TP1 - Árbol de punto Óptimo

Integrantes:

Cristian Martín Westergaard 80483
Maximiliano Alexis Montiel 93157

Cátedra:

Organización de Datos 75.06 - Curso Servetto

Tutor:

Nicolás Fernández Theillet

Descripción General

El árbol de punto óptimo es un árbol orientado a facilitar la resolución de los problemas de proximidad en un espacio métrico arbitrario. Consiste básicamente en un árbol cuyos nodos dividen el espacio en 2 partes: las entidades que se encuentran dentro de una burbuja, próximos a un “pivote”, y los que no.

Dado que utilizan la función distancia, y no las coordenadas para la toma de caminos en el árbol, el espacio métrico no necesita ser del tipo euclideano, por lo que se puede aplicar en pseudo-espacios métricos, donde otros árboles métricos no pueden, como por ejemplo los k-d, que dada su naturaleza sí requiere de espacios euclideanos.

Se atribuye la autoría a Jeffrey Uhlmann, quien publicó un paper del mismo en 1991. En el mismo, describe el árbol de punto óptimo como una solución mejorada al problema de la búsqueda de vecinos. En el mejor de los casos, la complejidad de búsqueda es de O(log(n)).

El árbol que presentamos a continuación presenta las siguientes particularidades:

- Todo árbol posee una única raíz.
- La raíz del árbol es un nodo.
- Todo método empieza con la raíz.
- Todo nodo tiene un valor de altura.
- Si el valor de la altura del nodo es igual a 0, entonces el nodo es una hoja.
- Todos los registros se guardan únicamente en las hojas.
- Los registros pueden ser de longitud fija o variable.
- Todo nodo interno es un nodo cuya altura es distinta de 0.
- Todo nodo interno se compone de una sucesión de registros internos con la siguiente composición:
    - (Pivote, radio, referencia a nodo).
- Todo nodo tiene, además, una referencia a un nodo sumidero.
- Los nodos tienen un factor de carga mínima de ½ de la capacidad máxima.
- En todo momento, ningún nodo, a excepción de la raíz, puede tener menos de la carga mínima.
- Todo pivote posee la misma estructura que cualquier identificador de registro.
- Un pivote no necesariamente coincide con un identificador de registro existente.
- Para cada registro interno, el contenido del nodos referenciado se encuentra a distancia menor al radio con respecto al pivote.

- El contenido del nodo sumidero se encuentra fuera de todas las burbujas delimitadas por los registros internos.
- Existe una función tal que, para todo par de identificadores de registro, se computa una distancia mayor o igual a cero. La distancia únicamente será cero si los identificadores son el mismo.
- La función distancia cumple con la desigualdad triangular.

Métodos ABM

Los métodos ABM corresponden a los modos de inserción (alta), eliminación (baja) y actualización (modificación) de registros en el árbol.

Se utilizan los identificadores de los registros para todas las operaciones.

Altas

Si la raíz tiene altura igual a 0, quiere decir que la raíz es hoja.
En este caso, se inserta directamente.

Caso contrario, la raíz es nodo interno.
Se calcula la distancia con cada uno de los pivotes, en orden.
Si para un pivote dado, la distancia es menor al radio del pivote considerado, se prosigue de manera recursiva con el nodo correspondiente.
Si ya se recorrieron todos los pivotes, se prosigue con el nodo sumidero.

Si se produce desborde en un nodo, se busca balancear.
Si no se puede se parte el nodo en 2.

Bajas

Se realiza el mismo procedimiento que para las altas, sólo que en este caso se busca eliminar el registro indicado. Si no se encuentra, no se hace nada.
Si un nodo queda por debajo de la carga mínima, se buscará balancear.
En caso de no poder, se fusiona.

Modificaciones

Se realiza una especie de baja/alta combinada del registro. Es decir, se navega hasta el nodo hoja correspondiente, y se modifica el registro indicado, en caso de existir.
Si los registros son de longitud variable, se puede llegar a generar tanto un desborde como una situación de underflow. En este caso, se procederá igual que en el caso de un desborde en alta o en un underflow en baja, según corresponda.

