1
Introdução

A quantidade de informação existente no mundo é grande e cresce a uma taxa exponencial a cada ano. Aplicações como engenhos de busca web, por exemplo, tem que lidar com uma quantidade de dados na ordem de peta bytes. Esse problema acarreta em sistemas de informação complexos que precisam de muito tempo para computar esses dados. Outra conseqüência é o alto custo financeiro associado. Geralmente, para processar essa massiva quantidade de informações, é preciso uma infra-estrutura de máquinas equipadas com processadores eficientes, mídias de armazenamento sofisticadas e as máquinas conectadas por uma rede de alta velocidade. Em resumo, existe uma necessidade em armazenar, comunicar e computar grandes quantidades de informações de forma eficiente, eficaz e viável financeiramente.

Em um recente estudo, Hilbert e Lopez (Hil2011) estimaram a capacidade tecnológica global de armazenar, comunicar e computar informação no período de 1986 a 2007. Em 2007, a humanidade conseguiu armazenar 2.9×10^{20} bytes comprimidos (290 exabytes), comunicar quase 2×10^{21} bytes e executar 6.4×10^{18} instruções por segundo. Eles também estimaram a taxa anual de crescimento da capacidade tecnológica global de armazenar, comunicar e computar informação no mesmo período em 23%, 28% e 58%, respectivamente.

Este crescimento exponencial da capacidade computacional, no entanto, apenas amortiza o problema e não o resolve por completo. Por exemplo, a capacidade de armazenamento de dados na memória principal (RAM) é muito inferior àquela da memória externa (disco), geralmente na ordem de 10^3 vezes. Outro exemplo é que existe uma disparidade no tempo de acesso à memória principal e à memória externa na ordem de 10^5 e pode piorar se compararmos o tempo de acesso aos dados em outros computadores ligados por uma rede.

Os problemas citados nos exemplos acima, em muitas situações, inviabilizam o uso de grandes quantidades de dados pelas aplicações seja: i) pelo longo tempo de processamento; ii) pelo aumento da complexidade da aplicação; iii) ou pelo alto custo financeiro.

Uma abordagem que pode evitar os três itens acima concomitantemente é a **compressão de dados**. Essa é uma área de estudo da teoria da informação

(Sha48). Muitas das técnicas dessa área já têm sido utilizadas na prática para viabilizar novos tipos de aplicações como, por exemplo, câmeras digitais de alta resolução e transmissão de vídeo pela Internet.

Compressão de dados tem a vantagem de aumentar a quantidade de dados nas memórias mais rápidas e diminuir o custo de armazenamento nas memórias mais lentas sem aumentar significativamente a complexidade da aplicação e o custo financeiro em hardware.

É bem conhecido, no entanto, o *tradeoff* entre o tempo de compressão/descompressão e a taxa de compressão. Codificadores aumentam a quantidade de padrões detectados de forma a aumentar a taxa de compressão. Porém, isso também implica no aumento do tempo de compressão. O inverso também é verdade: para diminuir o tempo, eles diminuem os padrões detectados implicando na diminuição da taxa de compressão. Portanto, *deve-se projetar um codificador de forma a encontrar o equilíbrio ideal entre o tempo e a taxa de compressão para cada tipo de aplicação.* Se isso ocorrer então o tempo de processamento geral da aplicação pode vir a diminuir e, muitas vezes, viabilizar o uso na prática da aplicação.

Para melhorar ambos o tempo e a taxa de compressão, podemos restringir o codificador a um tipo de dados na qual se está interessado. Existem vários esquemas de compressão para diferentes tipos de dados mais populares como, por exemplo, mp3 para música, jbig2 para documentos monocromáticos, jpeg para imagens coloridas e mpeg para vídeos. Como regra geral, *não existe um esquema de compressão genérico que é eficaz para todos os tipos de dados.* Assim, conhecendo as características e padrões dos dados, é possível explorá-los para obter um esquema de compressão melhor.

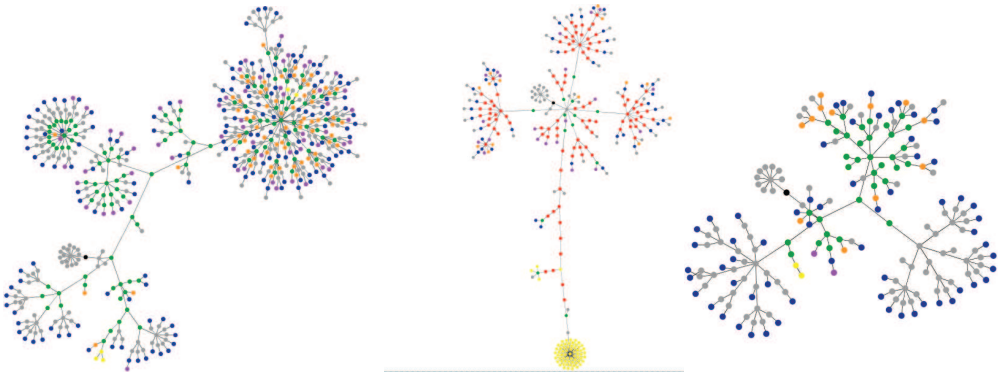
1.1 Tipos de Dados

Esta tese foca em esquemas de compressão para os seguintes tipos de dados menos populares: cadeias binárias e grafos web. Apresentamos apenas o terceiro tipo de dados em mais detalhes a seguir.

