

CMSC 818S LegionFS

Alan Sussman
October 22, 2002

Administrivia

- DataCutter project proposal due tomorrow
 - Remember, keep it short (1-2 pages), and describe both the data and the queries/application
- Need a volunteer for next Tuesday
 - Distributed file systems and OceanStore

LegionFS

- Fully integrated file system infrastructure, with several desirable features:
 - location-independent naming
 - security
 - scalability
 - extensibility
 - adaptability

Objects

- In Legion, everything is an object
 - independent, separate address space, active objects
 - that communicate via RPC
 - support for persistent objects
- File abstraction is a *BasicFileObject*
 - has methods with Unix-like functionality – read, write, seek, etc.
- *ContextObject* to manage name space
- *ProxyMultiObject* for aggregating files/contexts on one host, for performance reasons

Naming

- Context names (user strings) mapped to Legion object identifiers (LOIDs)
 - *ContextObject* does the mapping, so acts as a directory service – but can also contain references to non-file objects
- For object-object communication (RPC), LOID bound to Object Address (OA), representing a communication endpoint
- Helps with fault tolerance (migration/restart) and replication (map one LOID to multiple OAs)

Security

- No trusted code base or kernel
- Each object makes and enforces own security policies
 - use public key in LOID to enable secure communication
- User authentication involves obtaining a short-lived, unforgeable credential that provides unique identification
 - authorization via access control list for each object of operations allowed for each (class of) user(s)

Scalability

- Files and contexts distributed across all available system resources
 - no centralized servers
- Scheduler objects make placement decisions when objects (e.g., files) created
 - based on host load, network connectivity, space availability, other system-wide metadata
 - user chooses the scheduler, so gets to make own policy decisions (e.g., place all files on local host, or on local cluster)
- LegionFS uses multiple levels of caching for fast file and directory access

Extensibility

- Objects differ based on exported interfaces, not implementations
 - so any object providing *BasicFileObject* interface looks like a file to LegionFS
 - allows specialization according to application or domain-specific requirements
- Legion has event-based protocol stack
 - can register handlers to intercept RPCs and exceptions and hand them off to other handlers

Adaptability

- LegionFS maintains system-wide metadata to all adaptation
 - every object contains a set of <key, value> pairs to hold static and dynamic attributes
 - E.g., for a host object, load average, architecture, OS, network interfaces and current bandwidths, disk configurations, etc.
- Metadata repository also stores the attributes – the *Collection*
 - a hierarchically distributed set of objects that can be queried (by schedulers) to find out object info
 - objects periodically push state info to the Collection