UNIVERSITY OF CALIFORNIA
UC RIVERSIDE

CS 202: Advanced Operating Systems

Andrew File System

AFS: Andrew File System

- Main Motivation: Scalability!!!

- Basic idea: whole-file caching
  - Fetch the whole file for the first time
  - Update on close

AFS version 1

- When open a file for the first time, cache it
- Next time, TestAuth to determine if the file has changed

- Performance is poor. Why?
  - Path-traversal costs are too high
  - Too many TestAuth messages

AFS version 2

- Solution
  - File identifier
    - Similar to file handle in NFS
  - A callback mechanism to reduce client/server interactions
    - An analogy to polling vs. interrupts

Client (C_1)	Server
fd = open("/home/remzi/notes.txt", ...);	
Send Fetch (home FID, "remzi")	
	Receive Fetch request look for remzi in home dir setup callback(C_1) on remzi return remzi's content/FID
Receive Fetch reply write remzi to local disk cache record callback status of remzi Send Fetch (remzi FID, "notes.txt")	
	Receive Fetch request look for notes.txt in remzi dir setup callback(C_1) on notes.txt return notes.txt's content/FID
Receive Fetch reply write notes.txt to local disk cache record callback status of notes.txt local `open()` of cached notes.txt return file descriptor to application	
read(fd, buffer, MAX);	
perform local `read()` on cached copy	
close(fd);	
do local `close()` on cached copy if file has changed, flush to server	
fd = open("/home/remzi/notes.txt", ...);	

Foreach dir (home, remzi)
if (callback(dir) == VALID)
use local copy for lookup(dir)
else
Fetch (as above)
if (callback(notes.txt) == VALID)
open local cached copy
return file descriptor to it
else
Fetch (as above) then open and return fd	

Client$_1$			Client$_2$		Server	*Comments*
P_1	P_2	Cache	P_3	Cache	Disk	
open(F)		-		-	-	File created
write(A)		A		-	-	
close()		A		-	A	
	open(F)	A		-	A	
	read() → A	A		-	A	
	close()	A		-	A	
open(F)		A		-	A	
write(B)		B		-	A	
	open(F)	B		-	A	Local processes
	read() → B	B		-	A	see writes immediately
	close()	B		-	A	
		B	open(F)	A	A	Remote processes
		B	read() → A	A	A	do not see writes...
		B	close()	A	A	
close()		B		~~A~~	B	... until close()
		B	open(F)	B	B	has taken place
		B	read() → B	B	B	
		B	close()	B	B	
		B	open(F)	B	B	
open(F)		B		B	B	
write(D)		D		B	B	
		D	write(C)	C	B	
		D	close()	C	C	
close()		D		~~C~~	D	
		D	open(F)	D	D	Unfortunately for P_3
		D	read() → D	D	D	the last writer wins
		D	close()	D	D	

AFS Crash Recovery

- If a client crashes, it treats all cache contents as suspect. Send TestAuth to the server.

- If the server crashes, it asks all clients to reconstruct the callback states

Discussion Again

- Throughput
- Latency
- Scalability
- Crash Recovery
- Fault Tolerance