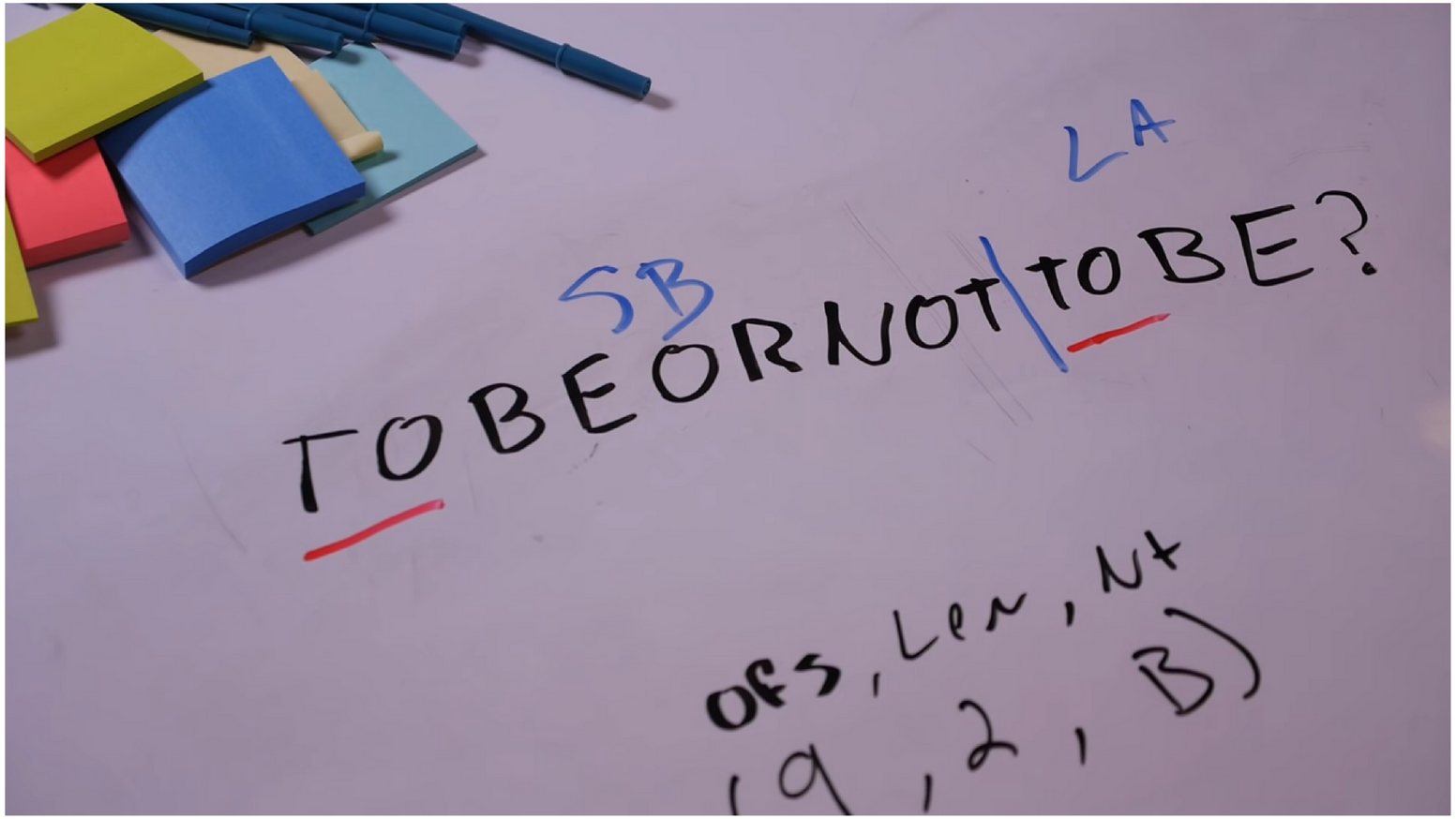


Elimina repetitiile de date folosind un sliding window pentru a construi un dictionar.

