Algoritmos para compressão de URLs

Ronan Loschi Rodrigues Ferreira, Fabrício Benevenuto
PPGCC - Programa de Pós-Graduação em Ciência da Computação
UFOP - Universidade Federal de Ouro Preto
Ouro Preto, Minas Gerais, Brasil
email: ronan.loschi@gmail.com, benevenuto@gmail.com

***Resumo*—O uso de mensagens curtas tem sido amplamente explorado em sistemas como o Facebook, Twitter e Orkut, permitindo que os usuários compartilhem informações com seus amigos. Em particular, o Twitter é um sistema social voltado unicamente para a postagem de mensagens curtas, que podem conter no máximo 140 caracteres. Com a grande popularidade desses sistemas o uso de encurtadores de URLs está se tornando cada vez mais comum. A compressão de URLs funciona da seguinte forma. A URL original (que pode consistir de centenas de caracteres) passa pela compressão e é traduzida em uma nova URL, tipicamente com poucos caracteres, que retorna os códigos HTTP 301 ou 302 de redirecionamento para a URL longa original. Apesar de extremamente úteis, os sistemas de compressão de URLs podem introduzir atrasos para seus usuários e têm sido amplamente utilizados como forma de ofuscar *spam*, *phishing* e *malware*. A ideia futura é utilizar um mecanismo descentralizado de compressão de dados no momento de envio da mensagem na rede social (o algoritmo de compressão encurta a URL e posteriormente a descompressão da URL ocorre nos clientes que recebem a URL).**

***Keywords*-Algoritmos, complexidade, compressão, URL, Huffman.**

I. Introdução

O uso de mensagens curtas tem sido amplamente explorado em sistemas como o Facebook, Twitter e Orkut, permitindo que os usuários compartilhem informações com seus amigos. Em particular, o Twitter é um sistema social voltado unicamente para a postagem de mensagens curtas, que podem conter no máximo 140 caracteres. Com a grande popularidade desses sistemas o uso de encurtadores de URLs (*Um URL (de Uniform Resource Locator), em português Localizador-Padrão de Recursos, é o endereço de um recurso disponível em uma rede; seja a Internet, ou uma rede corporativa, uma intranet.*) está se tornando cada vez mais comum. A compressão de URLs funciona da seguinte forma. A URL original (que pode consistir de centenas de caracteres) passa pela compressão e é traduzida em uma nova URL, tipicamente com poucos caracteres, que retorna os códigos HTTP 301 ou 302 de redirecionamento para a URL longa original [1].

Apesar de extremamente úteis, os sistemas de compressão de URLs podem introduzir atrasos para seus usuários e têm sido amplamente utilizados como forma de ofuscar *spam*, *phishing* e *malware* [2], [3]. A ideia futura é utilizar um mecanismo descentralizado de compressão de dados que no momento de envio da mensagem na rede social o algoritmo de compressão encurta a URL e posteriormente a descompressão da URL ocorre nos clientes que recebem a URL [1].

Neste momento pretendemos investigar se os algoritmos atuais de compressão, considerados estado-da-arte, conseguem comprimir URLs de forma razoável. Sendo assim, neste trabalho, propomos analisar a complexidade, implementar e testar o algoritmo de compressão de dados Huffman em (1952). O método de Huffman é uma codificação utilizada para a compressão de dados sem perdas. A idéia do método é atribuir códigos mais curtos a símbolos com frequências altas. Um código único, de tamanho variável, é atribuído a cada símbolo diferente do texto. Métodos de Huffman baseados em caracteres comprimem o texto para aproximadamente 60 %, enquanto os métodos de Huffman baseados em palavras comprimem o texto para valores próximos de 25 % [4].

A compressão de dados é o ato de reduzir o espaço ocupado por dados num determinado dispositivo, com os objetivos de reduzir a quantidade de Bytes para representar um dado e de retirar a redundância através de uma regra, chamada de código ou protocolo, que elimina os bits redundantes. Além disso, os dados são comprimidos pelos mais diversos motivos. Entre os mais conhecidos estão economizar espaço em dispositivos de armazenamento, como discos rígidos, ou ganhar desempenho (diminuir tempo) em transmissões. [5].

O restante desta proposta está organizada da seguinte forma. A Seção II apresenta a justificativa para este trabalho. Os objetivos são apresentados na Seção III. A Seção IV descreve os passos para a realização deste trabalho. Finalmente a Seção V oferece os resultados esperados.

