INFORMATION TECHNOLOGY

Трансформерная технология таймерного кодирования и сжатие информации

Е.А.Осадчий[1], А.Е. Осадчий[2], О.А. Горбунов[1], Р.В. Скуратовский[3]

https://doi.org/10.31174/SEND-NT2018-157VI17-15

[1]Киевский национальный университет имени Тараса Шевченка, Киев, Украина
[2]Университет Финикса, Аризона, США
[3]Межрегиональная Академия управления персоналом, Киев, Украина
Corresponding author. E-mail: sanaltd@hotmail.com

Paper received 27.01.18; Accepted for publication 02.02.18.

Аннотация. Наводятся методы и средства таймерного кодирования уменьшающие комбинаторные значения двоичных чисел начиная с наполнения ими байта. Для этого, используются понятие таймерная метка и вводятся дополнительные унарные коды, адекватные по количественному значению тем, которые содержатся в кодировочной таблице ASCII. В результате, оптимизируется не только интеллектуальная человеко-машинная обработка символов языка коммуникации, а и процесс сжатия данных.

Ключевые слова: *таймерная метка, язык коммуникации, унарный код, количественное значение данных, таблица ASCII*

Введение. Для изложения результатов данного исследования используем принятые нами ключевые определения. Для концентрации на изложении сути исследования не будем анализировать и доводить их истинность. Остальные определения, в рассматриваемой предметной области (ПО), семантически однозначны и поэтому не наводятся.

Цифровые компьютеры (ЦК), предназначены для трансформации (превращения) цифровой информации. *Информация* – осмысленный интеллектом и представленный на языке коммуникации (ЯК) результат преобразования и анализа данных. В ЦК с любым основанием, системой счисления (СЧ), например, двоичной ($СЧ_2$), *данные* - количественное значения натурального числа.

Объем обрабатываемой информации существенно влияет на производительность компьютера. Поэтому, проблема повышения эффективности сжатия данных (уменьшением количественного значения натурального числа) всегда будет актуальной задачей компьютеризации.

Краткий обзор публикаций по теме. Существует обоснованный пессимизм ожидания прогресса в рассматриваемой ПО, т.к. считается, что теоретические пределы сжатия информации уже практически достигнуты применением существующих методов (алгоритмов) и средств (реализующих программ). Развитие компьютерной техники уменьшило значимость сжатия данных, но остаются проблемные места при их распределенной обработке, где оно все еще крайне востребовано. Объясняется относительно низкой скоростью обработки данных в их хранилищах и каналах коммуникации. Основные из них:

1. Передача по электронной почте (особенно для мобильных устройств).

2. Накопление данных на интернет-сайтах и порталах.

3. Экономия свободного места на носителях информации.

В перечисленных направлениях существует множество прикладных публикаций и новых технических решений. Но мы сконцентрируемся на анализе теоретических и практических достижений непосредственно в ПО сжатия информации.

Выделяют несколько способов сжатия. Их можно разделить на две основные категории. Это сжатие без потерь и сжатие с определенными потерями. Первая категория актуальна только тогда, когда есть необходимость восстановить данные с высокой точностью, не потеряв ни одного бита исходной информации. Считается, что приоритетным случаем, в котором необходимо его использование, это сжатие текстовых документов. Но этот вывод некорректен, т.к., всегда может возникнуть ситуация, когда потребуется полная адекватность исходной информации. Например, при восстановлении заархивированных данных. Причиной популярности алгоритмов сжатия с потерями является простота их реализации. Еще такие алгоритмы обеспечивают относительно высокую степень сжатия, при этом сохраняя достаточное количество исходной информации для ее идентификации. Использование подобных алгоритмов в большинстве случаев подходит для сжатия оцифрованных аналоговых данных, например, звуков или изображений. В таких случаях конечный результат может сильно отличаться от оригинала и все же он, как правило, удовлетворяет критериям их дальнейшей интеллектуальной обработки. Алгоритмы сжатия без потери информации позволяют обеспечить максимально точное восстановление исходных данных, когда любые потери исключены. Однако у данного метода есть один существенный недостаток: при их использовании классические методы и средства сжатия, мягко говоря, неэффективны. Он основывается на вероятностно-статистическом подходе и условно разбивается на два типа: блочные и адаптивные методы. При использовании блочных алгоритмов происходит высчитывание каждого отдельного блока информации с последующим добавлением его к блоку, который уже прошел сжатие. Адаптивные алгоритмы предусматривают вычисление вероятностей по той информации, которая уже была обработана в процессе сжатия. К этому типу методов можно отнести адаптивный алгоритм Шеннона-Фано. Это — один из первых алгоритмов

