**44-560 Advanced Topics in Database Systems**

**Distributed Databases WS 02 KEY**

From *Modern Database Management, 9th edition*, by Hoffer, Prescott, and McFadden, published by Pearson Prentice-Hall and online at <http://wps.prenhall.com/wps/media/objects/5485/5616991/Hoffer_CH14_Online.pdf>. The reference to Date’s work is from *An Introduction to Database Systems, Volume 2*, published by Addison-Wesley, 1983.

*With distributed databases, the response to a query may require the DBMS to assemble data from several different sites…A major decision for the DBMS is how to process the query, which is affected by both the way a user formulates a query and the intelligence of the distributed DBMS to develop a sensible plan for processing. Date (1983) provides an excellent yet simple example of this problem. Consider the following situation adapted from Date. A simplified procurement (relational) database has the following three relations:*

***SUPPLIER(SUPPLIER\_NUMBER, CITY)*** 10,000 records stored in Detroit

***PART(PART\_NUMBER, COLOR)*** 100,000 records stored in Chicago

***SHIPMENT (SUPPLIER\_NUMBER, PART\_NUMBER)*** 1,000,000 records stored in Detroit

*A query is made (in SQL) to list the supplier numbers for Cleveland suppliers of red parts:*

***SELECT SUPPLIER.SUPPLIER\_NUMBER***

***FROM SUPPLIER, SHIPMENT, PART***

***WHERE SUPPLIER.CITY = ‘Cleveland’***

***AND SHIPMENT.PART\_NUMBER = PART.PART\_NUMBER***

***AND SHIPMENT.SUPPLIER\_NUMBER = SUPPLIER.SUPPLIER\_NUMBER***

***AND PART.COLOR = ‘RED’;***

*Each record in each relation is 100 characters long, there are ten red parts, a history of 100,000 shipments from Cleveland, and a negligible query computation time compared with communication time. Also, there is a communication system with a data transmission time of 10,000 characters per second and one second access delay to send a message from one node to another.*

*Date identifies six plausible query-processing strategies for this situation and develops the associated communication times; these strategies are summarized in the [table below]. Depending on the choice of strategy, the time required to satisfy the query ranges from one second to 2.3 days!*

|  |  |
| --- | --- |
| *Method* | *Time* |
| *Move PART relation to Detroit, and process whole query at Detroit computer.* | *16.7 minutes* |
| *Move SUPPLIER relation to Chicago; then move SHIPMENT relation to Chicago; process whole query at Chicago computer.* | *2.8 hours* |
| *Join SUPPLIER and SHIPMENT at the Detroit computer, project these down to only tuples for Cleveland suppliers, and then for each of these check at the Chicago computer to determine if associated PART is red. For each tuple, Chicago will send a 1-charater response (Y or N).* | *2.3 days* |
| *Project PART at the Chicago computer down to just the red items, and for each, send to the Detroit computer. Process query at Detroit.* | *10 seconds* |
| *Join SUPPLIER and SHIPMENT at the Detroit computer, project just SUPPLIER\_NUMBER and PART\_NUMBER for only Cleveland SUPPLIERs, and move this qualified projection to Chicago for matching with red PARTs.* | *16.7 minutes* |
| *Select just red PARTs at the Chicago computer and move the result to Detroit for matching with Cleveland SUPPLIERs.* | *1 second* |

*Note 1: The term “tuple” means the same as “record”.*

*Note 2: The phrase “for each” means each record is processed separately. If the processing required involves network access then this means there will be a one second access delay for each record.*

*Example: The phrase “move {a set of tuples} to Chicago” means the network is accessed and the entire set of tuples is moved, so there is a total of one second access delay time. The phrase “for each tuple, move to Chicago” means that each tuple will be moved separately, so there is a one-second access delay for each tuple in the set.*

*Note 3: Unless specifically, noted, no response is returned from the final location of execution.*

***Your task: Show how each of the results from the table above is obtained.***

* 1. Move PART relation to Detroit, and process whole query at Detroit computer.

**PART** has 100,000 records \* 100 characters per record = 10,000,000 characters

10,000,000 characters / 10,000 characters per second = 1,000 seconds

1,000 seconds transmit time + 1 second access delay time = 1,001 seconds = **16.7 minutes**

* 1. Move SUPPLIER relation to Chicago; then move SHIPMENT relation to Chicago; process whole query at Chicago computer.

