SUNDR:
Secure Untrusted Data Repository

Jinyuan Li, Maxwell Krohn, David Mazieres, Dennis Shasha

NYU Department of Computer Science

Presented by: Hussein Nagree

December 7th, 2015

Motivation

- Data needs to protected on storage servers.
- Traditionally done by restricting external access to users and software.
- Can be thought of as building a fence around the servers; however,
  - Fences are often not high enough
  - People inside the fence may not be completely trustworthy
  - Fences impede useful traffic to the servers

Overview

- Reduce the need to trust storage servers.
- File system contents are cryptographically protected, and can be verified by any other user.
- Fork consistency is guaranteed.
  - Strongest consistency guarantee possible without any trusted parties.
  - Can be extended to fetch-modify consistency by maintaining a trusted consistency server, or by communication between users.
- Properties are maintained even in the face of a complete server takeover.

Fork Consistency

- Correct view of action history:

fetch(f_2) user A sig	mod(f_3) user B sig	fetch(f_3) user A sig	mod(f_2) user A sig	fetch(f_2) user B sig

- Forked view of action history:

user A:

fetch(f_2) user A sig	mod(f_3) user B sig	fetch(f_3) user A sig	mod(f_2) user A sig

user B:

fetch(f_2) user A sig	mod(f_3) user B sig	fetch(f_3) user A sig	fetch(f_2) user B sig

Basic Architecture

- One SUNDR server can support multiple file systems.
- Each file system has one super-user, who can set user and group privileges.
- Thus, a file system administrator is different from the server administrator.
- There are also multiple users that communicate using a SUNDR client.
- Users have the ability to switch between clients.

Basic Architecture

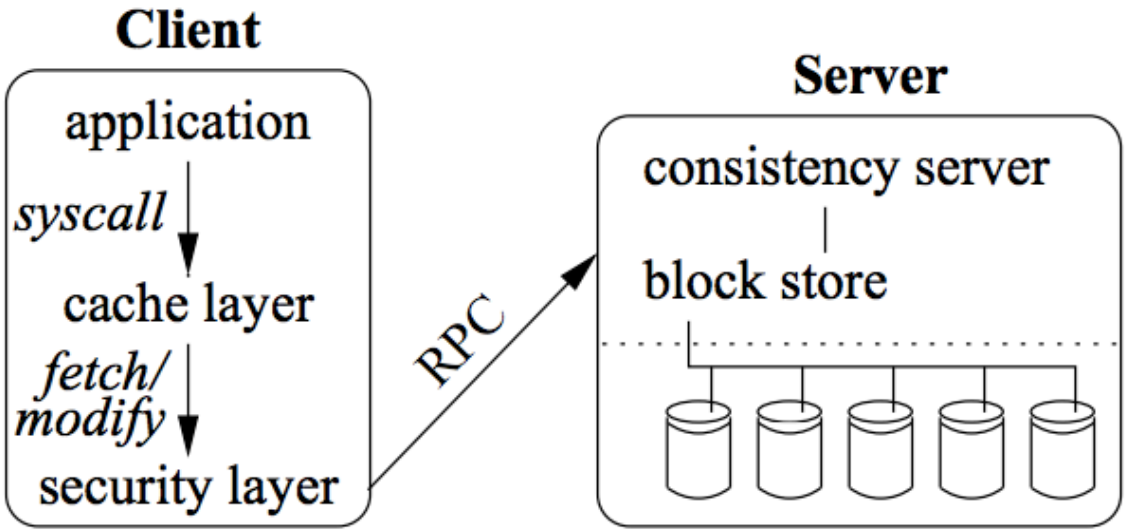

Figure 1: Basic SUNDR architecture.

Straw Man System Protocol

- Stores a complete ordered list of all operations in history.
- Each operation has a signature that covers all preceding events.
- Clients acquire a global lock, download the file system history, and validate each user's previous signature (including their own).
- Clients then construct a local copy of the file system, add their own operation, and sign and send that to the server.

Serialized and Concurrent SUNDR

- The previous version of SUNDR is impractical.
- I-handles are used to avoid having to recreate the entire file system.
- Clients download a Version Structure List (VSL) and make changes to the version structures that hold the relevant i-handles.
- Update certificates allow parallelization without having to wait for VSL updates; these are reflected in the Pending Version List (PVL).

