



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC3413 — Implementación de Sistemas de Base de Datos — 1' 2020

LABORATORIO 2

Laboratorio: Índices.
Publicación: Viernes 17 de abril.
Ayudantía: Viernes 17 de abril.
Entrega: **Viernes 1 de mayo a las 23:59 horas.**

En este laboratorio estudiaremos los índices de un sistema de bases de datos, modificando y extendiendo el funcionamiento de un B+tree. En particular, modificaremos la inserción del B+tree en IIC3413-DB como también, opcionalmente, implementaremos la eliminación en un B+tree.

Redistribución en BPlusTree durante insert

IIC3413-DB tiene un método de inserción de records del índice B+tree. Este método simplemente recorre el árbol buscando la hoja donde se debe insertar el record, lo inserta y, si es necesario, balancea el árbol haciendo “split” de los nodos. Sin embargo, hay veces que es posible redistribuir los datos localmente para no generar “splits” innecesarios. Para que el sistema de índices mejore, es necesario hacer redistribución de los data entries e index entries cuando sea posible.

Para esta tarea, usted debe modificar las clases y métodos dentro del directorio `storage/index/bplus_tree` para así extender la inserción de datos en un B+tree incorporando la redistribución de datos en el árbol.

Para programar su solución se sugiere que durante la inserción y antes que el método vaya a hacer “split” de una página del índice, se compruebe con el vecino adyacente (del mismo padre) si existe la posibilidad de redistribuir los datos. De ser posible, se debe ejecutar un método `redistribution` que los redistribuya para así evitar una acción de “split”. Específicamente, se sugiere que los pasos a seguir sean los siguientes:

1. Revisar las clases y código en el directorio `storage/index/bplus_tree` para entender el funcionamiento del `BPlusTree` en IIC3413-DB. Se recomienda fuertemente entender primero el almacenamiento de los directorios y hojas del índice en páginas, y el funcionamiento recursivo de `insert` en las clases `BPlusTreeDir` y `BPlusTreeLeaf`.
2. Extender los métodos `insert` en `BPlusTreeLeaf` y `BPlusTreeDir`, para pasar como referencia al directorio padre del nodo.
3. Implementar un método en `BPlusTreeDir` que, dado una posición/puntero en su directorio y una dirección izquierda o derecha, entregue al hermano vecino a la izquierda o derecha, respectivamente, de ese nodo.
4. Implementar la redistribución de valores en `BPlusTreeLeaf`, modificando la inserción de tal manera que, antes de hacer split, intente hacer una redistribución con los nodos vecinos. En caso de no poder hacer una redistribución debe continuar con el split. Notar que para la redistribución usted puede suponer que los nodos vecinos de un `BPlusTreeLeaf` también serán del mismo tipo.

5. Implementar la redistribución de valores en `BPlusTreeDir`, modificando la inserción similar al caso de `BPlusTreeLeaf`. Para este caso, usted también puede suponer que los nodos vecinos serán del mismo tipo que `BPlusTreeDir`.

Es importante recalcar que la redistribución se hace solamente con los nodos vecinos que tienen el mismo padre. En particular, si el nodo es el de más de la izquierda del padre, entonces este solo puede redistribuir con su hermano de la derecha.

Implementación de eliminación simple en `BPlusTree` (bonus 1 punto)

Para esta tarea, usted deberá implementar la eliminación simple en `BPlusTree`. Para esto, usted deberá implementar un método `remove(const Record& record)` que encuentre el record (si existe) y lo elimine del B+tree, siempre y cuando no sea necesario hacer un `unsplit`. Esto es, si hay espacio suficiente en el nodo o se pueda hacer redistribución con los vecinos, entonces se eliminará el record. En caso que esto no se pueda y sea necesario hacer un `unsplit`, entonces su código deberá levantar una excepción de que no se pudo hacer la eliminación. Notar que para esta tarea usted debe hacer la eliminación, pero también hacer una redistribución de eliminación si es necesario.

Implementación de eliminación con `unsplit` en `BPlusTree` (bonus 1 punto)

Para esta tarea, usted deberá extender la eliminación simple anterior para también soportar la eliminación con `unsplit`. Para esto, usted deberá responder al hecho de no poder hacer una eliminación en el caso anterior y hacer los `unsplits` necesarios para rebalancear el árbol. Notar que su solución también debe soportar la reducción de la altura del árbol (`unsplit` de la raíz) en caso de ser necesario.

Bonus por bugs encontrados/solucionados en IIC3413-DB

Se darán los siguientes bonus por encontrar o solucionar un bug de IIC3413-DB :

1. **Bug encontrado:** se darán 2 décimas en la nota final del laboratorio si encuentran un bug. Para ser efectivo este bonus, deben publicar una issue¹ del repositorio del proyecto sobre el bug, esto es, la descripción del bug, junto con el input que se le está entregando al sistema más el output “errado” que entrega.
2. **Bug solucionado:** se darán 2 décimas adicionales por entregar el código que soluciona este bug en el sistema (no necesariamente tiene que ser el mismo estudiante). Para esto deben dar una explicación de como solucionar el problema en el mismo issue y hacer un pull request con el código que soluciona el problema (no importa si es una línea o muchas líneas). Aparte de tener las dos décimas, su usuario GitHub quedará como contribuidor al proyecto.

Una vez publicado el bug o solución, los ingenieros del proyecto revisarán la publicación y determinarán si el bonus corresponde o no. Por último, dado un bug o solución se dará el bonus al primer estudiante que lo publicó en las issues del repositorio.

Bonus por uso de nueva versión de IIC3413-DB (0.2 puntos)

Existe una nueva versión de IIC3413-DB en el repositorio del proyecto, para la cual se arreglaron algunos errores y se extendieron algunas funcionalidades del sistema. En particular, tanto la versión nueva como la versión usada en el laboratorio anterior tienen el mismo código para el índice de B+tree.

Para esta entrega usted puede usar la versión antigua o la versión nueva de IIC3413-DB . En caso de hacer su solución en la versión nueva del sistema, se le otorgará 0.2 puntos de bonus en su nota final (2 décimas). También en caso de encontrar un bug en la versión antigua que ya fue solucionado en la versión nueva, entonces este bonus por bug encontrado no será válido.

¹<https://github.com/PUC-IIC3141/IIC3413-DB/issues>

Evaluación y entrega

El día límite para la entrega de esta tarea será el Viernes 1 de mayo a las 23:59 horas. Para ello se utilizará el repositorio privado en GitHub que fue proporcionado por el curso. Para la entrega el alumno deberá crear una nueva rama en git a partir de la rama principal del proyecto con nombre “laboratorio2”, para que así pueda seguir desarrollando sus próximos laboratorios sin considerar las modificaciones del actual. La evaluación se realizará en base a test que debe pasar su código.

Ayudantía y preguntas

El día Viernes 17 de abril se realizará una ayudantía donde se darán más detalles sobre IIC3413-DB y el laboratorio. Para preguntas se pide usar el foro del curso o enviar un correo a iic3413@inc.puc.cl. De preferencia, se sugiere escribir al foro del curso para que todos los estudiantes estén al tanto de las dudas.