Thor: A Persistent Object Storage System

Barbara Liskov
Laboratory for Computer Science
Massachusetts Institute of Technology

GOAL: Provide a convenient high-level platform for implementing applications in a heterogeneous distributed network.

(picture of clients in different programming languages using a global object store)

Approach: Objects and Transactions

- system stores persistent objects
- applications access objects by method calls within atomic transactions

Thor interface:

(picture showing application communicating with Thor through simple interface with method calls and commits)

Thor architecture:

(picture showing client/server architecture with a network)

Application runs as:

- a thread at each client
- runs a sequence of transactions
- all communication through Thor

Implementation issues:

(picture showing connection between single client and server)

- Efficient support for safe sharing
- Reducing the cost of running methods
- Efficient concurrency control
- Reducing load at servers
- Efficient management of server disks for fine-grained sharing
- Efficient replication algorithms

An important technique: caching at clients

Accomplishments:

- Safe sharing techniques
- Hybrid, adaptive cache with adaptive prefetching
- OCC, a new very efficient optimistic concurrency control algorithm
- Modified Object Buffer (MOB) architecture at servers
- Garbage collection techniques for very large distributed heaps
- Cooperative caching with fragment reconstruction

Future directions (SafeBase):

- Richer model of computation
- From fault-tolerance to survivability
- Weaker consistency models
- Cooperative caching in wide area nets
- New replication algorithms