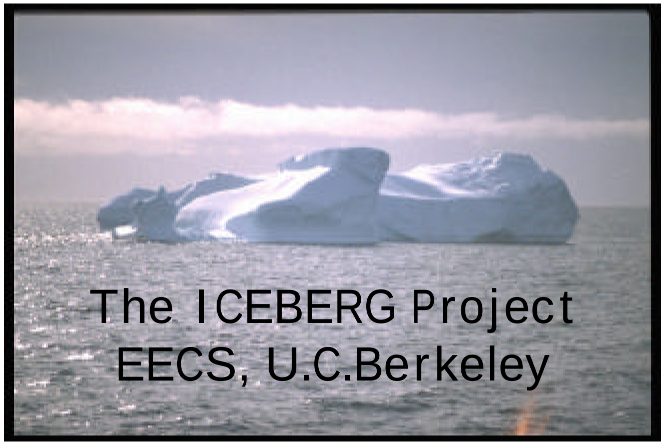
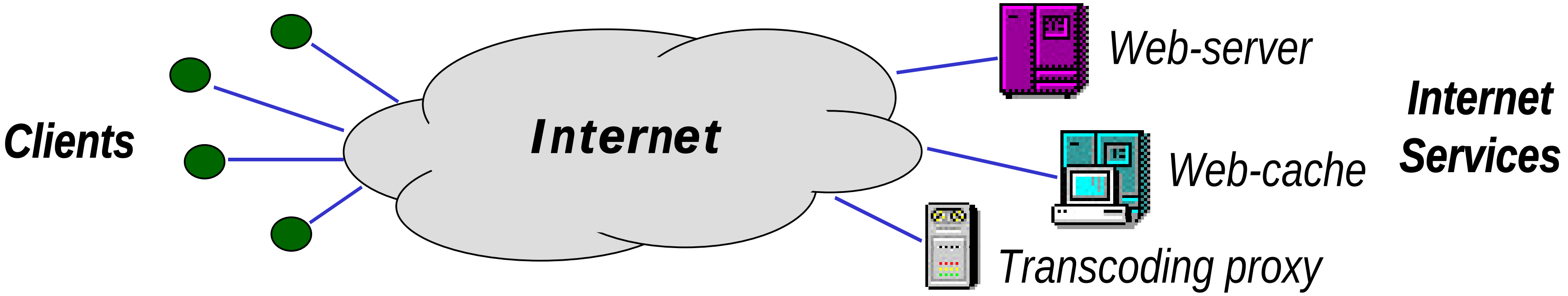
A Framework for Highly-Available Session-Oriented Internet Services

Bhaskaran Raman, Prof. Randy H. Katz
{bhaskar, randy}@cs.berkeley.edu



The ICEBERG Project
EECS, U.C.Berkeley

Resilient Session-Oriented Internet Services



- For Internet services (e.g., web-servers, proxies), robustness & high-availability are very important
- Session-Oriented Services
  - Client sessions live for long: many minutes to hours
  - E.g., Audio/Video transcoding proxies, Internet Video-on-demand, Game servers
  - Important, growing class of Internet applications
- For such services, high-availability means:
  - No session interruption at client in presence of failures

Motivation and Problem Scope

Motivation – Internet components subject to failure

- – Should be hidden from end-clients
- – Services often critical to users
- – Session re-instantiation on failure little studied so far

