Informe sobre Árboles B (B-Trees).

Autores:

1. Yuniel Silva.
2. Alejandro Veitía.

Funcionamiento:

Los árboles B-Trees permiten un ordenamiento de los nodos, desde el menor al más extremo izquierdo y el mayor hacia el extremo derecho. La principal característica con respecto a los árboles búsquedas es que crece hacia arriba, en dependencia del valor mínimo establecido para los nodos padres. Un nodo raíz se considera lleno en dependencia del valor establecido, al insertar un nuevo valor, su compara con las llaves almacenadas en el nodo o si está vacío. Si está lleno, la media entre los valores sube, convirtiéndose en el nuevo padre, de la lista de nodos hijos.

El algoritmo para insertar, se basa en comparar la llave almacenada con el nodo raíz, si este es menor se inserta hacia la izquierda, si no, a la derecha. Así, recursivamente.

Propiedades de los B-Trees:

1. Todas las hojas están al mismo nivel.
2. B-Trees se define por el término grado mínimo ' t '. El valor de ' t ' depende del tamaño del bloque de disco.
3. Cada nodo, excepto la raíz, debe contener al menos t-1 claves. La raíz puede contener un mínimo de 1 clave.
4. Todos los nodos (incluido la raíz) pueden contener como máximo (2*t – 1) claves.
5. El número de hijos de un nodo es igual al número de claves en él más 1.
6. Todas las claves de un nodo se ordenan en orden creciente. El hijo entre dos claves k1 y k2 contiene todas las claves en el rango de k1 y k2.
7. B-Trees crece y se encoge desde la raíz, a diferencia del Binary Search Trees. Los árboles de búsqueda binaria crecen hacia abajo y también se encogen hacia abajo.
8. Al igual que otros árboles de búsqueda binarios equilibrados, la complejidad de tiempo para buscar, insertar y eliminar es O (log n).
9. La inserción de un nodo en B-Trees ocurre solo en el nodo hoja.

Ventajas:

1. Los B-Trees tienen una complejidad de tiempo garantizada de O(log n) para operaciones básicas como inserción, eliminación y búsqueda, lo que los hace adecuados para grandes conjuntos de datos y aplicaciones en tiempo real.
2. Los B-Trees se equilibran a sí mismos.
3. Alta concurrencia y alto rendimiento.
4. Utilización eficiente del almacenamiento.

Desventajas:

1. Los B-Trees se basan en estructuras de datos basadas en disco y pueden tener un alto uso de disco.
2. No es el mejor para todos los casos.
3. Lento en comparación con otras estructuras de datos.

Principales aplicaciones:

1. Se utiliza en grandes bases de datos para acceder a los datos almacenados en el disco.
2. La búsqueda de datos en un conjunto de datos se puede lograr en mucho menos tiempo usando el B-Trees.
3. Con la función de indexación, se puede lograr la indexación multinivel.
4. La mayoría de los servidores también utilizan el enfoque de árbol B.
5. Los árboles B se utilizan en los sistemas CAD para organizar y buscar datos geométricos.
6. Los B-Trees también se utilizan en otras áreas, como el procesamiento del lenguaje natural, las redes informáticas y la criptografía.

Ejemplo práctico:

Para poder obtener el ejemplo practico y poder revisar el código se podrá descargar desde la siguiente url.

https://github.com/YuniGoldenBoY30/B-tree