Komprese dat

David Bařina

2. listopadu 2024

ZIP, gzip

PKWARE®
ENCRYPTION WITHOUT COMPROMISE

- Deflate vyvinul v roce 1991 americký programátor Phil Katz
- použit ve formátu ZIP (PKZIP, původně), gzip z roku 1993 / zlib z roku 1995, 7-Zip z roku 1999
- jedná s o kombinaci LZ77 a Huffmanova kódování
- existují nástroje jako Zopfli, který podstatně zvyšuje kompresní poměr za cenu extrémně pomalé rychlosti komprese
- vyhledávací buffer 32 KiB
- rozdělení do bloků při kompresi je složitá optimalizační úloha (Zopfli)

bzip2

- bzip2 byl prvně vydán v roce 1996 britským programátorem Julianem Sewardem
- je založen na BWT, MTF a Huffmanovu kódování
- bzip2 komprimuje většinu souborů efektivněji než Deflate (.zip, .gz), ale je znatelně pomalejší
- bzip3 je nástupce bzip2 z roku 2022 dosahující vyšších kompresních poměrů a pomalejší rychlosti
- je založen na BWT, LZ77 a aritmetickém kodéru

7-Zip

- především Lempel–Ziv–Markov chain algorithm (LZMA)
- algoritmus vyvinutý programátorem Igorem Pavlovem pro jeho archivační program 7-Zip v roce 1999
- skládá se z LZ77, Markovových řetězců a aritmetického kodéru (range coder)
- použit také v XZ Utils (liblzma, xz)

LZ4, lzop

- LZ4 se slovníková metoda zaměřená na rychlost komprese a dekomprese z roku 2011
- implementuje LZ77 a na rozdíl od jiných algoritmů nepoužívá entropické kódování
- existuje odnož LZ4 HC, která dosahuje vyšších kompresních poměrů za cenu rychlosti, bitstream je plně kompatibilní s LZ4
- lzop implementuje slovníkovou metodu LZO (Lempel–Ziv–Oberhumer, vychází z LZ77), je zaměřena na rychlost dekomprese
- lzop byl poprvé vydán v roce 1997, nemá oficiální specifikaci

Brotli

- vyvinut v roce 2013 společností Google
- používá kombinaci LZ77, Huffmanova kódování a kontextového modelování
- posuvné okno až do velikosti 16 MiB
- lepší kompresní poměr než Deflate především nad soubory HTML, CSS a JavaScript

Zstandard

- kombinace LZ77 a entropického kódování
- uvolněn v roce 2015 společností Facebook
- běžně dosahuje podobných kompresních poměrů jako Deflate
- oproti knihovnám implementující Deflate dosahuje vyšší kompresní i dekompresní rychlosti
- při maximální kompresi Zstd dosahuje kompresních poměru blízko LZMA

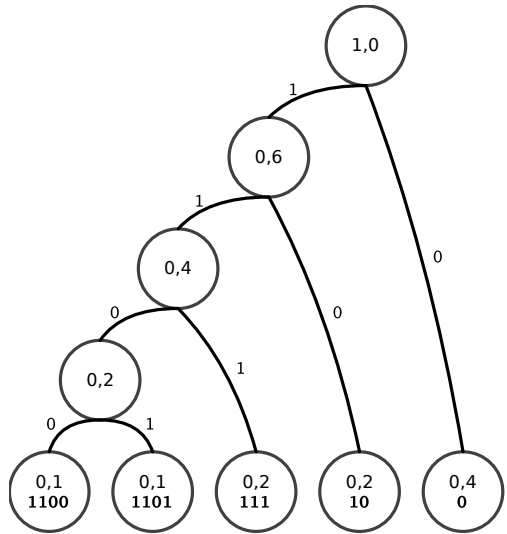
Huffmanovo kódování

- populární metoda
- adaptuje se na data
- nejlepší výsledky pro pravděpodobnosti rovny záporné mocnině 2
- symboly jsou listy v binárním stromě s hranami udavajícími kód
- použití: bzip2, Deflate
- konstrukce stromu:
  1. seřadit symboly sestupně dle jejich pravděpodobností
  2. vyber 2 symboly s nejnižší pravděpodobností, spoj je do nového uzlu
  3. pokračuj krokem 2, dokud je co spojovat
- u adaptivní varianty je nutné opravovat strom

Huffmanovo kódování

příklad

p					kód
0,4				0	0
0,2			0	1	10
0,2		1	1	1	111
0,1	1	0	1	1	1101
0,1	0	0	1	1	1100

