Common Internet File System (CIFS)

- Protocol description
- Samba
- Discussion on course project
- Advanced file caching and prefetching

CIFS

- Features:
  – File access, mandatory file and record locking
  – Safe caching, read-ahead and write-behind
    - Very similar to Sprite
  – File change notification
    - Register with file server for event notification
  – Protocol version negotiation
  – Extended file attributes

Features (Cont.)

– Distributed replicated virtual volume
  - Similar to AFS
  - Virtual volumes --- unified name space
  - Volume can be moved
  - Volume can be transparently replicated; referrals redirect clients to servers

– Batched requests
  - Multiple requests can be put in a single message

– Server name resolution independence

Operation Overview

- Given a full file name, determine the server name and server transport address, and the relative name in the server
  – Example: u:\foo.txt, or file://xxx.com/filepath, …
  – Name resolution: DNS or NetBios
- Connect to the server
- Exchange CIFS messages
  – Connection is kept around
  – Both server and client can torn down the connection

Sample Message Flow

- SMB_COM_NEGOTIATE
  – include list of SMB dialects
- SMB_COM_SESSION_SETUP_ANDX
  – include user's name and credentials; server set uid field in SMB header
- SMB_COM_TREE_CONNECT_ANDX
  – include the file system ID (disk share); server set TID field

Sample Message Flow

- SMB_COM_OPEN_ANDX
  – transmit name of the file; server sets FID
- SMB_COM_READ
  – supply TID, FID, file offset and # of bytes to read; server responds with data
- SMB_COM_CLOSE
  – Close the file (give TID, FID)
- SMB_COM_TREE_DISCONNECT
  – Disconnect from the disk share

Connection Management

- Message transport: assume reliable transport mechanism; failure detection for client and server, and deliver the failures
- Server can terminate connections if:
  - client establishes a new connection
  - client generating malformed requests
  - error on transport
  - connection idle

Caching Reads and Writes

- Oplocks
  - Level II oplock
    - shared read locks; can cache reads, and can read ahead
  - Exclusive oplock
    - only one that has the file open; can cache read and write, but when close, have to reflect change to server
  - Batch oplock
    - No need to contact server when read/write, open/close.

Safe Caching

- If a client holds no oplock
  - Read can be performed using cached or read-ahead data if the byte range has been locked by the client
  - All other operations must be sent to server
- Oplocks can be requested when opening the file
- Oplocks can be broken

Oplock Transformations

- Exclusive to Level II oplock (lock downgrade)
- Exclusive to no oplock
- Batch to no oplock or Level II oplock

Referrals

- A flag in the SMB header indicates that the client wants the server to resolve the SMB's path
- Server return: STATUS_DFS_PATH_NOT_COVERED
- Clients then sends: TRANS2_DFS_GET_REFERRAL

Message Format

- Multi-byte values are transmitted least significant byte first (little Endian)
- All fields are aligned on byte boundary; no padding
- bit field laid out as little Endian too

SMB Header

- Protocol (4 bytes)
- Command (1 byte)
- Status (4 bytes)
- Flags (1 byte): CASELESS, SERVER_REP
- Flags2 (2 bytes)
- padding of 6 bytes

SMB Headers

- TID (2 byte): authenticated client ID
- PID (2 bytes): “process” ID at the client
- UID (bytes): User id
- MID (2 bytes): request/response identifier;
- WordCount (1 byte)
- ParemeterWords
- Bytecount (2 bytes)
- Buffer

Batching of Requests

- Multiple requests can be sent in one message
- No need to repeat headers
- But, server must respond with one messages to all the requests
- All requests and responses must fit in one packet
- Error processing

SMB Requests

- Dialect negotiation
- Session_Setup, Logoff
- Tree connect, Tree disconnect
- Query_FS_Info, Ping
- Create, read/write, locking, flush, delete, rename, move, copy
- Directory create/delete

SMB over TCP

- one byte of zero followed by length, then SMB message