Questions to think about

From short in-class quizzes

Memory, disks, buffers

What are the three key components of disk latency?

A. Reading time, buffering time, replacement delay

B. Rotating time, stop time, transfer time

C. Seek time, rotational delay, transfer time

D. Warming up time, search time, transfer time

What is an advantage of magnetic disks over RAM? Check all that apply.

A. Multiple storage surfaces (platters)

B. Larger capacity

C. Faster random access

D. Lower price

E. Persistence of data

What is a dirty buffer page?

A. The page with pin count > 0

B. The page that contains data that changed since it was read from disk

C. The page that contains incorrect data

D. The page that got corrupted in memory

Which of the following classes of buffer pages are written to disk if we need to free some buffer space?

A. Pages with pin count zero, and dirty flag zero (0 0)
B. Pages with pin count greater than zero, and dirty flag one (>=1 1)
C. Pages with pin count zero, and dirty flag one (0 1)
D. Pages with pin count greater than zero, and dirty flag zero (>=1 0)

External-memory sorting

How big a relation (in bytes) that we can sort in 2 passes

M=5 GB = 5,000,000,000 Bytes = $5*10^9$Bytes

B = 10 KB

1 KB	10^3
1 MB	10^6
1 GB	10^9
1 TB	10^{12}
1 PB	10^{15}

A. 250 TB

B. 2.5 TB

C. 25 TB

D. 2.5 PB

B - block size in bytes.
M - main memory in bytes.

What is the state after the next step of 2PMMS?

- **Phase 2.**
- On disk:
  - Sub-list 1: 20 25 27 29 30
  - Sub-list 2: 18 23 35 45 65
  - Sub-list 3: 21 22 29 34 39
- Main Memory (4 buffers)
  - Input Buffer1: 20 25
  - Input Buffer2: 18 23
  - Input Buffer3: 21 22
  - Output Buffer:
- Sorted list:

A. Output buffer: 20, sorted list: empty

B. Output buffer: 18, sorted list: empty

C. Output buffer: 20, sorted list: 18

What software and hardware methods we can use to improve performance of multi-way sort, given that the available memory is constrained and cannot be increased? Check all that applies

- Double buffering
- Cylindrification
- Replacing hard disks with tapes
- Multiple disks
- Stronger CPU

Disk files

The record header may contain a directory of field offsets. What problems does it solve? Check all that apply.

- Minimizing an overall space occupied by the record
- Efficient access to the beginning of the field data
- Defining the order of fields within each record
- Efficient representation of nulls

Indexes

An index on a search key K can be created even if the data file is not sorted by K. Such an index can be dense or sparse.

B-trees

In order to maintain the pre-defined capacity range, internal nodes of B-tree must be joined or split. The insertion may cause the splitting of internal nodes.

The deletion of a key from a B*-tree may result in the following tree modifications (check all that apply):

- The structure of the tree remains unchanged
- A key from one sibling is transferred to another sibling
- A parent and a child merge into a single node
- Two siblings merge into a single node

Implementing relational operators

Given two relations R and S, B(R) = 20,000 and B(S) = 10,000, how many memory buffers do we need (at least) in order to perform a join with a single pass over R and S

A. 30,000
B. 10,000
C. 20,001
D. 10,001

Zigzag join can be performed if both R and S have Hash-based indexes on join attributes

- True
- False

Given 2 relations R and S of sizes B(R) = 1000, B(S) = 500, what is the cost of optimized sort-merge join given enough memory?

- 4500
- 7500
- 3000
- 2500
- 1500

Given two relations R and S with B(R)=1000 and B(S)=1000, it is possible to perform join in 3 passes if memory buffers M = 500 blocks

- True
- False

Query optimization

Why do we push selections?

A. To decrease the size of the participating relations

B. To make a RA expression more concise

C. To decrease the depth of the RA tree

When projecting out redundant attributes, we need to leave untouched (select 2):

A. attributes that participate in joins
B. attributes that represent numeric fields
C. attributes that are required for the final output
D. attributes that represent the meaning of the RA query

Intersection is not a core operator of the relational algebra, because it can be expressed using other core operators

- True
- False

What of the following are the core operators of the relational algebra (check all that apply)

A. Selection
B. Join
C. Renaming
D. Projection
E. Union
F. Difference
G. Duplicate elimination

What is an estimated size of the selection **S** = $\sigma_{A=c \text{ and } B=d}$ (**R**)?

