Mariposa: a wide area distributed database system

M. Stonebraker, P.M. Aoki, W. Litwin, A. Pfeffer,
A. Sah, J. Sidell, C. Staelin, A. Yu

Assumptions - Requirements

Traditional distributed database systems

- Static data allocation
- Single administrative structure
- Uniformity

Mariposa: a DBMS for non-uniform, multi administrator WAN environments

- Scalability to a large number of cooperating sites
- Data mobility
- No global synchronization
- Total local autonomy
- Easy configurable policies

A microeconomic paradigm

- All clients and serves have an account with a network bank
- A user allocates a budget in the currency of this bank for each query
- Goal: solve the query within the allotted budget
- Contract with various sites to perform portions of the query
- Each query is administrated by a broker, which obtains bids for pieces of the query from various sites

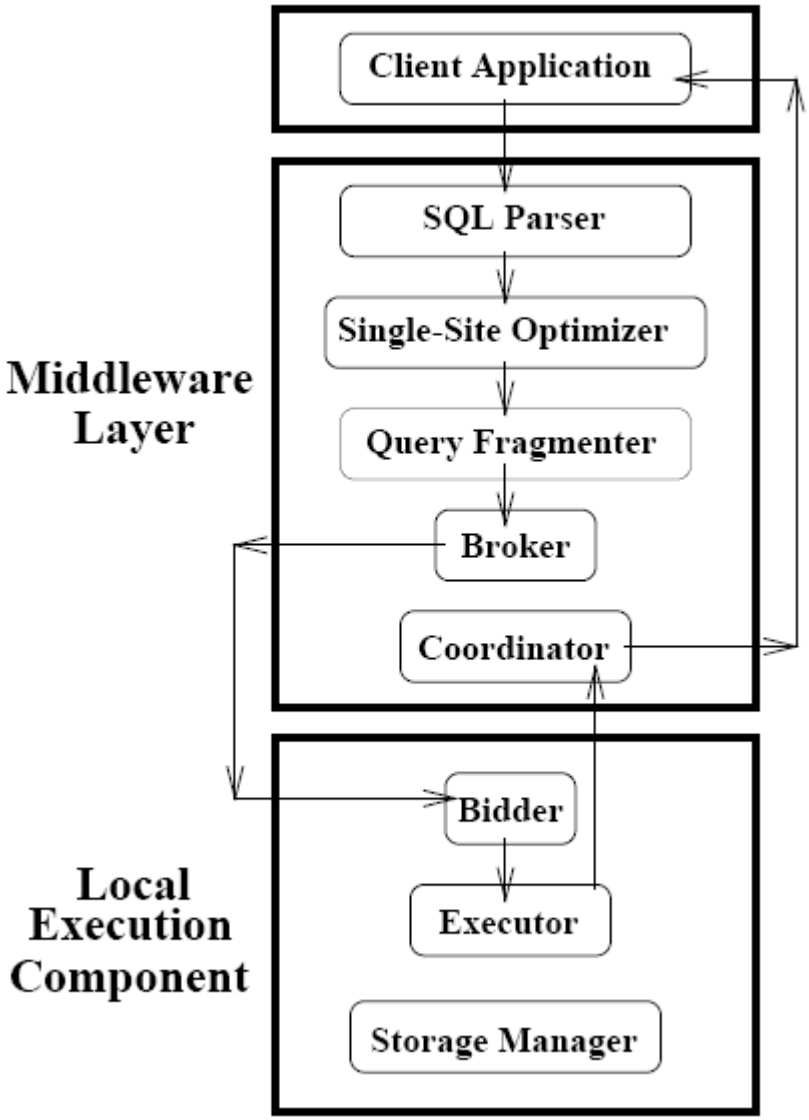
Architecture

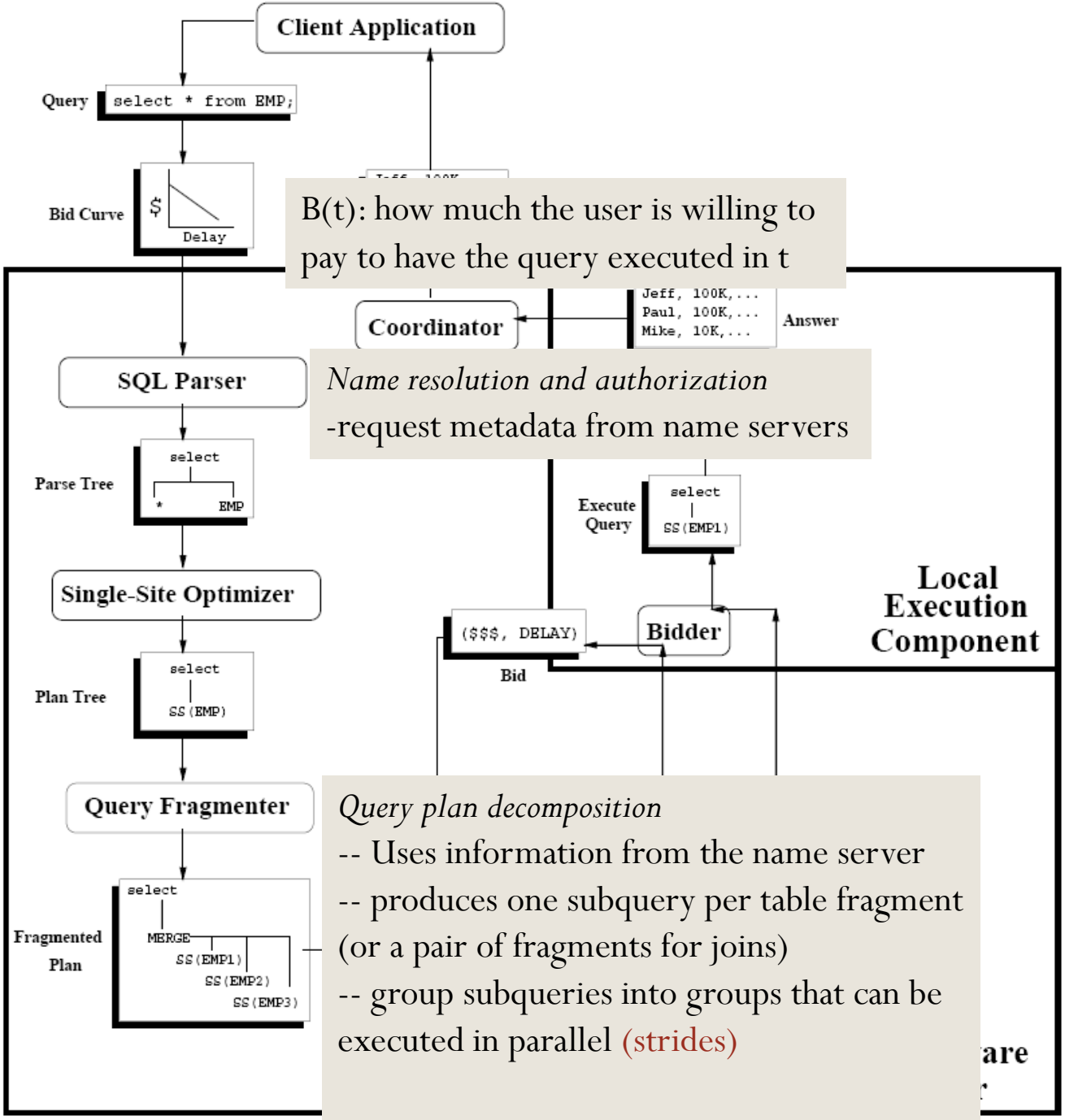
Transparent fragmentation of tables across sites

