Tagging Responses for Disaster Recovery

Avinash Kulkarni
CS 6464
Project Demo
Enterprise storage requires fault tolerance

- One solution is the Primary-Backup approach
- For better fault tolerance, Primary and Backup are geographically separated
- Primary and Backup are synchronized through replication
- Consistency through replication
- But replication affects performance – there's a trade off to be made
- Can we get good performance without sacrificing consistency?
Replication Strategy

- Synchronous replication
  - Maintains data consistency
  - Poor performance for high latency links

- Asynchronous replication
  - Good performance
  - Danger of data inconsistency
Existing approach

Use of asynchronous replication

- SMFS (Hakim W, Lakshmi G et al)
  - Exploit large network bandwidth delay product
  - Network serves as a data store
  - Risk of packet drops – redundant packets are sent for error recovery
  - Achieves good performance, at the cost of extra bandwidth usage
Exploit client caches for recovering from disasters.

Tag responses to writes from clients, directing them to cache data.

Data in client caches is the delta between primary-backup.

We gain the performance benefits of asynchronous.

- Without sacrificing consistency.
Steady State
Disaster Recovery
Possible issues

- Unbounded log growth
  - Checkpointing: Primary requests clients to flush logs after synchronizing with backup
  - Requires clients to be modified!
- Susceptible to client crashes
  - Solution: Send a random combination of data as response to client
- Clients need to discover primary after crash