UFMG - Departamento de Ciência da Computação

Indexador de páginas da Web

Gabriel Miranda Pedrosa

29 de maio de 2017

1 Introdução

A recuperação de informação é uma tarefa importante quando tratamos de coleções de dados. Quando a quantidade de dados é grande, boas estratégias devem ser adotadas para que as buscas possam ser feitas em tempo hábil. Os dados passam a ser organizados na forma dos chamados índices invertidos de documento, no qual uma palavra referencia os documentos nos quais aparece, em vez do contrário.

A tarefa do segundo trabalho prático da disciplina de Recuperação da informação é implementar um indexador de páginas Web. O objetivo é realizar a análise das páginas HTML, extrair os termos indexáveis e gerar o índice invertido usando memória externa.

2 O indexador

2.1 Extração dos termos

A criação do índice começa com a análise das páginas Web da coleção. Essa tarefa é auxiliada pela biblioteca htmlcxx [1] que gera árvores de análise sintática do HTML da página e facilita a extração dos termos indexáveis. Termos extraídos são normalizados e reescritos em caixa baixa.

[1] Disponível em: `http://htmlcxx.sourceforge.net`

2.2 Geração das tuplas

A indexação continua iterando sobre os termos indexáveis de cada página (documento) e armazenando tuplas de códigos que referenciam o termo, o documento, a frequência daquele termo no documento e as posições onde aparece. Para limitar o uso de memória, pré-determinamos o número de tuplas de termos que podem ser armazenadas em memória principal. Quando essa quantidade é atingida, o conjunto de tuplas é ordenado e armazenado em memória secundária. Este limite é calculado considerando que a tupla ocupa 212B, que é o quanto um termo que aparece em 50 posições de um documento ocuparia em memória.

2.3 Ordenação externa

Os arquivos de tuplas ordenadas são mesclados através do algoritmo heapsort multi-way, respeitando o limite do número de tuplas estabelecido. O algoritmo é executado até que sobre um arquivo somente, com todas as tuplas da coleção. Este arquivo, somado dos arquivos que relacionam termos e URLs de páginas com seus respectivos códigos, forma o índice de documentos invertido. A mesclagem dos arquivos não é feita in-place, o que significa que a mesclagem de um arquivo A com o arquivo B gera um terceiro arquivo C com o conteúdo de A e B.

3 O Buscador

Como parte da tarefa, um buscador que utilizasse o índice gerado deveria ser implementado. A busca recuperava a lista de documentos que cada um dos termos da consulta aparecia e retornava a interseção entre todas as listas. A interseção era calculada ordenando as listas pelo tamanho delas. A menor lista era tomada como referência e a presença de seus elementos nas outras listas era verificada, de forma que somente eram retornados documentos que apareciam nas listas de todos os termos da busca.

4 Análise de complexidade

4.1 Tempo

As tarefas que mais demandam tempo na indexação é a geração de termos indexáveis e a ordenação externa. A primeira gera uma árvore sintática que é usada para navegar pelo documento em busca dos termos. Considerando que a construção é feita em tempo linear, a extração de termos é feita em $O(C)$, em que L é o tamanho do documento em número de caracteres. A segunda tarefa executa um heapsort multi-way que tem ordem de complexidade $O(Tlog_W T)$, em que T indica o número de tuplas a serem ordenadas e W o número de tuplas que vão para a memória principal.

4.2 Espaço

Os módulos mais custosos em espaço são o vocabulário e as tuplas armazenadas em memória principal. O vocabulário ocupa espaço $O(V)$, em que V indica o tamanho do vocabulário. Uma tupla armazena também as posições onde aparece no documento. Por isso, a complexidade média de espaço é da ordem $O(P)$, onde P indica a frequência média de termos em documentos da coleção.