1.1.1 Grafos web

O grafo web é um grafo direcionado e sem pesos tal que seus nós representam páginas web e cada aresta representa um *link* entre duas páginas web. Este grafo representa a estrutura topológica da Web (Figura 1.1).

O grafo web é utilizado em uma variedade de aplicações como, por exemplo, estratégias para *crawlers*, detecção de web spam, localização de



1.1(a): www.betanews.com 1.1(b): www.puc-rio.br 1.1(c): www.inf.puc-rio.br

Figura 1.1: Cortesia de `http://www.aharef.info/static/htmlgraph/`.

páginas relacionadas, e foi popularizado pela sua principal aplicação: o *ranking* de páginas web.

Uma característica dos grafos web que se torna um problema para aplicações que o usam é o seu potencial tamanho. Grafos sociais são mais populares, mas são limitados ao tamanho da população mundial que é de cerca de 7 bilhões de pessoas. Grafos web, no entanto, são estimados em possuírem pelo menos 1 trilhão de nós (Goo08). Um grafo web deste tamanho mesmo compactado não cabe em memória principal (computadores modernos suportam memória RAM com até 192 GB). Portanto, existe a motivação de construir representações compactas para discos de modo a viabilizar o uso de grafos web massivos na prática.

Limitamos nossas pesquisas às páginas estáticas, ou seja, páginas que não são geradas dinamicamente com dados provindos de um banco de dados. Isto porque páginas web dinâmicas induzem um subgrafo característico de seus dados e não daquele induzido por páginas web estáticas. Por exemplo, *sites* de redes sociais geram uma página web para cada usuário cadastrado em seu banco de dados e cria links de acordo com as relações sociais (amigos) do usuário. Portanto, o subgrafo induzido é típico dos grafos sociais e não de um grafo web. Esses grafos sociais, em particular, são difíceis de comprimir (Chi2009).

1.2 Objetivos

O escopo desta tese concentra-se na intersecção entre ciências da computação e teoria da informação. Ao nos concentrarmos nessa intersecção, os avanços alcançados beneficiam ambos os lados. Assim, esta tese tem o objetivo de estudar e aprimorar:

1) a compressão de cadeias binárias através da relação entre algoritmos de

intercalação e codificadores de fonte binária;

2) as representações compactas de grafos web construídas para memórias externas (discos).

1.3 Contribuições

Para o problema de compressão de cadeias binárias, demonstramos a relação entre algoritmos de intercalação e codificadores de fonte binária. Em específico, provamos que *qualquer* algoritmo de intercalação pode ser convertido em um codificador de fonte binária. Em específico, se um algoritmo de intercalação é ótimo no modelo de comparações então o codificador de fonte binária também é ótimo. Em seguida, mostramos que os algoritmos de intercalação binário (Hwang e Lin, 1972), recursivo (Dudzinski, 1981) e probabilístico (Vega, 1993), geram respectivamente os codificadores binários de entropia baseado em comprimentos de carreiras codificados com o código de Rice, o codificador de intercalação binária (Moffat, 2000) e o codificador de Rice aleatório, na qual é um novo variante do código de Rice.

Para o problema de compressão de grafos web, propomos uma nova representação compacta para grafos web, intitulada *árvore-w*, construída especificamente para memória externa (disco), sendo a primeira nesse gênero. Propomos também um novo tipo de layout projetado especificamente para grafos web, intitulado *layout escalado*. Além disso, mostramos como construir um layout *cache-oblivious* para explorar a hierarquia de memórias, sendo a primeira desse tipo. Apresentamos vários tipos de consultas que podem ser executadas e é a primeira representação a suportar execução de consulta de leitura aleatória em lote e a otimização de consultas avançadas, inclusive em memória principal. Por fim, executamos uma série de experimentos que mostra que a árvore-w apresenta taxas de compressão e de tempo de execução competitivas com outras representações compactas em memória principal. Assim, demonstramos empiricamente a viabilidade de uma representação compacta de grafos web para memória externa, contrariando a afirmação de alguns pesquisadores (Suel, 2001) (Buehrer, 2008).

1.4 Organização da Tese

Esta tese está organizada em cinco capítulos, além desta introdução e das referências bibliográficas. No Capítulo 2, revisamos os conceitos básicos de teoria da informação que são utilizados nesta tese. No Capítulo 3, abordamos o problema da compressão de cadeias binárias através da demonstração da

relação entre algoritmos de intercalação e codificadores de fonte binária. No Capítulo 4, abordamos o problema da compressão de grafos web. Por fim, no Capítulo 5, apresentamos nossas conclusões e algumas propostas de trabalhos futuros.