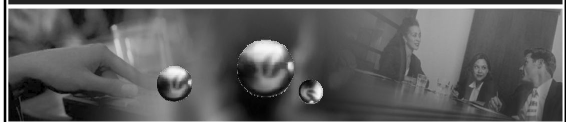
Mariposa:
A wide-area distributed database



Presentation: Shahed

Discussion: Dutch

Outline

1. Motivation
2. Assumptions for DDBMS
3. Economics in Mariposa
4. Mariposa architecture
5. Bidding process
6. Storage and Name resolution
7. Experiment and Conclusion

2

Outline

1. Motivation

3

Motivation

- 1) Build a wide-area Distributed database system
- 2) Apply principles of economics
  - Example: Demand and supply of data, advertisements in yellow pages
- How is it different from a local-area Distributed database system?

4

Outline

1. Motivation
2. Assumptions for DDBMS

5

(wrong) Assumptions in Distributed DBMS

- Static data allocation
  - No handling of changing access patterns
  - Manual transfer of data from site to site
- Single administrator
  - Site selection done by optimizer
  - But what if site belongs to another? Chance of being refused?
- Uniformity
  - Different hardware, network connections, hard disk space
- Assumptions hold for LAN but not multi-admin WAN

6

Assumptions for WAN based DDBMS

- Scalability
  - More than 1000 sites
- Data mobility
  - Change “home” of object. (Take it closer to query)
- No global synchronization
  - Schema changes should not cause sync
- Total local autonomy
  - Sites control own local resources. What objects to store, what queries to run
- Easily configurable policies
  - Easily change individual rules of sites by local administrators

7

Outline

1. Motivation
2. Assumptions for DDBMS
3. Economics in Mariposa

8

Application of economics to Mariposa

- Clients and servers have accounts with a network bank
- User allocates budget to each query
- Query administered by broker which obtains bids
- Fragments (objects) are the units of storage that are bought and sold
- Servers buy objects, advertise its services, bids on queries, leaves by selling objects
  - Goal: optimize revenue

9

… more economics

- Objects have “current owner” which changes as they are moved
- Object replication based on payment for frequency of updates among copy holders
  - Name servers use the same policy for metadata
- Each site has a bidder and storage manager
  - Which objects to buy/sell, which queries to execute

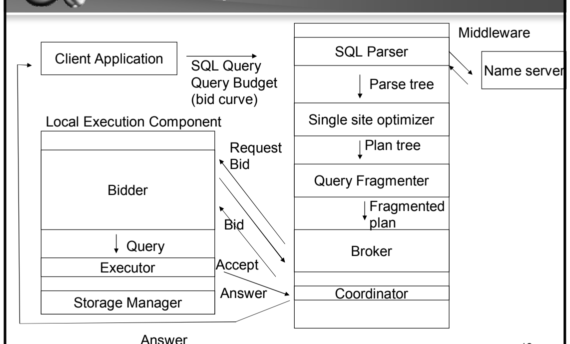
10

Outline

1. Motivation
2. Assumptions for DDBMS
3. Economics in Mariposa
4. Mariposa architecture

11

Mariposa architecture



12

A few more details…

- Rush
  - A rule language
  - Ever mariposa entity has a rush interpreter
  - Storage manager, bidder, broker coded in rush

13

Outline

1. Motivation
2. Assumptions for DDBMS
3. Economics in Mariposa
4. Mariposa architecture
5. Bidding process

14

Bidding process

- Each query has a budget B(t)
- Each query is fragmented into sub-queries
- Strides
  - Multiple fragmented subqueries that can be executed in parallel
- Broker solves sub-queries using
  - Expensive bid protocol
  - Purchase order protocol

15

Expensive Bid protocol

- 2 phases
- 1. Request for bids
  - Send portion of query plan being bid
  - Bidder sends back a triple (C,D,E)
    - C= Cost
    - D= Delay (time to process query)
    - E= Expiration date of offer
- 2. notify the winning bidder
- This process used only for complex queries as it is expensive (too many messages). Solution?
- Use Purchase order protocol for simple queries

16

Purchase order protocol

- Send subquery to bidder with highest likelihood of winning anyway
  - Keep track of query-history
- Site processes request and sends a “bill”
- Con: Probable budget deficit

17

More on bidding

- Finding bidders
  - Servers post “advertisements” with nameservers.
  - Name servers store “ad tables”
    - Advertisements in form of “yellow pages”
    - Example: date of advertisement, sale price, coupons
  - Brokers examine ad tables to locate bidders
  - Brokers remember sites that bid successfully

18

Setting the bid price

- Remember, bidder sends reply in form (Cost, Delay, Expiration) to broker
- Cost
  - CPU, I/O (naive), Network resource
  - Optimization: Billing rate per fragment, Adjust cost based on current load, bid on hot list items even if server does not have data
- Delay
  - Time to process under zero load or current load + safety factor
- Expiration
  - Set arbitrarily

Outline

Storage Management

- Manages fragments to maximize profits in local execution component
- Buying and selling fragments
  - Maintains history of each fragments revenue
  - Contact current owner for fragment revenue before buying (remember : maximize profit)
  - Performs bidding process to sell fragments that it does not want by sending revenue history to bidders
- Splitting or coalescing fragments
  - Break fragments that have high revenues, to lower copies (to redirect traffic to oneself)
  - Coalesce copies if it takes more processing than is required

Naming and Name service

- Unlike traditional centralized name servers, Mariposa has a DECENTRALIZED name registration system
- Names are unordered sets of attributes
- Each object has four structures for naming
  - Internal names
  - Full names
  - Common names
  - Name contexts

Name resolution and discovery

- Every client-server has local name cache to resolve object names
- Broker queries name-server if match not found
- There exists multiple name-servers
- Broker choose name-server based on quality-of-service (staleness of metadata) required

Outline

Experimental Evaluation

- Environment
  - 3 relations in 3 sites, 11MB data
- Test Purchase order Vs Expensive Bid in LAN vs WAN environment
  - Result
  - Broker: 4.52 (s) for PO Vs 14.08 (s) for EB
- Test Expensive Bid to show how data is moved to a closer site for repeated-query
  - Result: all 3 tables move to site that starts the query

25

Epilogue

- Where is Mariposa now?
  - Mariposa -> Cohera -> PeopleSoft -> Oracle
- How many DDBMS commercially available?

26