

FARSITE and AFS

Arun Balasubramanian
CMSC 818
Department of Computer Science
University of Maryland

FARSITE:

Federated, Available and Reliable storage for an
Incompletely Trusted Environment
Microsoft Research

Introduction

- Intended to provide benefits of a central file server.
i.e.
(shared namespace, location transparent access and reliable storage).
- And local desktop file system
(low cost, privacy from sys-admins and resistance to geographically localized faults).

A Central Server based file system

- Pros
 - Shared namespace among users.
 - Greater file reliability than local file systems.
 - Location transparent access.
- Cons
 - Dedicated administrative staff (reliability and security concerns)
 - Vulnerability to geographically localized faults.

Goal

- Harness the collective resources of loosely coupled, insecure and unreliable machines to provide logically centralized secure and reliable file-storage service.

System Overview

- Enabling Technology Trend:
 - A general increase in unused disk capacity.
 - Decrease in computational cost of cryptographic operations.
- Namespace roots
 - Multiple roots.
 - Set of machines manage a root.
 - These machines form a byzantine fault tolerant group.

Trust and Certification

- Public Key cryptographic certificates
 - Namespace certificates
 - User certificates
 - Machine certificates
- Certification authorities (CA) play a major role.
- CA private keys are kept offline.

System Architecture

- Consists of Clients, Directory groups and file hosts.
- The directory group – set of machines.
- Each member stores a replica of the information about files.
- Clients contact the directory group to read a file.

System Enhancements

- Local caching of file content.
- Directory group issues leases on files.
- Delays pushing updates to directory groups.
- A directory group can delegate a portion of namespace to another directory group.

Reliability and Availability

- Metadata is preserved if no more than $(R-1)/3$ of machines die.
- Directory migration is performed more aggressively than file host migration.
- Directory group membership biased towards high availability machines.

File system features

- Security
 - Access control.
 - Privacy
 - Integrity
- Durability – updates are logged locally, pushed back to directory groups periodically.
- Consistency – content leases.
- Scalability – delayed updates.

Scale and Performance in a Distributed System

The ANDREW FILE SYSTEM

THE PROTOTYPE SYSTEM

- Comprises of Clients and servers. (running the 4.2 Berkeley s/w distribution)
- Comprises of trusted servers – Vice.
- Venus- User level process on workstations.
- Paper- Focuses on scalability.

Overview of the System

- Venus on a Client interacts with Server via TCP.
- User level file locking – dedicated lock server process.
- Data and status information stored separately.
- Access list stored in shadow directories (.admin)
- Stub directories – namespace located on other servers.
- Cached copies considered suspect by venus (checks timestamp with server).

Evaluating Performance of Prototype

- CPU bound operations were fast.
- Other operations like recursive directory listing of subtree of files took longer if subtree was in vice.
- Acceptable limit 20 users per server.
- However, difficult to operate and maintain. (excessive context switching overheads and virtual memory demands).

Performance results

- Significant performance improvement possible.
- Reduce frequency of cache validation checks.
- Reduce number of server processes.
- Require workstations rather than the servers to do pathname traversals.
- Balance server usage by reassigning users.

The Changes

- Cache Management.
- Name resolution.
- Communication and server process structure.
- Low-level storage representation.

Cache Management

- Venus caches contents of directories and symbolic links in addition to files.
- Modifications to files are done locally. Reflected back to vice when file is closed. (LRU used to maintain files)
- Modifications to directories are made directly on the server.
- Callback eliminates checking for timestamps with server.

Name Resolution

- Two level names are introduced. Fids and pathnames.
- Each vice file or directory is represented by Fid.
- Vice only knows Fids.
- Venus maps vice pathnames to fids.
- Requests the fids to the server.

Communication and Server Process Structure

- Eliminate separate Server Process per client.
- Threads were introduced. Non-preemptive LWPs.
- Context switching between LWPs is only a few procedure calls.

Low Level Storage Representation

- Files are accessed by inodes rather than pathnames.
- Inode interfaces are not visible to user level processes.
- System calls were implemented.
- The vnode number in fid represents the inode of the file storing the data.

Effect of Changes for performance

- See results on paper.
- Overall the modified system performed better than the prototype.
- 70% faster than prototype.
- Prototype – 4 times as long at a load of 10 as at a load of 1.
- Modified system- 2 times as long at a load of 20 as at a load of 1.

THANK YOU