Storage Systems

NPTEL Course

Jan 2012

(Lecture 09)

K. Gopinath

Indian Institute of Science

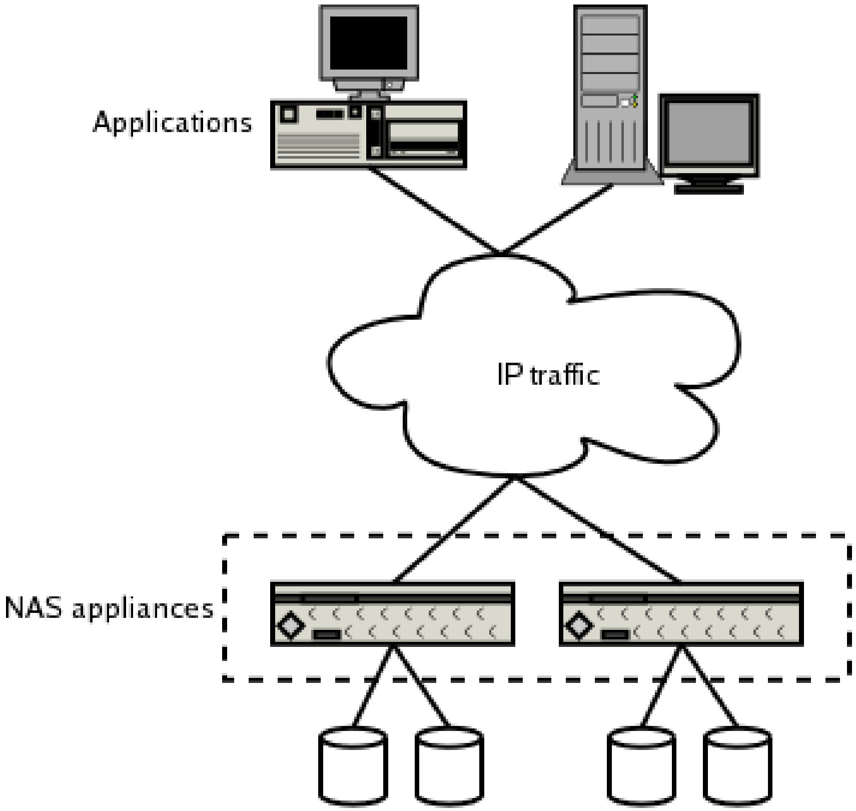
SAN-NAS comparison

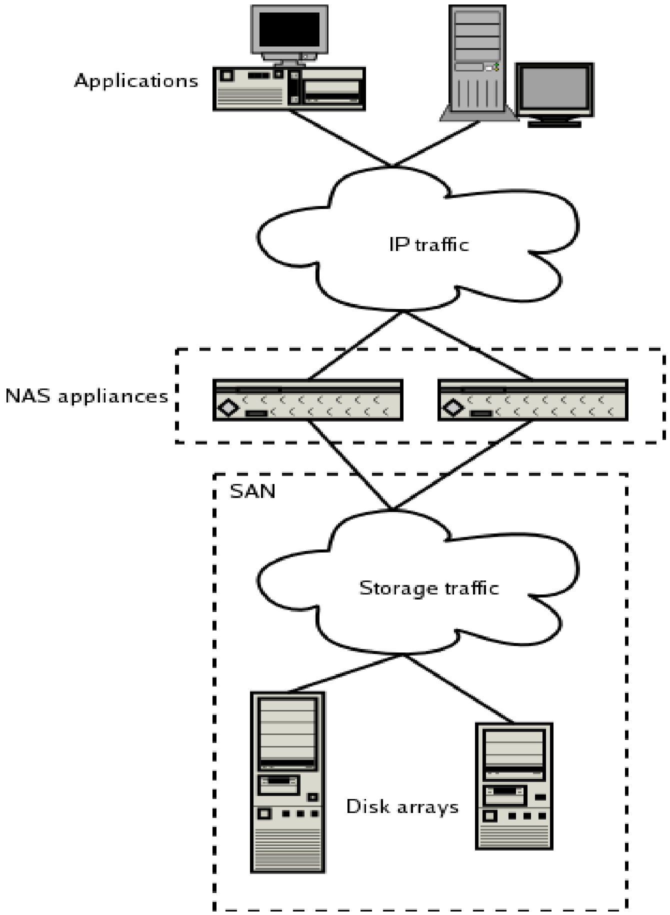
	SAN	NAS
Abstraction	Raw block device	File (byte-stream)
Access model	SCSI command set	File operations <offset, range>
Consumer	FS, DBMS	Application, DBMS
Naming & discovery	SCSI ITL nexus	Pre-configured names / DNS / WINS
Security	Transport layer	Transport layer / Independent mechanisms

NAS overview

- Application requirements
  - Consolidation, sharing, databases
  - Performance, resilience, scalability, manageability
- File-level access
  - Unix/NFS, Windows/CIFS
- Client-side I/O redirectors
  - VFS/vnode framework, IFS
- Server-side appliance model
  - Special-purpose systems, e.g. NetApp
- Why not cluster FS?
  - POSIX semantics across clusters