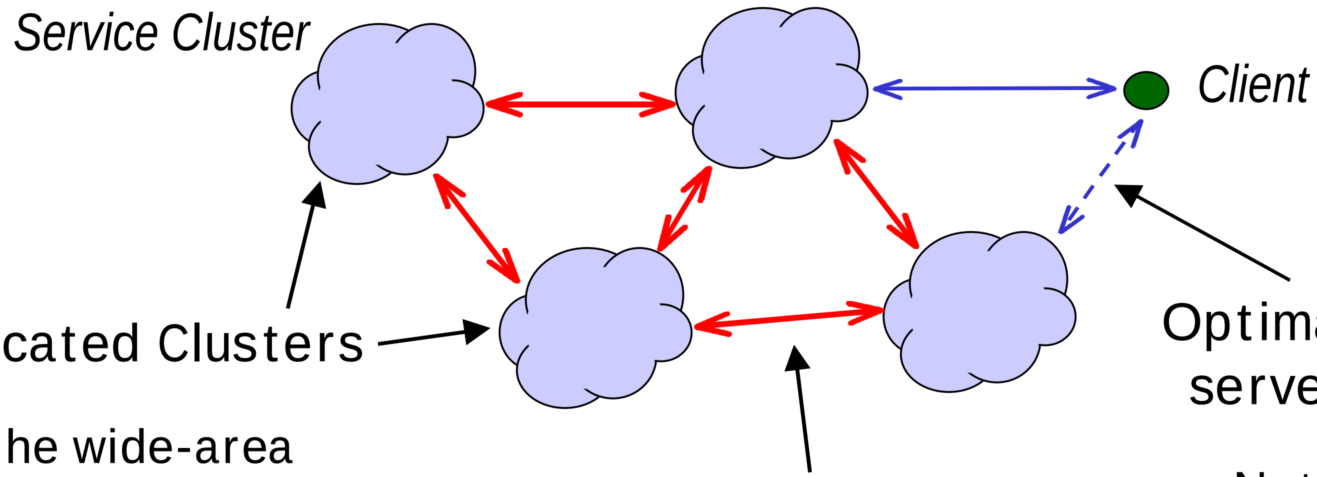
Goal – A framework for robust session-oriented services:

- – Handle failures at all levels
  - Process, machine, site-failure, network partition
- – **Quick recovery** from failure
  - Minimal service interruption for interactive sessions

Assumption: **Stateless** services

- – Little or no state build-up at service **during session**
  - All state at client(s) – problem of re-instantiating sessions is tractable
- – Examples: transcoding proxies, video-on-demand servers
- – We do not consider "stateful" services
  - Stateful ➔ Entirely different semantics for moving sessions

Our Framework



Replicated Clusters

- In the wide-area
- Tolerance to site/network failures
- Clusters could belong to independent service providers
  - Orthogonality good for fault-tolerance

Monitoring Mechanism

- **Peer-clusters** monitor each other
- Keep track of live-ness and performance behavior
  - For backup/fail-over purposes

Optimal choice of server instance

- Not only for session **setup** but for the **entire session** duration
- **Session-transfer** on detecting poor performance or failure