File System Implementation

- The server consists of a consistency server and a block store.
- Built upon an xfs device driver.
- Uses E-sign instead of RSA for performance improvement.
- Consistency server stores changes made to the VSL and PVL to block storage (or NVRAM if available) before responding to client RPCs.
- A block store daemon bstor handles all disk storage.

Bstor Disk Storage

- Interacts directly with clients as well as the consistency server.
- Writes all data to disk, to allow for crash recovery.
- Heavily optimized to support writes to disk.
- Uses a temporary log to store incoming writes
  - Sector aligns data blocks to improve synchronous write latency.
  - Improves throughput even under heavy load using batch writes.

Performance Evaluation

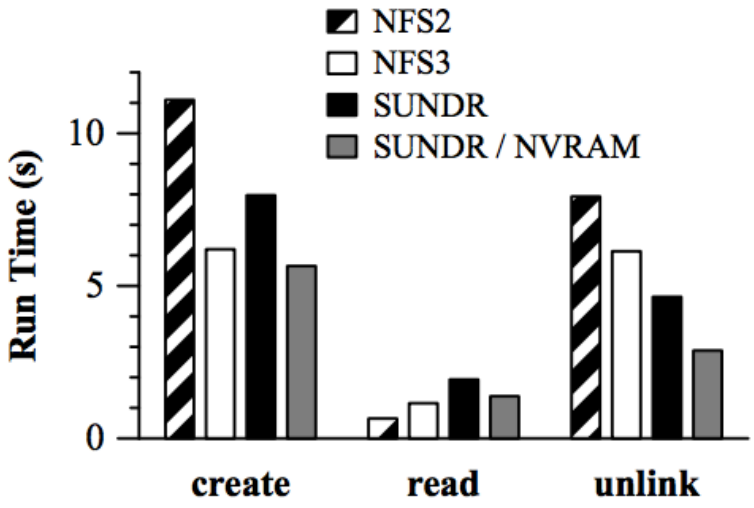

Figure 9: Single client LFS Small File Benchmark. 1000 operations on files with 1 KB of random content.

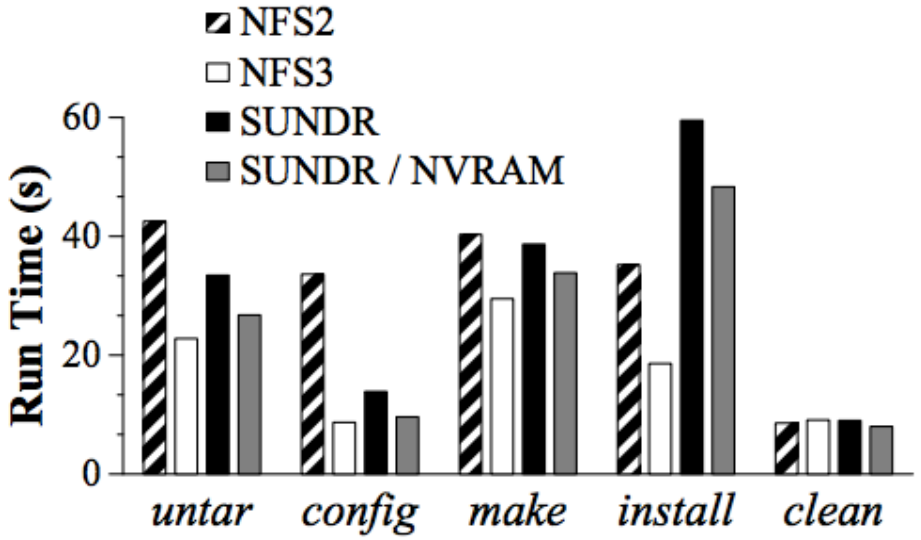

Figure 11: Installation procedure for `emacs-20.7`

Strengths/Limitations

Strengths

- Provides integrity during system recovery, even from untrusted clients.
- Enables hassle-free outsourcing of storage management.
- Performance is at par with similar NFS systems.

Weaknesses

- Can only detect attacks, but does not resolve them.
- Does not offer read protection or confidentiality.
- Leaves room for optimization and compression.
- Cannot protect against file changes made by a malicious user.

Discussion

Thoughts or questions?