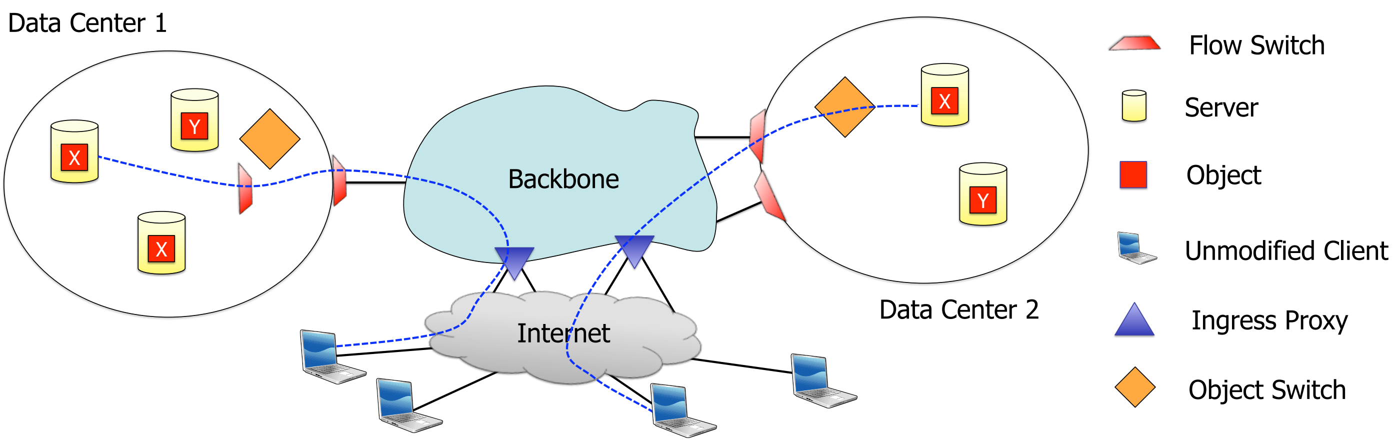
A SCAFFOLD for Wide-Area Distributed Services

Michael J. Freedman, Jennifer Rexford, Steven Y. Ko, Prem Gopalan, and David Shue
Princeton University

SCAFFOLD (Service-Centric Architecture For Flexible Object Localization and Distribution)



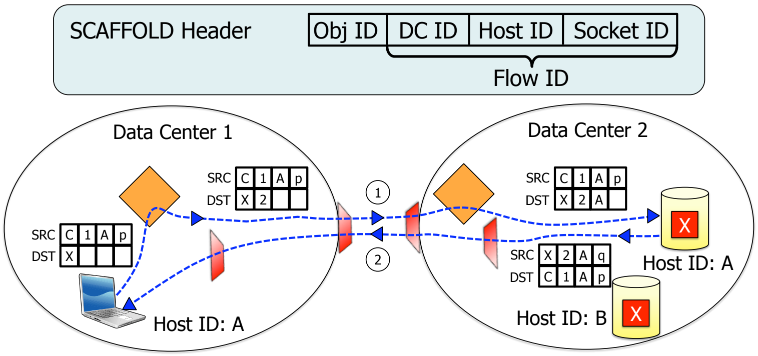
- Designed to support wide-area services
  - Users access service objects.
  - Objects are replicated.
  - Many types of churn arise.
    - Server and route failures
    - Object migration and user mobility
    - Planned downtime
    - Load balancing
- Designed to overcome the limits of the current Internet
  - DNS
    - Domain names are too *coarse*-grained.
    - DNS binds too *early*.
  - IP
    - IP addresses are too *fine*-grained as service ids.
    - IP anycast binds too *late*.
  - L2 solutions (e.g., ARP spoofing) inadequate in wide-area
- Applications
  - Content distribution networks
  - Distributed storage services
  - Virtual worlds
- Environments
  - VM-based cloud platforms
  - Legacy enterprise applications
  - Mobile users

- Existing solutions require complex combinations of various tricks.

SCAFFOLD Features

- Combining naming and routing
  - Flat naming for objects (e.g., services, files, etc.)
  - Routing based on object names
- Anycast with flow affinity as a first-class primitive
  - Anycast to locate an object
  - Flow affinity to one server for stateful services
  - No per-flow state via per-packet flow id
- Failover and migration support from the network
  - Transparent redirection of flows upon failure or migration
- Clean-slate, yet incrementally-deployable
  - Ingress proxies to interoperate with unmodified clients

SCAFFOLD Packet Flow: Successive Refinement



Components

- Object switch
  - Stores (obj id -> DC/host) mappings
  - Resolves object ids
- Flow switch
  - Performs routing on DC & host IDs
  - Acts as a gateway for end-hosts
- Controller
  - Controls object and flow switches
  - Handles migration and failover

Integration

- SCAFFOLD socket API
  - Binds on object names
  - Generates SCAFFOLD packets
  - Mimics the existing socket API
- Ingress proxy
  - Interacts with unmodified clients
  - Performs IP ↔ SCAFFOLD conversion
  - Lowers the barrier for new services

Building upon GENI Technologies

- OpenFlow switches for object and flow switches for both “L2/L3” routing
- NOX for the controller
- Click for the end-host SCAFFOLD network stack
- VINI backbone for inter-domain routing
- Transit Portal for global Internet connectivity
- PlanetLab control framework for network and end-host configuration