- T(S) = T(R) / [V(R,A) * V(R,B)]
- T(S) = T(R) / [V(R,A) + V(R,B)]
- T(S) = T(R) / V(R,A) + T(R)/3

What are the histograms useful for? (check all that apply)

- To give a better estimate for a selection
- To know how many blocks we need to access during index lookup
- To see a clear picture of what is going on in our database
- To give a better estimate of an output for a join

Estimate the size of the following extended relational algebra expression: $\mathbf{S} = \delta\ (\pi_{A}\ (\mathbf{R}))$, if **T(R)**=10,000 and **V(R,A)** = 500.

- 20
- 10,500
- 500
- 5,000,000

Map-reduce

When performing join with map-reduce, why do we include the name of a source table as part of the value?

- The reduce phase needs to join pairs of tuples -- one tuple from each relation. An explicit label provides a robust way to differentiate the two kinds of tuples.
- The key-value pair format requires the label.
- MapReduce is a unary operation - it only takes one dataset as input. The table labels help us implement binary operations.
- Adding a label reduces the running time of the computation

The advantages of implementing query in map-reduce over using DBMS implementations are:

- Map-reduce program scales to much larger inputs
- Map-reduce heavily uses indexes for increased query performance
- Map-reduce program is generic and once implemented can be used for different types of queries
- Map-reduce program can be implemented by a single programmer
- Map-reduce applies algebraic query optimization

Transactions

What does *isolation* mean in the context of ACID transactions?

A. The transaction appears to each user as if it was executed in isolation, and no other transactions were running concurrently

B. The transaction must be isolated from other database operations by running in a separate thread

C. The transaction has its own temporary space on disk

D. The transaction isolates all the required database objects and locks them

Why do we need to execute multiple transactions concurrently?

A. Because disk I/Os for multiple requests can be better optimized

B. Because we want to implement serializable schedules

C. Because users cannot wait until one long transaction finishes, before running their own transactions

D. Because concurrent execution of multiple transactions leaves database in a consistent state

If there is a cycle in the precedence graph, then the schedule is serializable

2PL protocol guarantees serializability

SOLUTIONS

What are the three key components of disk latency?

A. Reading time, buffering time, replacement delay

B. Rotating time, stop time, transfer time

[✓ C. Seek time, rotational delay, transfer time]

D. Warming up time, search time, transfer time

What is an advantage of magnetic disks over RAM? Check all that apply.

A. Multiple storage surfaces (platters)

B. Larger capacity

C. Faster random access

D. Lower price

E. Persistence of data

What is a dirty buffer page?

A. The page with pin count > 0

B. The page that contains data that changed since it was read from disk

C. The page that contains incorrect data

D. The page that got corrupted in memory

Which of the following classes of buffer pages are written to disk if we need to free some buffer space?

A. Pages with pin count zero, and dirty flag zero (0 0)

B. Pages with pin count greater than zero, and dirty flag one (>=1 1)

C. [✓ Pages with pin count zero, and dirty flag one (0 1)]

D. Pages with pin count greater than zero, and dirty flag zero (>=1 0)

How big a relation (in bytes) that we can sort in 2 passes

M=5 GB = 5,000,000,000 Bytes = $5*10^9$Bytes

B = 10 KB

1 KB	10^3
1 MB	10^6
1 GB	10^9
1 TB	10^{12}
1 PB	10^{15}

A. 250 TB

B. 2.5 TB

[✓ C. 25 TB]

D. 2.5 PB

B - block size in bytes.
M - main memory in bytes.

What is the state after the next step of 2PMMS?

- **Phase 2.**
- On disk:
  - Sub-list 1: 20 25 27 29 30
  - Sub-list 2: 18 23 35 45 65
  - Sub-list 3: 21 22 29 34 39
- Main Memory (4 buffers)
  - Input Buffer1: 20 25
  - Input Buffer2: 18 23
  - Input Buffer3: 21 22
  - Output Buffer:
- Sorted list:

A. Output buffer: 20, sorted list: empty

[✓ B. Output buffer: 18, sorted list: empty]

C. Output buffer: 20, sorted list: 18

What software and hardware methods we can use to improve performance of multi-way sort, given that the available memory is constrained and cannot be increased? Check all that applies

- [✓ Double buffering]
- [✓ Cylindrification]
- Replacing hard disks with tapes
- [✓ Multiple disks]
- Stronger CPU

The record header may contain a directory of field offsets. What problems does it solve? Check all that apply.