сжатия, который впервые сформулировали эти американские учёные. Данный метод сжатия имеет большое сходство с алгоритмом Хаффмана, который появился на несколько лет позже и является логическим усовершенствованием предыдущего. Среди профессионалов он известен как «адаптивный жадный алгоритм оптимального префиксного кодирования алфавита с минимальной избыточностью» [1]. Он позволяет строить кодовую схему в поточном режиме (без предварительного сканирования данных), не имея никаких начальных знаний из исходного распределения, что позволяет за один проход сжать данные. Преимуществом этого способа является возможность быстрого кодирования. Он до сих пор считается непревзойденным по эффективности в рассматриваемой ПО. Для дальнейшей оптимизации оперирования сжатыми данными вся преобразовываемая информация распределяется на несколько отдельных блоков. Происходит целостное трансформирование информации. В конечном итоге, все будет зависеть только от программы, которая используется для сжатия информации. При выборе утилиты для сжатия, учитывается то, что эффективность сжатия зависит от типа преобразуемой информации. Так, например, эффективность сжатия текстовых файлов и документов может достигать 90%. А вот при сжатии изображений - всего в несколько процентов. Самым легковесным (занимает всего1,31 Мб) и доступным в мире архиватором, полностью совместимым с актуальной на сегодня Windows 10, считается 7-Zip [2].

Сжатие информации является составляющей важнейших методов и средств цифровой обработки - кодирования и шифрования. Первичное прямое и обратное кодирование (информация ↔ данные) обеспечивается применением таблиц символов ASCII (7 битной, однобайтовой), Unicode (двух байтовой) и всех их производных. Это кодирование осуществляется на памяти прямого доступа ограниченного размера, что обеспечивает относительно незначительные и одинаковые интервалы времени доступа ко всем ее элементам. В ней, в качестве адресных индексов, использовано количественное значение каждого разряда двоичного числа, которое может быть воспроизведено в 7 битах, 1 или 2 байтах. Оно расписано последовательно по мере их получения двоичным сложением. В результате, строки двоичных кодов таблиц поочередно содержат: 128, 256, 65536 записей из 7, 8, 16 бит. Каждому двоичному коду поставлено в соответствие обозначение символа/символов алфавита ЯК. Прямое и обратное кодирование осуществляется аппаратно-алгоритмическими средствами периферийных устройств (клавиатура, дисплей и другими). Преобразование последовательностей из 16, 8 и 7 бит в символы происходит на базовом уровне, так что при работе на ЦК пользователь больше не сталкивается с двоичным представлением. Такое преобразование задействуется сразу же после тестирования BIOS. В качестве международного стандарта признано кодирование 128 битной таблицы ASCII. Оно охватывает символы алфавитов основных ЯК ($СЧ_{10}$, английские буквы, управляющие и другие символы). Производные таблицы от ASCII и Unicode покрывают все остальное многообразие ЯК. Ограниченные возможности таймерного кодирования (ТК) используются в кодировочных таблицах типа ASCII (прямой доступ к их кодам) и компьютерном таймере (прерывание «по таймеру»). Создание дополнительных возможностей ТК, для решения проблем трансформации информации возможно при внесении ряда архитектурных изменений в двоичный ЦК. Концептуально они изложены в [3, 4, 5], но применительно к очерченной проблеме сжатия информации будут детализированы нами в содержательной части публикации.

Цель проводимого исследования показать, как методы и средства ТК способствуют повышению, эффективности сжатия данных не зависимо от величины количественного значения числа и СЧ в которой оно представлено.

