Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC2413 - Bases de Datos

Ayudantía I3

treisenegger@uc.cl

12 de noviembre de 2018

Problema 1: Recuperación de Fallas

- Suponga que su sistema tuvo una falla. Si su política es la de Undo Logging, indique qué parte del *log file* debo leer y qué operaciones se deben deshacer para los siguientes archivos de *logs*.

Log
<START T1>
<T1, A, 5>
<START T2>
<T2, B, 10>
<START CKPT(T1, T2)>
<T2, C, 15>
<START T3>
<T1, D, 20>
<COMMIT T1>
<T3, E, 25>
<COMMIT T2>
<END CKPT>
<T3, F, 30>

Log
<START T1>
<T1, A, 5>
<START T2>
<T2, B, 10>
<START CKPT(T1, T2)>
<T2, C, 15>
<START T3>
<T1, D, 20>
<COMMIT T1>
<T3, E, 25>

- Suponga que su sistema tuvo una falla. Si su política es la de Redo Logging, indique qué parte del *log file* debo leer y qué operaciones se deben rehacer para los siguientes archivos de *logs*.

Log
<START T1>
<T1, A, 5>
<START T2>
<COMMIT T1>
<T2, B, 10>
<START CKPT(T2)>
<T2, C, 15>
<START T3>
<T3, D, 20>
<END CKPT>
<COMMIT T2>
<COMMIT T3>

Log
<START T1>
<T1, A, 5>
<START T2>
<COMMIT T1>
<T2, B, 10>
<START CKPT(T2)>
<T2, C, 15>
<START T3>
<T3, D, 20>

Problema 2: Transacciones (I3 2017-2)

Considere el Schedule del cuadro 1. Diga si es o no *conflict serializable*. En caso de que no lo sea, explique por qué e indique cómo Strict-2PL puede resolver el problema.

T1	T2	T3	T4
R(a)			
	R(b)		
		W(a)	
			R(a)
	R(a)		
	R(d)		
W(d)			

Cuadro 1: schedule problema 2

Problema 3: MongoDB (I3 2016-2)

Piense en una base de datos en MongoDB con dos colecciones, una de personas, otra de compras y otra de posts en una red social. Se muestra a continuación una instancia de documento para cada colección:

```
// Persona
{
     "id_persona": 1,
     "name": "Adrián Soto"
}
```

```
// Post
{
     "id_post": 1,
     "id_persona": 1,
     "content": "Mañana juego una pichanga!"
}

// Compra
{
     "id_compra": 1,
     "id_persona": 1,
     "fecha": "10-09-2016",
     "compras": [
          {
               "producto": "Pelota",
               "tipo": "Deporte",
               "valor": 15000
          },
          {
               "producto": "Café con leche",
               "tipo": "Comida",
               "valor": 1000
          }
     ]
}
```

- Entregue los posts que contengan la palabra "Gol" y la frase "Vamos Chile" pero no la frase "Vamos Colombia" junto al nombre del usuario que lo emitió.
- Entregue cada persona junto al nombre de cada producto tipo "Deporte" que ha comprado.
- Para cada persona que haya emitido un post que contenga la frase "vamo a Calmarno" y no la frase "vamo a evolucionarlo" indique el nombre de la persona junto al total gastado comprando productos tipo "Pokémon".

Problema 4: External Merge Sort (Guía I3)

Considere una relación de 3.000.000 páginas.

- Indique el número de fases y de I/O del External Merge Sort no optimizado.

Solución: Las siguientes son las fórmulas ocupadas para encontrar la cantidad de fases y de I/O del External Merge Sort.

- Fases: $1 + \lceil \log_2(N) \rceil = 1 + \lceil \log_2(3000000) \rceil = 23$
- I/O: $2 \cdot N \cdot (1 + \lceil \log_2(N) \rceil) = 2 \cdot 3000000 \cdot (1 + \lceil \log_2(3000000) \rceil) = 138000000$

- Indique el número de fases y de I/O del External Merge Sort optimizado que imprime directamente el resultado final, sin materializar el resultado final. Suponga buffer óptimo.

  Solución: El número de fases para External Merge Sort optimizado con un buffer óptimo es siempre igual a 2. Además, el número de I/O en este caso es de $4 \cdot N$. Sin embargo, dado que no se materializa el resultado final, este número disminuirá a $3 \cdot N$. Así, las cantidades pedidas son las siguientes.

  - Fases: 2
  - I/O: $3 \cdot N = 3 \cdot 3000000 = 9000000$

- ¿Qué significa que el buffer sea óptimo? ¿Por qué un buffer en el que caben 3.000.000 de páginas no es necesario? Demuestre su respuesta.

  Solución: Que el buffer sea óptimo signfica que podemos realizar el algoritmo en el menor número de fases posible. En otras palabras, el buffer es suficientemente grande como para que podamos ejecutar el algoritmo en dos fases. Para esto necesitamos que se cumpla la siguiente condición.

$$\begin{array}{rcl} 2 & = & 1 + \lceil \log_B(\lceil \frac{N}{B+1} \rceil) \rceil \\ & \Downarrow & \\ B & \geq & \sqrt{N} \end{array}$$

  Así, bastará con que en el buffer quepan 1733 páginas.

Problema 5: Map Reduce (I3 2017-2)

Considere un esquema de dos tablas `A(id int, name varchar(10))` y `B(id int, name varchar(10))`. Suponga que quiere hacer la consulta en álgebra relacional $\pi_{A.name,B.name}(A \bowtie B)$, pero solamente dispone de un archivo que se ve de la siguiente forma:

```
A,1,palabra1
A,2,palabra2
A,3,palabra3
B,1,palabra1
...
```

El primer término antes de la coma representa la tabla, el segundo el id y el tercero el name. Explique con palabras un algoritmo Map - Reduce que ejecute la consulta deseada.

Solución: El esquema Map - Reduce se basa en las dos etapas que indica su nombre. Un ejemplo de algoritmo que podría servir para resolver este problema se detalla a continuación.

- *Map:* En primer lugar, debemos encontrar un par *key - value* para cada valor. Cada entrada será mapeada a un nodo *reducer* según su llave, por lo que tenemos que escoger alguna llave tal que solo se necesiten realizar opearaciones entre entradas de la misma llave. En este caso, utilizaremos la columna `id` de cada fila. El valor de cada fila será el resto de los valores que la componen. En este caso, estos corresponden al nombre de la tabla y al `name`. Así, los pares *key - value* se verán de la forma

$$\begin{aligned} 1 &\rightarrow \texttt{A}, \texttt{palabra1} \\ 2 &\rightarrow \texttt{A}, \texttt{palabra2} \\ 3 &\rightarrow \texttt{A}, \texttt{palabra3} \\ 1 &\rightarrow \texttt{B}, \texttt{palabra1} \\ &\dots \end{aligned}$$

- *Reduce:* Como mencionamos antes, a un reducer le llegarán todas las filas que presenten la misma llave. Así, lo que se hará en esta etapa será, en primer lugar, separar las filas por la tabla de la cual provienen. En segundo lugar, se hará el producto cruz entre todas las filas de cada tabla. Estas operaciones serán de la siguiente forma

$$\begin{array}{cc} \texttt{A}, \texttt{palabra1} & \texttt{B}, \texttt{palabra1} \\ \dots & \dots \end{array}$$

$$\Downarrow$$

$$\begin{array}{c} \texttt{palabra1}, \texttt{palabra1} \\ \dots \end{array}$$