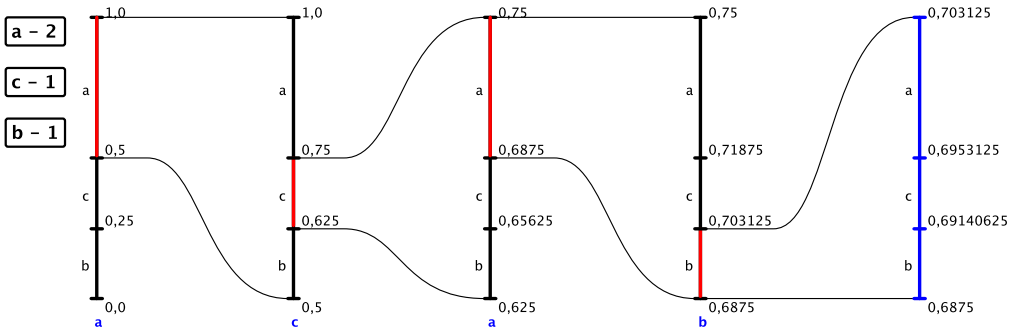


Aritmetické kódování

- optimální kódy pro libovolné pravděpodobnosti výskytu symbolů
- metoda přidělí jeden kód celému kódovanému souboru dat
- začíná se s intervalem, který se podle pravděpodobností kódovaných symbolů neustále zužuje
- zužování intervalu vyžaduje další bity, takže délka kódu postupně roste
- myšlenka komprese: symbol s vyšší pravděpodobností zúží interval méně (přidá méně bitů) než symbol s pravděpodobností nižší
- praktické implementace musejí pracovat s celými čísly
- použití: kontextové kodéry, multimédia, bzip3

Aritmetické kódování

příklad



LZ77

- publikovali A. Lempel a J. Ziv v roce 1977
- mnoho modifikací
- použita v Deflate
- pohyblivé okno (sliding window)
- dvě části: vyhledávací a předvídací buffer
- v praxi tisíce vs. desítky bajtů
- tvoří značky `(offset, délka, symbol)`

←...`east#easily#t`|`eases`|...←

- 2× shoda „eas“ na pozicích 8 a 13
- vytvoří se značka `(13, 3, 'e')`
- prvky značky se zakódují na odpovídajícím počtu bitů $\lceil \log_2 S \rceil, \lceil \log_2 (L-1) \rceil, \lceil \log_2 A \rceil$, kde A je velikost abecedy

LZ77

←...east#easily#t|rashe...←

- při nenalezení shody se generuje značka `(0, 0, 'r')`

←...east#easily#t|tttt...←

- shoda může překročit hranici vyhledávácího bufferu, zde `(1, 5, 't')`
- to je také důvod pro počet bitů $\lceil \log_2(L-1) \rceil$ délky
- LZ77 předpokládá, že fragmenty se vyskytují blízko u sebe
- existuje množství vylepšení této metody: např. zrušení posledního pole značky

Burrowsova–Wheelerova transformace (BWT)

- publikovali ji v roce 1994 M. Burrows a D. J. Wheeler
- bloková transformace, data nekomprimuje, pouze změní pořadí symbolů
- stejné symboly se budou vyskytovat za sebou
- vytvoří se všechny rotace vstupního řetězce, které se seřadí, výstupem je řetězec posledních znaků se seřazené posloupnosti
- výstup se podrobí Move-to-front (MTF) transformaci, jejíž výstup se následovně zakóduje Huffmanovým kodérem, který přiřazuje symbolům blízkým nule krátké bitové kódy
- MTF nahrazuje symboly vstupní abecedy za jejich indexy ze seznamu symbolů, aktuálně kódovaný symbol je přesunut v tomto seznamu vždy na začátek; často vyskytující se symboly jsou umístěny blíže začátku seznamu
- použita v bzip2 a bzip3

Burrowsova–Wheelerova transformace (BWT)

- vstup: "swiss miss$"
- seřazené rotace: " miss$swiss", "iss miss$sw", "iss$swiss m", "miss$swiss ", "s miss$swis", "ss miss$swi", "ss$swiss mi", "swiss miss$", "s$swiss mis", "wiss miss$s", "$swiss miss"
- výstup: "swm sii$sss"
- výstupní bajty:
  115, 119, 109, 32, 115, 105, 105, 255, 115, 115, 115
- po MTF transformaci:
  115, 119, 111, 35, 3, 108, 0, 255, 2, 0, 0

Dataset

Silesia compression corpus

dickens	Collected works of Charles Dickens	text
mr	Medical magnetic resonanse image	picture
ooffice	A dll from Open Office.org 1.01	exe
osdb	Sample database in MySQL format	database
reymont	Text of the book CHŁOPI by W. Reymont	pdf
sao	The SAO star catalog	bin data
xml	Collected XML files	html