Материалы и методы. После первичного кодирования натуральных двоичных чисел в ASCII ограниченной размерами 7, 8 или 16 битных записей (преимущественно - байтами) двоичный ЦК, как правило, оперирует с количественно структурированными с возрастанием в 10^3 или $1024_{(2)}$ размерами записей линейной памяти производными от базового байта. Указанные размеры памяти обозначают приставками (кило, мега, гига, тера, …, иотта). В результате, приходится оперировать большими и даже сверх большими двоичными числами. Это ограничивает возможности распараллеливания вычислений и применение памяти прямого доступа. Все операции над такими числами осуществляются исключительно их последовательным преобразованием операциями производными от затратного двоичного сложения. Поэтому, имеющиеся аппаратно-временные преимущества памяти прямого доступа нижнего (базового) уровня кодирования ASCII тут обесцениваются. Существенно не помогают технические и алгоритмические уловки в виде использования сумматоров, других производных устройств, а также программ для ускорения обработки больших объёмов информации. Базовая операция сложения, реализуемая программно-аппаратными средствами, при определении количественное значения числа позиционной СЧ характеризуется степенной зависимостью увеличения времени счета на каждом старшем разряде. Подтверждением сказанного, является визуально наблюдаемая временная задержка при осуществлении счета различных интервалов времени таймером ЦК. Вместе с тем, известно, что чем проще операция, тем меньше затрат требуется на ее реализацию. Такое утверждение в ПО ЦК основываться на гипотезе многомерности и взаимодействия: материи (символов, устройств, …), пространства (бит, байт, …) и времени (секунда, минута, …). Известно, что базовой операцией сложения, используемой в ЦК для выявления свойства количества данных (натуральных чисел), является операция сдвига. Она реализуется линией задержки, в простейшем случае, это обычное электрическое сопротивление. Простота технической реализации и быстродействие операции (сегодня, она ограничена скоростью света) дают значительное преимущество при альтернативной «таймерной» идентификации количественного значения числа, записанного в единичной (унарной) СЧ. Оно может быть отображено позицией единичной ячейки в линейной памяти и затем однозначно иден-

тифицироваться в требуемой СЧ генератором стабильных временных интервалов. Сегодня, всемирным стандартом признана частота атомных часов по стабильно минимальной частоте которых синхронизируют работу всех производных частотных генераторов. В ЦК основными частотными генераторами являются процессоры. На определенных частотах, они легко синхронизируются с эталонной. При представлении количественного значения отдельного байта в ASCII, но уже в производной $СЧ_{1\text{-}2}$, потребуются единственный символ «1» и шесть символов «0». Позиция этого байта в строке таблицы, как и при кодировании в $СЧ_2$, определяется механизмом доступа к 16 записям (байтам) ее памяти. Таким образом, по сравнению с наполнением количественного содержания байта в $СЧ_2$ бинарная комбинаторика упрощается. На примере модернизированной 128 битной кодировочной таблицы ASCII покажем преимущество ТК при сжатии количественного значения содержащихся данных. В ней, 7 битное количественное наполнение байта представлено полной комбинаторикой бит. Она отображается на линейной памяти (записи) из128 ячеек.

По аналогии с методом адаптивных алгоритмов, наведенным в [1], проведем вероятностно- статистический анализ символов входного, генерируемого и выходного (количественно сжатого) двоичного кода. Как показывает статистика, полная комбинаторика 8 разрядного двоичного кода, представленная в ASCII для входного числа в $СЧ_{(1\text{-}2)}$ скорее исключение, чем правило. Но именно для нее возможно самое эффективное количественное сжатие. В качестве примера рассмотрим число 128_{10} с признаками количества (по умолчанию, это пробелы до и после слитного написания цифр числа). Тогда, осмысливаемое нами количественное значением числа 128 → 1, 2, …, 128, так как мы его считаем до 128. В ЦК для этого используют смешанные СЧ, например, запись числа в $СЧ_{(2\text{-}8)}$ кодировке. Но они уже требуют использования производных операций двоичного сложения.

