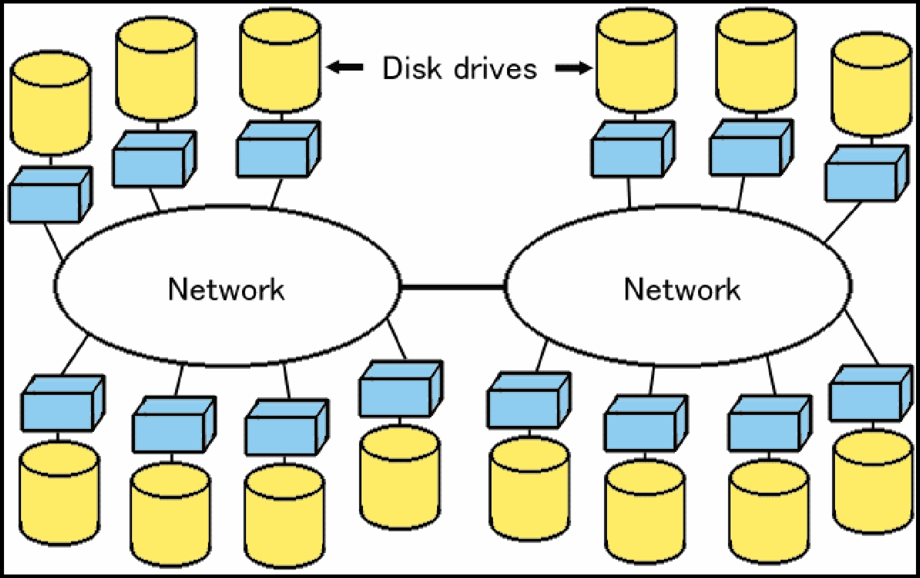
Lock Services in Distributed File Systems

Shaan Mahbubani
Anshuman Gupta
Ravi Vijay
Anup Tapadia

UCSD CSE 221 Operating Systems - Winter 07

Distributed File System Lock Service

- Synchronization
- Consistency
- Access control
- Shared access



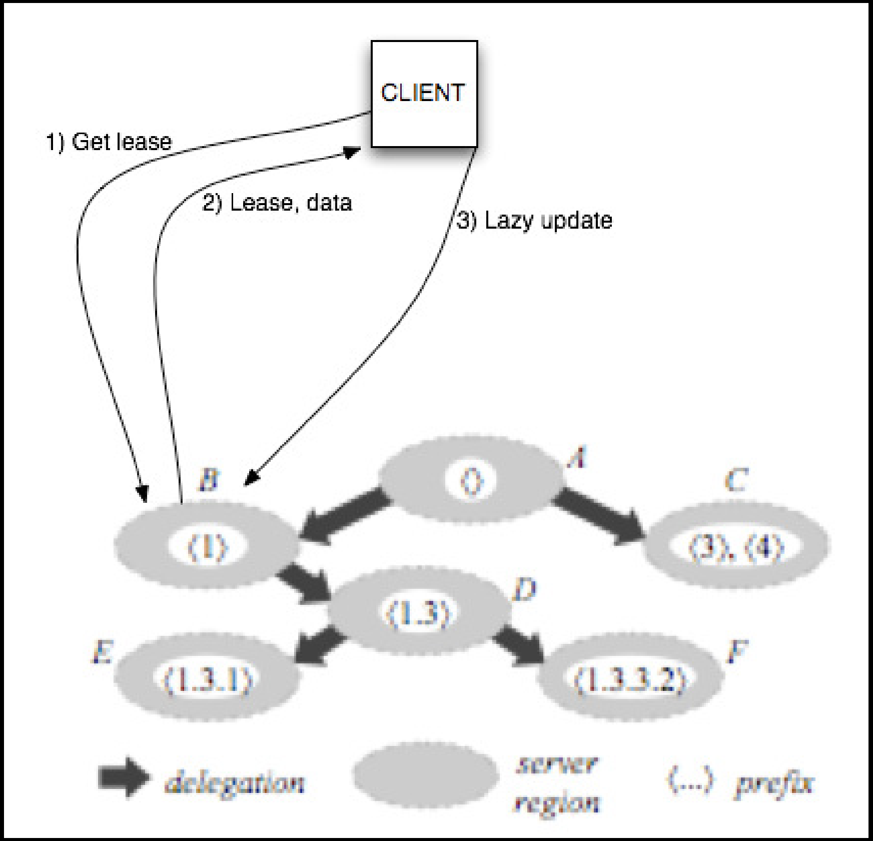
Farsite - Overview

Goals:

