The Sun Network Filesystem: Design, Implementation and Experience

RUSSEL SANDBERG

Presented by – Aniruddh Adkar
CSE 710 Parallel and Distributed File Systems (Spring 2016)
SUNY, University at Buffalo

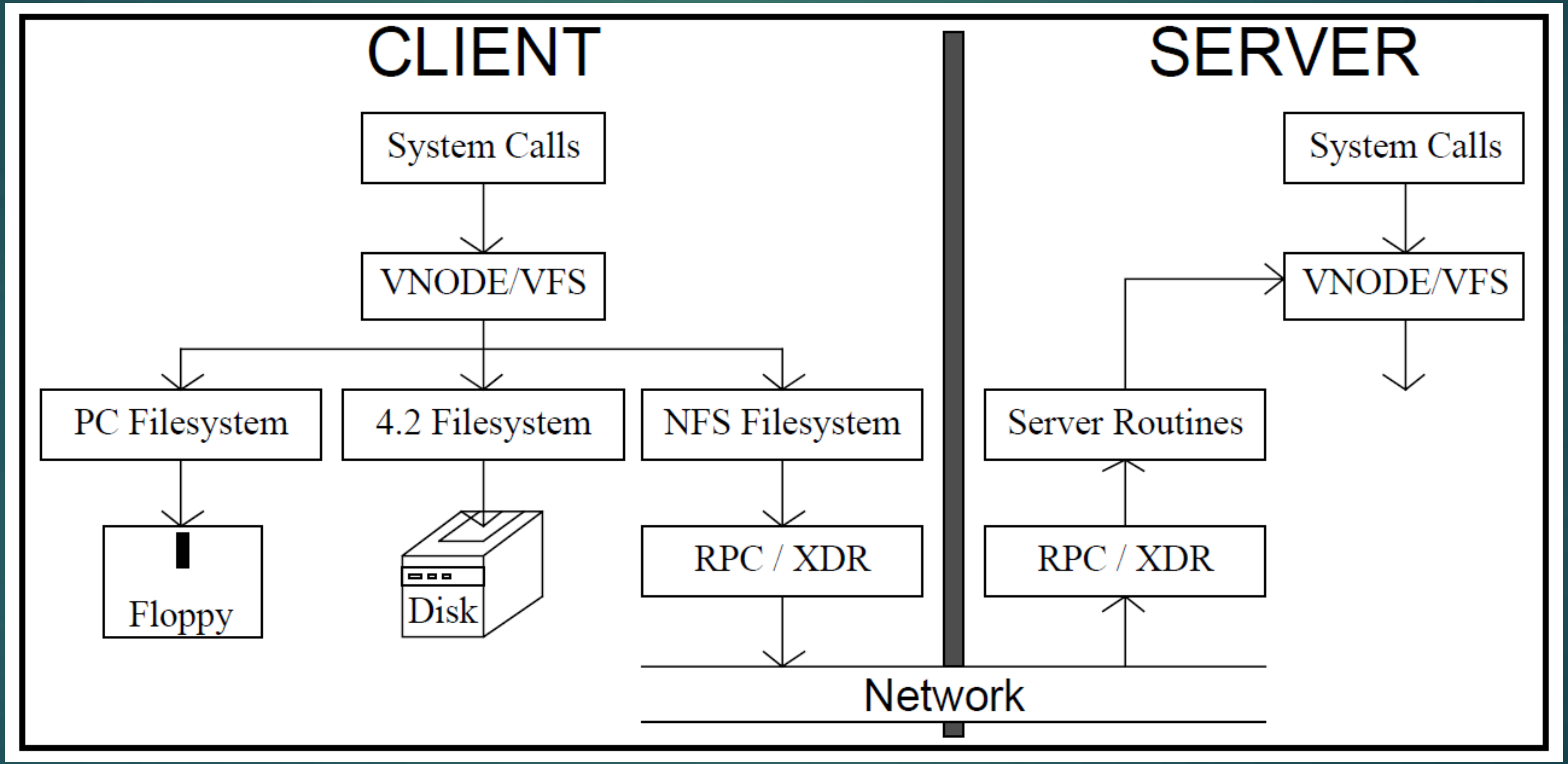
NFS - Why, What, How ?

- Why do we need File system over network ?
- Introduction to NFS
- What are expectations from NFS ?
  - Access Transparency
  - Location Transparency
  - Consistency
  - Fault tolerance
  - Heterogeneity

How ? Design and Architecture

- NFS Implementation
  - Design goals
  - NFS protocol
    - over SUN RPC (UDP and IP)
    - XDR specification
  - Server
  - Client
  - File system Interface

Design and Architecture contd.



Server

- Stateless server
- Synchronous operation, flushing buffer caches before returning
- File handle = FS id + i-node no + i-node generation no

Client

- Abstraction to FS
- Treats NFS as local directory
  - Virtual File System support in Kernel
  - Distinguish between local vs network file handle
  - VFS Interface - vnode wrapper for i-node

Challenges and changes to kernel

- lookup()
- Rewritten FS routines to support vnodes
  - namei, direnter, getdirentries
- Synchronous write requests
- Modifications in MOUNT - Soft, Hard
- Modifications in /etc/fstab and /etc/mtab
- nfsd system call - user context to kernel NFS server

Authentication and Security

- uid,gid permissions model
  - Flat uid, gid across network
- Yellow pages
- root is no more omnipotent in NFS !
- Impersonations issues
- Solution - Kerberized NFS

Optimizations

- No state ? Can we optimize little ?
  - Server caching
  - Client caching

- Better in many aspects when AT&T RFS vs NFS
  - Networking – special purpose protocol vs UDP/IP
  - System call over network vs RPC
  - Ties to UNIX vs Heterogeneous, Machine and OS independent
  - Crash Recovery – Stateful vs stateless matters !

Issues, Limitations and Criticism

- No concurrent access and File locking
  - May get intermixed data
- No complete coverage of UNIX file semantics
- Time Skew
- Write is slow (Excuse me its only 5% of total !)

Its 2016 ! Wake up from hibernation

- NFS v3
  - 64-bit file sizes and offsets added (Files > 2GB)
  - Support for Async writes on server – Performance gain
  - Readdirplus – Get file handles while reading dir
  - Support for TCP – Larger read and write sizes, use over WAN
- NFS v4
  - Stateful ! It sounds weird right ?
  - Recovery – client helps server, gets NO_GRACE in return !
  - Sharing for OPEN

Conclusion

- Early and flexible Distributed File System
- Can work with large network, mixed protocols, machine types and OS
- First widely used IP based network file system
- New features, optimizations and Extensions

References

- The Sun Network Filesystem: Design, Implementation and Experience - *Russel Sandberg Sun Microsystems Inc.*
- http://docs.oracle.com/cd/E19253-01/816-4555/6maoquifj/index.html
- https://en.wikipedia.org/wiki/Network_File_System#NFSv4
- http://www.ibm.com/developerworks/aix/library/au-aix7networkoptimize2/
- https://www.ietf.org/rfc/rfc3530.txt
- Distributed Systems: Concepts and Design By George Coulouris, Jean Dollimore, and Tim Kindberg
- Unix Concepts and Applications By Sumitabha Das
- http://www.ibm.com/developerworks/library/l-virtual-filesystem-switch/