УДК 004

Сравнительный анализ алгоритмов сжатия данных

Ленкин Алексей Викторович
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье рассмотрены основные алгоритмы и методы сжатия информации, проведен сравнительный анализ некоторых методов на примере сгенерированных данных.
Ключевые слова: сжатие данных, 7-Zip, LZ, PPM, BWT, префиксные коды, арифметические коды.

Comparative analysis of data compression algorithms

Lenkin Aleksei Viktorovich
Sholom-Aleichem Priamursky State University
Student

Abstract
In this article, the main algorithms and methods of information compression are considered, a comparative analysis of some methods is carried out using the example of generated data.
Keywords: data compression, 7-Zip, LZ, PPM, BWT, prefix codes, arithmetic codes.

Научный руководитель:
Лучанинов Дмитрий Васильевич
Приамурский государственный университет имени Шолом-Алейхема
старший преподаватель кафедры информационных систем, математики и правовой информатики

На сегодня дата-центрам необходимо с каждым годом покупать всё больше аппаратных средств для хранения данных, это связано с тем, что поток клиентов и их требования в размерах доступной памяти всегда растут. В связи с этим расходы компании иногда растут быстрее, чем получение прибыли с новых пользователей. Также это касается и обычных пользователей, которым необходимо сохранить важную информацию на ограниченном дисковом пространстве.

Одним из возможных решений для оптимизации процесса хранения информации является внедрение различных алгоритмов сжатия на имеющихся объемах данных, что позволит архивировать большинство информации, доступ к которой не нужен мгновенно.

Исследованиями в области алгоритмов сжатия занимались следующие авторы. Е.Г. Жиляков, А.В. Болдышев и Е.И. Прохоренко вывели «Алгоритм сжатия речевых данных на основе двумерной обработки данных» [1]. «Применение алгоритмов сжатия для оптимизации данных в системах управления базам и данных» было описано А.С. Карпенко и И.В. Крысовой [2]. Ф.А. Данилкин и М.Л. Гришин описали использование «Комбинаторного кодирования для сжатия данных» [3].

Существует уже довольно большое число алгоритмов сжатия данных, которые разрабатываются и улучшаются до сих пор, но все они направлены на увеличение эффективности использования дискового пространства. Так по типу различают следующие методы сжатия [4]:

1. Современные методы кодирования. Включает в себя более тысячи различных методов, но выделяют следующие большие группы:

- Алгоритмы статического моделирования. Сжатие происходит путем предсказания появления следующего символа в последовательности на примере последовательностей в уже зашифрованной части сообщения. Самое лучшее качество сжатия, но большое время для сжатия и большие затраты ресурсов. Сюда входят методы PPM (Prediction by Partial Matching), DMC (Dynamic Markov Compression), ACB (Associative Coding by Buyanovski).
- Алгоритмы словарного сжатия. Сжимает сообщение заменяя последовательности символов на символы в установленном словаре. Самый быстрый и менее ресурсоемкий, сжимает хуже статистического моделирования в 1.6 раз. Представителями являются алгоритм LZ (Lempel-Ziv) и его модификации.
- Алгоритмы сжатия сортировкой блоков. Разбивают информацию на блоки символов, затем преобразуют так, что появляется много повторений каждого символа и сжимает любым простым способом. Не является сжатием, но позволяет подготовить данные к более эффективной архивации. Совместно с простыми методами сжатия по сравнению со статистическим моделированием сжимает в 1.25 раз меньше, по скорости близок к алгоритмам словарного сжатия. Сюда относится BWT (Burrows-Wheeler transform).

2. Методы энтропийного кодирования. Применяются в совокупности с современными методами кодирования. Заменяет всю информацию кодовыми словами из 1 и 0, а после меняет наиболее часто встречаемые на короткие фразы. Выделяют следующие группы:

- Префиксные коды. Заменяет кодовые слова так, что ни одно не является началом другого. Самыми известными являются код Хаффмана и код Шеннона.
- Арифметические коды. Разбивает отрезок от 0 до 1 и ставит на нём точки таким образом, что длины отрезков равна частоте использования символа и каждый отрезок соответствует одному символу, далее берёт уже распределенные символы и начиная с первого заново распределяет его на отрезке, после проведения этой операции для нескольких последовательных

символов выбирается любое число на полученном отрезке, что и является результатом алгоритма.

