Advanced Operating Systems
Lecture notes

Dr. Dongho Kim
Dr. Tatyana Ryutov
University of Southern California
Information Sciences Institute

Review: Transparency in Locus

- Network Transparency
  - Ability to hide boundaries
- Syntactic Transparency
  - Local and remote calls take same form
- Semantic Transparency
  - Independence from Operand Location
- Name Transparency
  - A name always refers to the same object
  - No need for closure, only one namespace

Review: Transparency in Locus (cont)

- Location Transparency
  - Location can't be inferred from name
  - Makes it easier to move objects
- Syntactic Transparency
  - Local and remote calls take same form
- Performance Transparency
  - Programs with timing assumptions work
- Failure Transparency
  - Remote errors indistinguishable from local
- Execution Transparency
  - Results don't change with location

Review: Replication in LOCUS

- File group replicated at multiple servers.
- Replicas of a file group may contain different subsets of files belonging to that file group.
- All copies of file assigned same descriptor (i-node #).
  - File unique name: <file group#, i-node #).

Review: Replica Consistency

- Version vectors.
  - Version vector associated with each copy of a file.
  - Maintain update history information.
  - Used to ensure latest copies will be used and to help updating outdated copies.
  - Optimistic consistency.
    - Potential inconsistencies.

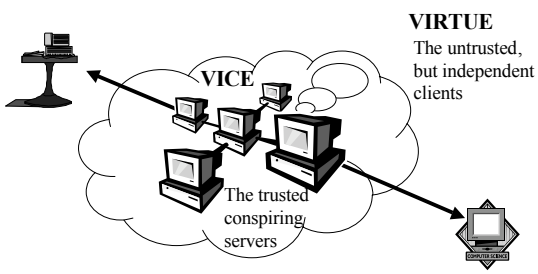
CSci555: Advanced Operating Systems
Lecture 10 – October 30,31 2003
Case Studies
(slides by Dr. Neuman and Dr. Obraczka)

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

Andrew System

- Developed at CMU starting in 1982
  - With support from IBM
  - To get computers used as a tool in basic curriculum
- The 3M workstation
  - 1 MIP
  - 1 MegaPixel
  - 1 MegaByte
  - Approx $10K and 10 Mbps network, local disks

Vice and Virtue



Andrew System (key contributions)

- Network Communication
  - Vice (trusted)
  - Virtue (untrusted)
  - High level communication using RPC w/ authentication
  - Security has since switched to Kerberos
- The File System
  - AFS (led to DFS, Coda)
- Applications and user interface
  - Mail and FTP subsumed by file system (w/ gateways)
- Window manager
  - similar to X, but tiled
  - toolkits were priority
  - Since moved to X (and contributed to X)

Project Athena

- Developed at MIT about same time
  - With support from DEC and IBM (and others)
    - _ MIT retained all rights
  - To get computers used as a tool in basic curriculum
- Heterogeneity
  - Equipment from multiple vendors
- Coherence
  - None
  - →Protocol
  - →Execution abstraction (e.g. programming environment)
  - Instruction set/binary

Mainframe/WS vs Unified Model (athena)

- Unified model
  - Services provided by system as a whole
- Mainframe / Workstation Model
  - Independent hosts connected by e-mail/FTP
- Athena
  - Unified model
  - Centralized management
  - Pooled resources
  - Servers are not trusted (as much as in Andrew)
  - Clients and network not trusted (like Andrew)

Project Athena - File system evolution

- Remote Virtual Disk (RVD)
  - Remotely read and write blocks of disk device
  - Manage file system locally
  - Sharing not possible for mutable data
  - Very efficient for read only data
- Remote File System (RFS)
  - Remote execution of file system calls
  - Target host is part of argument (no syntactic transparency).
- SUN's Network File System (NFS) - covered
- The Andrew File System (AFS) - covered

Project Athena - Other Services

- Security
  - Kerberos
- Notification/location
  - Zephyr
- Mail
  - POP
- Printing/configuration
  - Hesiod-Printcap / Palladium
- Naming
  - Hesiod
- Management
  - Moira/RDIST

Heterogeneous Computer Systems Project

- Developed
  - University of Washington, late 1980s
- Why Heterogeneity
  - Organizational diversity
  - Need for capabilities from different systems
- Problems caused by heterogeneity
  - Need to support duplicate infrastructure
  - Isolation
  - Lack of transparency

HCS Aproach

- Common service to support heterogeneity
  - Common API for HCS systems
  - Accommodate multiple protocols
- Transparency
  - For new systems accessing existing systems
  - Not for existing systems

HCS Subsystems

- HRPC
  - Common API, modular organization
  - Bind time connection of modules
- HNS (heterogeneous name service)
  - Accesses data in existing name service
  - Maps global name to local lower level names
- THERE
  - Remote execution (by wrapping data)
- HFS (filing)
  - Storage repository
  - Description of data similar to RPC marshalling

CORBA (Common Object Request Broker Architecture)

- Distributed Object Abstraction
  - Similar level of abstraction as RPC
- Correspondence
  - IDL vs. procedure prototype
  - ORB supports binding
  - IR allows one to discover prototypes
  - Distributed Document Component Facility vs. file system

Microsoft Cluster Service

- A case study in binding
  - The virtual service is a key abstraction
- Nodes claim ownership of resources
  - Including IP addresses
- On failure
  - Server is restarted, new node claims ownership of the IP resource associated with failed instance.
  - But clients must still retry request and recover.