

# Ch 9 – Distributed Filesystem

- Reading Assignment – Summary due 10/18 (CODA paper)
- Goals
  - Network Transparency – you don't know where in the system the file is.
  - High Availability – Users should get same filesystem irrespective of location, and maintenance/failure should cause graceful degradation.

- Overall Architecture
  - Client, Fileserver, client cache/local storage, server cache, name server, cache manager.
- Mechanisms
  - Mounting
    - Binding together different file namespaces to form a single hierarchically structured namespace.
  - Caching
    - Hints
  - Bulk transfer
  - Encryption

# Design Issues

- Naming and Name Resolution
  - Single Namespace ? Location Dependent ? Location Transparent ?
  - Context as a component of the namespace ?
  - Name server maps names to stored objects
    - Single centralized server ?
    - Distributed ? – the actual resolution may involve several servers
- Caches
  - Disk vs Memory
    - Disk is larger, simpler memory management, portable systems (volatility?)
    - Memory is faster, client and server caching schemes can be similar, allows diskless clients to participate

- Cache Writing Policy – when should changes made by a client committed at the server ?
  - Write through ? Delayed Write ? Write on Close ?
- Cache Consistency
  - Server Initiated – server informs clients of changes, they can then update or invalidate
  - Client Initiated – client side cache checks with server before returning data to client.
  - Both are problematic and expensive.
  - Caching for concurrent write sharing – don't cache!
  - Caching for sequential write sharing
    - Outdated file on client.
    - Written blocks not yet flushed.