

АНАЛИЗ ПРИМЕНИМОСТИ АЛГОРИТМОВ СЖАТИЯ ДЛЯ РЕГИСТРАЦИИ ПОТОКОВ РАДИОЛОКАЦИОННОЙ ИНФОРМАЦИИ

П.А. Малашкин
Кафедра ВТ, Пензенский Государственный Университет
Пенза, Российская Федерация
E-mail: zodissey@gmail.com

В данной статье анализируются два алгоритма сжатия на пример пригодности для использования при регистрации большого объема информации.

Для работы с радиолокационными станциями (РЛС) требуется специальный аппаратно-программный комплекс регистрации информации, задачей которого является получение в реальном времени потока радиолокационной информации от РЛС и запись её на носитель. [1]

В процессе регистрации радиолокационной информации возникает необходимость в получении, обработке и записи на носитель большого объема информации[2]. Из-за этого требуется носитель с широкополосной шиной и большим размером. Для решения данной задачи можно использовать алгоритмы сжатия данных, которые должны удовлетворять следующим требованиям:

- Обеспечивать сжатие в «реальном» времени;
- Обеспечивать хороший коэффициент сжатия для потока радиолокационной информации;
- Обеспечивать полное восстановление исходных данных из сжатых данных, то есть сжатие без потерь.

Для нашей задачи больше всего подходят словарные методы сжатия, так как статистические методы сжатия требуют анализа всей сжимаемой информации, что невозможно для данного случая, так как мы получаем информации потоком и не можем полностью оценить вероятности появления элементов во входящей последовательности[3].

Для работы с потоком данных необходимо успевать производить обработку входящей информации до момента поступления новой партии информации, в нашем случае за 10 с, поэтому необходимо буферизировать поступающие данные, а также выполнять чтение определенного количества байт входящей информации для сжатия. Это количество должно обеспечивать хорошую степень сжатия и одновременно хорошую скорость работы, а также быть минимальным по размеру для бесперебойности работы алгоритма[4].

Чтобы определить размер буфера для каждого алгоритма была написана тестовая программа на C++. Программа предварительно копирует файл с эталонными радиолокационными данными в оперативную память компьютера.

Затем устанавливает значение буфера для сжатия, после этого производит сжатие всего эталонного файла. Результаты сжатия, такие как время сжатия и объем сжатых данных сохраняются в отдельный файл для логирования. Затем буфер увеличивается, и операция повторяется вновь (рис. 1).

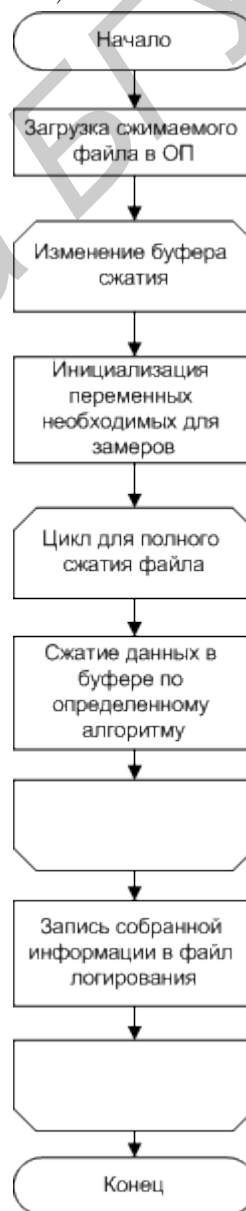


Рис. 1 – Блок-схема программы анализа алгоритмов сжатия

Для анализа взяты реализации следующих алгоритмов:

1. Алгоритм DEFLATE из библиотеки zlib 1.2.3 (в данном эксперименте используется режим с самой высокой скоростью сжатия);
2. Алгоритм LZ4 от корпорации Google.

Эксперимент производился для ПК с процессором Intel Atom N550 с частотой 1,5 GHz и ОП - 2 GiB, на ОС – Windows XP, входной файл для сжатия – файл с эталонными данными размером 371 MiB.

Задача эксперимента была определить:

1. Минимальное значение размера буфера для обоих алгоритмов с целью уменьшения затрат на выделение памяти для оптимального сжатия;
2. Оптимальное значение времени сжатия, позволяющая нам эффективно сжимать данные и укладываться в ограничения по времени.

По результатам эксперимента получены следующие данные для реализации алгоритма DEFLATE в библиотеке zlib, при режиме самого быстрого сжатия (рис. 2,3).



Рис. 2 – График зависимости времени сжатия от размера буфера

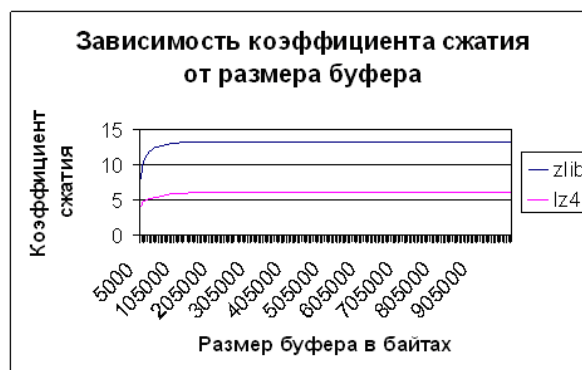


Рис. 3 – График зависимости коэффициента сжатия от размера буфера

Из графиков видно, что с увеличением буфера суммарное время сжатие стремится к значению 11 секунд для библиотеки zlib, и 1,5 для lz4, а коэффициент сжатия 13 для zlib и 6 для lz4. Но, так как нам надо выбрать минимальное значение размера буфера, то оптимальным будет значение в промежутке от 86 000 байт до 106 000 байт для библиотеки zlib, и значение в промежутке от 91 000 байт до 110 000 байт для lz4.

В результате мы видим, что, несмотря на высокую скорость сжатия, реализация алгоритма DEFLATE не укладывается в заданный промежуток времени, в отличие от реализации LZ4, которая укладывается с большим запасом времени, но обеспечивает вдвое меньший, но достаточный, коэффициент сжатия. В результате, при наших условиях, нужно использовать реализацию LZ4. А реализацию в библиотеке zlib эффективно использовать при меньших требованиях ко времени сжатия и в случае реализации алгоритма на аппаратной платформе.

1. Васильев А.В., Чернышев М.И., Коннов Н.Н., Гурин Е.И., Москвитин А.О., Логунов М.В. Аппаратно-программный комплекс для регистрации и имитации радиолокационной информации / «Новые информационные технологии и системы»: Труды VII Международной научно-технической конференции, Ч.2 – Пенза, 2006.
2. Коннов Н.Н., Таранцев Е.К. «Способы повышения производительности программно-аппаратных комплексов РЛС импульсно-доплеровского типа»/ Телекоммуникации. № 5. С. 25-33. - 2011
3. Bell T., Witten I, Cleary J. «Modeling for Text Compression» ACM Computing Surveys, Vol.21, No.4, pp.557-591, Dec. 1989.
4. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. - М.: ДИАЛОГ-МИФИ, 2002. - 384 с.