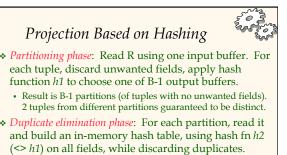


Database Management Systems 3ed, R. Ramakrishnan and Johannes Gehrke



- If partition does not fit in memory, can apply hash-based projection algorithm recursively to this partition.
- Cost: For partitioning, read R, write out each tuple, but with fewer fields. This is read in next phase. Database Management Systems 3ed, R. Ramakrishnan and Johannes Gehrke

Discussion of Projection

- Sort-based approach is the standard; better handling of skew and result is sorted.
- If an index on the relation contains all wanted attributes in its search key, can do *index-only* scan.
- Apply projection techniques to data entries (much smaller!)
 If an ordered (i.e., tree) index contains all wanted
- attributes as *prefix* of search key, can do even better: • Retrieve data entries in order (index-only scan), discard
 - unwanted fields, compare adjacent tuples to check for duplicates.
- Database Management Systems 3ed, R. Ramakrishnan and Johannes Gehrke

Set Operations

- Intersection and cross-product special cases of join.
- Union (Distinct) and Except similar; we'll do union.
- Sorting based approach to union:
 - Sort both relations (on combination of all attributes).
 - Scan sorted relations and merge them.
 - Alternative: Merge runs from Pass 0 for both relations.
- Hash based approach to union:
- Partition R and S using hash function *h*.
- For each S-partition, build in-memory hash table (using h2), scan corr. R-partition and add tuples to table while discarding duplicates.
- atabase Management Systems 3ed, R. Ramakrishnan and Johannes Gehrke

Aggregate Operations (AVG, MIN, etc.)

- Without grouping:
 - In general, requires scanning the relation.
 - Given index whose search key includes all attributes in the
- SELECT or WHERE clauses, can do index-only scan.
- With grouping:

Summary

- Sort on group-by attributes, then scan relation and compute aggregate for each group. (Can improve upon this by combining sorting and aggregate computation.)
- Similar approach based on hashing on group-by attributes.
- Given tree index whose search key includes all attributes in SELECT, WHERE and GROUP BY clauses, can do index-only scan; if group-by attributes form prefix of search key, can
- retrieve data entries/tuples in group-by order.

Impact of Buffering

- If several operations are executing concurrently, estimating the number of available buffer pages is guesswork.
- Repeated access patterns interact with buffer replacement policy.
 - e.g., Inner relation is scanned repeatedly in Simple Nested Loop Join. With enough buffer pages to hold inner, replacement policy does not matter. Otherwise, MRU is best, LRU is worst (sequential flooding).
 - · Does replacement policy matter for Block Nested Loops?
- What about Index Nested Loops? Sort-Merge Join?
 tabase Management Systems 3ed, R. Ramakrishnan and Johannes Gehrke



- A virtue of relational DBMSs: *queries are composed of a few basic operators;* the implementation of these operators can be carefully tuned (and it is important to do this!).
- Many alternative implementation techniques for each operator; no universally superior technique for most operators.
- Must consider available alternatives for each operation in a query and choose best one based on system statistics, etc. This is part of the broader task of optimizing a query composed of several ops. Database Management Systems 3ed, R. Ramakrishnan and Johannes Gehrke