University of Waterloo
Midterm Examination #1 Model Solution

Spring, 2007

1. (12 marks)

Suppose that a database includes two relations, R and S. Relation R has N_R tuples and occupies B_R blocks on the disk. Relation S has N_S tuples and occupies B_S blocks on the disk.

The database system needs to process a query which involves an equijoin of R and S. The system is considering 4 different plans for processing this query. For each of these plans, which are listed below, give an estimate of the total I/O cost of the plan, ignoring the cost of materializing the join output. In each case, you should assume that the join operator has sufficient memory available to hold M blocks of data, where $B_R < M < B_S$ and $B_S < M^2$. Express your answers in terms of B_R, B_S, N_R, N_S, and M. Briefly explain your answers.

Plan A: (3 marks) Block-oriented nested-loop join with S as the inner relation. Both R and S are accessed using table scans.

All of R is read into memory, and then all of S is read.

$$B_R + B_S$$

Plan B: (3 marks) Block-oriented nested-loop join with R as the inner relation. Both R and S are accessed using table scans.

S is read into memory M blocks at a time. For each M blocks, R is scanned.

$$B_S + \lceil \frac{B_S}{M} \rceil B_R$$

Plan C: (3 marks) Basic hash join (i.e., Grace hash join, not hybrid hash join) with R as the build relation and S as the probe relation. Both R and S are accessed using table scans.

Each relation is read in, partitioned, and written out to temporary storage, at a total cost of $2(B_R + B_S)$. Pairs of R and S partitions are then read in and joined, at an additonal cost of $B_R + B_S$. Thus, the total cost is $3(B_R + B_S)$.
A more general implementation might recognize that partitioning is not necessary at all because $B_R < M$, allowing the entire build relation to fit into memory. In this case, the build relation (R) is read once and hashed in memory. The probe relation is then read once and its tuples are matched to R using the has table. In this case, the total cost is $B_S + B_R$

Plan D: (3 marks) Basic hash join with S as the build relation and R as the probe relation. Both R and S are accessed using table scans.

All of the data from both relations are read, partitioned, written out to temporary files, and then read back in one partition at a time.

$$3(B_S + B_R)$$

2. (8 marks)

For this question, consider a B+-tree defined on a database relation. The B+-tree's leaves hold RIDs, not the actual tuples. The size of the index key values is such that N (key,pointer) or (key,RID) pairs can be stored in each block of the index.

a. (3 marks)
What is the maximum relation size (i.e., the maximum number of tuples) that can be indexed using a B+-tree of height 3, and what is the total number of index blocks in such a B+-tree?

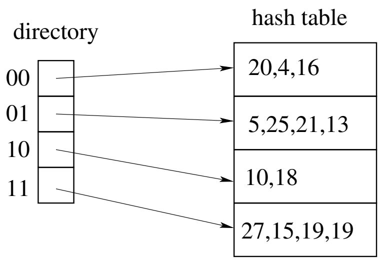
There will be one root block, up to N blocks in the second level, and up to N^2 blocks in the leaf level. Each leaf block refers to up to N tuples. The maximum indexable relation size is N^3 and the maximum number of index blocks is $1 + N + N^2$.

b. (5 marks)
Suppose that the B+-tree is modified to that it implements prefix compression of the index keys. The effect of the compression is that αN ($\alpha > 1$) (key,pointer) values can fit into an index node. Suppose that such a prefix-compressed index is used to index a relation of the same maximum size that you determined in part (a). Assuming that all index nodes are completely full (except possibly the root node), what is the the total number of index blocks in the prefix-compressed index?

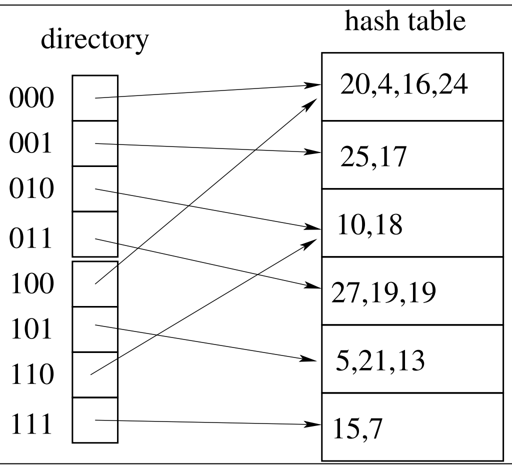
Since only non-leaf blocks are compressed, there will be N^2 leaf blocks, as in part (a). $N^2/(\alpha N)$ will be required at the next level, plus a single root node. Thus, the total number of blocks required will be $1 + N/\alpha + N^2$.

3. (10 marks)

The diagram below shows an extensible hash table with four hash buckets. Each number x in the buckets represents an entry for a record for which the hashed key value is x. For example, the number 20 in the first hash bucket represents a record for which the hashed key value is 20.



Draw a similar diagram showing what this extensible hash table will look like after the following sequence of records is inserted: 17, then 24, then 7. Again, each number in the insertion sequence represents the hashed key value of a record to be inserted.



4. (5 marks)

a. (3 marks)

Briefly explain the distinctions between the RAID0, RAID1, and RAID5 disk organizations.

In RAID0, the data are striped across all of the available disks. In RAID1, each data block is replicated on two disks. RAID5 is like RAID0, but parity blocks are used to improve fault tolerance.

b. (2 marks)

Briefly explain the difference between a clustered index and an unclustered index.

In a clustered index, the logical order of the index key values is similar to the physical order of the tuples in storage. In an unclustered index, this is not the case.

5. (5 marks)

Suppose that a database management system uses a B+-tree index of height 3 to index a relation. Assume that the leaves of the index hold tuples from the relation, not RIDs. In the *worst case*, what is the I/O cost of inserting a single record into the relation using this B+-tree? Indicate separately how many block read operations and how many block write operations will be required, and justify your answer. Assume that the system has a large buffer cache, but that initially the cache does not contain any of the B+-tree's nodes.

The worst case situation occurs when the B+-tree splits all of the way to the root and grows by one level. In this case, one index block at each of the original 3 levels will be read. Each of these blocks will be split, with the resulting pair of blocks being written to the disk. In addition, the new root block must be wrritten. Thus, there will be a total of 3 disk read operations and $2(3) + 1 = 7$ disk write operations, for a total cost of 10.