

Relatório Trabalho de Escalonamento de Transações - Banco de dados

Cristiano Creppo Mendieta, Guilherme Carbonari Boneti

Estruturas de dados

- Estrutura de Escalonamento: um Escalonamento possui dois inteiros begin e end que identificam o espaço que cada escalonamento possui na lista dada na entrada.
- Estrutura de Transação: cada Transação possui os inteiros 'tempo_chegada', 'identificador', 'qtd_operacoes' e os char 'operacao' e 'atributo'.
- Estrutura de Nó: um Nó possui um inteiro 'index' que identifica o nó, uma lista de nós vizinhos 'neighbours' e uma flag 'visited' que indica se o nó já foi percorrido.

Separa Escalonamentos

A função "separaEscalonamentos" recebe uma lista de transações que é dada na entrada do programa e identifica onde começa e termina cada escalonamento da lista, armazenando essas informações em uma lista de escalonamentos, onde cada posição da lista corresponde a um escalonamento na lista de transações. Retorna também o número total de escalonamentos.

Algoritmo de Teste de Seriabilidade quanto ao conflito

A função "seriabilidade" recebe como parâmetro a lista de transações do escalonamento e o número total de operações. É criado um grafo, que consiste em uma lista de nós, onde cada nó possui uma lista de ponteiros para seus nós vizinhos. Inicialmente, é criado um nó para cada transação do escalonamento.

Em seguida, o algoritmo executa dois laços aninhados, onde o laço exterior percorre os nós do grafo e o laço interno percorre a lista de transações do escalonamento.

Dentro desse laço, o algoritmo executa 3 operações principais:

- cria uma aresta $T_i \rightarrow T_j$ para cada $r(x)$ em T_j depois de $w(x)$ em T_i ;
- cria uma aresta $T_i \rightarrow T_j$ para cada $w(x)$ em T_j depois de $r(x)$ em T_i ;
- cria uma aresta $T_i \rightarrow T_j$ para cada $w(x)$ em T_j depois de $w(x)$ em T_i ;

Essas operações computam de forma similar entre si. Cada uma possui um teste para identificar se T_i executa a operação correspondente e internamente executa um laço que percorre as operações seguintes, buscando uma transação que execute a operação desejada pelo teste.

Quando é encontrada uma operação que passa nos testes, é criada uma aresta entre os dois nós do grafo que correspondem às transações T_i e T_j envolvidas. Dessa forma, para cada nó do grafo, os 3 testes são executados percorrendo seus vizinhos.

Após percorrer todos os nós, a função "DFS" é utilizada para identificar um ciclo no grafo. "DFS" é uma busca em profundidade que percorre as arestas do grafo marcando os nós visitados e caso chegue em um nó já visitado é identificado um ciclo. Se há ciclo no grafo, a função "Seriabilidade" retorna "NS" indicando que o escalonamento não é serial, caso contrário retorna "SS" indicando que o escalonamento é de fato serial.

Algoritmo de Visão por equivalência

A função "visão" recebe como parâmetro a lista de transações do escalonamento e o número total de operações, então dentro da função as transações são separadas por id, organizando-as em lista, desse modo temos quais operações são referentes a cada transação. Em seguida, serão criadas duas matrizes que irão armazenar os atributos escritos e lidos de cada transação presente no escalonamento. Também são criados dois vetores para armazenar o número de operações de leitura e escrita. Então, é verificado se todas as operações de leitura de cada transação já foram escritas por outras transações. Isso é feito verificando as operações de leitura de cada transação na matriz de leitura. Para cada operação de leitura, todas as transações são percorridas, exceto a transação que executou a operação de leitura, para ver se o valor foi escrito. Se o valor foi escrito, a variável "escrito" é definida como verdadeiro e interrompe o loop interno. Se o valor não foi escrito, a função retorna "NV", indicando que as transações não são equivalentes por visão. Por fim, o código verifica se todas as operações de escrita de cada transação foram as últimas operações de escrita para os valores que eles escreveram. Isso é feito percorrendo as operações de escrita de cada transação na matriz de escrita. Para cada operação de escrita, o código percorre todas as transações, exceto a transação que executou a operação de escrita, para ver se o valor foi escrito por outra transação. Se o valor foi escrito por outra transação, a função retorna "NV", indicando que as transações não são equivalentes por visão.

Se a função chegar ao final, significa que as transações atendem às condições e ela retorna "SV", indicando que as transações são equivalentes por visão.