- Meta-data partitioning
- Namespace consistency
- Scalable

Design:

- Recursive path leases
- File-field leases
- Disjunctive leases



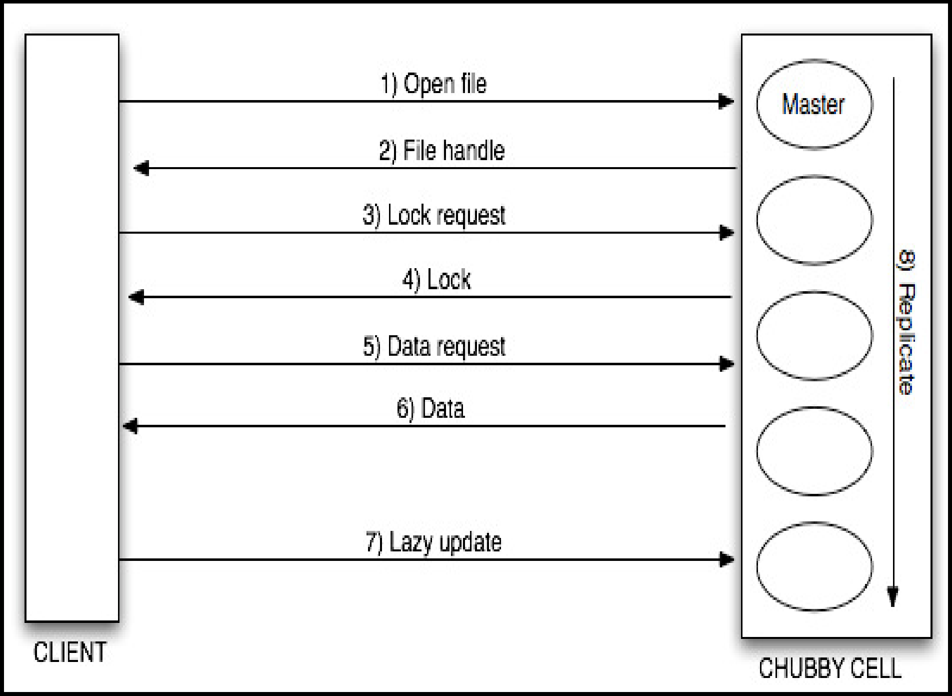
Chubby - Overview

Goals:

- Coarse-grained synchronization
- Primary election

Design:

- High capacity
- Server, client library
- Sessions, events, client caching



Analysis - Farsite

- Fully functional rename
- Identifier based partitioning
- Fine grained synchronization: leases
- Logical atomicity
- Optimizations:
  - Recursive leases
  - Lazy updates
- No fault recovery

Analysis - Chubby

- Low volume storage
- Flat access control lists
- Lock service in lieu of consensus service
- File handles and advisory locks
- Event notification service
- KeepAlive piggybacking
- Client side caching

Comparison

	Farsite	Chubby
Meta-data distribution	Dynamic	Centralized
Partitioning	File ID	None
Rename	Fully functional	Not across dir
Update policy	Operation replay	Update as deltas

More Comparison

	Farsite	Chubby
Locking	Mandatory	Advisory
Load Balancing	Delegation	None
Fault Recovery	None	Reconstruction
Bandwidth utilization	Low	High

Even More Comparison

	Chubby	Farsite	NFS	AFS
Meta-Data partitioning	✗	File ID	Pathname	File ID
Full rename	✗	Yes	✗	✗
Access Control	Flat	Hierarchical	Hierarchical	Hierarchical

Questions

- Farsite
  - Scalability
  - Lease overhead penalty
  - Network robustness
  - Effect of malicious machines
  - Time to recovery
- Chubby
  - API problems
  - Throughput bottlenecks due to keepalives
  - System capacity

Conclusion

- Farsite locking
  - Distributed directory service avoids hotspots
  - Fine grained locking avoids false sharing
  - No fault tolerance
- Chubby locking
  - Lock service is better than consensus service
  - Coarse grain locking more useful
  - More reliable across failovers

Thank You !

Questions ?