Здравствуйте, Владимир!

Прикрепляю описание программы с индексом, инструкцию по использованию и полученные результаты.

Я выполнила все пункты задания на 25 баллов: работающий обратный индекс, потоковая обработка дерева запросов, алгоритм сжатия (varbyte), выводы и результаты

1) Как запустить?

* Сначало надо скачать 8 документов с текстами по ссылке <https://cloud.mail.ru/public/FnMq/qCNif6bFG/dataset/>
* И назвать их dataset1.txt,…dataset8.txt, и положить в директорию, откуда вы будете запускать программу.
* Убедиться, что в этой же директории лежат документы:

launch\_prog3NotCompress.sh

create\_index3NotCompress.cpp

create\_tree3NotCompress.cpp

launch\_prog4Compress.sh

create\_index4Compress.cpp

create\_tree4Compress.cpp

varbyte.cpp

* Если хотите запустить программу без сжатия индекса-то запустите вбейте в командной строке ./launch\_prog3NotCompress.sh
* Если хотите запустить программу со сжатием индекса-то запустите вбейте в командной строке ./launch\_prog4Compress.sh

Программа сначала выводит немного полезной информации:

Hello

amt\_docs=10001

amt\_tokens=91768

SizeMyindex=91767

Потом выводит время, затраченное на потоковую и непотоковую обработку.

Потом выводит дерево запросов, потом сам запрос, потом количество подходящих документов, потом их урлы в столбик

2)Описание программы

Если ./launch\_prog3NotCompress.sh:

То программа create\_index3NotCompress.cpp соберет индекс (предварительно приведя токены к нижнему регистру, обрубит всякие запятые и окончания) и запишет его в файл indexNotCompress.txt, а потом программа create\_tree3NotCompress.cpp откроет этот индекс, построит дерево запросов и подберет подходящие документы 2 способами: потоковой обработкой и непотоковой, и сравнит их времена.

ЕСЛИ ВЫ ХОТИТЕ ПОМЕНЯТЬ ЗАПРОС - то это надо сделать на строчке 368 в мейне в файле create\_tree3NotCompress.cpp

Если ./launch\_prog4Compress.sh: все то же самое, только индекс считывается сжатый, и разжимается при считывании.

3)Сравнение результатов

От сжатия оказалась только одна польза - оно сжало индекс больше, чем в два раза: indexNotCompress.txt весит 8.2МБ, а indexCompress.txt весит 3.8МБ.

А вот от потоковой обработки польза очень большая.

Точнее так: для коротких запросов (типа путин & медведев) непотоковая обработка немного быстрее, а вот на длинных (типа (власти & (бельгии | парижа)) & !теракт) потоковая обработка быстрее в очень большое число раз.

А именно: для путин & медведе получаем:

duration\_nepotok=3055ns

duration\_potok=4119ns

путин & медведев

А для (власти & (бельгии | парижа)) & !теракт получаем:

duration\_nepotok=254403ns

duration\_potok=3143ns

То есть чем длинее запрос - тем сильнее выигрывает потоковая обработка!

4)Выводы:

Потоковую обработку стоит делать, она существенно уменьшает время выполнения длинных запросов.

Сжатие индекса тоже стоит делать, это экономит место для хранения индекса.