A continuación mostraremos los métodos de balanceo, partición y fusión de nodos.

Balanceo de nodos

En el caso de balanceo, se busca siempre balancear con el hermano derecho.
Para el caso del nodo sumidero, que se encuentra a la derecha de cualquier pivote, se usará el hermano izquierdo.
Entiéndase por izquierdo y derecho al orden en el cual se encuentran las referencias a los nodos internos.
Hay 2 tipos de balanceo. Balanceo por desborde (overflow) y balanceo por carga mínima (underflow)

Balanceo por desborde

En el caso de balanceo por desborde, queremos repartir parte de la carga de nodos a otro nodo. Para garantizar que se pueda realizar la acción, se necesita que el hermano posea carga mínima. Al finalizar el balanceo, ambos quedarían con ¾ de la capacidad máxima.

El procedimiento es el siguiente:

Se conserva el pivote del hermano más a la izquierda.
Se calculan las distancias de todos los elementos al pivote mencionado.
Mediante un heap de mínimos, se seleccionan los elementos más próximos,
cuidando de lograr dividir al grupo de registros en 2 partes de similar tamaño.
Luego se selecciona un radio tal que separe al último del primer grupo con el primero del segundo.

Balanceo por carga mínima

En este caso, lo que buscamos es un nodo con más de la carga mínima. Esto nos garantiza que al final de la operación, ambos queden, o bien con carga mínima, o sólo uno con carga mínima.

Se procede exactamente igual que en el caso de los balanceos por desborde.

Partición de nodos

En la partición de nodos, se genera un pivote nuevo, con la consecuente repartición de los registros entre el nodo desbordado y el nuevo.

Para la partición, se debe seleccionar un pivote y un radio tales que dividan a los registros en 2 partes de similar tamaño.

El mejor método para determinar el pivote que indica el autor antes mencionado, es calcular un rincón del menor hipercubo que contenga a todos los registros. Luego, sólo queda seleccionar el radio y repartir como en el caso de los balanceos por desborde.
Es de notar, sin embargo, que en la partición, se modifica el radio del primero, y el radio del segundo es simplemente el menor que contenga a todos los registros excluídos por el primer pivote.

Debido a la naturaleza recursiva de los métodos sobre el árbol, se puede llegar a generar una cadena de desbordes, que al llegar a la raíz, termina con el incremento en altura.
Como regla general, la partición de un nodo genera 2 nodos, tanto el original como el nuevo, con la misma altura.

En el caso de se parta a la raíz en estado de nodo hoja, se promueve a nodo interno, incrementando el contador interno de altura. Se almacenará entonces un pivote y un sumidero, siendo ambos nodos hojas los contenedores de los registros antes almacenados en la raíz.

Fusión de nodos

En la fusión de nodos, lo que se hace es juntar los contenidos de 2 nodos, eliminando 1 pivote en el proceso. Para poder realizarlo, se requiere que la suma de registros no exceda la capacidad máxima de un nodo.

Lo que se hace es simplemente extender el radio del primero hasta que incluya a todos los registros del nodo a fusionar.

En el caso de que se deba fusionar el un único pivote del nodo con el nodo sumidero, simplemente se traen los contenidos al nodo y se decrementa el contador de altura.
Si esto ocurre en la raíz, se dice que el árbol decreció en altura.

Persistencia en disco

Para poder persistir el árbol de manera eficiente en disco, se utilizan nodos con un tamaño fijo de bloque, consistente en una potencia de 2 igual o mayor a 9 en bytes. Con esto nos aseguramos de que se lean múltiplos de 512 bytes. 512 bytes corresponden a la menor unidad de asignación física del disco. Debido a eso, los nodos poseen una gran capacidad, por lo que cada nodo hoja puede contener más de un registro. Asimismo, se pueden incluir una mayor cantidad de registros internos en cada nodo interno.