Применительно к кодировке в ASCII это может выглядеть так. Выбирается бинарный код, идентифицируемый в строке символов ЯК как нечитаемое (запорченное) содержания байта, которому присваивается значение ТМ. Этот код записывается в байт который дописывается возле байта с последней значащей позицией «1» идентифицирующей количество числа в $СЧ_{(1\text{-}2)}$. При условии идентификации в памяти прямого доступа позиции записи байта с признаком количественного значения числа, легко восстановить и само количественное значение числа. Если у ЦК имеется возможность идентифицировать ТМ на уровне бита, например, как «Ŧ», тогда фактически читаемый двоичный код будет содержать 6 символов «0», пустой (запорченный) символ «Ŧ» и символ «1». ТМ является признаком воспроизведения числа и всего ряда предшествующих двоичных кодов. Посимвольно, это будет выглядеть так: 0000000Ŧ1. В результате, для рассматриваемого варианта, нам удалось последовательность из 128 символов (1, 2, …, 128) записать комбинаторикой из 8 символов по 3 (6→ «0», 1→ «Ŧ» и 1→ «1»). Налицо количественное сжатие исходного числа в 16!!! раз. Количественное сжатие других 8 разрядных, но фактически 7 битных входных двоичных кодов конечно могут быть менее эффективными, но и они впечатляют, так как возможность количественного сжатия содержания байта классическое сжатие вообще исключает. Нами, с использованием ТК, разработаны алгоритмы обеспечивающие сжатия на один бит содержание байта кодировочных таблиц. А это уже позволяет архивировать любое наполнение байта, в том числе и с ранее архивированной информацией. В связи с трудоемкостью их изложения, здесь они не наводятся. Сжатие ТК основывается на предположении «избыточности» полной комбинаторики двоичного кодирования заложенной в ASCII. Не составляет особого труда, даже без предварительного статистического анализа использования в сообщениях ЯК всех их символов, исключить из всего перечня адекватных двоичных кодов, как минимум 64 комбинации символов, как практически неиспользуемых для определенных видов информации. Это возможно, так как невозможно себе представить один код сжатия для различного количественного значения числа. Особенно эффективным, как и для стандартных методов сжатия, оказалось сжатие определенных видов информации, но в отличии от них потери памяти и времени не пропорциональны размерам сжимаемой информации. Для этого, также предварительно проводится вероятностно статистический анализ входного структурированного, например, байтами, большого двоичного числа. В результате создается такой генератор содержащихся в нем символов ЯК, который воспроизводит их в порядке встречаемости в этом входном сообщении. Например, для текстовой информации, математически доказана в [4] возможность его создания.

Результаты и их обсуждение. Существует дальнейшая перспектива использования статистически вероятностных методов для преобразования двоичных кодов увеличенной размерности к 7 битной кодировке ASCII, а также для оптимизации генераторов сообщений ЯК. Критерием последних, может служить следующая выявленная нами закономерность, используемая для оценки наибольшего количества шагов перестановочного генератора CG при восстановлении входного сообщения по таймерной метке. Пусть, например, известно, что средне статистический текст имеет частоты появления каждого символа x_i из алфавита A равные f_i. Соответственно, в тексте X длина которого равна m символ x_i содержится $k_i = nx_i$ раз. Тогда, в наихудшем случае перестановочный символьный генератор сделает *N* шагов для восстановления текста.

$$N = \frac{m!}{k_1!k_2!\ldots k_m!}$$

При этом выполняется дополнительное ограничение $\sum_{i=1}^{I} k_i = m$.

Используя ТК в ПО текстовой информации нами предложены и новые аппаратные средства [7], приближающие решения проблем межязыковой коммуникации, в частности, перевода с одного ЯК на другой.

Выводы. В результате проведенного нами иссле-

дования, показана возможность повышения эффективности сжатия информации с применением методов и средств ТК. Приведено описание основных из них. Они могут оказаться полезными для создания аппаратной, алгоритмической и программной реализации эффективного сжатия. А оно, всегда будет актуальной задачей компьютеризации.