II. Justificativas

Acreditamos que o desenvolvimento de um sistema encurtador de URLs eficiente e que dispensa a existência de um servidor central para a realização da compressão da URL pode possuir um grande impacto para os usuários desse tipo de sistema. Parte deste impacto é devido a uma potencial redução no tempo para resolver a URL e a redução da

proliferação de *spam* e *phishing* em sistemas como Twitter. Também o ganho de espaço obtido por um método de compressão pode apresentar ganhos consideráveis, por exemplo, se o arquivo não comprimido possuir 100 bytes e o arquivo comprimido resultante possuir 30 bytes, então o ganho de espaço é de 30%, segundo a razão de compressão definida pela porcentagem que o arquivo comprimido representa em relação ao tamanho do arquivo não comprimido.

Além do exposto acima, o desenvolvimento deste trabalho, também pode ser visto como uma oportunidade de integração entre os objetivos acadêmicos, da disciplina Projeto e Análise de Algoritmos, e os objetivos de produção científica, dos envolvidos, como o desenvolvimento da dissertação.

III. Objetivos

Pretende-se analisar a complexidade, implementar e testar o algoritmo de compressão de dados proposto por Huffman em (1952). Predente-se investigar através da compressão de URLs a eficiência deste método de compressão sem perdas.

IV. Metodologia

Como metodologia vamos seguir os seguintes passos:

Levantamento bibliográfico: Nesta etapa vamos buscar bibliografias e artigos acadêmicos que tratem dos problemas relacionados a compressão de textos, tais como os referentes ao método de codificação de Huffman. **Investigação de estratégias de compressão de URLs**. Esta etapa consiste em investigar formas de se encurtar URLs, com foco no método de codificação de Huffman. Para textos em linguagem natural, a técnica de compressão mais eficaz é a codificação de Huffman baseada em palavras. O método considera cada palavra diferente do texto como um símbolo, conta suas frequências e gera um código de Huffman para as palavras. A seguir comprime o texto substituindo cada palavra pelo seu código. Assim a codificação é realizada em duas passadas sobre o texto. O codificador realiza uma primeira passada sobre o texto para obter a frequência de cada palavra diferente e faz a compressão em uma segunda passada. Parte deste processo pode ser observado na Figura 1. Pretendemos implementar, testar e analisar a complexidade deste algoritmo de compressão. Nossa abordagem pode utilizar o fato de que URLs não aceitam todos os tipos de caracteres, mas texto no Twitter aceita qualquer caractere utf8. [4]

V. Resultados Esperados

Como resultados pretendemos submeter 1 artigo científico para o seminário da disciplina Projeto e Análise de Algoritmos, do programa de Pós Graduação em Ciência da computação da Universidade Federal de Ouro Preto. Além disso, pretendemos realizar uma apresentação oral, mostrando o desenvolvimento deste trabalho, seu vínculo com a disciplina Projeto e Análise de Algoritmos e sua contribuição para a dissertação.

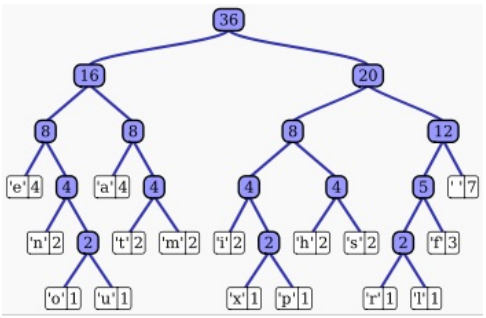

Figura 1. Exemplo: Árvore de Huffman gerada pelas frequências exatas do texto "this is an example of a huffman tree".

Referências

[1] D. Antoniades, I. Polakis, G. Kontaxis, E. Athanasopoulos, S. Ioannidis, E. Markatos, and T. Karagiannis, "we.b: The web of short urls," in *Int'l Conference on World Wide Web.*, 2011, pp. 715–724.

[2] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on twitter," in *Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS), Redmond, Washington, USA. July*, 2010, pp. 1–9.

[3] C. Grier, K. Thomas, V. Paxson, and M. Zhang, "@spam: The underground on 140 characters or less," in *ACM conference on Computer and communications security (CCS)*, 2010, pp. 27–37.

[4] N. Ziviani, *Projeto de Algoritmos com Implementações em Pascal e C*, 3rd ed. Cengage Learning, 2011, iSBN: 978-85-221-1050-6.

[5] Wikipedia, "Compressão de dados," 2011, http://pt.wikipedia.org/wiki/Algoritmodecompressao, Acesso em 06/06/2011.