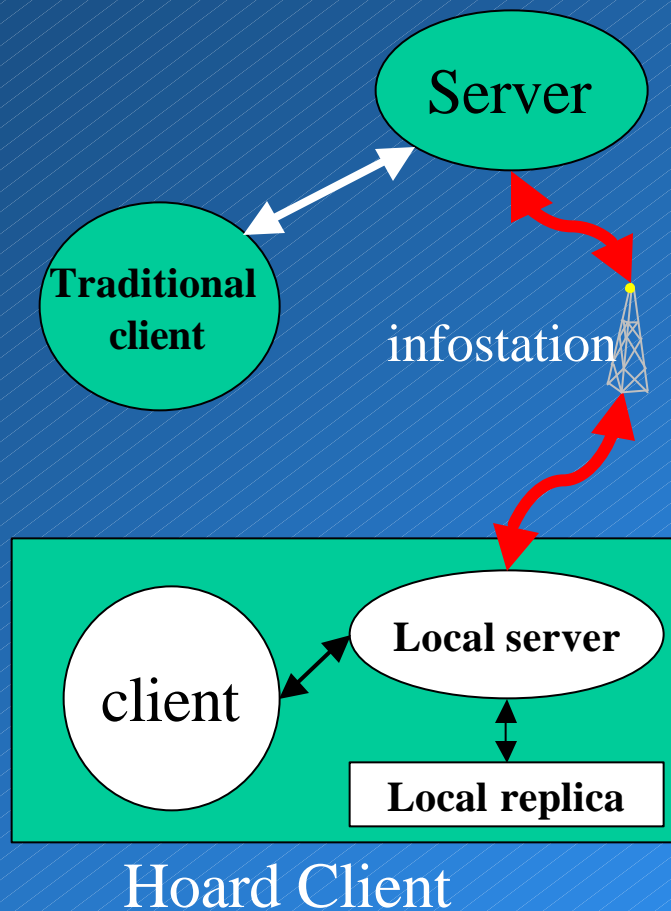


Data Partitioning for Disconnected Client Server Databases

Shirish H. Phatak

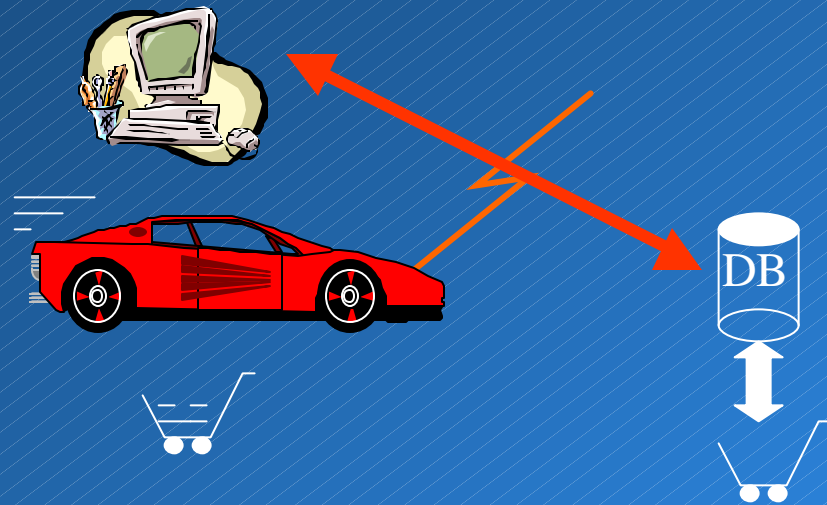
B. R. Badrinath

The disconnected client server database



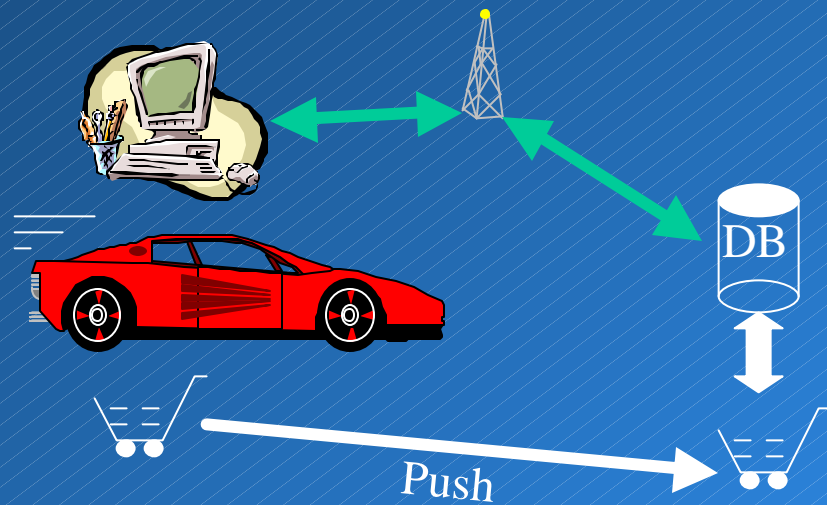
- The disconnected client server model
 - Already adapted for file systems and commercial databases
 - e.g. CODA, SQL Anywhere, Oracle Lite
- Disconnectable clients replicate server data while connected
 - Connected clients operate as traditional clients
- On disconnection a local server takes over serving local clients from the local replica

