Comparativo entre técnicas de tratamento de colisão em Tabelas Hash

Fernando Concatto

Centro de Ciências Tecnológicas da Terra e do Mar
Universidade do Vale do Itajaí (UNIVALI)
Itajaí – SC – Brasil

fernandoconcatto@edu.univali.br

Resumo. *As tabelas hash são estruturas de dados que armazenam dados identificados por chaves. As chaves devem ser discernidas pela função hash da tabela, que tenta atribuir um valor único para cada chave; entretanto, a função pode atribuir valores iguais para chaves diferentes, um fenômeno que precisa ser tratado. Este trabalho analisou os métodos de endereçamento aberto e encadeamento para tratamento de colisões, e identificou que o encadeamento oferece desempenho notavelmente superior ao concorrente.*

1. Introdução

Tabelas hash são estruturas de dados que armazenam pares de chaves e valores em um vetor de tamanho fixo. Seu funcionamento é baseado em uma função denominada *função hash*, que mapeia uma chave para um valor inteiro. A saída desta função é utilizada para determinar o índice do vetor à qual aquela chave pertence; consequentemente, as chaves de uma tabela hash devem ser únicas. Entretanto, é possível que a função hash atribua o mesmo índice para duas chaves diferentes, causando um efeito denominado *colisão*. Existem diversas formas de tratar colisões em tabelas hash, cada uma com suas vantagens e desvantagens. Este trabalho analisou dois métodos de tratamento de colisão conhecidos como endereçamento aberto e encadeamento, identificando qual técnica obtém melhor desempenho nas operações de inserção e pesquisa.

O trabalho foi dividido em mais três seções: a segunda define as estruturas e funcionamento dos tipos de tabela hash analisados, a terceira expõe a análise comparativa realizada e a quarta seção apresenta as conclusões obtidas sobre o trabalho.

2. Estruturas e operações

Ambos os tipos de tabela hash utilizam vetores para armazenar os pares de chaves e valores. Cada posição do vetor pode estar preenchida ou vazia; inicialmente, todas as posições devem ser marcadas como vazias. Quando uma inserção ou pesquisa é realizada, a chave recebida é aplicada à função hash, que determina o índice inicial do elemento. Os passos conseguintes são específicos de cada método de tratamento de colisão, descritos nas subseções seguintes.

2.1. Endereçamento aberto

A versão da tabela hash que utiliza o método de endereçamento aberto é assim chamada pois o índice gerado pela função hash não será obrigatoriamente o índice onde o elemento será armazenado. As colisões na operação de inserção são resolvidas da seguinte forma: a partir do índice gerado pela função hash, o vetor é percorrido até que uma posição livre seja encontrada, e o elemento é inserido naquela posição. Caso um elemento com a mesma chave seja encontrado durante a operação ou se não existem mais posições livres, a inserção é rejeitada.

A operação de pesquisa acontece de uma forma similar: a partir do índice produzido pela função hash, o vetor é percorrido exatamente da maneira que a operação de inserção até que a chave do elemento naquela posição seja igual à chave sendo procurada. Caso uma posição vazia seja encontrada durante o percurso ou se o vetor inteiro for percorrido e a chave não for encontrada, então a chave desejada não está presente na tabela.

2.1. Encadeamento

A técnica de encadeamento para tratamento de colisões funciona armazenando um ponteiro para o próximo par em cada par de chave e valor, como uma lista encadeada. Assim, cada posição do vetor pode conter zero ou vários elementos. A operação de inserção funciona de uma forma similar à inserção no final de uma lista encadeada: primeiramente, o índice é gerado a partir da função hash; em sequência, os pares naquela posição são visitados em sequência até que um elemento sem próximo par definido seja encontrado, e então o novo par se torna o próximo deste elemento. Caso uma chave igual seja encontrada durante as visitas, o par não é inserido.

A pesquisa funciona de forma parecida: a partir da chave fornecida, o índice é gerado pela função hash e os pares são percorridos do mesmo modo da operação de inserção. Quando a chave do elemento sendo visitado for igual à chave desejada, o elemento é retornado. Se um elemento nulo for visitado durante o procedimento, a pesquisa falha, pois aquela chave não existe na tabela.

