Advanced Operating Systems Lecture notes

Dr. Clifford Neuman
Dr. Dongho Kim
University of Southern California
Information Sciences Institute

CSci555: Advanced Operating Systems

Lecture 13 – November 19, 2004
Scalable Systems
(slides by Dr. Neuman)

Dr. Clifford Neuman
University of Southern California
Information Sciences Institute

Hints for building scalable systems

- From Lampson:
  - Keep it simple
  - Do one thing at a time
  - If in doubt, leave it out
  - But no simpler than possible
  - Generality can lead to poor performance
  - Make it fast and simple
  - Don't hide power
  - Leave it to the client
  - Keep basic interfaces stable

Hints for building scalable systems

- From Lampson:
  - Plan to throw one away
  - Keep secrets
  - Divide and conquer
  - Use a good idea again
  - Handle normal and worst case separately
  - Optimize for the common case
  - Split resources in a fixed way
  - Cache results of expensive operations
  - Use hints

Hints for building scalable systems

- From Lampson:
  - When in doubt use brute force
  - Compute in the background
  - Use batch processing
  - Safety first
  - Shed load
  - End-to-end argument
  - Log updates

Scale in Distributed Systems - Neuman

- A system is said to be scalable if it can handle the addition of users and resources without suffering a noticeable loss of performance or increase in administrative complexity.

Three dimensions of scale

- Numerical
  - Number of objects, users
- Geographic
  - Where the users and resources are
- Administrative
  - How many organizations own or use different parts of the system

Effects of Scale

- Reliability
  - Autonomy, Redundancy
- System Load
  - Order of growth
- Administration
  - Rate of change
  - Heterogeneity

Techniques - Replication

- Placement of replicas
  - Reliability
  - Performance
  - Partition
  - What if all in one place
- Consistency
  - Read-only
  - Update to all
  - Primary Site
  - Loose Consistency

Techniques - Distribution

- Placement of servers
  - Reliability
  - Performance
  - Partition
- Finding the right server
  - Hierarchy/iteration
  - Broadcast

Techniques - Caching

- Placement of Caches
  - Multiple places
- Cache consistency
  - Timeouts
  - Hints
  - Callback
  - Snooping
  - Leases

Review for Final

- General
  - Operating Systems Functions
  - Kernel structure - microkernels
  - What belongs where
- Communication models
  - Message Passing
  - RPC
  - Distributed Shared Memory
  - Other Models

Review for Final

- Synchronization - Transactions
  - Time Warp
  - Reliable multicast/broadcast
- Naming
  - Purpose of naming mechanisms
  - Approaches to naming
  - Resource Discovery
  - Scale

Review for Final

- Security - Requirements
  - Protection
  - Authentication
  - Authorization (ACL, Capabilities)
  - Scale

Review for Final

- Distributed File Systems - Caching
  - Replication
  - Synchronization
    _voting,master/slave
  - Distribution
  - Access Mechanism
  - Access Patterns
  - Availability
- Other file systems
  - Log Structured
  - RAID

Review for Final

- Case Studies
  - Locus
  - Athena
  - Andrew
  - V
  - HCS
  - Amoeba
  - Mach
  - CORBA
- Resource Allocation
- Real time computing
- Fault tolerant computing

SCALE

Advanced Operating Systems
Lecture notes

Dr. Clifford Neuman
Dr. Dongho Kim
University of Southern California
Information Sciences Institute

CSci555: Advanced Operating Systems

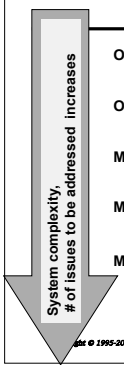
Lecture 13 part 2 – November 21, 2003
Review slides by Dr. Neuman)

Dr. Clifford Neuman
Dr. Tatyana Ryutov
University of Southern California
Information Sciences Institute

