CS522 Advanced Database Systems

Query Processing

Chengyu Sun
California State University, Los Angeles

SQL Query

- <SFW> syntax

SELECT <SelList> FROM <FromList> WHERE <Condition>

SQL Query Example

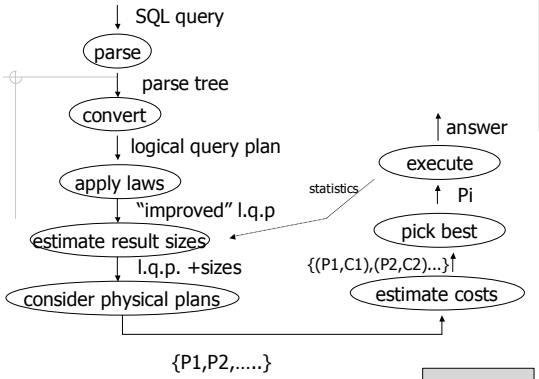
R

A	B	C
a	1	10
b	1	20
c	2	10
d	2	25
e	3	45

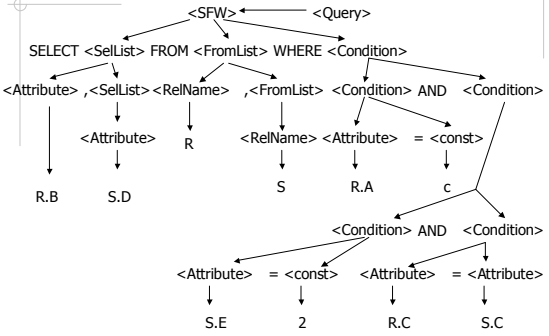
S

C	D	E
10	x	2
20	y	2
30	z	2
40	x	1
50	y	3

```
select B, D
       from R, S
       where R.A = c and S.E = 2 and R.C = S.C;
```



From Query to Parse Tree



Relational Algebra Notations

$\cup$	UNION
$\cap$	INTERSECTION
–	DIFFERENCE
σ_c	$SELECT_c$
π_L	$PROJ_L$
X	*
$\triangleright\triangleleft$	JOIN
$\triangleright\triangleleft_c$	$JOIN_c$
ρ	RENAME

δ	DELTA
τ	TAU
γ	GAMMA
π_L	$PROJ_L$
$\triangleright^O\triangleleft$	OUTERJOIN
$\triangleright^O\triangleleft_L$	LEFTJOIN
$\triangleright^O\triangleleft_R$	RIGHTJOIN
SUM, AVG, COUNT,MIN,MAX	

- Review Chapter 5

Sets and Bags

- Bag (multi-set) semantics
  - Allow duplicates
  - default for SQL
- Set semantics
  - No duplicates
  - More expensive to implement in general

From Parse Tree to Logical Query Plan

- With no subquery
  - Start with a product of all relations in the <FromList>
  - Add σ_c for each condition C
  - Add π_L for all attributes in the <SelList> L
- With subquery
  - see Query Optimization

From LQP to Best LQP

- Plan rewriting
  - Algebraic laws
  - Heuristics
- Choose the best plan
  - Cost estimation – result size

From Logical Query Plan to Physical Query Plan

- Plan enumeration
- Choose the right physical operator
  - Estimate algorithmic cost
- Other considerations

Cost Model for Physical Operators

- B data pages
- M buffer pages
- T tuples
- V(a) or V(a_1,a_2,...,a_n) distinct values

- Simplifications
  - in-memory operations are free
    - Computation and *storage*
  - Output cost does not count

Algorithms for Physical Operators

- One-pass algorithms
- Two-pass algorithms
- Multi-pass algorithms
- Index-based algorithms

One-pass Algorithms

- Simple
- Efficient
- *May* depends on the size of memory buffer
  - e.g. apply only to relations with B < M
  - Potential performance degradation if M is not estimated correctly

Selection and Projection

- One tuple at a time
- *Restriction on relation size*: none

Duplicate Elimination and Grouping

- Duplicate elimination
  - In-memory data structure??
  - *Relation size*??
- Grouping
  - MIN, MAX, COUNT, SUM, AVG
  - *Relation size*??

Union, Intersection, and Difference

- with Bag semantics
  - Union: trivial
  - Intersection: min(|R|,|S|) < M
  - Difference: ??
- with Set semantics
  - Union: ??
  - Intersection: ??
  - Difference: ??