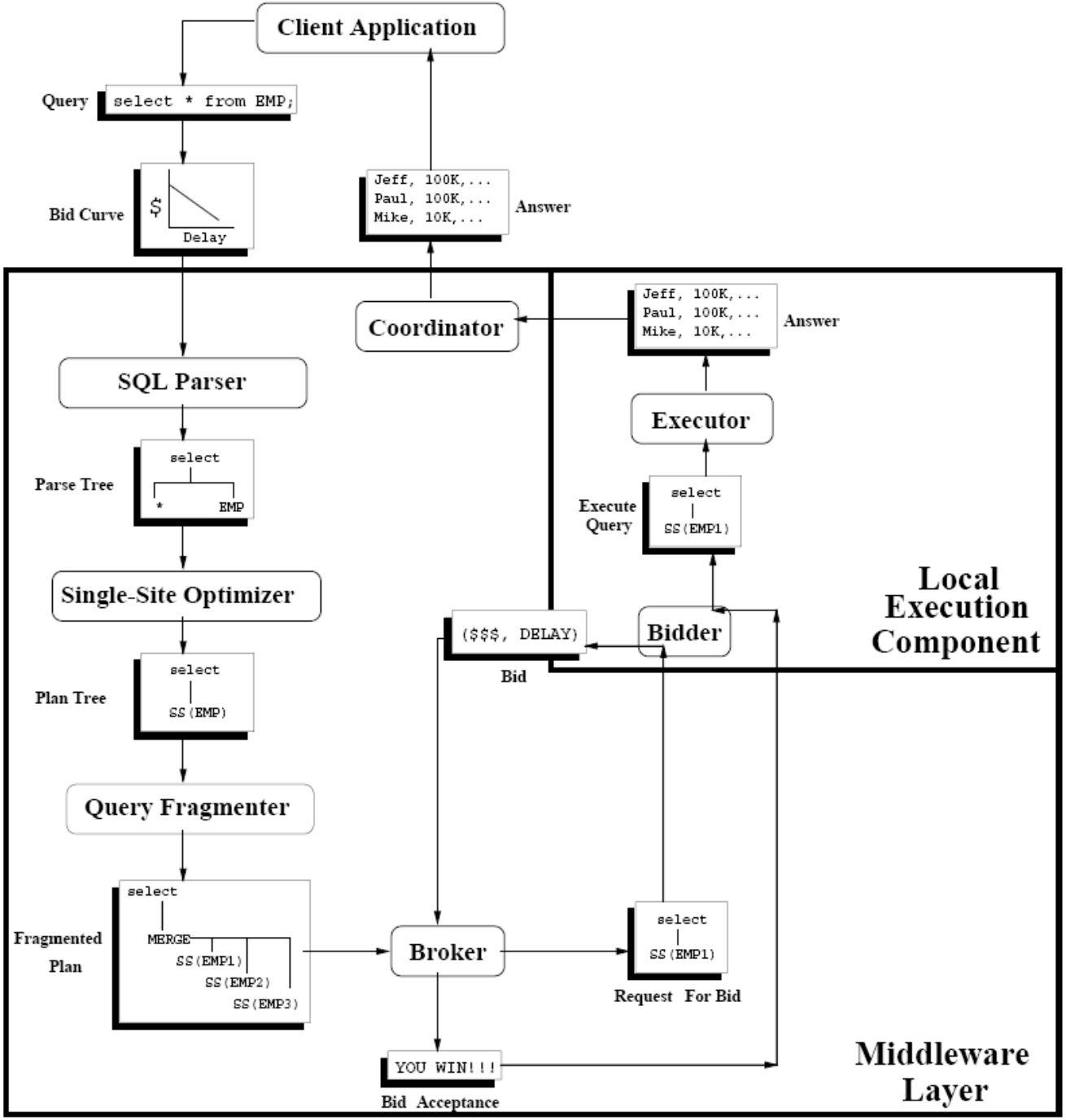
- Range / hash-based distribution

Fragments are the units of storage

Queries are processed on fragments




Client Application
Query
select * from EMP;
Bid Curve
$
Delay
B(t): how much the user is willing to pay to have the query executed in t
Coordinator
Jeff, 100K, ...
Paul, 100K, ...
Mike, 10K, ...
Answer
SQL Parser
Name resolution and authorization
-request metadata from name servers
Parse Tree
select
*
EMP
Execute Query
select
SS(EMP1)
Single-Site Optimizer
Local Execution Component
($$$, DELAY)
Bidder
Bid
Plan Tree
select
SS(EMP)
Query Fragmenter
Query plan decomposition
-- Uses information from the name server
-- produces one subquery per table fragment (or a pair of fragments for joins)
-- group subqueries into groups that can be executed in parallel (strides)
Fragmented Plan
select
MERGE
SS(EMP1)
SS(EMP2)
SS(EMP3)