5 Experimentos

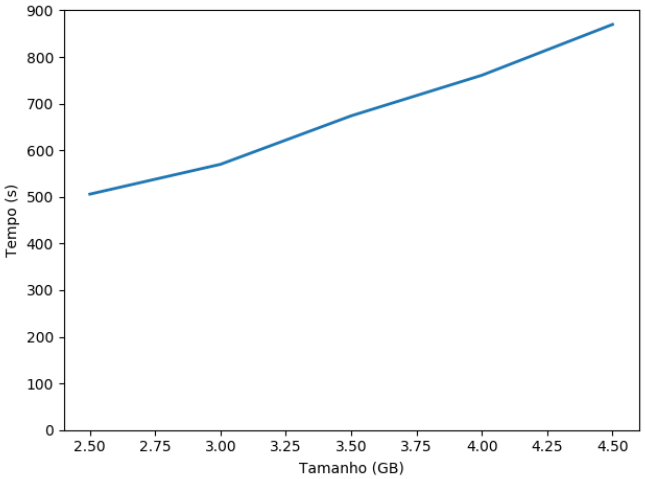
Para avaliar o desempenho do indexador implementado, foram realizados testes para analisar a resposta quando variamos o número de tuplas disponíveis em memória principal e o número de documentos para indexar. Foi usada uma coleção de 5GB, com 8608 termos e 21924 documentos, segundo o índice gerado pelo indexador. A coleção corresponde aos 10 primeiros documentos da coleção completa disponibilizada em sala. O computador utilizado para os experimentos foi:

- 1.4 GHz Intel Core i5
- 4 GB 1600 MHz DDR3
- OS X El Capitan
- SSD 128GB

5.1 Tamanho da coleção

Fixando o tamanho da memória disponível em 200 megabytes, o tamanho da coleção foi variado em 2,5GB e 4,5GB. O resultado de tempo de indexação pode ser observado no gráfico 5.1. É possível validar a ideia de que quanto maior a coleção, mais tempo o indexador leva.

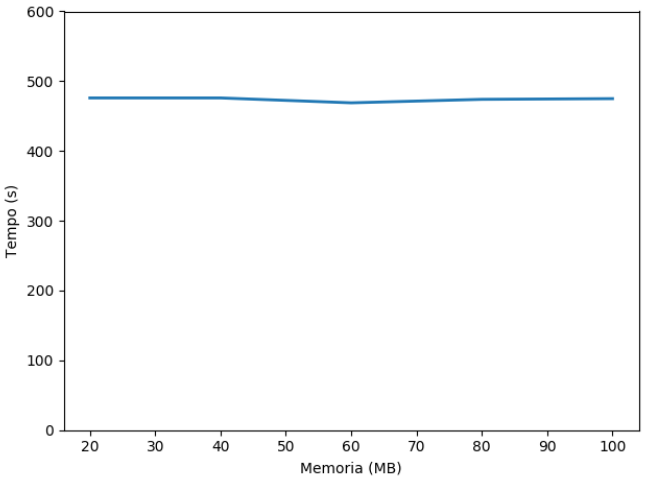
Figura 5.1: Tempo de indexação em função do tamanho da coleção



5.2 Tuplas

Foi passada a memória disponível como parâmetro variando entre 20, 40, 60, 80 e 100 megabytes. O tamanho da coleção foi fixado em 2,5GB. O resultado, porém, contradiz a análise de complexidade, pois o tempo de indexação não responde às variações de tamanho da memória disponível. Isso ocorre, provavelmente, porque o controle de memória não é estrito, o que permite que se use a mesma quantidade de memória nas cinco execuções. O resultado pode ser observado no gráfico 5.2.

Figura 5.2: Tempo de indexação em função da memória principal



6 Melhorias

Para o uso do indexador implementado em situações reais, deveria haver remoção de acentos nos termos da coleção. Isso diminuiria a quantidade de termos indexados e melhoraria a qualidade das recuperações com o índice.

O controle de memória pecou, por não ter sido feito de forma estrita e sim com estimativas de consumo de memória por tupla. Isso resultou em usos descontrolados de memória o que impede o uso do indexador para coleções realmente grandes.