WatDFS
A Project for Understanding Distributed Systems

Michael Abebe
Brad Glasbergen
Khuzaima Daudjee

tiny.cc/watdfs
SIGCSE 2019
Build Your Own Distributed File System
Project Goals

Cover wide range of course material

Interaction with common systems & applications

Provide high-quality and timely feedback
Distributed File Systems

Client

RPC

Server

App

File
Distributed File Systems

Lacks access transparency
Distributed File Systems

App

File System

RPC

File

Client

Server
Distributed File Systems

Provides access transparency
Distributed File Systems

Client

VFS

FS Client

Userspace

Kernel

Server

FS Server

File

RPC

App
Distributed File Systems

Kernel implementation
Distributed File Systems

Client

Userspace

Kernel

App

WatDFS

VFS

FUSE

Server

RPC

WatDFS

File
Provides access transparency

Userspace implementation
WatDFS Project

Implement WatDFS client and server:

Support: file creation, open, close, read, write, truncate, and metadata operations

Using two distributed file systems models
Remote-Access Model

**Forward** operations to server

Client  

File stays at server  

Server
Remote-Access Model

File stays at server

Forward operations to server

Increases Latency

File stays at server
Remote-Access Model

Learning Goals

Introduce RPCs and file I/O

Familiarize tools (libfuse, gdb, strace)
Upload-Download Model

Download file from server

Client ➔ Server

File
Upload-Download Model

Download file from server

Client
Perform operations at client

File

Server
File
Upload-Download Model

Download file from server

Client
Perform operations at client

Upload file to server

Server
Upload-Download Model

**Download** file from server

Client
Perform operations at client

File

Upload file to server

File

Server
Upload-Download Model

Download file from server

File

→

File

Reduces Latency

Perform operations at client
Upload-Download Freshness

**Download** file from server

Client \[\rightarrow\] Server

- Client
- Server
- File
Upload-Download Freshness

Download file from server

Client
Perform operations at client

Server
Upload-Download Freshness

Download file from server

Client
Perform operations at client

Server
Upload-Download Freshness

Download file from server

File

Client
Perform operations at client

File

Server

Clients see stale state
How to ensure freshness?

Periodically upload and download
Upload-Download Freshness

**Download** file from server

Read Client ➔ Server
Upload-Download Freshness

Download file from server

File

Read Client

File

Server
Upload-Download Freshness

Download file from server

File

Read Client
Perform operations at client

File

Server
Upload-Download Freshness

**Download file from server**

File

Read Client
Perform operations at client

File

Server

University of Waterloo
Upload-Download Freshness

Download file from server

Read Client
Perform operations at client

Server

Download file from server
Upload-Download Freshness

Download file from server

Read Client

Perform operations at client

Download file from server

File

File

Server
Upload-Download Freshness

Periodically upload and download using timestamp-based cache consistency
Upload-Download Freshness

Clients see fresher state

Reduces Latency

Details at tiny.cc/watdfs
Upload-Download Model

Learning Goals

Manage distributed state with cache consistency

Use locks for atomicity and mutual exclusion
Experiences with WatDFS

Provide students with:

- Detailed specification and Q&A forum
- Public and release tests (Marmoset)
- Starter code: ~300 lines of code
Experiences with WatDFS

Students implement

~760 lines of code for remote-access model

~1425 lines of code for upload-download model

Design document about upload-download model
Experiences with WatDFS

95% passed all remote-access model tests

80% passed majority of upload-download model tests

Most common errors and questions about timestamp-based cache consistency
WatDFS Project Summary

Implement two distributed file systems models

Covers wide range of course material

FUSE allows usage of existing applications

Tests provide high-quality & timely feedback

Details at tiny.cc/watdfs