Client Application
Query
select * from EMP;
Bid Curve
$
Delay
Jeff, 100K, ...
Paul, 100K, ...
Mike, 10K, ...
Answer
Coordinator
Jeff, 100K, ...
Paul, 100K, ...
Mike, 10K, ...
Answer
SQL Parser
Executor
Parse Tree
select
*
EMP
Execute Query
select
SS(EMP1)
Single-Site Optimizer
Local Execution Component
($$$, DELAY)
Bid
Bidder
Plan Tree
select
SS(EMP)
Query Fragmenter
Fragmented Plan
select
MERGE
SS(EMP1)
SS(EMP2)
SS(EMP3)
Broker
select
SS(EMP1)
Request For Bid
YOU WIN!!!
Bid Acceptance
Middleware Layer

Local Execution Component

Bidder:

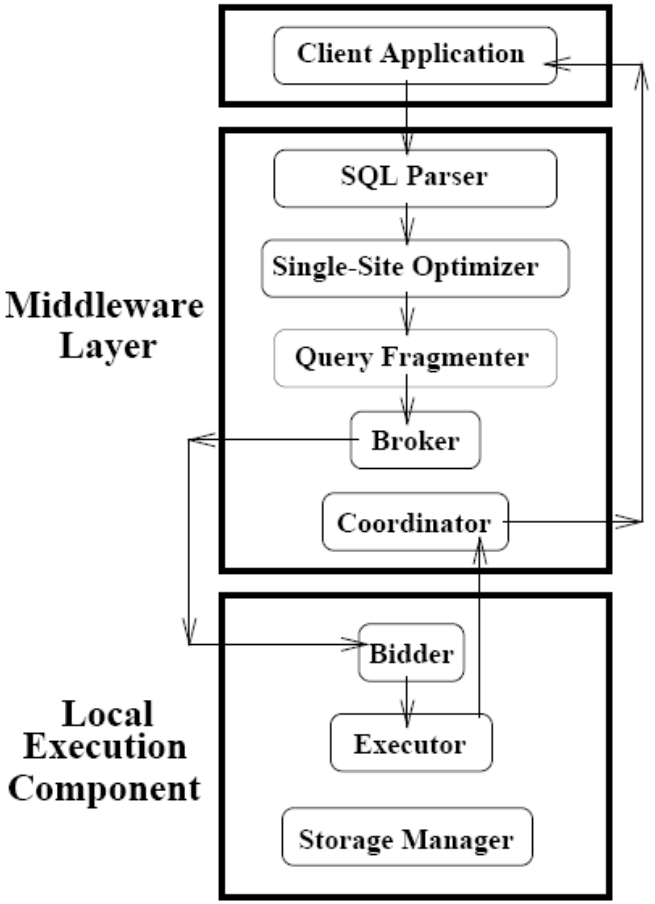
- responds to bid requests
- formulates the bid price and speed for processing a subquery based on local resources: CPU time, disk I/o bandwidth, storage, etc.