Review for Final

- General
  - Operating Systems Functions
  - Distributed System Structure
  - Kernels
    - _ Functions
    - _ Structure
      - microkernels vs monolithic kernels
  - E2E argument
  - What belongs where

Review for final



One user, one site, one process

One user, one site, multiple processes

Multiple users, one site, multiple processes

Multiple (users, sites and processes)

Multiple (users, sites, organizations and processes)

Review for Final

- Communication models
  - Message Passing
  - RPC
  - Distributed Shared Memory
    - _approaches
  - Other Models

Review for Final

- Synchronization
  - Transactions (local, distributed, nested)
    - Atomicity
    - Concurrency control in DT
    - Deadlock detection
  - Time Warp
  - Reliable multicast/broadcast

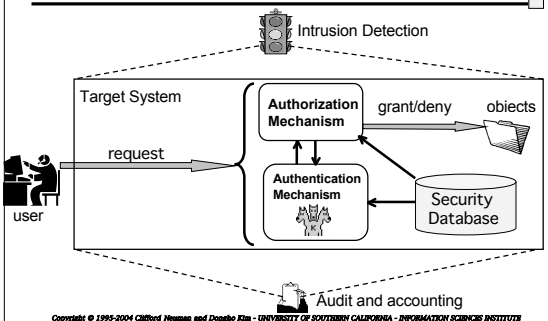
Review for Final

- Naming
  - Purpose of naming mechanisms
  - Approaches to naming
  - Resource Discovery
  - Scale

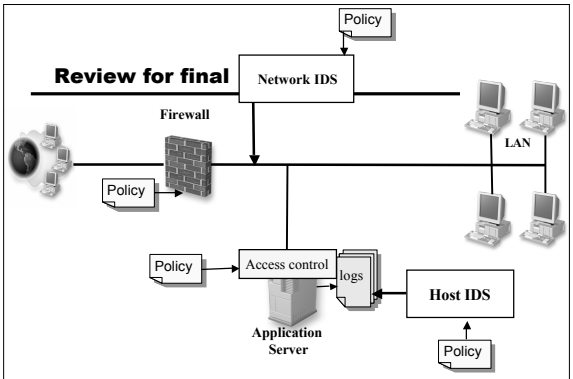
Review for Final

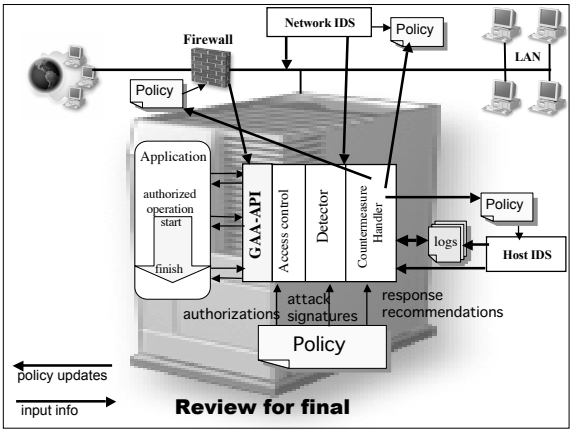
- Security – Requirements
  - Confidentiality
  - Integrity
  - Availability
- Security mechanisms (prevention/detection)
  - Protection
  - Authentication
  - Authorization (ACL, Capabilities)
  - Intrusion detection
  - Audit
- Cooperation among the security mechanisms
- Scale

Review for final



Review for final





Review for Final

- Distributed File Systems
  - Caching
    - _Cache consistency
  - Replication
  - Synchronization
    - _voting,master/slave

Review for Final

- Distributed File Systems
  - Stateful/statless
  - Distribution
  - Access Mechanism
  - Access Patterns
  - Availability
  - Security
- Other file systems
  - Log Structured
  - RAID

Review for Final

- Case Studies
  - Locus
  - Athena
  - Andrew
  - V
  - HCS
  - Amoeba
  - Mach
  - CORBA

Review for Final

- Resource Allocation
- Real time computing
- Fault tolerant computing
- Scale