Locality Aware Mechanisms for Large-scale Networks

Ben Y. Zhao
Anthony D. Joseph
John D. Kubiatowicz
UC Berkeley

Future Directions in Distributed Computing
June 2002

Global Scale Applications

- Clear demand for global scale applications
  - Exploiting collective resources
  - File sharing, data dissemination, shared computation
- Wide-area issues
  - Scalability, fault-handling, adaptability, manageability
- Decentralized Object Location and Routing (DOLR)
  - Provides scalable message routing to object location
  - Spawns numerous apps:
    file systems, multicast, web caches, mobility systems
- Can we make them (& their apps) more scalable?

Outline

- Decentralized location and scalability
- Locality-awareness in design and policy
  - Proximity metrics
  - Data replication
  - Service replication
  - State maintenance
- The Gaia network model

Decentralized Location Review

- Goal: route message to object given unique OID
- Key Properties:
  - Scalable overlay routing mesh (metric: RDP)
  - (sub-)Logarithmic storage per node
  - (sub-)Logarithmic overlay hops per route
  - *Scalable algorithmic core*
- One perspective on differentiation
  - Proximity routing versus random route choice
  - Where and how object location info is stored

Wide-area Scalability

- Large number of 2nd generation P2P apps
  - Multicast: Bayeux, CAN multicast, Scribe
  - FS: OceanStore, PAST, Mnemosyne, CFS
  - Others, I3, Mobile Tapestry, Squirrel, etc…
- Apps differ in scalability properties
  - Question:
    Are they truly scalable?
    Or …
    Why is a wide-area application or networking substrate more scalable than another?

Locality Awareness

- Optimizing infrastructure load for scalability
- "*The ability to exploit local resources over remote ones whenever possible*"
- "-Centric" approach
  - Client-centric, server-centric, data source-centric
- Intuition: raindrops rippling in a pond
  - Client requests result in load on overall network
  - Dampen / confine the impact of network operations: storage / communication costs, faults

Another Perspective

- More transparent layering beneficial
  - DiffServ, resource allocation
- Overlays / IP → inefficient interaction
  - Expose IP level network scale
  - Attempt to correlate interlayer behavior
- Old concept, new application (WANs)

Outline

- Decentralized location and scalability
- Locality-awareness in design and policy
  - Proximity metrics
  - Data replication
  - Service replication
  - State maintenance
- The Gaia network model

Routing Proximity

- Building a routing table w/ constraints
- Proximity neighbor selection (*Tapestry / Pastry*)
  - Integrate proximity metrics into routing construction
  - Distributed algorithms approx. global knowledge
  - "Locally optimal" routing tables maintained
- Proximity routing (*CAN / Chord*)
  - Local heuristics determine preference among routes
  - Performance gain for long-running stable networks
- ✓ Locality awareness
  - Minimize wide-area traffic, bandwidth utilization, congestion, and sensitivity to wide-area faults

Data Replication

- Storage and consistency vs. performance
- Providing local resolution of client requests
  - Caching: Akamai
  - Streaming media: multicast
  - Process and file migration
- In context of object location systems
  - # of replicates (metadata or data)
  - Placement of replicates

Replication and Placement

- Largely independent of actual network substrate
- CAN:
  - Replicated data propagates back to hotspot source
  - Long term performance gain for immutable data
- Pastry / Chord / Kademlia:
  - Distribute replicates of immutable objects randomly
- Tapestry / PRR (Plaxton, Rajamaran, Richa)
  - Replicate *Log(N)* object pointers
  - More replicates around location of object
- Application level: OceanStore, SCAN, ...

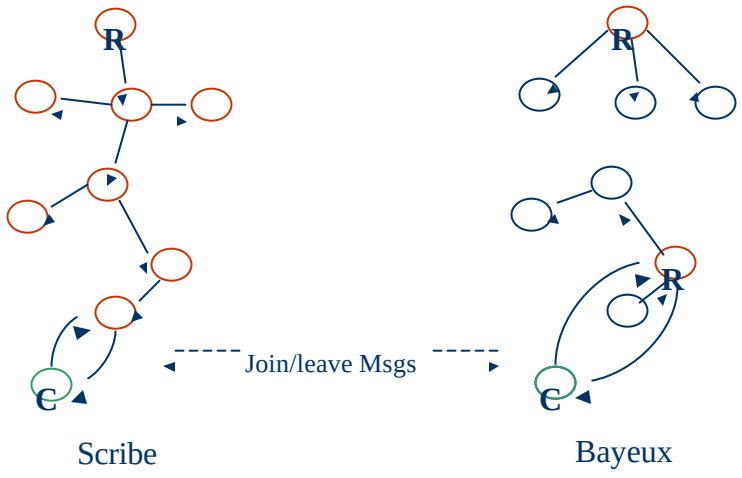
Implications

- Static approaches vs. run-time optimizations
  - Overlays increasingly dynamic in membership
  - Run-time schemes too slow to be effective
  - Approach must consider underlying network scale
    - node-to-node object replicate propagation is high in storage cost and slow to stabilize
- Locality awareness of replication schemes
  - Object replication vs. object pointer replication
  - Indirection allows mutability / flexibility