NAS implementation





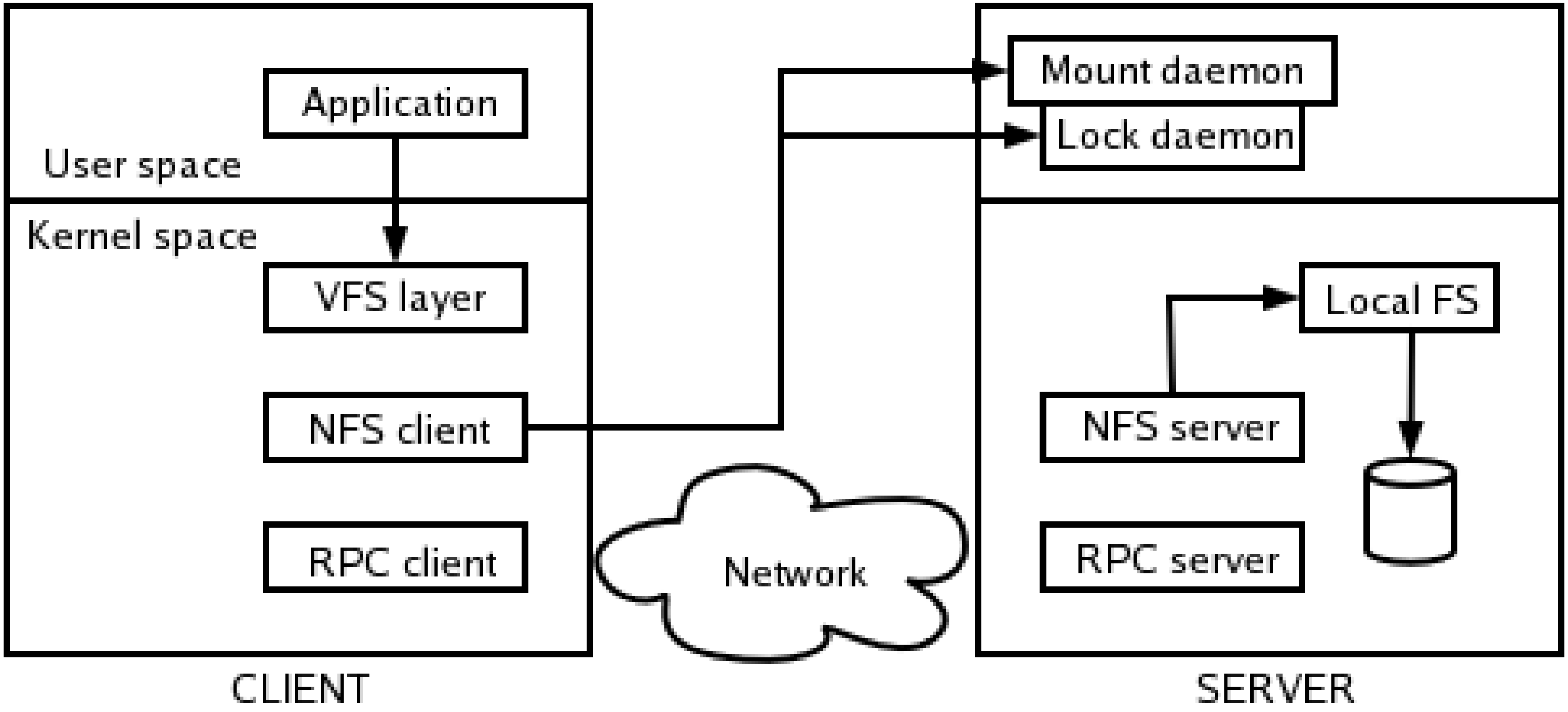
NFS protocol

- Stateless protocol
  - “Smart client, dumb server”
  - Failure handling & crash recovery
- File handles
  - Mounting file-systems
  - Stale file handles
- Error handling
- Transport-independence
  - RPC/XDR, TCP/UDP
- Typical operation example

NFS protocol requests

- Data operations: READ, WRITE
- Directory operations: LOOKUP
  - READDIR, MKDIR, RMDIR
- File management
  - CREATE, REMOVE, RENAME
  - LINK, SYMLINK, READLINK
- File information: GETATTR
  - SETATTR
  - STATFS
- Mount operations
  - MNT, UMNT, EXPORT

NFS architecture



RPC/XDR overview

- RPC services – RFC 1057
  - Request-response protocol
  - Reliable transmission
    - At least once/ At most once semantics
  - Message formats, marshalling, transmission
  - Authentication schemes: none, UNIX-based, key-based, Kerberos-based
  - RPC compilers
- Portmapper daemon
  - maps RPC prog # to TCP/IP ports where servers listen
- XDR services – RFC 1014
  - Byte ordering (big-endian)
  - Date types & formats

NFSv2 implementation

- Components
  - Server & client daemons
  - Mount daemon
  - Lock manager
- Client-side caches
  - Attribute & data caches
- Client-side asynchronous I/O
  - Read-ahead/write-behind
- Server-side re-transmission (xid) cache
  - Idempotent vs. non-idempotent operations
  - Extreme: cache replies too
- Security: Authentication & access control

NFSv2 problems

- Maintaining UNIX semantics
  - Open file access permissions
    - Posix checks on 1st access; NFSv2 on every access
  - Atomic I/O operations
  - Deletion of open files: what if server deletes file?
- Cache consistency guarantees
  - NFS 2 checks if mod time of client cached data diff from server mod time. Works if server only making changes
- Security
  - Access control: User credentials
  - Securing data traffic
- Performance: UDP storms, Synch writes: ad-hoc opts.
- Needs Portmapper, mountd, lockd, statd
- Functionality: 4GB file size limit