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

Отчет по лабораторной работе №6
«Ускорение, прыжки по индексу»
по курсу
«Информационный поиск»

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

Москва, 2018

Задание

В полученный в предыдущих лабораторных работах индекс нужно добавить специальную информацию, позволяющую выполнить «прыжки по индексу», чтобы ускорить пересечение высокочастотных терминов.

Нужно выбрать расстояние, на которое можно выполнить прыжок, таким образом, чтобы получить максимальное ускорение на исследуемом пуле поисковых запросов. Продемонстрировать, что на другом, тестовом пуле поисковых запросов, ускорение тоже есть. Если ускорение будет разным, объяснить причины.

В отчёте должна быть представлена побитовая схема хранения индекса, с дополненными полями для эффективного выполнения прыжков по индексу. Должна быть показана зависимость между размером прыжка и средней скоростью выполнения запросов.

Решение

Была выбрана следующая схема для аккуратного внедрения прыжков.

1. Прыгать будем по документам
2. Пусть L длина списка документов в которых есть термин, тогда введем 2 функции
    1. p = P(L) — частота прыжка, то есть если номер элемента в списке длины L кратен p то за этим элементом находится блок прыжка.
    2. o = O(L) — длина прыжка.
3. p должно быть примерно равно корню L. Я выбрал след формулу P(L)=round_up(sqrt(L-1)).
4. Довольно очевидно что o <= p, иначе мы прыжком перепрыгиваем потенциальный прыжок.
5. Если o < 3 не будем прыгать и вставлять прыжки. Понятно что прыжок должен выполняться быстрее чем проход по элементам через которые мы прыгаем.
6. Первоначальная формула для o: O(L) = round_up((L-1) / (P(L)*3))

Такая схема позволяет тратить на прыжок всего 4 байта места. Из-за того что я не храню значение на которое буду прыгать, для определения успешности прыжка нужно прочитать значение в точке назначения и не хранить номера статей как список дельт.

Прыжок хранится в виде смещения на другой блок, в остальном схема идентична предыдущей лабораторной работе.

В теории все звучит очень сладко, на один результат мы можем сделать не более одного неуспешного прыжка, а успешные прыжки только ускоряют пересечение.

Модификация исходных формул

Лучше всего такие прыжки должны ускорять запросы на пересечение высокочастотного термина и средне-/низкочастотного. Поскольку индекс строится не мгновенно тесты проводились на 1e4 документах.

- Запрос «Black and», 500 найденных результатов, 41 попытка прыжка, 3 успешных прыжка.
- Запрос «parent than», 408 найденных результатов, 60 попыток прыжка, 1 успешный прыжок.
- Запрос «the company», 500 найденных результатов, 26 попыток прыжка, 1 успешный прыжок.
- Запрос «the and», 3000 найденных результатов, 0 попыток прыжка.

Скорость обработки запроса в среднем около 100мс(от момента отправки запроса до получения результата), время от приема запроса до отправки ответа порядка 20мс.

Выводы — нужно уменьшать частоту и длину прыжка, для высокочастотных термов не работает. Попробуем увеличить частоту в 3 раза и ограничим длину прыжка десятью сверху.

- Запрос «Black and», 500/130/51.
- Запрос «parent than», 300/135/20.
- Запрос «the company», 1000/133/17.
- Запрос «the and», 1000/2/0.

Выводы — наблюдается движение в правильном направлении, замедление/ускорение обработки не наблюдается, но процент прыжков/успешных прыжков вырос.

Проиндексируем 1e5 статей:

- Запрос «Black and», 500/62/30.
- Запрос «parent than», 500/112/22.
- Запрос «the company», 500/31/5.
- Запрос «the and», 1000/3/0.

Выводы — время обработки запроса 20-40мс, зависимость от числа успешных прыжков не прослеживается. Скорее всего стоит отказаться от верхнего лимита на длину прыжка и заменить функцию на логарифмическую с небольшим основанием, например 1,3.

Возьмем случайное предположительно очень редкое слово и высокочастотное and

- Запрос «rotelliform and», нашли всего 1 результат и 598 из 598 прыжков успешны, время выполнения запроса 1,7 сек.
- Запрос без прыжков 1,8 сек, стабильно.

Теперь перестроим индекс с новой функцией длины прыжка и проверим те же запросы:

- Запрос «rotelliform and» 1/598/597, и ускорение до 1 сек.
- Запрос «Black and», 500/62/3.
- Запрос «parent than», 500/112/1.
- Запрос «the company», 500/31/0.
- Запрос «the and», 1000/3/0.

Выводы

Данная реализация прыжков по индексу ускоряет «больное» место, то есть запросы с редкими словами, при этом не замедляя другие запросы.