- Minimizing an overall space occupied by the record
- [✓ Efficient access to the beginning of the field data]
- Defining the order of fields within each record
- [✓ Efficient representation of nulls]

An index on a search key K can be created even if the data file is not sorted by K. Such an index can be dense or sparse.

- False – only dense

In order to maintain the pre-defined capacity range, internal nodes of B-tree must be joined or split. The insertion may cause the splitting of internal nodes.

- True

The deletion of a key from a B*-tree may result in the following tree modifications (check all that apply):

- [✓ The structure of the tree remains unchanged]
- [✓ A key from one sibling is transferred to another sibling]
- A parent and a child merge into a single node
- [✓ Two siblings merge into a single node]

Given two relations R and S, B(R) = 20,000 and B(S) = 10,000, how many memory buffers do we need (at least) in order to perform a join with a single pass over R and S

A. 30,000

B. 10,000

C. 20,001

D. [✓ 10,001]

Zigzag join can be performed if both R and S have Hash-based indexes on join attributes

- True
- [✓ False]

Given 2 relations R and S of sizes B(R) = 1000, B(S) = 500, what is the cost of optimized sort-merge join given enough memory?

- [✓ 4500]
- 7500
- 3000
- 2500
- 1500

Given two relations R and S with B(R)=1000 and B(S)=1000, it is possible to perform join in 3 passes if memory buffers M = 500 blocks

- [✓ True]
- False

Why do we push selections?

[✓ A. To decrease the size of the participating relations]

B. To make a RA expression more concise

C. To decrease the depth of the RA tree

When projecting out redundant attributes, we need to leave untouched (select 2):

[✓ A. attributes that participate in joins]

B. attributes that represent numeric fields

[✓ C. attributes that are required for the final output]

D. attributes that represent the meaning of the RA query

Intersection is not a core operator of the relational algebra, because it can be expressed using other core operators

- [✓ True]
- False

What of the following are the core operators of the relational algebra (check all that apply)

[✓ A. Selection]

B. Join

[✓ C. Renaming]

[✓ D. Projection]

[✓ E. Union]

[✓ F. Difference]

G. Duplicate elimination

What is an estimated size of the selection **S** = $\sigma_{A=c \text{ and } B=d}$ (**R**)?

- T(S) = T(R) / [V(R,A) * V(R,B)]
- T(S) = T(R) / [V(R,A) + V(R,B)]
- T(S) = T(R) / V(R,A) + T(R)/3

What are the histograms useful for? (check all that apply)

- To give a better estimate for a selection
- To know how many blocks we need to access during index lookup
- To see a clear picture of what is going on in our database
- To give a better estimate of an output for a join

Estimate the size of the following extended relational algebra expression: $\mathbf{S} = \delta\,(\pi_A\,(\mathbf{R}))$,
if $\mathbf{T(R)}$=10,000 and $\mathbf{V(R,A)}$ = 500.

- 20
- 10,500
- [✓ 500]
- 5,000,000

When performing join with map-reduce, why do we include the name of a source table as part of the value?

- The reduce phase needs to join pairs of tuples -- one tuple from each relation. An explicit label provides a robust way to differentiate the two kinds of tuples.
- The key-value pair format requires the label.
- MapReduce is a unary operation - it only takes one dataset as input. The table labels help us implement binary operations.
- Adding a label reduces the running time of the computation

The advantages of implementing query in map-reduce over using DBMS implementations are:

- Map-reduce program scales to much larger inputs
- Map-reduce heavily uses indexes for increased query performance
- Map-reduce program is generic and once implemented can be used for different types of queries
- Map-reduce program can be implemented by a single programmer
- Map-reduce applies algebraic query optimization

What does *isolation* mean in the context of ACID transactions?

[✓ A. The transaction appears to each user as if it was executed in isolation, and no other transactions were running concurrently]

B. The transaction must be isolated from other database operations by running in a separate thread

C. The transaction has its own temporary space on disk

D. The transaction isolates all the required database objects and locks them

Why do we need to execute multiple transactions concurrently?

[✓ A. Because disk I/Os for multiple requests can be better optimized]

B. Because we want to implement serializable schedules

[✓ C. Because users cannot wait until one long transaction finishes, before running their own transactions]

D. Because concurrent execution of multiple transactions leaves database in a consistent state

If there is a cycle in the precedence graph, then the schedule is serializable

- False

2PL protocol guarantees serializability

- True