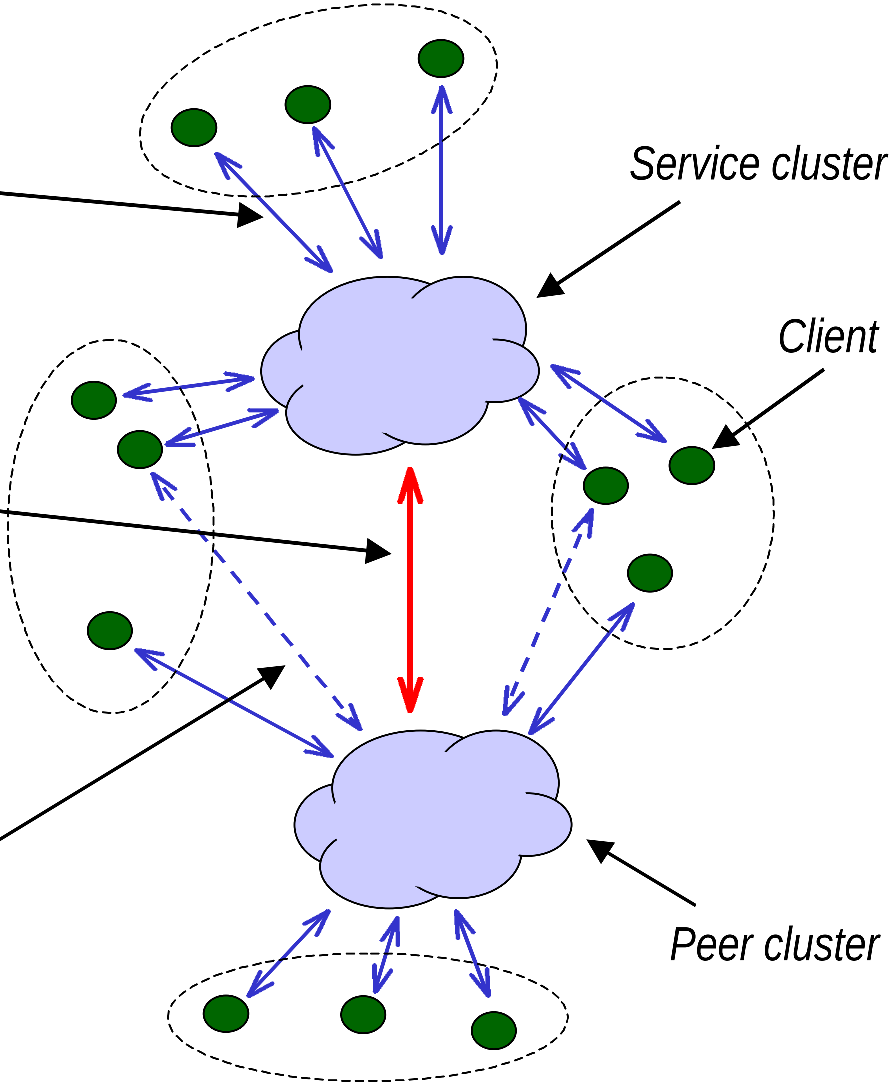
Support for Session-transfer

Cluster monitors performance of its clients – or gets performance reports from clients

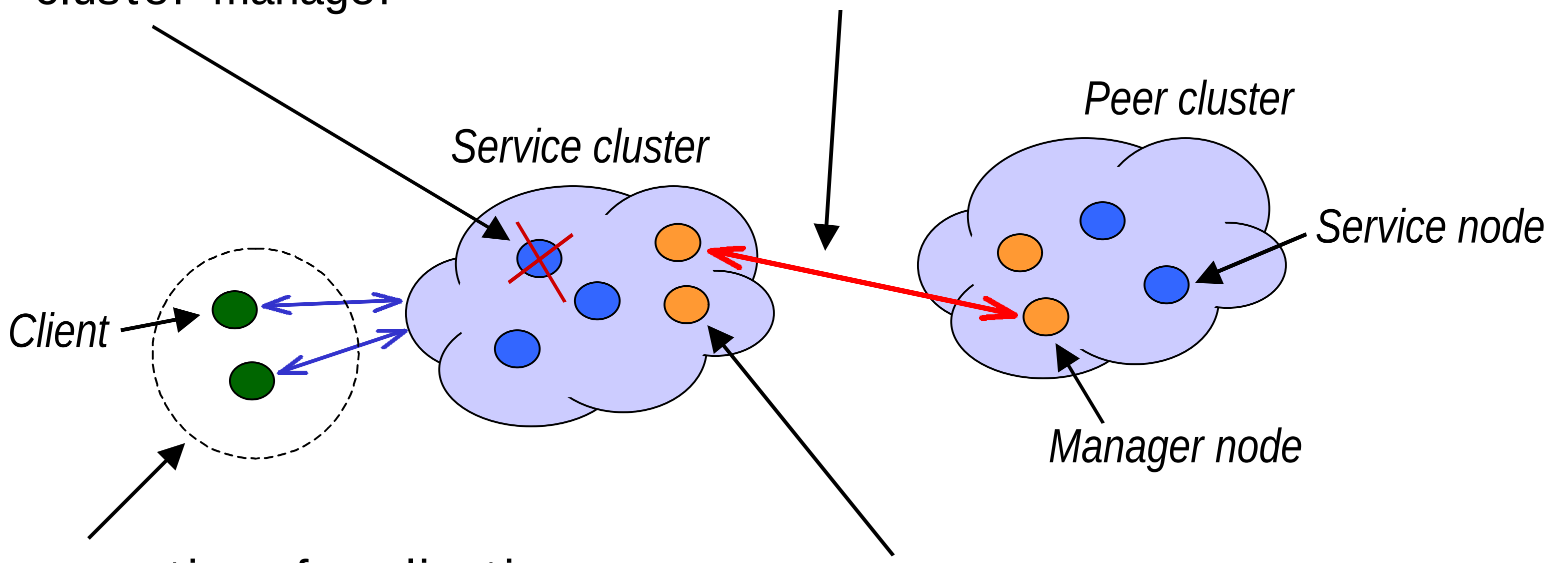
Periodic heart-beats to keep track of liveness – timeouts to detect failure