Fiecare element ajunge sa fie codat intr-o pereche (d, l, c):

- d - distanta saltului in urma
- l - lungimea repetitiei
- c - caracterul ce urmeaza

lz77 2

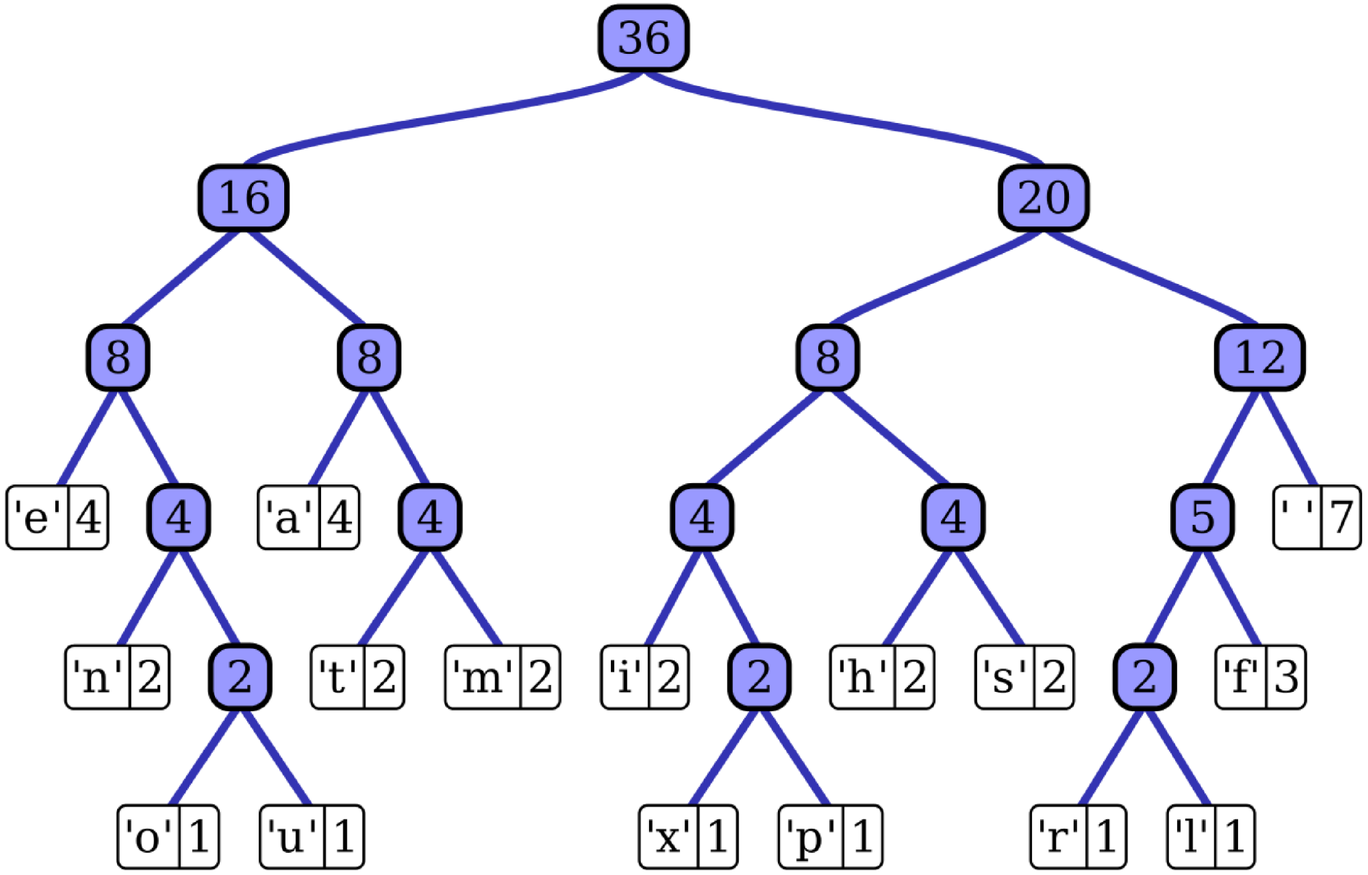


SB - search buffer
LA - look ahead buffer

<offset, length, nextToken>

algoritmul lz77 folosit in gzip foloseste un sliding window de marime 32 kilobytes

Huffman coding



"this is an example of a huffman tree"

Codificarea datelor prin coduri de lungime variabila pe baza probabilitatii de aparitie a acestora. Elementele mai frecvente avand coduri mai scurte.

huffman 2

Pasul de Huffman coding folosit in gzip se foloseste de output-ul comprimat in urma algoritmului LZ77 si creeaza doua tabele

Cate un tabel pentru:

- Literals + Lengths
- Distances

Cyclic redundancy check

- Functie de tip hash
- Asigura ca datele comprimate raman intacte in timpul transmiterii
- Detecteaza modificarile accidentale, coruperea sau erorile din fisierul comrpimat

Cum?