Service Replication

- Distribute server processing close to clients
- Overlay multicast example: Scribe and Bayeux
- Scribe: membership handled by any node
  - New members find nearest node to attach to via Pastry location
  - Localize membership traffic, reduce server load
- Bayeux: explicitly replicated root nodes
  - Members find nearest root, then attach to MC tree
  - Self-organizing trees
- Other examples: dynamic transcoding

Service Replication …



State Maintenance

- Maintenance of link conditions, routing state
- Example: soft-state beacons measure link conditions between neighbors (Tapestry, Scribe)
- Locality awareness
  - Naïve: messages sent across overlay hops at same rate regardless of actual network distance
  - Traffic can scale with length of overlay hop, possibly causing congestion
- Alternatives:
  - Scale soft-state frequency w/ length of overlay hop
  - External fault-detection / measurement platform

Outline

- Decentralized location and scalability
- Locality-awareness in design and policy
  - Proximity metrics
  - Data replication
  - Service replication
  - State maintenance
- The Gaia network model

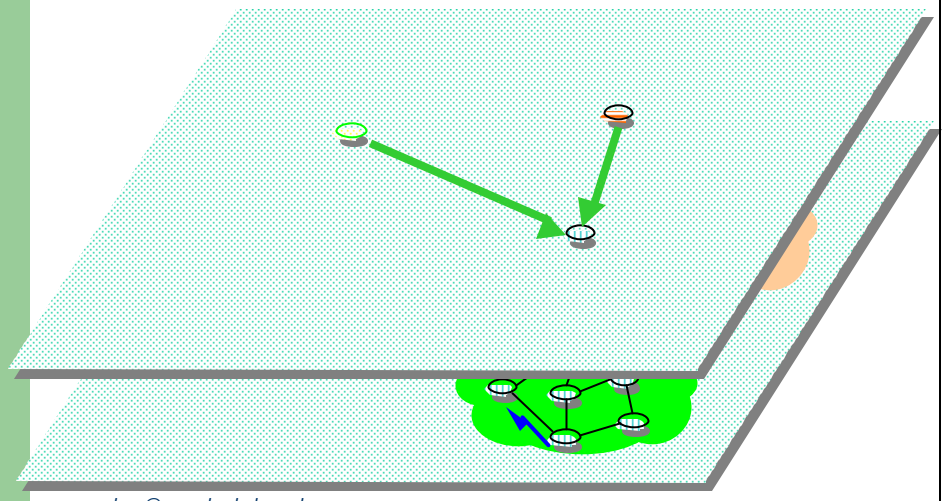
Scalable Network Core for UbiComp

- My (infrastructure-based) perspective:
  - Highly available (scalable) backbone infrastructure
  - Service large # of heterogeneous clients
- Solving some problems
  - Devices decreasing in size
  - Devices Increasing in power (Moore's law)
  - Network connectivity expanding
- Still need: management (availability/performance)
  - Shorter MTBF, changing network, security / attacks
  - Sheer # of devices makes normal solutions infeasible

Layered Management

- Proactive redundancy, exploit plentiful resources
  - Fault-detection not enough
    - Precomputation of redundancy + rapid adaptation
    - E.g. more replication for slow-changing data, active duplication of messages
  - Continuous probing for alternatives
  - Minimal degradation of availability during faults
- Making maintenance scalable
  - Stacked layers of maintenance domains
  - Each layer uses DOLRs for self-organization / comm.
  - Aggregate data at each level (GRID, SDS)

Layered Management



Scalable Protection

- Availability mechanisms need protection
  - Redundancy generation
  - Detection and triggering of faults
- Intelligent exception triggering
  - Pattern-based heuristics, tracebacks detect attack flows
  - Group agreement protocols make authoritative decisions
  - Aggregate warnings propagate up notification hierarchy
- Active countermeasures
  - Isolate and remove malicious nodes
  - Switch to redundant path / plane
    (GUID aliasing in Tapestry, realities in CAN)

The Living Network Model

- Gaia: a living network
  - James Lovelock [1979]
- Large scale self-management
  - Locally constrained interactions → scalability, performance
  - Layered control structure
  - Upward propagation of aggregate data
- Survival via active redundancy & self-repair
  - Catastrophic failures handled by top level control (human interaction)

For More Information

- CAN / CAN multicast
  - http://www.icir.org/sylvia
- Chord / CFS
  - http://www.pdos.lcs.mit.edu/chord
- Pastry / Past / Scribe / Squirrel
  - http://www.research.microsoft.com/~antr/pastry/
- Tapestry / OceanStore / Bayeux / Brocade
  - http://www.cs.berkeley.edu/~ravenben/tapestry
  - http://oceanstore.cs.berkeley.edu
- Other relevant material:
  - http://www.cs.berkeley.edu/~ravenben