ЛИТЕРАТУРА

1. Donald E. Knuth, «Dynamic Huffman Coding», Journal of Algorithm, 6(2), 1985, pp 163–180.
2. Salomon, D. and Bryant, D. and Motta, G. Handbook of Data Compression. — Springer London, 2010. — P. 411-414. — 1361 p. — ISBN 9781848829039.
3. Осадчий Є.О. Трансформерні технології побудови машин і механізмів. - К.: Науковий світ, 2004.- 167 с.
4. Метод быстрого таймерного кодирования текстов / Р.В. Скуратовский // Кибернетика и системный анализ. — 2013. — Т. 49, № 1. — С. 154-160.
5. Декларативний патент України на корисну модель № 56185 МПК6 A61 H 3/00, A 62 D 7/00. Інерційний лічильник / Анісімов А.В., Гриценко В.І, Осадчий О.Є., Осадчий В.Є., Осадчий Є.О.- Опубл. 11.01.11.- Бюл. № 1.- 8 с.
6. Turing A. M. On Computable Numbers, with an Application to the Entscheidungsproblem. A Correction // Proceedings of the London Mathematical Society — 1938. — Vol. s2-43, Iss. 6. — P. 544–546. — ISSN 0024-6115; 1460-244X — doi:10.1112/PLMS/S2-43.6.544.
7. Патент України на корисну модель № u 121740, МПК G 06 F 15/38 Пристрій для перетворення кодів з однієї мови на іншу / Крак Ю.В., Терещенко В.М., Осадчий Є.О., Горбунов О.А.- Опубл. 11.12.17.- Бюл. № 23.- 7с.

REFERENCES

1. Donald E. Knuth, «Dynamic Huffman Coding», Journal of Algorithm, 6(2), 1985, pp 163–180.
2. Salomon, D. and Bryant, D. and Motta, G. Handbook of Data Compression. — Springer London, 2010. — P. 411-414. — 1361 p. — ISBN 9781848829039.
3. Osadchy E.O. Transformer technologies for the construction of machines and mechanisms. - K .: Scientific World, 2004. - 167 p.
4. Method of fast timer encoding of texts / R.V. Skuratovskyy // Cybernetics and system analysis. - 2013. - Vol. 49, No. 1. - P. 154-160.
5. Ukraine's Declarative Patent for Utility Model No. 56185 MPK6 A61 H 3/00, A 62 D 7/00. Inertia counter / Anisimov A.V, Gritsenko V.I., Osadchyy O.Y., Osadchyy V.Y., Osadchyy Y.O. - Publ. 11.01.11.- Bul. No. 1.- 8 p.
6. Turing A. M. On Computable Numbers, with an Application to the Ent scheidungs problem. A Correction // Proceedings of the London Mathematical Society — 1938. — Vol. s2-43, Iss. 6. — P. 544–546. — ISSN 0024-6115; 1460-244X — doi:10.1112/PLMS/S2-43.6.544.
7. Patent of Ukraine for utility model № u 121740, IPC G 06 F 15/38 Device for converting codes from one language to another / Krak Yu.V., Tereshchenko V.M., Osadchyy Y.O., Gorbunov O.A.- Pubwished 11.12.17.- Bul. No. 23.- 7p.

Transforming technology of timer coding and data compression

Y. O. Osadchyy, O. Y. Osadchyy, O. A. Gorbunov, R. V. Skuratovskyy

Annotation. The methods and services of timer coding that reduce the combinatorial values of binary numbers starting with filling the byte by them are introduced. The concept "timestamp" (time marker) is used and additional unary codes are introduced (bring in) that are adequate in quantitative meaning to those that contained in the ASCII coding table for this purpose. As a result, not only the intelligent human-machine processing of the communication language symbols is optimized, but also the process of data compression

Keywords: *timestamp, communication language, unary code, quantitative data value, ASCII table*

Трансформерная технология таймерного кодирования и сжатие информации

Е. А.Осадчий, А. Е. Осадчий, О. А. Горбунов, Р. В. Скуратовский

Аннотация. Наводятся методы и средства таймерного кодирования уменьшающие комбинаторные значения двоичных чисел начиная с наполнения ими байта. Для этого, используются понятие таймерная метка и вводятся дополнительные унарные коды, адекватные по количественному значению тем, которые содержатся в кодиро-вочной таблице ASCII. В результате, оптимизируется не только интеллектуальная человеко-машинная обработ-ка символов языка коммуникации, а и процесс сжатия данных.

Ключевые слова: *таймерная метка, язык коммуникации, унарный код, количественное значение данных, таблица ASCII*