3. Comparativo

Na tabela 1, estão dispostas as funções de complexidade pessimista para as operações de inicialização, inserção e pesquisa em tabelas hash utilizando endereçamento aberto e encadeamento. A operação de inicialização em ambos os casos requer c instruções, onde c é a capacidade do vetor, pois todas as posições necessitam ser marcadas como vazias. Já as operações de inserção e pesquisa para ambas as versões, no pior caso, requerem tempo linear em relação à quantidade de elementos presentes na tabela. Esta situação ocorre quando absolutamente todas as inserções resultam em colisão; ou seja, sempre que um novo elemento for inserido, todos os elementos inseridos até o momento deverão ser visitados, pois é necessário verificar se a chave sendo inserida já existe na tabela, enquanto a operação de pesquisa fica reduzida à uma simples busca linear em um vetor, no caso do endereçamento aberto, ou em uma lista encadeada, no caso do encadeamento.

Tabela 1. Complexidade pessimista das operações da tabela hash

	Endereçamento aberto	Encadeamento
Inicializa	f(n) = c	f(n) = c
Inserir	f(n) = n	f(n) = n
Pesquisar	f(n) = n	f(n) = n

Apesar do pior caso apresentar um desempenho que deixa a desejar, tal circunstância raramente ocorre. Através dos experimentos realizados, foi possível determinar que, em média, a complexidade de inserção na tabela hash utilizando endereçamento aberto é constante enquanto a tabela possuir menos do que aproximadamente dois terços de sua capacidade preenchida; após este ponto, a quantidade de instruções necessárias para realizar uma operação cresce rapidamente, sendo impossível realizar inserções após ocupar todas as posições do vetor. No caso do método de encadeamento, a complexidade se mostrou praticamente constante até mesmo depois da quantidade de elementos na tabela ultrapassar a capacidade do vetor. Estes resultados são apresentados na tabela 2.

Tabela 2. Média de instruções na inserção por taxa de ocupação

	Endereçamento aberto	Encadeamento
10%	1,106	1,008
25%	1,359	1,073
50%	2,363	1,270
75%	7,322	1,573
100%	321,843	1,880
200%	1	2,772

Na tabela 3, estão listadas as médias de instruções na operação de pesquisa nas duas versões da tabela hash. Os valores seguem um padrão muito similar à inserção, pois o procedimento em ambos os casos é bastante semelhante. A maior diferença se localiza na pesquisa na tabela hash utilizando endereçamento aberto após o vetor ser completamente preenchido, pois a busca por um elemento que não está presente na tabela resulta na necessidade de analisar todos os elementos da tabela.

Tabela 3. Média de instruções na pesquisa por taxa de ocupação

	Endereçamento aberto	Encadeamento
10%	1,107	1,133
25%	1,361	1,397
50%	2,373	1,807
75%	7,380	2,250
100%	332,909	2,262
200%	968,674	3,572

A média de instruções na operação inserção está apresentada em forma de gráfico na figura 1. O eixo vertical está em escala logarítmica para facilitar a visualização da grande variação de valores observada no endereçamento aberto quando a taxa de ocupação se aproxima de 100%.

