МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)
Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

Отчет по лабораторной работе №5
«Сжатие»
по курсу
«Информационный поиск»

Группа: М80-108М-17
Выполнил: Забарин Н.И.
Преподаватель: А.Л. Калинин

Москва, 2018

Задание

В этом задании необходимо применить алгоритмы сжатия к координатным блокам. Исследовать изменения в размерах частей индекса, влияние на скорость индексации и поиска. В отчёте нужно указать:

- Выбранный метод сжатия. Привести побитовую схему хранения данных в индексе.
- Описать причины, по которым был выбран именно этот метод сжатия.
- Влияние сжатия на размер и скорость прохождения по координатным блокам всех
- терминов, редких терминов, терминов средней частотности и высокочастотных терминов.
- Обосновать, почему поиск после внедрения сжатия работает корректно. Как производилось тестирование?

Решение

Используемый мной метод сжатия — VarByte. Причины:

- Неудобно использовать битовые методы сжатия
- Простой в реализации
- Считается довольно эффективным

По мимо сжатия на этапе написания этой лабораторной работы была добавлена лемматизация, а так же я стал хранить списки в виде списка дельт.

Побайтовая схема идентична схеме из лабораторной работы №4, за исключением того что все 4хбайтные числа теперь сжимаются.

Производительность

Скорость индексации выросла в примерно 2 раза, это связано с размером индекса, который в свою очередь уменьшился в 2 раза.

На скорости выполнения поисковых запросов сжатие данных не отразилось, были проверены различные термины. Небольшие различия были замечены на высокочастотных терминах при цитатном поиске, чтение огромных координатных блоков стало в 2 раза быстрее, но это «капля в море» на фоне перебора в цитатном поиске.

Поскольку изменены были только базовые операции чтения/записи числа/списка, то тестировались только эти функции отдельно от остальной программы.

Простые тесты в ipython, сгенерировать 1e5-6 чисел, закодировать, раскодировать, проверить на совпадние.

Заметки

Изначально хотел реализовать 3 вида varbyte 4хбитный, 8мибитный и 16тибитный и использовать их в зависимости от длины координатного блока. То есть чем чаще в статье встречается термин, тем меньше разности между позициями в координатном блоке, тем более оптимальный метод мы будем использовать. На деле оказалось что 16тибитный вариант мертв, тк проигрывал по всем параметрам 8мибитному. А для 4хбитного не удалось подобрать хорошую границу когда он становится эффективным, на самых высокочастотных термах он давал порядка 3-5% выигрыша, а код с ним уже переставал быть прозрачным. Поэтому я решил использовать только 8мибитный вариант.

Выводы

В ходе выполнения данной работы мной были изучены и реализованы несколько вариантов сжатия списков чисел, один из них был интегрирован в поисковую систему.

Сжатие очень важный элемент работы с большими данными, тк современные системы хранения не поспевают даже за довольно старыми процессорами.