ISSN 2079-8490

Электронное научное издание
«Ученые заметки ТОГУ»
2018, Том 9, № 1, С. 25 – 28

Свидетельство
Эл № ФС 77-39676 от 05.05.2010
http://pnu.edu.ru/ru/ejournal/about/
ejournal@pnu.edu.ru

УДК 338.2

© 2018 г. **А. В. Левенец,** канд. техн. наук,
А. А. Равский
(Тихоокеанский государственный университет, Хабаровск)

ЧАСТНЫЙ СЛУЧАЙ АЛГОРИТМА СЛОВАРНОГО СЖАТИЯ

В статье описаны методы сжатия с использованием словаря. Алгоритмы рассмотрены в хронологическом порядке. В конце представлен один из вариантов словарного метода.
Ключевые слова: словарные методы сжатия, алгоритмы сжатия без потерь, алгоритмы сжатия.

A. V. Levenets, A. A. Ravsky

PRIVATE CASE OF ALGORITHM OF DICTIONARY COMPRESSION

The article describes methods of compression using a dictionary. Algorithms are considered in chronological order. At the end one of the variants of the dictionary method is presented.
Keywords: dictionary compression methods, lossless compression algorithms, compression algorithms.

Введение

Одним из наиболее важных способов сжатия без потерь является метод LZW на основе словаря. Он появляется во множестве программ сжатия - ZIP, Compress, Deflate и в файлах формата GIF и PNG.

Рассмотрим любой текст как объект для сжатия. Вместо того, чтобы писать каждое слово, написанное буквами, можно указать номер страницы и номер строки слова в стандартном словаре. Это легко можно представить, и вычисление покажет, как уменьшится объем данных, необходимых для хранения текста, используя словарные ссылки. Это основа словарного метода. Проблема в том, какой словарь использовать. Если используется стандартный словарь, можно откорректировать содержание словаря, чтобы уменьшить необходимое хранилище до минимума, но есть скрытые недостатки. Для того, чтобы разобраться нужно отдельно рассмотреть словарь и данные необходимые для хранения. Можно сохранить текст всего за несколько килобайт, используя словарь, но для хранения словаря может потребоваться несколько мегабайт, необходимого для его декодирования.

Обзор основных методов

Одним из первых алгоритмов сжатия текстовой информации стал LZ77, разработанный в 1977 году. Этот алгоритм представил новую концепцию «скользящего окна», позволившую значительно улучшить сжатие данных. LZ77 использует словарь, содержащий тройки данных – смещение, длина серии и символ расхождения. Смещение – как далеко от начала файла находится фраза. Длина серии – сколько символов, считая от смещения, принадлежат фразе. Символ расхождения показывает, что найдена новая фраза, похожая на ту, что обозначена смещением и длиной, за исключением этого символа. Словарь меняется по мере парсинга файла при помощи скользящего окна. Например, размер окна может быть 64Мб, тогда словарь будет содержать данные из последних 64 ме-габайт входных данных. К примеру, для входных данных «abbadabba» результат будет «abb(0,1,'d')(0,3,'a')» [2]

Самый популярный вариант алгоритма LZ - LZ78, разработанный Лемпель-Зив-Велч в 1984 году, несмотря на запатентованность. Алгоритм избавляется от лишних символов на выходе и данные состоят только из указателей. Также он сохраняет все символы словаря перед сжатием и использует другие трюки, позволяющие улучшать сжатие – к примеру, кодирование последнего символа предыдущей фразы в качестве первого символа следующей.

Алгоритм среди новых возможностей которого была функция разбиения архива на тома был DEFLATE. Эта версия до сих пор повсеместно используется, несмотря на почтенный возраст. Основной смысл алгоритма сжатие в два этапа, первый замена повторяющихся строк указателями(LZ77), второй замена символов, основываясь на частоте их использования(Хаффман).

Алгоритм, который давал максимальную скорость распаковки был LZO (Lempel-Ziv-Oberhumer). Данный алгоритм сжатия данных разработан в середине 1990-х годов. Алгоритм сжимает данные без потерь и его базовая реализация поддерживает многопоточное исполнение. Алгоритм является одним из самых быстрых по скорости распаковки, наряду с созданным на его основе методом LZ4 (LZ4 HC)

В 1998 году в архиваторе 7-zip, который демонстрировал сжатие лучше практически всех архиваторов появился алгоритм LZMA (Lempel-Ziv-Markov chain-Algorithm).

Алгоритм использует цепочку методов сжатия для достижения наилучшего результата. Вначале слегка изменённый LZ77, работающий на уровне битов (в отличие от обычного метода работы с байтами), парсит данные. Его вывод подвергается арифметическому кодированию. Затем могут быть применены другие алгоритмы. В результате получается наилучшая компрессия среди всех архиваторов. [2]

Компания Microsoft в 1996 купила алгоритм LZX, который был разработан в 1995 году Дж. Форбсом и Т.Потаненом. Потом Форбс устроился туда работать над ним, в результате чего улучшенная его версия стала использоваться в файлах CAB, CHM, WIM и Xbox Live Avatars.

Еще один вариант алгоритма LZ77, алгоритм ROLZ расшифровывается как «Лемпель-Зив с уменьшенным смещением», уменьшая смещение, чтобы уменьшить количество данных, необходимого для кодирования пары смещение-длина. Впервые был представлен в 1991 году в алгоритме LZRW4 от Росса Вильямса. Другие вариации — BALZ, QUAD, и RZM. Хорошо оптимизированный ROLZ достигает почти таких же степе-ней сжатия, как и LZMA.

Алгоритм словарного сжатия

Одним из вариантов словарного метода сжатия может служить алгоритм, базирующийся на естественной структуре файловой системе компьютера. Предложенный алгоритм может быть написан следующими основными шагами:

1) Необходимо выделить не изменяемые сектора памяти
2) Подготовить имеющуюся информацию в секторах для создания словаря (дефрагментация диска)
3) Разбить необходимые данные на блоки
4) Записать адреса встречающихся блоков в подготовленных секторах

Особенности, достоинства и недостатки:

Привязка к носителям, при декодировании, необходимо с файлом сохранять таблицу с кодированием или декодировать данные перед переносом их физически на другие носители, так же и по сети. Анализ всех данных на массиве носителей достаточно длительный процесс. Создание правил или таблица кодирования в двух режимах в зависимости от использования (с адресами, с таблицей закодированных адресов для однозначного декодирования). Основные временные затраты на алгоритмы поиска и перебора. Такой способ хорошо подойдет для систем с постоянным потоком данных, которые не будут изменены или удалены (например данные с коллайдера). Способ похож на стеганографию применимую к носителю информации. Такой метод можно назвать частным случаем словарного метода, где словарем является имеющиеся данные.

Заключение

Таким образом, можно проследить как постепенно развивались алгоритмы сжатия с использованием словаря. Каждый метод имеет свои достоинства и недостатки, а правильное комбинирование алгоритмов может принести лучшие результаты. Так, предложенный в статье алгоритм может показывать хорошие результаты, при имеющихся больших объемах данных, без затрат на создание отдельного словаря.

Список литературы

[1] Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. – М.: ДИАЛОГ-МИФИ, 2003. – 384 с

[2] «Хабрахабр» [Электронный ресурс] / Алгоритмы сжатия данных без потерь, часть 2. - Режим доступа : http://habrahabr.ru/post/235553/ (22.01.2017)