Move **SUPPLIER** to Chicago, then move **SHIPMENT** to Chicago.

**Result:** 1,000,000 records in SHIPMENT and 10,000 records in SUPPLIER = 1,010,000 records; 100 characters per record = 101,000,000 characters to transmit

Transmit time = 101,000,000 characters/ 10,000 characters per second

= 10,100 seconds. In addition there is a one second access delay at Detroit. Since SUPPLIER and SHIPMENT are sent separately, there is a 2 second access delay. Total transmission time is 10,102 seconds for this step.

Process query at Chicago.

**Result:** Query time is negligible, so required time is 10,102 seconds = **2.8 hours**.

* 1. Join SUPPLIER and SHIPMENT at the Detroit computer, project these down to only tuples for Cleveland suppliers, and then for each of these check at the Chicago computer to determine if associated PART is red. For each tuple, Chicago will send a 1-charater response (Y or N).

Join **SUPPLIER** and **SHIPMENT** at Detroit.

**Result:** There are 1,000,000 records in **SHIPMENT**, and each **SHIPMENT** record is joined with exactly one **SUPPLIER** record, so there will be 1,000,000 records in the join.

Project the tuples for Cleveland.

**Result:** 100,000 records

For each of the 100,000 records we query Chicago to see if the part is red. We perform a separate query for each record.

**Result:** 1 second access delay for each record = 100,000 seconds.

Chicago returns result, resulting in another 100,000 seconds of access delay time.

With each query, we send the record we are checking; each record has 100 characters and there are 100,000 records, so we are transmitting 100,000 \* 100 = 10,000,000 characters. Transmit speed is 10,000 characters per second.

**Result:** 10,000,000 characters / 10,000 characters per second = 1000 seconds.

The result returned from Chicago for each query will be small. Assume one character per record , so a total of 100,000 characters will be returned. 100,000 characters /10,000 characters per second = 10 seconds. Access delay time was accounted for above.

**Final Result:** 200,000 seconds access time + 1,010 seconds transmit time = 201,010 seconds = **2.3 days**.

* 1. Project PART at the Chicago computer down to just the red items, and for each, send to the Detroit computer. Process query at Detroit.

Project PART at Chicago computer down to just the red items.

**Result:** There are 10 red parts, so this will result in 10 records. Query time is negligible.

For each of the 10 parts, send to Detroit.

**Result:** There are 10 red parts, and we are sending a query for each, so there will be **10** seconds access delay time.

To send the part records: 10 records \* 100 characters per record = 1000 characters.

1000 characters / 10,000 characters per second = **0.1** time to send.

Total time = 10 seconds + 0.1 = **10.1** = **10** **seconds** (rounded to the nearest second)

* 1. Join SUPPLIER and SHIPMENT at the Detroit computer, project just SUPPLIER\_NUMBER and PART\_NUMBER for only Cleveland SUPPLIERs, and move this qualified projection to Chicago for matching with red PARTs.

Join SUPPLIER and SHIPMENT at the Detroit Computer.

**Result:** There are 1,000,000 records in shipment, so there will be 1,000,000 records in the join. Query time is negligible.

Project just SUPPLIER\_NUMBER and PART\_NUMBER for only Cleveland suppliers.

**Result:** 100,000 shipments are from Cleveland, so this will result in 100,000 records. Query time is negligible.

Move the qualified projection to Chicago.

**Result:** 100,000 records \* 100 characters per record = 10,000,000 characters.

10,000,000 characters / 10,000 characters per second = 1,000 seconds plus 1 second access delay time = 1,001 seconds = **16.7 minutes**. Query time at Chicago is negligible.

* 1. Select just red PARTs at the Chicago computer and move the result to Detroit for matching with Cleveland SUPPLIERs.

Select red parts at Chicago.

**Result:** There are 10 red parts, so 10 records will be selected. Query time is negligible.

Move result to Detroit.

**Result:** 10 records \* 100 characters per record = 1,000 characters.

1,000 characters / 10,000 characters per second = 0.1 second plus one second access delay = 1.1 second = **1 second**, rounded to nearest second.