Networking Applications

Martin Děcký

DEPARTMENT OF DISTRIBUTED AND DEPENDABLE SYSTEMS
http://d3s.mff.cuni.cz/

CHARLES UNIVERSITY IN PRAGUE
FACULTY OF MATHEMATICS AND PHYSICS

Networking stack

- Queuing architecture
  - Design follows the network layers
  - Communication between layers using packet queues
    - Avoiding data copying
      - Buffers with preallocated space for headers and footers
      - Scatter & gather hardware
    - Excessive data dispatching
      - Computational and interrupt overhead
      - Processing of multiple packets at once
      - Scatter & gather, hardware off-loading

Packet scheduling & policing

- Packet policing
  - Discarding excessive input packets
    - Usually on system overload, does not save network bandwidth
- Packet scheduling
  - Time/discard output packets
  - Stochastic Fair Queuing
    - Many flows competing for a common bandwith
      - Each flow modelled as a queue, sending in round-robin fashion
      - Hashing into a small set of output queues, hash function reset

Packet scheduling & policing (2)

- Token Bucket
  - Regulation of the bandwidth of a single flow
    - Tokens added to the bucket by a constant speed
      - Full bucket → no more tokens added
      - Token inflow speed: Bandwidth limit
    - Tokens removed from the bucket as data is sending out
      - Empty bucket → no data is sent out
      - Bucket capacity: Fluctuation limit
  - Hierarchical Token Bucket
    - For multiple flows
- Class-Based Queuing
  - For complex situations and multiple flows
    - Bandwidth borrowing, etc.

Packet scheduling & policing (3)

- Queue Tail Drop
  - Dropping of packets which don't fit into a fixed size queue
    - Naive approach, little overhead, works well for underutilized networks
    - Pathological situations on overutilization
      - Congestion
      - Late detection
      - Global flow synchronization (simultaneous hold back)
- Random Early Detection
  - Dropping packets randomly with the probability proportional to a weighted average of the queue length
    - Early warning of congestion
    - Slower, graceful degradation

Network filesystems

- File based
  - VFS-level access, usual file operations
  - Pitfalls
    - Network reliability
      - Packet loss, packet duplication
      - Connection loss
    - Distribution of state
      - Open files, file position, file size
      - Locking, security
    - Consistency
      - Access to the same file from multiple nodes

File-based network filesystems

- NFS (v2, v3)
  - Originally by Sun Microsystems
    - Straitforward mapping to the Solaris VFS
    - Common security/trust realm (access control on clients)
    - Based on Sun RPC and XDR
  - Network encapsulation
    - Usually over UDP
    - Packets limited to 8 KB
      - Problems with the atomicity of some operations

File-based network filesystems

- NFS (v2, v3)
  - Stateless operation
    - No explicit open/close operation
      - File handles are opaque 32 B identifiers
        - Usually containing: filesystem ID, i-node, generation ID
      - Identpotent operations
        - Protection againsts packet duplication or replaying
        - File offset always stated explicitly
        - Complicates some operations (end-of-file append)
    - Stateful extensions (separate protocols)
      - Mount protocol
      - Locking protocol
        - Periodical lock lease renewal

File-based network filesystems

- NFS (v4)
  - Abandoned the stateless design
    - Mount and lock protocol integrated
    - Open/close explicit with leases
      - Consistency on close/flush
    - TCP operation, some firewall friendly
    - External authentization possible (Kerberos, etc.)
  - Generally very similar to CIFS (SMB)

File-based network filesystems

- AFS
  - Single global name space divided into cells
    - Each cell managed by a set of servers
      - Files stored on multiple volumes
      - Support for read-only volume replicas (load ballancing, backup, manual replication)
    - Client caches file contents
      - Propagation to the server on close/flush
    - Server sends data with a callback
      - On external modification the server notifies about the changed data
    - Uses Rx over UDP (proprietary RPC mechanism)

Block-based network filesystems

- GFS
  - A single filesystem image accessed on a shared block device
    - iSCSI, multipath SCSI, Fibre Channel, etc.
    - On-disk layout can be similar to classical disk filesystems
      - Storage pools, bitmaps, i-nodes, directory entries, journal
      - Most structures occupy entire blocks (sectors)
    - Distributed locking protocols for locking of blocks
      - NOLOCK (for local use only)
      - Distributed LM (distributed architecture, migrating lock instances)
      - Grand Unified LM (client/server, replicated lock servers with majority quora)