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

Отчет по лабораторной работе №4
«Поиск цитат, координатный индекс»
по курсу
«Информационный поиск»

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

Москва, 2018

Задание

Необходимо расширить язык запросов булева поиска новым элементом — поиском цитат. Синтаксис этого элемента следующий:

- «что где когда»/2 - такому запросу удовлетворяют документы содержащие в себе термины что, где и когда на расстоянии не более 2, без нарушения порядка. Если элемент /2 отсутствует то термины должны идти подряд.

Новый элемент должен комбинироваться со стандартными средствами булева поиска.

Для реализации цитатного поиска нужно использовать координатный индекс, т.е. для каждого вхождения термина в документ построить и сохранить список позиций внутри документа, где этот термин встречался.

В отчёте нужно описать формат координатного индекса. Привести статистические данные:

- Размер получившегося индекса.
- Время построения индекса.
- Общее количество позиций. Среднее количество позиций на термин и на пару термин-документ.
- Скорость индексации (кб входных данных в секунду)
- Время выполнения поисковых запросов.
- Примеры долговыполняющихся запросов.

Кроме того, нужно привести примеры запросов и результаты их выполнения. В выводах должны быть указаны недостатки работы, приведены примеры их решения. Что можно сделать, чтобы ускорить «долгие» запросы?

Решение

Блок в файле-индексе имеет следующую структуру:

1. 4 байта, длина терма, l
2. l байт под сам терм
3. 4 байта, размер блока, sz
4. 4 байта, длина блока, n
5. n «вхождений» следующего вида
    1. 4 байта, номер документа
    2. 4 байта, размер «вхождения», k
    3. k байт, содержат список координат

Решенные проблемы

В процессе отладки было обнаружено что считывать целиком блок в память не получается в случае высоко частотных термов(предлоги, союзы и т. п.).

Для решения этой проблемы был написан древовидный итератор. На этапе обработки обратной польской записи термины преобразовываются в итераторы первого типа(идут по файлу, пока не найдут следующий документ в котором встречается термин), а операторы в итераторы второго типа(содержат в себе операцию и набор дочерних итераторов, итерируются по дочерним пока не находят подходящий документ).

Таким образом реализация стала немного медленнее(порядка 15%), но заметно эффективнее по памяти.

Статистика индексации

Логи обработки 100 000 документов:

Total time: 1506.6748433113098
Total data: 480.365 Mb
Total docs: 100000
Average time: 0.015066748433113099
Speed: 318.825 Kb/sec
Total entrances: 58525849
Average token entrances: 70.81535881479617
Average token/doc entrances: 2.344759501476706

Размер индекса 440Mb.

Статистика поиска

Лог обработки запроса:

[2018.05.10 14:03:47] [api_search] {'query': '"black and"'}
"black and"
[2018.05.10 14:03:47] Converted query: ['black', 'and', '/2_1']
[2018.05.10 14:03:47] Search for "black" in ../data/index_001
[2018.05.10 14:03:47] Block found for 0.0004363059997558594 sec
[2018.05.10 14:03:47] Search for "and" in ../data/index_001
[2018.05.10 14:03:47] Block found for 0.00035381317138671875 sec
[2018.05.10 14:03:47] Query searched for 0.0013129711151123047 sec

Лог поиска результатов:

[2018.05.10 14:03:47] [api_get_results] id: 8yEAy1kN79, page: 0
[2018.05.10 14:03:47] Result found for 0.030297517776489258 sec

При добавлении еще одного слова в запрос время создания итератора возрастает до 2мс, время поиска до 350мс.

Возможные оптимизации

Поиск цитаты «A B C D» сейчас происходит след образом, находится документ в котором есть «C D» и позиции C в этом документе, далее пытаюсь подставить B, если не получается то ищу новое вхождение «C D». Затем пытаюсь добавить A. То есть время обработки запроса сильно вырастает.

Сходу в голову приходит идея о поиске документа A & B & C & D, его проверке на наличие цитаты и дальнейший перебор по таким документам, эта операция будет работать быстрее(особенно в свете 6 лабораторной работы которая ускоряет операцию &).

Выводы

В ходе выполнения данной работы было решено несколько интересных задач, а так же был построен координатный индекс для 1e5 документов, из-за сжатия в 90% было решено не строить индекс для всей википедии.