Product and Join

- Product
- Nested loop join
  - One relation fits in memory buffer
  - Neither relations fits in memory buffer – *one and a half pass*

Summary of One-pass Algorithms

Operators	M Required	Disk I/O
σ, π	1	B
γ, δ	B	B
$\cup$, $\cap$, $-$	min(B(R),B(S))	B(R)+B(S)
$\times$, $\triangleright\triangleleft$	min(B(R),B(S))	B(R)+B(S)
$\times$, $\triangleright\triangleleft$	M≥2	B(R)B(S)/M

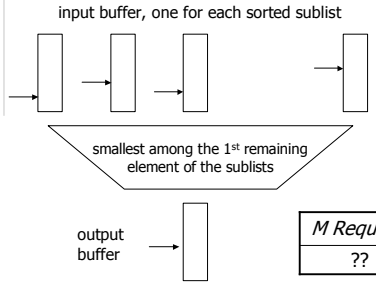
Two-pass Algorithms

- Can handle very large relations
- Sort-based algorithms
- Hash-based algorithms

Two Phase Multiway Merge Sort (TPMMS)

- Phase 1: fill memory buffer, in-memory sort
- Phase 2: merge *sorted sublists*, one block from each

TPMMS Merge Phase



M Required	*Disk I/O*
??	??

Duplicate Elimination and Grouping

- Example
  - R: {2, 5, 2, 1, 2, 2, 4, 5, 4, 3, 4, 2, 1, 5, 2, 1, 3}
  - 2 tuples per page
  - 3 buffer pages

Union, Intersection, and Difference

- Example
  - R: {2, 5, 2, 1, 2, 2, 4, 5, 4, 3, 4, 2}
  - S: {1, 5, 2, 1, 3}
  - 2 tuples per page
  - 3 buffer pages
- Bag or Set??

Simple Sort Join

- Sort R, sort S, output matching tuples
- Example
  - B(R) = 1000
  - B(S) = 500
  - M = 101
- vs. Nested Loop Join??

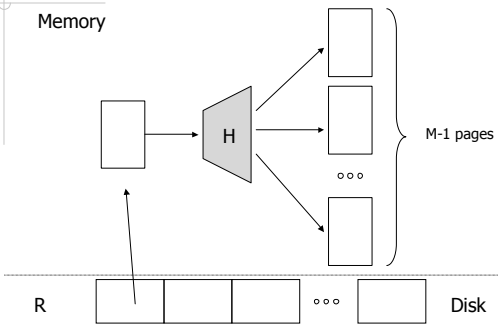
Sort Merge Join

- Join in the merge phase
- vs. Simple Sort Join??

Summary of Two-pass Algorithms (Sorting)

Operators	M Required	Disk I/O
γ, δ		
$\cup$, $\cap$, $-$		
$\triangleright\triangleleft$ (simple sort)		
$\triangleright\triangleleft$ (merge sort)		

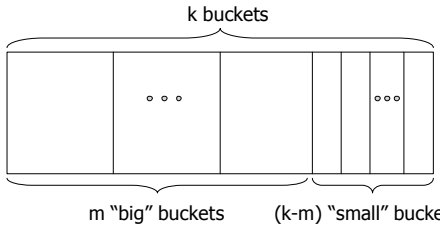
Hashing



Hash-based Algorithms

- Duplicate eliminations
- Grouping and aggregation
- Union, intersection, and difference
- Join

Hybrid Hash Join



- one buffer page for a "small" bucket
- multiple buffer pages for a "big" bucket s.t. *all tuples in the bucket can be kept in memory*

Hybrid Hash Join Parameters

- The smaller k, the better
- k cannot be too small

Summary of Two-pass Algorithms (Hashing)

Operators	M Required	Disk I/O
γ, δ		
$\cup$, $\cap$, $-$		
$\triangleright\triangleleft$ (hash)		
$\triangleright\triangleleft$ (hybrid hash)		

Sorting vs. Hashing

- Similar complexity and limitations
- Hashing looks better on paper
  - What's the catch??
- Sorting allows more optimization

Multi-pass Algorithms

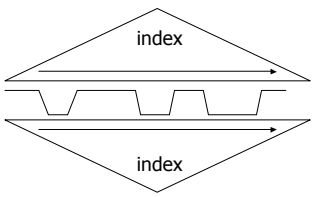
- kPMMS
  - What's the largest relation we can sort in k passes??
  - What's the I/O complexity of kPMMS??

Index-based Algorithms

- Clustered index vs. Un-clustered index

Indexed Join

- Indexed Nested Loop Join
- Zig-zag Join



Reading

- Stanford book: Chapter 15
- [Join survey paper]