Se incepe cu un polinom de ordin mare

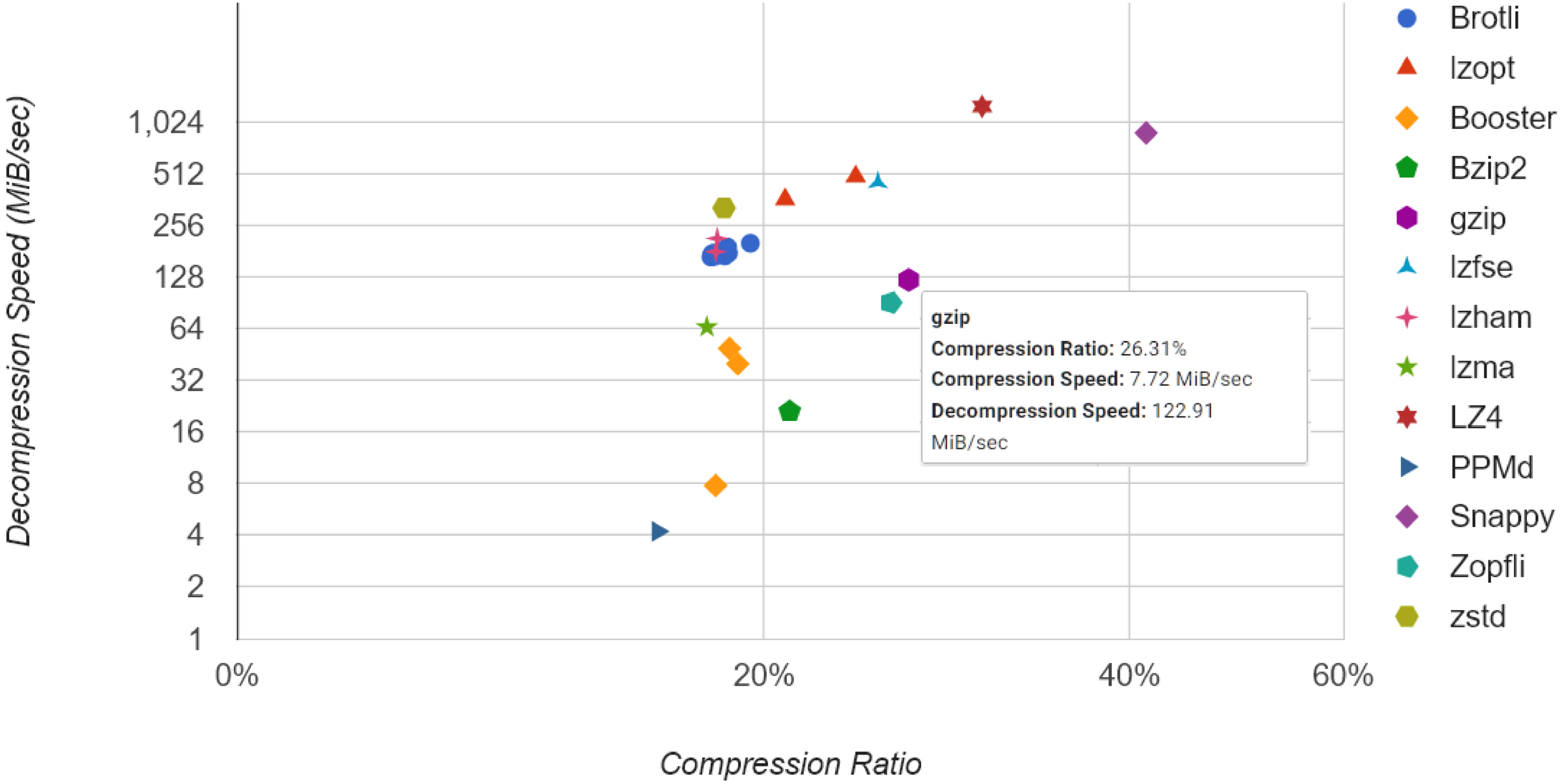
$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

Pentru fiecare byte din fisierul necomprimat aplicam operatii de XOR succesive si updatam valoarea CRC-ului

```python
def create_table():
    a = []
    for i in range(256):
        k = i
        for j in range(8):
            if k & 1:
                k ^= 0x1DB710640
            k >>= 1
        a.append(k)
    return a

def crc_update(buf, crc):
    crc ^= 0xFFFFFFFF
    for k in buf:
        crc = (crc >> 8)
              ^ crc_table[(crc & 0xFF) ^ k]
    return crc ^ 0xFFFFFFFF

crc_table = create_table()
print(hex(crc_update(b"The quick brown fox
                       jumps over the lazy dog", 0)))
```


Decompression Speed (MiB/sec)
1,024
512
256
128
64
32
16
8
4
2
1
0%
20%
40%
60%
Compression Ratio
gzip
Compression Ratio: 26.31%
Compression Speed: 7.72 MiB/sec
Decompression Speed: 122.91 MiB/sec
Brotli
lzopt
Booster
Bzip2
gzip
lzfse
lzham
lzma
LZ4
PPMd
Snappy
Zopfli
zstd