Compressing Integers for Fast File Access

Hugh E. Williams
Justin Zobel

Benjamin Tripp
COSI 175a: Data Compression
October 23, 2006

Introduction

- Many data processing applications depend on access to integer sets of data, such as in scientific and financial data
- Compression schemes allow for faster retrieval of stored text in document databases, since computational cost of decompressing can be offset by reductions in disk seeking and transfer costs

- This paper set out to see if similar gains could are possible with integer sets of data
- Experimented using multiple compression technique: Elias gamma and delta codes, and Golomb codes, and variable-byte

Variable-Byte Coding

- 7 bits in each byte are used to code an integer, and the last bit is a zero to indicate short, or a 1 to indicate there are more digits
- Useful for storing small data sets, or with data sets where the structure of data is unknown and other coding techniques cannot be selectively applied
- Variable-Byte coding requires few CPU operations to decode

Elias Gamma Code

- A positive integer x is represented by 1 + floor($\log_2 x$) in unary (which is floor($\log_2 x$) 0 bits followed by a 1 bit) followed by the binary representation of x without its most significant bit
- Efficient for small integers, but not suited to large integers

Elias Gamma Code (cont.)

- Example: 9 is represented as 0001001, since 1 + floor($\log_2 9$) = 4, or 0001 in unary and 9 is 001 in binary with the most significant bit removed.

Elias Delta Code

- For an integer *x*, a delta codes stores the gamma code representation of $1 + \log_2 x$, followed by the binary representation of *x* without the most significant bit
- Example: 9 is represented 00100001, since the Gamma code of $1 + \log_2 x$ is 00100 and 9 is 001 in binary with the most significant bit removed

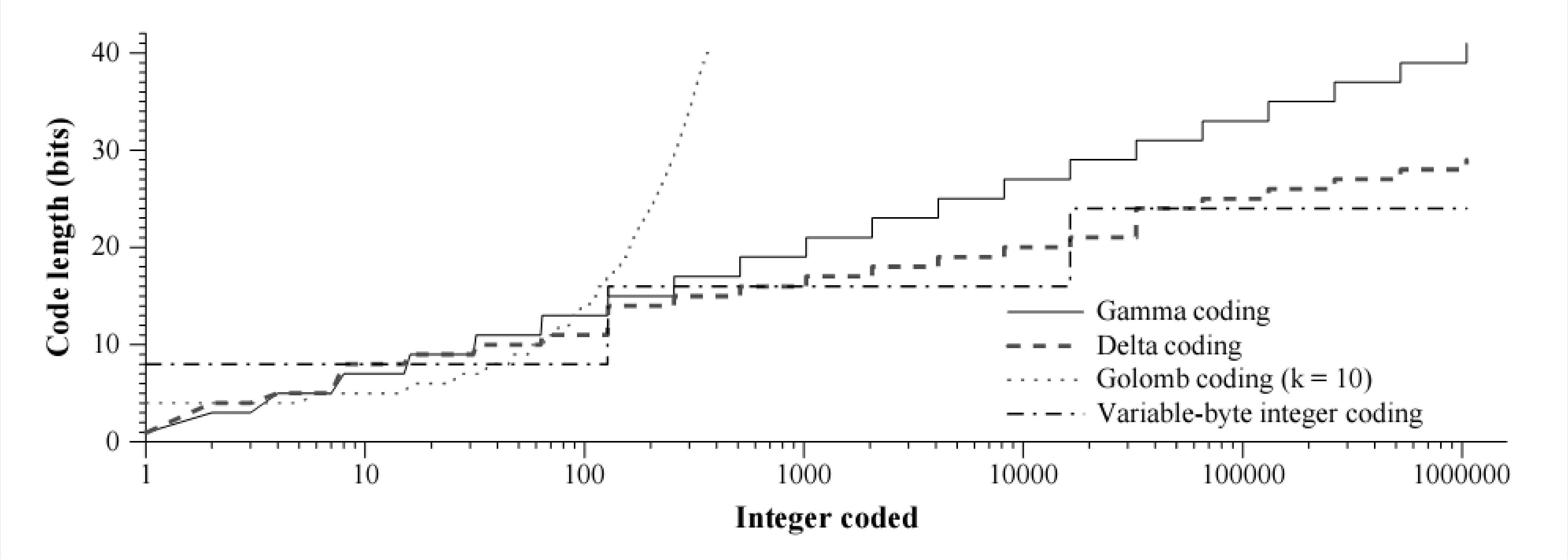
Golomb Codes

- Compression uses a parameter *k* in algorithm
- Parameter *k* must be calculated and often stored with each array coded integers. The choice *k* has a significant impact on the compression

Golomb Codes (cont.)

