Large-Scale Systems: WebOS

- Access to geographically distributed data-dissemination *and* computing resources.
- Resource discovery, persistent storage, process control, security and authentication
- WebOS architecture : Smart clients, smart proxies, WebFS
- Applications: Intenet Chat, Remote Compute Engine, Wide-Area Cooperative Cache, Rent-A-Server
- *Reading:*
  - *A. Vahdat et al. "WebOS: Operating System Services for Wide Area Applications"* **(http://now.cs.berkeley.edu/WebOS/papers/webos-overview.ps)**

Access to Geographically Distributed Computing Resources

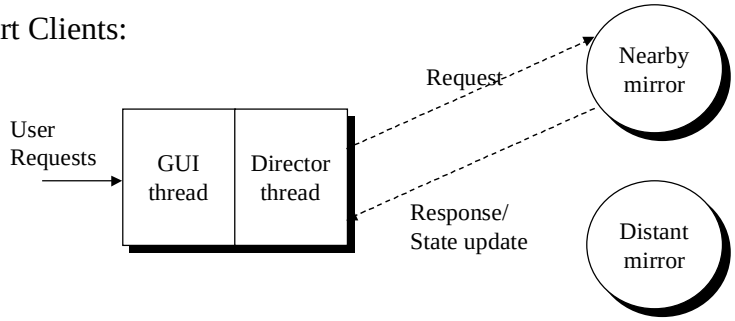
- WWW makes access to distributed read-only data easy
- What about distributed computing resources?
- Alta Vista, Netscape download page
  - geographically replicated to improve bandwidth, reduce latency, improve availability
  - manually managed, at server and client

WebOS Components

- Resource Discovery
  - Client Application must be able to dynamically locate server with best QoS
  - Mapping of name to service; load balancing; state management
  - Smart Clients, Smart Proxies
- Wide Area File System
  - To support replication and wide-scale sharing.
  - WebFS: cache coherent wide area file system.
- Security and Authentication
  - To support applications operating across organizational boundaries.
- Process Control
  - Execute a process on remote node.
  - Control to safeguard local system integrity and resource usage.

Resource Discovery

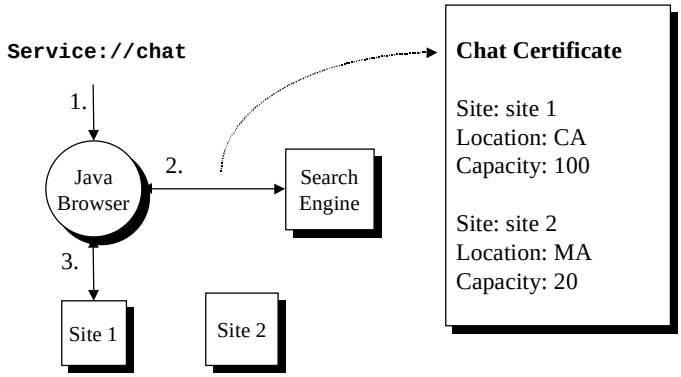
- How does a client locate a representative of geographically distributed and dynamically reconfiguring service?
- Current Approaches:
  - HTTP Redirection
  - DNS Aliasing
- Smart Clients:



- Bootstrapping applet retrieval: *meta-applet* in browser.

Resource Discovery (2)

- Bootstrapping applet retrieval through **meta-applet** in browser.



1. User request
2. Certificate request
3. Applet request

Resource Discovery (3)

- Smart Cients: load balancing, naming, fault tolerance
- Limitations of Smart Clients:
  - Clients must choose applets with potentially dangerous behavior: connect to arbitrary network hosts.
  - Load information quickly outdated.
- Smart Proxies:
  - aggregate behavior of multiple clients.
  - Security concerns addressed only once by system administrator

Persistent Storage

- Many distributed applications share state and transfer control using **communication** abstraction.
- Simplify such applications by using global cache coherent **file system** for communication and synchronization.
- Example: encapsulate conversation in internet chat in secure, globally accessible files.
- WebFS: URL-based cache coherent file system
  - URL's as name space

  e.g. **`ls /http/www.usenix.org/event`**
  - File system semantics:
    - consistency policies are application controllable.
    - support for optimistic re-integration to deal with disconnected operations

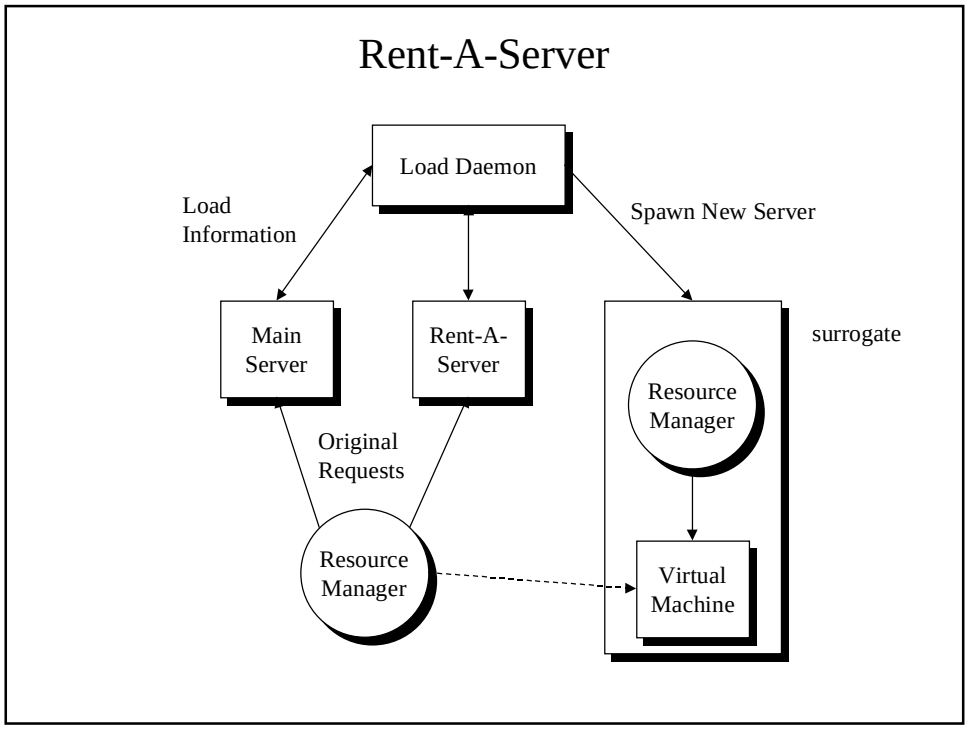
Process Control

•Execution of processes on remote nodes simple.

•Safety?
- Execution in separate address space.

•Fairness?
- Resource manager in each WebOS machine.
- Process runs in virtual machine (created by Janus(*))
- Resource allocation:
  - **`priocntl`** and **`setrlimit`** to set maximum amount of memory and CPU usage for process.
  - Solaris **`/proc`** file system to intercept problematic system calls.

(*) Goldberg, Wagner, Thomas, Brewer. "A Secure Environment for Untrusted Helper Applications". *Proceedings of the Sixth USENIX Security Symposium*, July 1996.


Rent-A-Server
Load Daemon
Load Information
Spawn New Server
Main Server
Rent-A-Server
surrogate
Resource Manager
Original Requests
Resource Manager
Virtual Machine