Kompresní poměr

	dickens	mr	ooffice	osdb	reymont	sao	xml
lzop	2.2803	2.4523	1.8408	2.3925	3.0753	1.2517	6.7689
lz4	2.3244	2.3784	1.7391	2.5489	3.2072	1.2787	7.0204
gzip	2.6461	2.7138	1.9907	2.7138	3.6396	1.3613	8.0709
zopfli	2.7783	2.9060	2.0565	2.7966	3.9169	1.3847	8.5281
rar	3.2698	3.5758	**2.6634**	3.0469	4.2394	1.3068	10.7838
lzma	3.6006	3.6237	2.5351	3.5462	5.0383	**1.6388**	12.2943
zstd	3.5770	3.2103	2.3672	3.2550	4.9177	1.4502	11.7940
brotli	3.6044	3.5317	2.4818	3.5812	4.9747	1.5812	12.4145
bzip2	3.6407	4.0841	2.1492	3.5984	5.3178	1.4678	12.1157
bzip3	**4.5625**	**4.7031**	2.4346	**4.4589**	**6.7591**	1.5517	**13.1697**

Rychlost dekomprese (MiB/s)

	dickens	mr	ooffice	osdb	reymont	sao	xml
lzop	264.8	357.4	252.8	421.8	311.3	352.8	490.1
lz4	**665.7**	**699.1**	**592.6**	**745.6**	**658.3**	**606.6**	**717.9**
gzip	148.4	148.3	123.5	153.6	170.3	127.6	229.6
rar	158.3	134.6	109.6	152.9	169.4	124.6	193.8
lzma	70.3	53.6	37.6	55.7	92.1	24.5	170.4
zstd	439.8	424.4	362.1	500.9	501.6	409.2	670.7
brotli	263.4	206.2	152.3	231.2	329.1	129.0	439.4
bzip2	27.1	35.8	21.2	28.1	32.7	18.5	44.2
bzip3	6.8	6.8	7.0	6.6	7.2	5.7	16.2

Rychlost komprese (MiB/s) se zaměřením na kompresní poměr

	dickens	mr	ooffice	osdb	reymont	sao	xml
lzop	6.2	4.1	7.8	12.8	3.8	7.7	15.7
lz4	11.6	8.5	14.6	24.5	9.3	17.4	22.0
gzip	11.4	7.7	11.4	23.1	6.7	14.4	29.8
zopfli	0.0	0.0	0.0	0.0	0.0	0.0	0.0
rar	29.9	31.9	34.9	37.5	39.7	35.2	49.9
lzma	1.5	1.6	2.5	2.2	1.6	2.3	3.2
zstd	1.7	2.3	3.0	2.5	1.9	3.0	2.0
brotli	0.5	0.4	0.3	0.5	0.5	0.4	0.8
bzip2	13.0	19.2	14.0	14.5	14.5	12.0	10.5
bzip3	6.3	6.5	6.2	6.1	6.1	5.2	11.6

Rychlost komprese (MiB/s) se zaměřením na rychlost

	dickens	mr	ooffice	osdb	reymont	sao	xml
lzop	249.2	365.7	293.3	356.2	300.9	276.6	463.4
lz4	**486.0**	**500.4**	**366.6**	**565.7**	**371.7**	**432.2**	**509.7**
gzip	66.1	78.5	44.1	66.3	73.4	38.4	115.8
rar	32.0	32.7	35.3	38.6	39.0	37.1	47.6
lzma	3.5	2.5	4.1	4.2	2.8	4.4	3.3
zstd	206.8	352.1	325.9	356.2	234.0	329.3	463.4
brotli	202.5	288.1	217.3	331.6	274.7	192.1	463.4
bzip2	13.8	19.6	13.4	14.1	15.5	10.5	12.6
bzip3	6.6	6.6	6.1	6.4	6.7	5.2	11.3

Závěry

- bzip3 je v současnosti nejúčinnější kompresor, přibližně o 15–25 % lepší než bzip2
- Zopfli zlepší kompresní poměr proti gzipu o 3–8 %, stále stejná kompresní metoda Deflate
- lzop, LZ4, gzip jsou relativně rychlé, ale s nízkým kompresním poměrem
- LZMA, Zstd, Brotli dosahují relativně vysokých kompresních poměrů, nicméně bzip2 a bzip3 jsou ještě o něco účinnější
- LZ4 má s náskokem nejrychlejší dekompresi (leč nízký kompresní poměr), Zstd je také poměrně rychlé
- LZMA, bzip2 a bzip3 jsou při dekompresi hodně pomalé
- nástroje lze zaměřit na rychlost komprese (přinese snížený kompresní poměr), tady vyhrává LZ4

Zobecnění

- LZ4 má velmi rychlou kompresi a dekompresi
- Deflate je středně účinný, zato všude podporovaný (starý standard)
- pro velmi vysoké kompresní poměry je tu bzip3

Dodatky a zdroje

- všechny zkomprimované soubory dekomprimovány a porovnány s originálem
- měřeno na AMD Ryzen Threadripper 2990WX 32-Core Processor
- jednovláknová komprese i dekomprese (ale některé nástroje umějí využít více vláken)

Zdroje:

- D. Solomon. Data Compression, The Complete Reference, Fourth Edition, 2007, ISBN 978-1-84628-602-5.
- `https://sun.aei.polsl.pl/~sdeor/index.php?page=silesia`