- A positive integer v is represented in two parts:
  - First is a unary representation of the quotient $q = \text{floor}((v-1)/k) + 1$
  - Second is a binary representation of the remainder $r = v - q * k - 1$

Comparing Sizes



Code lengths in bits of Elias gamma and delta codes, a Golomb code with k = 10, and variable-byte integer codes for integers in the range 1 to around 1 million

Examples

Decimal	Uncompressed	Elias Gamma	Elias Delta	Golomb ($k = 3$)	Golomb ($k = 10$)	Variable-byte
1	00000001	1	1	1 10	1 001	0000001 0
2	00000010	0 10	0 100	1 11	1 010	0000010 0
3	00000011	0 11	0 101	01 0	1 011	0000011 0
4	00000100	00 100	0 1100	01 10	1 100	0000100 0
5	00000101	00 101	0 1101	01 11	1 101	0000101 0
6	00000110	00 110	0 1110	001 0	1 1100	0000110 0
7	00000111	00 111	0 1111	001 10	1 1101	0000111 0
8	00001000	000 1000	00 100000	001 11	1 1110	0001000 0
9	00001001	000 1001	00 100001	0001 0	1 1111	0001001 0
10	00001010	000 1010	00 100010	0001 10	01 000	0001010 0
11	00001011	000 1011	00 100011	0001 11	01 001	0001011 0
12	00001100	000 1100	00 100100	00001 0	01 010	0001100 0
13	00001101	000 1101	00 100101	00001 10	01 011	0001101 0
14	00001110	000 1110	00 100110	00001 11	01 100	0001110 0
15	00001111	000 1111	00 100111	000001 0	01 101	0001111 0
16	00010000	0000 10000	00 1010000	000001 10	01 1100	0010000 0
20	00010100	0000 10100	00 1010100	0000001 11	001 000	0010100 0
25	00011010	0000 11001	00 1011001	000000001 10	001 101	0011010 0
30	00011110	0000 11110	00 1011110	00000000001 0	0001 000	0011110 0

Test Data

- WEATHER: A collection of weather station measurements
- TEMPS: Smaller temperature data set from single weather station
- MAP: Elevation levels for all points on a land contour map
- LANDSAT: Frequency spectrum of layered satellite data
- PRIME: Collection of the first one million prime numbers
- VECTOR: Collection of sorted integer arrays from file indexes

Selected compression

- More efficient representation is possible by selectively applying variable-bit codes to the VECTOR, TEMPS, and PRIME collections.
- VECTOR: Use separate local Golomb parameters for each list of document identifiers and word positions, and gamma codes for storing counts of identifiers in each list
- TEMPS: Use two different Golomb parameters for time values and for temperature values

Compression Performance

Scheme	TEMPS	PRIME	WEATHER	LANDSAT	MAP	VECTOR
Integers ($\times 10^6$)	0.72	1.00	10.00	41.01	197.80	165.29
Entropy	12.57	19.93	2.91	6.02	6.50	17.40
Elias gamma coding	33.50	44.65	16.57	8.42	11.02	11.42
Elias delta coding	23.80	30.84	12.82	8.09	10.19	9.78
Golomb coding	26.54	24.36	13.64	6.60	7.50	13.47
Variable-byte coding	22.11	30.74	12.59	8.00	8.63	11.97
GZIP	10.21	10.91	3.00	4.53	0.24	11.82
Selected compression	7.14	5.52	12.59	6.60	7.50	7.87

Compression performance of integer coding schemes, in bits per integer. The first line shows the size of each data set.

Sequential Retrieval

Scheme	TEMPS	PRIME	WEATHER	LANDSAT	MAP	VECTOR
Uncompressed 32-bit integers	2.34	2.31	2.19	2.39	2.48	1.98
Elias gamma coding	1.05	1.03	1.96	3.08	2.49	2.24
Elias delta coding	1.40	1.42	2.29	2.86	2.46	2.47
Golomb coding	1.77	1.85	2.31	3.25	3.13	2.30
Variable-byte coding	2.12	1.42	3.67	4.45	5.41	2.69
GZIP	3.83	4.14	12.72	9.25	25.68	4.50
Selected compression	2.42	2.72	3.67	3.25	3.13	2.78

Sequential stream retrieval performance of integer coding schemes, in megabytes per second. In each case data is retrieved from disk and, in all bu the first case, decompressed.

Random Access

- For random access a separate file of offsets for each collection
- Each offset represents a file position in the collection that is the begging of a block of 1,000 integers
- Report the speed of randomly seeking to 10% of the offsets in each collection and retrieving blocks of 1,00 integers at each offset

Random Retrieval

Scheme	TEMPS	PRIME	WEATHER	LANDSAT	MAP	VECTOR
Uncompressed 32-bit integers	0.31	0.49	0.39	0.33	0.34	0.70
Elias gamma coding	0.23	0.33	0.33	0.61	0.58	0.67
Elias delta coding	0.32	0.45	0.33	0.50	0.48	1.00
Golomb coding	0.34	0.49	0.46	0.68	0.54	0.83
Variable-byte coding	0.35	0.58	0.58	0.51	0.49	0.75
Selected compression	0.92	0.83	0.58	0.68	0.54	1.29

Random-access retrieval performance of integer coding schemes, in megabytes per second. In each case data is retrieved from disk and, in all but the first case, decompressed.

Conclusion

- Storing integers in compressed form improves the speed of disk retrieval for both sequential and random access to files.
- Best performance is achieved by selecting a compression scheme that specific to the data.
- Disk retrieval costs are reduced by compression since the cost of retrieving a compressed representation from the disk and the CPU cost of decompressing is less than just retrieving an uncompressed representation.