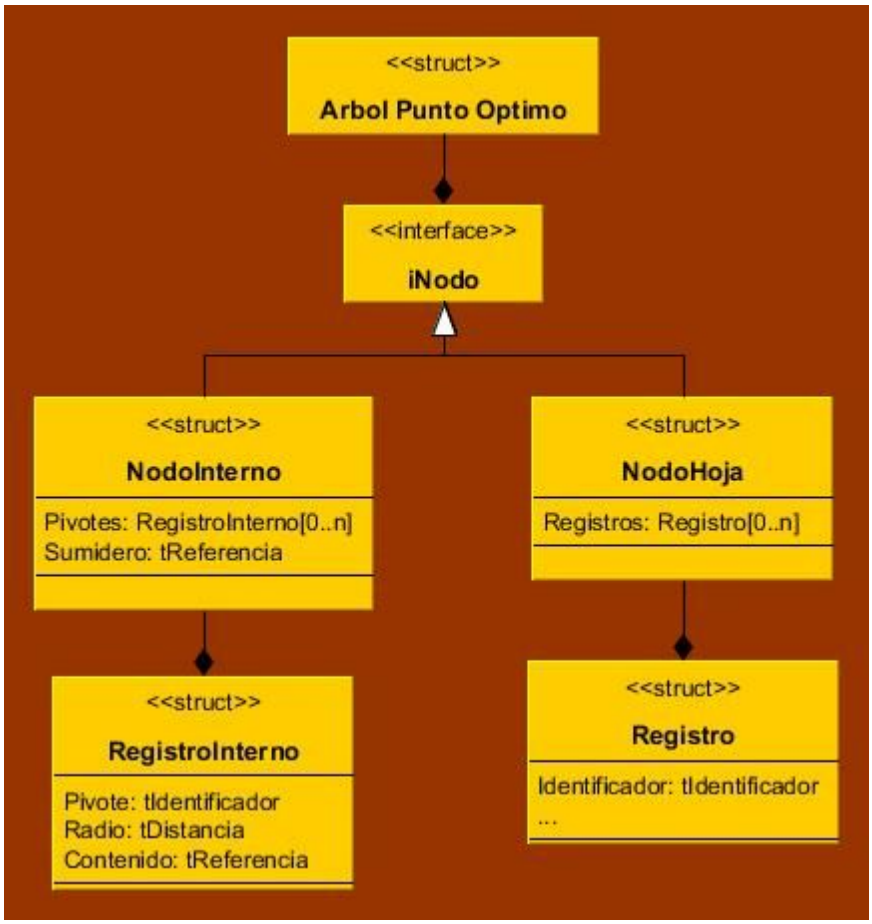
En la persistencia a disco, se garantiza lo siguiente:

- Todos los nodos tienen el mismo tamaño.
- La raíz se guarda siempre en el bloque lógico número 0 del archivo.
- El tamaño de un nodo es una potencia superior a 8 de base 2.

- Cada nodo se guarda con el contador de altura.
- Los nodos hojas guardan, además, todos sus registros.
- Los registros de las hojas se guardan ordenados por identificador.
- Los nodos internos guardan la colección de sus registros internos más el nodo sumidero, manteniendo el mismo orden que en memoria.

Diagrama de alto nivel

Esquema de la estructura del árbol a nivel lógico:



Donde tIdentificador es el tipo de identificador de registro.
tReferencia es un tipo que permite almacenar números de nodo.
tDistancia es un tipo que permite almacenar y comparar distancias.

Conclusiones

El árbol de punto óptimo no está pensado para guardar datos maestros o inclusive, para un uso intensivo de las altas y bajas, y en el caso de registros de longitud variable, de modificaciones. Como se pudo ver en los apartados de los métodos ABM, los desbordes,

particiones y fusiones son operaciones muy costosas, dado que se debe evaluar la función distancia al menos una vez por cada registro involucrado.

Asimismo, el rendimiento del árbol está condicionado por la definición de la función distancia, la distribución de los registros y la selección de los pivotes del espacio métrico.

Bibliografía

https://en.wikipedia.org/wiki/Vantage-point_tree
http://pnylab.com/pny/papers/vptree/main.html
http://lbd.udc.es/Repository/Thesis/1348132590960_tese_luis_g_ares.pdf