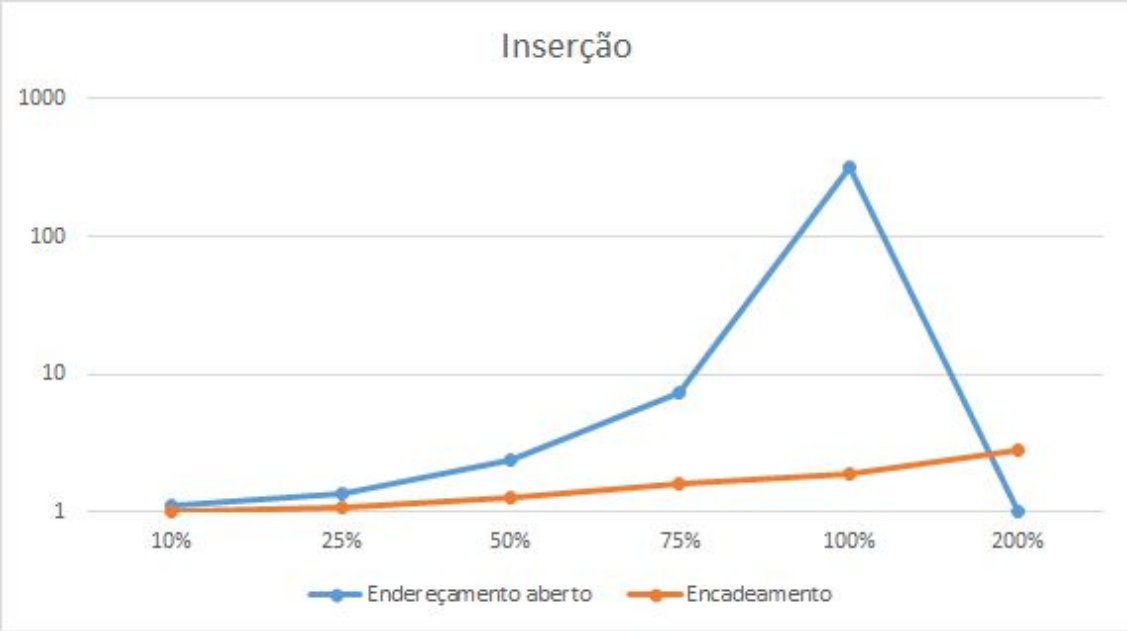

Figura 1. Gráfico da média instruções na inserção por taxa de ocupação

Na figura 2, estão dispostas as médias de instruções na operação de pesquisa em ambos os métodos de tratamento de colisão. Novamente, o eixo vertical foi configurado em escala logarítmica, pois o endereçamento aberto gera valores ainda mais discrepantes em relação ao encadeamento.

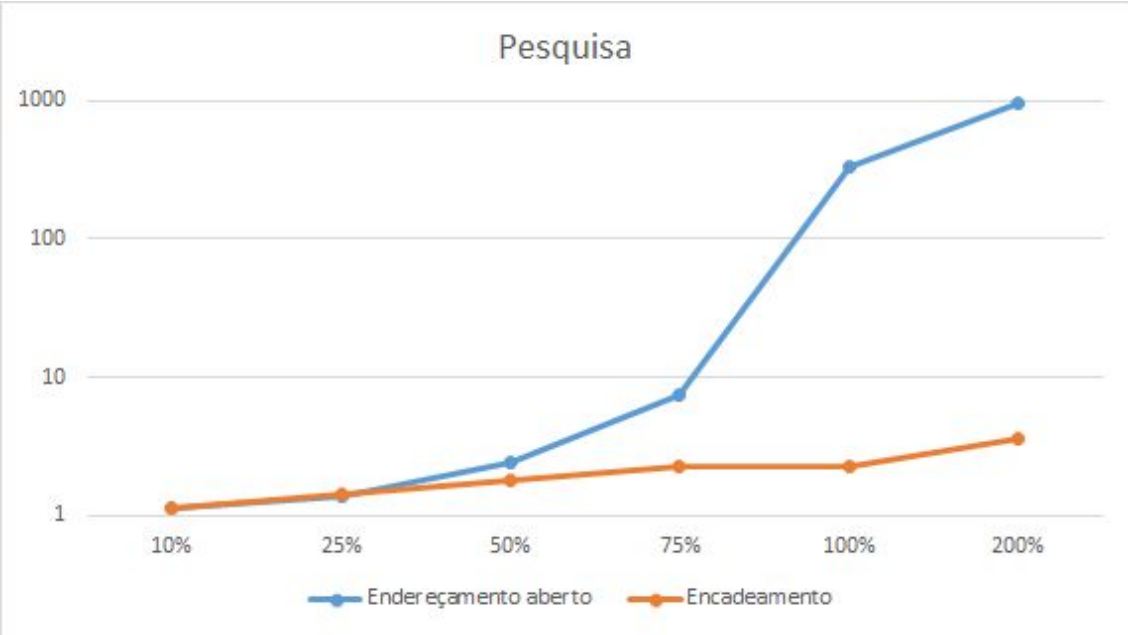

Figura 2. Gráfico da média de instruções na pesquisa por taxa de ocupação

4. Conclusões

Através da análise comparativa, foi possível observar que o método de encadeamento para tratamento de colisões se mostra superior que o endereçamento aberto em todas as circunstâncias avaliadas. Apesar das médias de instruções nas operações de inserção e pesquisa serem bastante similares inicialmente, a complexidade de tempo de ambas as operações na tabela hash empregando endereçamento aberto aumenta substancialmente após o vetor estar 75% preenchido.

Além da performance superior, o método de encadeamento também permite que elementos continuem a ser inseridos na tabela após o esgotamento de posições livres no vetor. As médias obtidas através dos experimentos também indicam que esta característica acarreta pouco impacto na quantidade de instruções necessárias para realizar operações sobre a tabela.