The shopping cart example



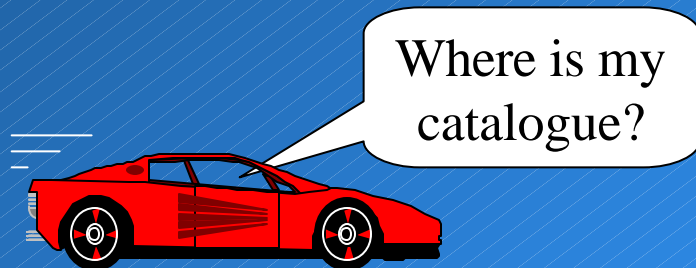
- A *disconnected* shopping cart for a book seller.
- An eager shopper replicates the book seller catalogue (database) on his/her laptop prior to disconnection.
 - *Hoarding* of the catalogue.
 - Local replica can also be from an offline external source, such as a catalogue on CD ROM.
 - Replication done at infostations.

The shopping cart example (contd.)



- The shopper makes purchases locally using the the local replica and the disconnected shopping cart.
- The cart is “pushed” on reconnection (at an infostation) and the purchases are completed.
 - *Reintegration and reconciliation.*

The replication problem



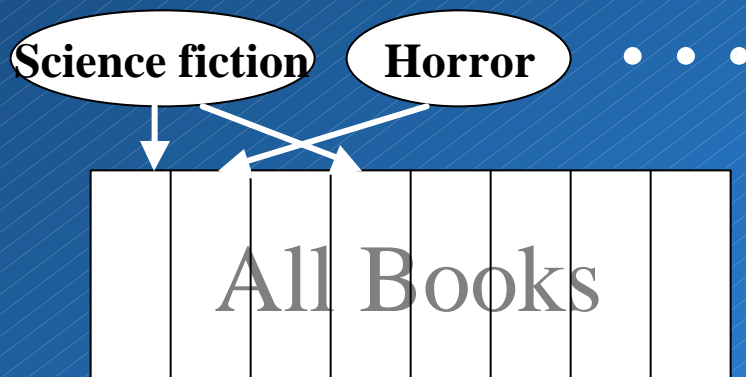
- How to control what to replicate?
 - Can't replicate the entire catalogue!
 - Ask the user?
 - Automated hoarding profiles.
 - Pre-processing of data.
- Speed of replication...what if you have one minute to replicate?
 - You are at an infostation!
- Specifying different consistency models for different subsets of the catalogue.
- We also need to support traditional clients.

Our solution



- **Partition the catalogue into logical horizontal fragments**
 - Attributes of the books along with qualifiers can be used for this
 - Locality of access
 - Customer profiling
 - e.g. science fiction books entries can be one fragment
- **Logical fragments represent the replication granularity**
 - e.g. replicate only the science fiction fragment
 - Use index data structures to identify data in various fragments
 - Traditional clients are not affected

Physical fragments



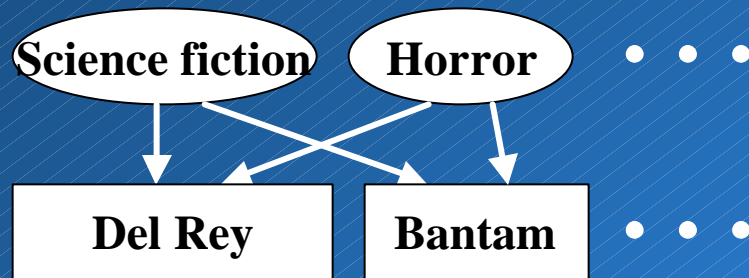
Logical Fragmentation



Physical Fragmentation

- But what about efficiency?
 - We can't afford to be inefficient with infostations.
- Realize some of the logical fragments as physical fragments on disk.
 - Data is contiguous.
- Identify physical fragments as logical fragments belonging corresponding to designated physical hoard attributes.
 - Create a separate sub catalogue for science fiction books.

Combined fragmentation



Combined Fragmentation

- We can have both logical and physical fragments on the same database.
- By maintaining correct balance between physical and logical fragments we can tune the system to various workload combinations.
 - Traditional and hoard clients provide distinct workloads.
 - Logical fragments help traditional clients while physical fragments optimize the performance of hoard clients.

Conclusions

- Organization aspects of mobile database servers have been addressed.
 - Architecture designed for infostations.
- Reconciliation (reintegration) is still a problem.
 - How do we detect conflicting updates , e.g. user buys a book that is out of print?
 - What do we do when conflicting updates are performed?
 - How do we incorporate semantics and application specific conflict resolution?
 - Reconciliation dealt with in ICDE '99 paper.