Ayudantía 9

Operadores físicos y optimización del plan físico

IIC2413, Bases de Datos
Benjamín Saavedra

En todas las preguntas el buffer tiene $B + 1$ páginas y los costos son en I/O, con las fórmulas de la clase.

Pregunta 1

Un archivo de N páginas se ordena con external merge sort (buffer de $B+1$ páginas, runs iniciales de $B + 1$ páginas, mezcla de B runs a la vez); su costo es $2N$ por el número de fases.

1. Con $B + 1 = 5$, entregue el mayor N que se ordena en dos fases y el menor N que necesita cuatro. Explique.
2. Una bitácora dice: la fase 0 produjo 37 runs, hubo 4 fases y 7.400 I/O. Muestre que no puede ser correcta. Si el número de runs y el de I/O fueran correctos, ¿cuántas fases hubo y cuánto valía B?
3. Sea $\text{cost}(N)$ el costo de ordenar N páginas. Demuestre que $\text{cost}(kN) \geq k \cdot \text{cost}(N)$ y dé N y B con desigualdad estricta para $k = 2$.

Adaptada de Garcia-Molina, Ullman y Widom (GMUW), Database Systems: The Complete Book*, 2.ª ed., §15.4, Ejercicio 15.4.6, p. 731.*

Pregunta 2

Se calcula $\delta(R)$ por hashing con un buffer de 3 páginas: R se reparte con una función de hash en $B = 2$ buckets; cada bucket se lee completo a memoria para eliminar sus duplicados y, si no cabe, se repite el reparto sobre él con otra función de hash.

1. Entregue una relación unaria R de 10 páginas para la que el algoritmo no termina, sea cual sea la función de hash. Explique.
2. Modifique solo la lectura de cada bucket para que termine con su R de (1), y diga cuándo todavía habría que repetir el reparto.
3. Sea $\text{pages}(R) = 2.000$, $\text{pages}(\delta(R)) = 600$ y $B + 1 = 101$. Para `SELECT DISTINCT a FROM R ORDER BY a`, calcule el costo de δ por hashing seguido de un sort y el de δ por sorting. Explique la diferencia.

Adaptada de GMUW §15.5, Ejercicio 15.5.4, pp. 738–739.

Pregunta 3

$R(\mathsf{id}, \mathsf{A}, \mathsf{B})$, con llave id, tiene 1.000.000 de tuplas, 100 por página, y B+trees unclustered sobre A y sobre B; cada atributo tiene 50 valores distintos. Para $\sigma_{A=a \,\wedge\, B=b}(R)$ se puede usar el full scan o ambos índices: cada índice cuesta 3 y luego se lee una página por cada tupla de la intersección.

1. Con S1 y S2, ¿qué costo estima el optimizador para cada plan? ¿Cuál elige?
2. Suponga que cada valor de A aparece el mismo número de veces y siempre con un único valor de B, y viceversa. Calcule el costo real del plan elegido y el menor número de tuplas del resultado para el que ese plan es peor que el full scan.
3. ¿Qué cifra del catálogo corrige la estimación de (2)? ¿Qué forma normal viola R en ese caso?
4. Si A y B fueran independientes con d valores distintos cada uno, y para $\sigma_{A=a \,\vee\, B=b}$ se usara $\mathrm{sel}_{\theta_1 \vee \theta_2}(R) = \mathrm{sel}_{\theta_1}(R) + \mathrm{sel}_{\theta_2}(R) - \mathrm{sel}_{\theta_1}(R)\mathrm{sel}_{\theta_2}(R)$, ¿cuál es el menor d para que usar ambos índices sea mejor que el full scan?

Adaptada de GMUW §16.7, Ejercicios 16.7.1 y 16.7.2, p. 838.

Pregunta 4

Con R como relación exterior y un buffer de $B+1$ páginas ($B \geq 2$), un block nested loops join cuesta $\mathrm{pages}(R) + \lceil \mathrm{pages}(R)/B \rceil \cdot \mathrm{pages}(S)$.

1. Entregue $\mathrm{pages}(R) < \mathrm{pages}(S)$ y B tales que poner S como exterior cueste estrictamente menos.
2. Demuestre que si $\mathrm{pages}(R) \leq \mathrm{pages}(S)$ y $\mathrm{pages}(R) \leq B$, poner R como exterior nunca es peor.
3. Demuestre que si $\lceil \mathrm{pages}(R)/B \rceil = \lceil \mathrm{pages}(S)/B \rceil = k > 1$, gana poner la mayor afuera, y diga por cuánto.
4. Explique cómo calcular R LEFT JOIN S con este algoritmo y qué memoria extra necesita. ¿Por qué no se puede igual con S como exterior?

Adaptada de GMUW §15.3, Ejercicios 15.3.2 y 15.3.3, p. 722, y §15.2, Ejercicio 15.2.4, p. 718; inspirada en preguntas de outer joins de la I2 de 2018 y del Control 2.

Pregunta 5

Sean R y S relaciones base con $\mathrm{pages}(R) \leq \mathrm{pages}(S)$. Un hash join cuesta $3(\mathrm{pages}(R) + \mathrm{pages}(S))$ y un block nested loops join con R exterior, $\mathrm{pages}(R) + \lceil \mathrm{pages}(R)/B \rceil \mathrm{pages}(S)$.

