Indexação

O problema fundamental associado à manutenção de um índice em disco é que o acesso é muito lento, o que coloca problemas para uma manutenção eficiente. O melhor acesso a um índice ordenado, até agora, foi dado pela Pesquisa Binária, porém:

- Pesquisa binária requer muitos acessos. 15 itens podem requerer 4 acessos, 1000 itens podem requerer até 11 acessos. Esses números são muito altos.
- Pode ficar muito caro manter um índice ordenado de forma a permitir busca binária. É necessário um método no qual a inserção e a eliminação de registros tenham apenas efeitos locais, isto é, não exija a reorganização total do índice.

Assim, foi preciso desenvolver estruturas que permitam recuperar esse mesmo registro em menos acessos. Essas estruturas devem permitir agrupar informações de modo que seja provável que toda a informação necessária possa ser obtida numa mesma operação de acesso. Por exemplo, se para um dado cliente precisamos do nome, endereço, telefone, saldo, número da conta, etc. é preferível obter toda essa informação de uma só vez em vez de ficar procurando em vários lugares. aquelas qualidades, pois os arquivos mudam, crescem e encolhem conforme algumas informações são adicionadas e outras removidas.

Árvore B

As árvores B são árvores balanceadas projetadas para trabalhar com dispositivos de armazenamento secundário como discos magnéticos. Elas visam otimizar as operações de entrada e saída nos dispositivos. O tempo de acesso às informações em um disco é prejudicado principalmente pelo tempo de posicionamento do braço de leitura. Uma vez que o braço esteja posicionado no local correto, a leitura pode ser feita de forma bastante rápida. Desta forma, devemos minimizar o número de acessos ao disco. Diferente das árvores binárias, cada nó em uma árvore B pode ter muitos filhos, isto é, o grau de um nó pode ser muito grande.

Uma árvore-B é definida em termos de sua ordem de capacidade, que expressa a capacidade mínima e máxima de armazenamento de cada um de seus nós. O tamanho dos nós das árvores B é determinado pela capacidade de armazenamento da memória principal. Cada nó de uma árvore B tem mais de uma entrada de dados (chaves) e múltiplas descendências (filhos). Árvores B têm vantagens substanciais em relação a outros tipos de implementações quanto ao tempo de acesso e pesquisa aos nós. Objetivo principal é minimizar o número de acessos ao disco para recuperar um registro.

Características principais:

- B-trees oferecem uma estrutura de acesso multinível que é uma estrutura de árvore balanceada em que cada nó está cheio pelo menos até a metade.
- Todos os valores de pesquisa na b-tree são únicos, pois consideramos que a árvore é usada como uma estrutura de acesso em um campo de chave.

- Se os valores se repetirem no arquivo (índice sobre campo não chave), os ponteiros de dados apontam para um bucket (depósito).
- Existe um número máximo e mínimo de filhos em um nó. Este número pode ser descrito em termos de um inteiro fixo **t** maior ou igual a 2 chamado grau mínimo.
- Cada nó de uma b-tree de ordem **t** pode ter no máximo **t** ponteiros de árvore, **t-1** ponteiros de dados, e **t-1** valores de campo de chave de pesquisa.
- Todas as folhas da árvore estão na mesma altura (que é a altura da árvore).
- Uma árvore B com **n** itens tem complexidade de E/S $O(\log_B n)$ para operações de pesquisa ou atualização e usa O(n/B) blocos, onde B é o tamanho de um bloco.
- A eficiência de localização de um registro é de $O(\log_2 n)$

Árvore B+

A árvore B+ é uma variação da estrutura básica da árvore B. Uma das maiores deficiências da árvore-B é a dificuldade de percorrer as chaves seqüencialmente. Uma variação da estrutura básica da árvore-B é a árvore-B+. Nesta estrutura, todas as chaves são mantidas em folhas, e algumas chaves são repetidas em nós não-folha para definir caminhos para localizar registros individuais. As folhas são ligadas através de uma lista duplamente encadeada, de modo a oferecer um caminho sequencial para percorrer as chaves na árvore. Esta lista é chamada de Conjunto de Seqüência (*Sequence Set*).

Características principais:

- Todas as chaves são mantidas em folhas;
- As chaves são repetidas em nós não-folha formando um índice;
- As folhas são ligadas oferecendo um caminho sequencial para percorrer as chaves.

Vantagens:

- Mantém a eficiência da busca e da inserção da árvore B;
- Aumenta a eficiência da localização do próximo registro na árvore de O(log2N) para O(1);
- Não é necessário manter nenhum ponteiro de registro em nós não-folha.
- As árvores B+ sempre mantêm uma cópia de todos os dados nas folhas, o que em caso de necessidade de imprimir toda ela, por exemplo, permite uma rápida busca linear, fazendo com que a Árvore B, em comparação, tenha menor performance.
- Nó folha e não folha são do mesmo tamanho

Desvantagens:

- Mais espaço em disco para armazenar o índice, já que as chaves se repetem nos nós intermediários e nas folhas;

- A busca na árvore sempre tem que chegar até o nó folha, que é onde realmente está o registro procurado.

A árvore-B+ é ideal para aplicações que requerem tanto acesso seqüencial quanto aleatório. Por isso e pelas vantagens citadas anteriormente, a árvore -B+ tornou-se bastante popular nos SGBDs comerciais.