Block Nested Join

Suppose smaller relation (say R) can be held in buffer w/ two pages left over.

Then can scan R once into memory and use remaining two pages to do join w/ S.

Total I/O cost M+N. (much better)

Even if don't have enough memory to completely hold smaller rel R could by holding B-2 pages of it, yielding:

```
For each block of B-2 pages of R do:
    foreach page of S do
    {
        for all matching in-memory tuples
        r ∈ R-block & s ∈ S-page output
        <r, s>
    }
```

Now S is scanned $\lceil \frac{M}{B-2} \rceil$ times

So total cost is $M + N \cdot \lceil \frac{M}{B-2} \rceil$

How do we do matching:

1. Build a hash table in memory for the block of R's page.
2. Use this to look up matches for an S.

Suppose can hold in hash table 100 pages of R
Then total cost is $1000 + 500 \cdot \left\lceil \frac{1000}{100} \right\rceil = 6000$ I/Os

Sort Merge Join

Algorithm for $R \bowtie_{a=b} S$

External merge sort R
External merge sort S } – result as partitioned into set on which join value is constant

Scan R (as much as can into buffer)

For each partition of R merge with corresponding partition of S

Cost If join condition is equality on key

$2M \log M + 2N \log N + M + N$

$2M \log M$ sort R, $2N \log N$ sort S

For running example
consider term →

$2 \cdot 2 \cdot 1000 = 4000$

$2 \cdot 2 \cdot 500 = 2000$

1000
500
7500

Hash Join (as with hash based projection)
have condition $B > \sqrt{f \cdot (M+N)}$

hash both relations on the join attribute using same hash fn

Produce B-1 partition files containing rec from R & S.

Now read in each partition apply a 2nd hash fn to B-1 partitions (hopefully < f·1 page in size). Merge & output tuples single

$3(M+N)$

Overview of Transaction Management

What properties do we want of transactions?

ACID

- atomic – each transaction has all its operations applied to DB or none of them
- consistency – transaction takes DB from consistent state to consistent state
- isolation – actions of a transaction should not be affected by concurrently running transactions.
- durability – affects of transaction should persist even if system crashes (recovery manager)

Transactions & Schedules

transaction – a list of operations to be applied to DB. Ex) withdraw $40 bank account.

operations – usually reads or writes of DB objects. Written as $R_T(O)$ or $W_T(O)$.

– also have commit (complete successfully) and abort operations $Commit_T$, $Abort_T$.

Assumptions – transactions do not communicate w/ each other

– A db is a fixed collection of independent objects