Так как было установлено, что на практике не применяются отдельно перечисленные выше алгоритмы, то используются их совокупности, так на примере архиватора 7-zip [5] используются следующие алгоритмы:

1. LZMA. Использует улучшенную модификацию LZ, в частности алгоритм LZ77 с использованием разновидности арифметического кодирования – интервальным кодированием.
2. LZMA2. Использует те же алгоритмы, что и у LZMA, но улучшен тем, что несжимаемые данные оставляются как есть, а также включена поддержка многопоточности при кодировании.
3. PPMd. Использует для сжатия модифицированный алгоритм PPM с применением наследования информации.
4. BPzip2. Является совокупностью использования методов BWT, простого алгоритма MFT (Move-to-Front) для улучшения энтропийного кодирования, которое также здесь используется, Хаффмана.

Проведем сравнение данных алгоритмов на примере сжатия двух сгенерированных файлов размером 100 Мб, один из файлов заполнен однотипными символами, другой случайными, занесем данные в сводную таблицу 1. Режим сжатия во всех экспериментах «Ультра».

Таблица 1. Сравнительный анализ алгоритмов сжатия программы 7-Zip

Наименование сжатия	**Качество сжатия (отношение размера сжатого файла к исходному)**	**Скорость кодирования и декодирования**	**Объем требуемой оперативной памяти**
Случайные символы			
LZMA2	0,0401	10 секунд	709 Мб
PPMd	0,921	47 секунд	223 Мб
BPzip2	1	1 минута 20 секунд	49 Мб
Одинаковые символы			
LZMA2	0,00015	5 секунд	709 Мб
PPMd	0,00038	1 секунда	223 Мб
BPzip2	0,00007	0,1 секунда	49 Мб

По полученным данным видно, что со случайными данными справляется лучше алгоритм LZMA2, в то время как с однотипными BPzip2. Алгоритм PPMd справился в обоих случаях хуже всех.

Но также следует помнить, что полученные данные были сделаны при сжатии сгенерированных файлов и могут существенно отличаться от реальных.

К тому же, существуют также сжатия направленные на уменьшение объема определенного типа файлов [6], так, например, для текста

используется метод ART, для звуковых файлов AC3 и MP3, для видеофайла совместно с его звуковой дорожкой применяют DVI, для хранения отсканированных страниц формат DJVU. Здесь перечислены лишь некоторые из специализированных алгоритмов, их существует огромное множество, но все они узконаправленны.

Таким образом, для обеспечения наилучшего способа хранения данных необходимо использовать сжатие данных, что позволит сохранять большие объемы данных, не увеличивая размеры для хранения аппаратно. В ходе проведенных опытов было установлено, что рекомендуется использовать сжатие LZMA2 и BPzip2. Но в зависимости от типа хранящейся информации может понадобиться использовать специфичные методы сжатия.

Библиографический список

1. Жиляков Е.Г., Болдышев А.В., Прохоренко Е.И. Алгоритм сжатия речевых данных на основе двумерной обработки данных // Вопросы радиоэлектроники. 2012. Т. 4. № 1. С. 27-33.
2. Карпенко А.С., Крысова И.В. Применение алгоритмов сжатия для оптимизации данных в системах управления базам и данных // Россия молодая: передовые технологии – в промышленность!. 2015. № 3. С. 102-104.
3. Данилкин Ф.А., Гришин М.Л. Комбинаторное кодирование для сжатия данных // Известия Тульского государственного университета. Технические науки. 2008. № 2. С. 251-258.
4. Алгоритмы сжатия [Электронный ресурс] URL: http://mf.grsu.by/UchProc/livak/po/comprsite/theory_classification_02.html#Co временные методы сжатия (датаобращения 03.09.2018)
5. 7-zip [Электронный ресурс] URL: https://www.7-zip.org/ (дата обращения 03.09.2018)
6. Сжатие данных [Электронный ресурс] URL: http://megabook.ru/article/Сжатие данных (дата обращения 03.09.2018)