Local execution engines:

- An idle engine is assigned the subquery

Storage Manager:

- watches the revenue stream generated by stored fragments
- buys/sells fragments



Rush

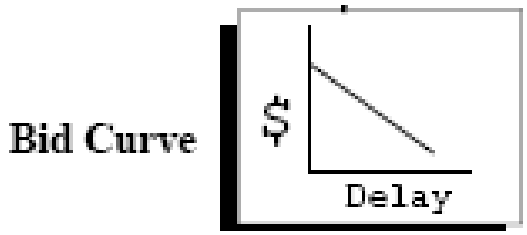
Rush: embedded scripting language and rule system

Storage managers, bidders and brokers are coded in Rush

on <*condition*> *do* <*action*>

Actions (messages)	Events (received messages)
Request bid	Receive bid request
Bid	Receive bid
Award contract	Contract won
Notify loser	Contract lost
Send query	Receive query
Send data	Receive data

Bidding



Budget: non-increasing function of time

For each subquery Q_i on a fragment of a table F (or a join of two fragments):

- *Expensive bid protocol*
- *Purchase order protocol*

Expensive bid protocol:

Phase 1: send requests to bidder sites – receive bids (C_i ,D_i ,E_i)

Phase 2: notify the winning bidders

Purchase order protocol:

Send each subquery to the site that would most likely win the bidding – receive answer with bill for services

Risk: exceeding the allotted budget

Bid acceptance

Strides: groups of subqueries that are executed in *parallel*

The next stride cannot begin until the previous one has been completed

Consider collections of bids for the subqueries in each stride

Choose a winning bid for each subquery with aggregate cost C and aggregate delay D such that the aggregate cost is less than or equal to B(t)

The estimated delay to process a stride is equal to the highest bid time in the corresponding collection

Choose a winning bid from a set of aggregated bids for each stride

If there is no bid for a subquery:

- Solicit additional bids
- Agree to perform the subquery
- Notify the user

Search for the best bid collection

$$difference = B(D) - C$$

Greedy heuristic:

1. Consider the bid collection with the smallest delay for each processing step and compute C and D
2. For each unused bid compute the cost gradient

Cost gradient: cost decrease for the processing step that would result by replacing the collection in the solution with the considered bid collection divided by the corresponding time increase from the substitution

3. Consider the processing step with the bid with the maximum cost gradient B'
4. Compute corresponding D' and C'
5. If B(D')-C'>B(D)-C, then replace B with B'
6. Recalculate all cost gradients and continue making substitutions until there is no increase in the *difference*

Finding Bidders

Advertising system:

- servers announce their willingness to perform services by posting *advertisements*
- Name serves keep a record of the advertisements in an *Ad Table*
- Brokers examine the table to find willing servers

Additional mechanisms:

- Coupons
- Bulk purchase contracts

query-template	A description of the service being offered. The query template is a query with parameters left unspecified. For example,
	`SELECT param-1`
	`FROM EMP`
	indicates a willingness to perform any SELECT query on the EMP table, while
	`SELECT param-1`
	`FROM EMP`
	`WHERE NAME = param-2`
	indicates that the server wants to perform queries that perform an equality restriction on the NAME column.

server-id	The server offering the service.
start-time	The time at which the service is first offered. This may be a future time, if the server expects to begin performing certain tasks at a specific point in time.
expiration-time	The time at which the advertisement ceases to be valid.
price	The price charged by the server for the service.
delay	The time in which the server expects to complete the task.
limit-quantity	The maximum number of times the server will perform a service at the given cost and delay.
bulk-quantity	The number of orders needed to obtain the advertised price and delay.
to-whom	The set of brokers to whom the advertised services are available.
other-fields	Comments and other information specific to a particular advertisement.

Setting bid prices

Bid: (C, D, E): (cost, delay, expiration time)

Naïve strategy: maintain a billing rate for CPU and I/O resources for each site

- Local administrator sets the constants based on local conditions
- Bidder estimates the amount of each resource needed for the subquery and calculates the bid
- If an object is not present at the site, the site declines to bid
- For a site to bid for a join it should:
  - Either possess one of the two objects
  - Had previously bid on a query whose answer formed on of the two objects

Improvements:

- Billing rate per fragment
- Decline bids below a site-specific threshold
- Adjust bids based on current site load :

$$actual\ bid = comp.\ bid * load\ average$$

- Bid on subqueries without possessing the referenced objects if these objects belong in the *hot list*