Peers know about each others' clients – info exchanged during session setup

On cluster/network-failure, or poor performance, session is taken over by peer cluster



Support for Session-transfer (Contd.)



Process/machine failures handled within cluster – by cluster-manager

Peer-monitoring done by manager-nodes – overhead amortized across many clients

Aggregation of application performance information – based on client “location” – for choosing “best” peer-backup-cluster

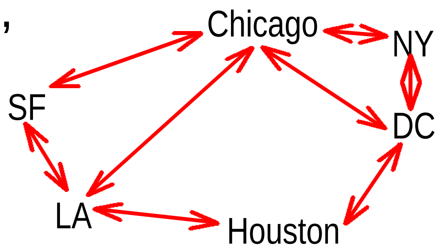
Replicated cluster-manager for redundancy

Open Research Questions

- Location of replicated services
  - How do clusters find each other?
  - How do they agree to *peer* ?
    - Who monitors who?
    - Who can serve as backup?
  - Which peer is the best backup for a particular client?
  - What are the trust relationships between service providers?
- Issues with inter-cluster wide-area monitoring
  - How to detect failures quickly in the wide-area Internet?
  - With low overhead?
  - Issue of false detection of failure due to latency variations
  - Stability: when site fails,
    - do not want to move all sessions to the same backup cluster
    - this may cause load-collapse of entire system

Next Steps…

- Cluster location and peering:
  - Graph of peering clusters should correspond to “network-closeness”. E.g.,



  - Peering can be similar to that for Internet backbone routers
- To identify “best” backup peer cluster for a client
  - Client clustering [Krishnamurthy & Wang, SIGCOMM’00]
  - Distance estimation [IDMaps, INFOCOM’99, INFOCOM’00]
- Design of heartbeat mechanism & timeouts
  - Studies of Internet delay behavior: [Allman & Paxson, SIGCOMM’99], [Acharya & Saltz, UMCP, ‘97]
    - RTT spikes are temporal ➔ possible to design **tight** and **reliable** heartbeat mechanism in the wide-area?

Related Work

- Summary of related work:
  1. Cluster-based approaches: TACC [A. Fox et.al. SOSP'97], LARD [V.S. Pai et.al. ASPLOS '98]
  2. Video transcoding proxy: Active Services [E. Amir et.al. SIGCOMM 1998]
  3. Web-mirror selection methods: SPAND [S. Seshan et.al. INFOCOM 2000]
  4. Fault-tolerant distributed computing (not Internet services)
  5. Our Approach: Wide-area replicated clusters + Monitoring

	(1)	(2)	(3)	(4)	(5)
Scope of failure recovery	Within cluster	Within cluster	In the wide-area	Local-area	Within and across wide-area clusters
Session-transfer capability	No	Yes	No	No	Yes

Session-recovery in the wide-area not considered so far

Summary

- Requirement for robust session-oriented services:
  - Arose from use of transcoding services for device communication in hybrid networks [ICEBERG, U.C.Berkeley]
  - Availability requirements especially stringent for communication services (e.g., the telephone system)
- Framework for high availability
  - Wide-area monitoring mechanism betn. service cluster peers
- Approach promises low-overhead, low-delay fail-over
  - Amortization of control overhead with use of clusters
  - Common failure cases handled within cluster
  - Backup cluster peers for quick recovery from site/network failures
- Next step: prototype implementation and evaluation…
  - Study trade-off between (a) service re-instantiation time, (b) monitoring overhead, and (c) amount of pre-provisioning