1. Demuestre que si $\lceil \mathrm{pages}(R)/B \rceil \leq 3$, el block nested loops no cuesta más que el hash join.

2. Entregue pages(R), pages(S) y B con $\lceil \text{pages}(R)/B \rceil = 4$ tales que el block nested loops sea estrictamente mejor. ¿Qué condición debe cumplir pages(R)/pages(S)?

3. Con un buffer de $B + 1$ páginas, el hash join particiona R y S en B buckets y luego, para cada i, carga R_i completo en memoria, lee S_i de a una página y escribe el resultado en otra. El sort-merge deja cada relación en runs de $B + 1$ páginas y luego mezcla todos los runs de R y de S a la vez, con una página por run y una de salida.

   (a) Derive la condición sobre B para que el hash join necesite una sola fase de partición.

   (b) Derive la condición sobre B para que el sort-merge mezcle todos los runs de una vez.

   (c) Con pages(R) = 30.000 y pages(S) = 90.000, calcule el menor B de cada uno y el costo que tendrían con ese B. Explique la diferencia.

4. S tiene 6.000 páginas, 27 tuplas por página y un B+tree clustered sobre su llave; R también tiene 27 por página, $B = 100$ y cada tupla de R se une con una de S. Un INLJ cuesta pages(R) + $3|R|$ (con la aproximación cost(I) ≈ 3 por búsqueda en el índice).

   (a) Dé $|R_1| < |R_2| < |R_3|$ tales que el INLJ gane en R_1 y en R_3 y pierda con el block nested loops en R_2.

   (b) Proponga un algoritmo que combine block nested loops e INLJ y que nunca cueste más que ninguno de los dos. Demuestre que lo cumple y calcule su costo para $|R_3|$.

Adaptada del resumen de costos de joins de la clase 19 y de GMUW §15.6, Ejercicio 15.6.5, pp. 745–746.

Pregunta 6

Se estima $|X \bowtie Y| = |X|\,|Y| / \max\{\text{distinct}\}$ sobre los atributos compartidos (S1 y S3), y el número de distintos de un atributo compartido en el resultado es el mínimo de los dos. El costo de un plan es la suma de los tamaños de sus resultados intermedios (sin el final) y no hay productos cruz. Considere $R(a, b)$, $S(b, c)$, $T(c, d)$ y $U(d, e)$. El algoritmo greedy une en cada paso los dos planes P, P' con menor $|P \bowtie P'|$.

1. Entregue tuplas y valores distintos para las cuatro relaciones tales que el plan de greedy no sea óptimo y el plan óptimo no sea left-deep.

2. Para su catálogo, entregue el plan de greedy, el óptimo y sus costos.

3. Explique por qué en una cadena de cuatro relaciones todo plan left-deep materializa un resultado de tres relaciones.

4. ¿Cuántos árboles de join hay para 5 relaciones y cuántos de ellos son left-deep?

Inspirada en GMUW §16.6, Ejercicios 16.6.4 y 16.6.5, pp. 825–826.

Pregunta 7

Un `BNL-JOIN` lee su relación interior Q, un subplan, una vez por cada bloque de la exterior. Sean $c = \text{cost}(Q)$ y $p = \text{pages}(Q)$.

1. Materializar Q significa calcularla una vez, escribirla y leerla k veces. Demuestre que conviene si y solo si $k\,(c - p) > c + p$, y diga cuándo nunca conviene.
2. Con $c = 23$, $p = 8$, una exterior de 3.000 páginas y un buffer de $B + 1$ páginas, ¿cuál es el menor B desde el cual ya no conviene materializar?
3. Con $B = 100$, la exterior (3.000 páginas) es mucho mayor que Q. ¿Qué regla de la clase 19 sugiere otro orden? Calcule su costo y compárelo con materializar.

Adaptada de GMUW §16.7, Ejercicio 16.7.3, p. 838.

Pregunta 8

Ventas24 y **Ventas25** tienen el esquema (id, cliente, monto), con id llave primaria en cada una, y ocupan 3.000 y 5.000 páginas.

1. ¿Basta eso para reemplazar `UNION` por `UNION ALL`? Justifique.
2. Entregue una condición que sí lo permita. Si la unión se hace por sorting, ¿cuánto se ahorra?
3. Si ambas estuvieran guardadas ordenadas por id, ¿cuánto se ahorraría con el reemplazo? Explique.

Idea propia, a partir del costo de `UNION` y `UNION ALL` de la clase 18.

Pregunta 9

El *semijoin* $R \ltimes S$ de $R(A, X)$ y $S(B, Y)$ entrega las tuplas de R que tienen al menos una pareja en S ($B = A$), cada una con su multiplicidad en R (en SQL, `WHERE EXISTS`).

1. Con $\text{pages}(R) = 10.000$, $\text{pages}(S) = 5.000$ y solo 50 valores distintos de B, ¿cómo se calcula con una sola pasada sobre R y cuánto cuesta? Compárelo con hash join.
2. ¿Es $R \ltimes S = \pi_{A,X}(R \bowtie S)$ con duplicados? Dé un contraejemplo y una condición que la haga cierta.
3. Con un B+tree unclustered sobre $S.B$ (40 tuplas de S por valor), $\text{pages}(R) = 2.000$ y $|R| = 54.000$, ¿cuánto cuesta el join y cuánto el semijoin con INLJ, si cada búsqueda cuesta 3 (aproximación $\text{cost}(I) \approx 3$) y leer cada pareja, 1?

Adaptada de GMUW §15.2, Ejercicio 15.2.4, p. 718, y de la pregunta de semijoin de la I2 de 2022. El semijoin no está en las clases 17 a 20.