Network Bidder

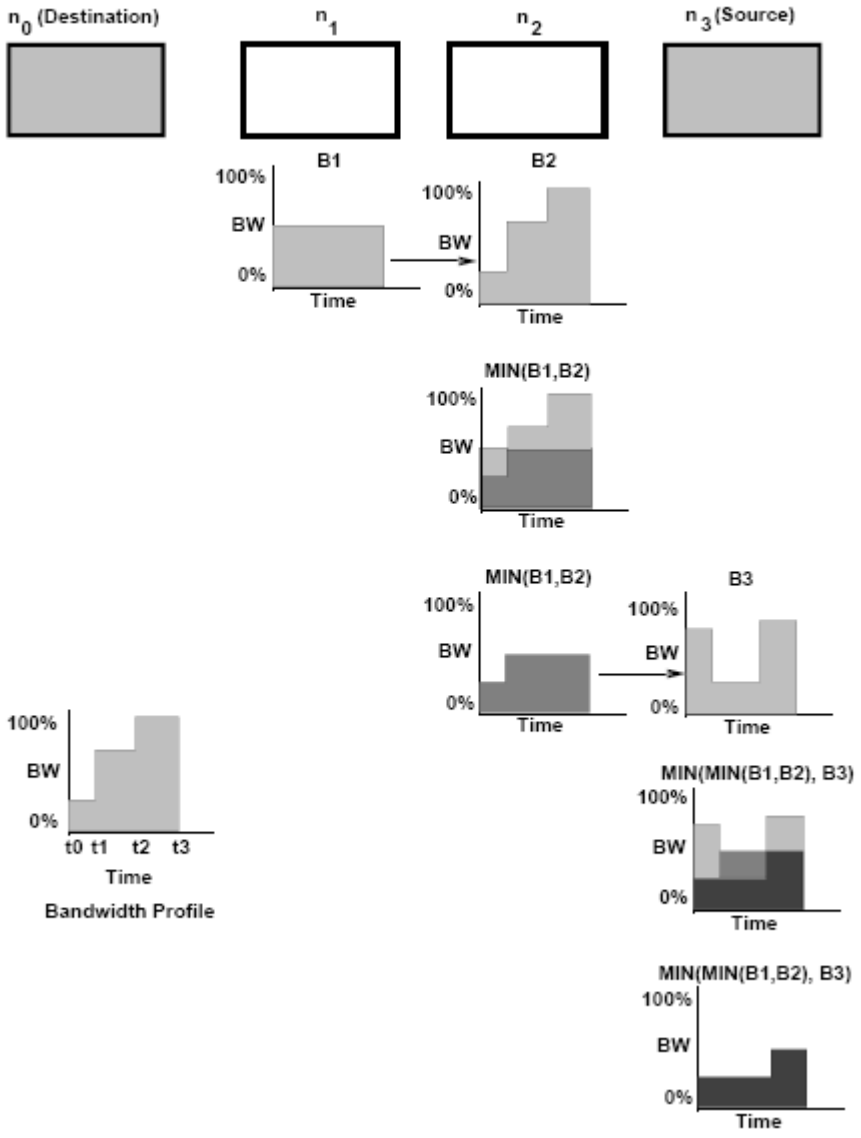
From destination node to source node:

In a forward pass: calculate bandwidth between two adjacent nodes

In a backward pass: calculate the price of the bandwidth

Bandwidth reserved:

1. While calculating prices
2. After the bids have been made



Storage Management

Buying and selling fragments

Each site maintains: *size* and *revenue history* of fragments

To purchase a fragment a site:

- locates its owner
- requests its revenue history
- places a price on the fragment

Splitting and Coalescing fragments:

- Large fragments have high revenue and attract many bidders for copies
- Small fragments have high processing overhead

Conclusions

- Scalability to a large number of cooperating sites
- Data mobility
- No global synchronization
- Total local autonomy
- Easy configurable policies

- No centralized metadata/ servers join and leave freely
- Buying / selling fragments – object owners (no homes)
- Replication system/ naming service…
- Each site is free to bid on any business of interest to maximize its individual profit
- Bidder and Storage manager in Rush