

Tapestry: Fault-resilient Wide-area Location and Routing

or, Providing High Availability and Adaptability in a Decentralized System



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

OSDI 2000

Issues Facing Wide-area Systems

1. Larger scale systems contain more heterogeneous components, MTBF decreases
2. More data on the WAN exacerbates scalability problems for points of centralization
3. More dynamic components complicate system management
4. Wide-area operation increases vulnerability to security attacks (e.g. Denial of Service)

What is Tapestry?

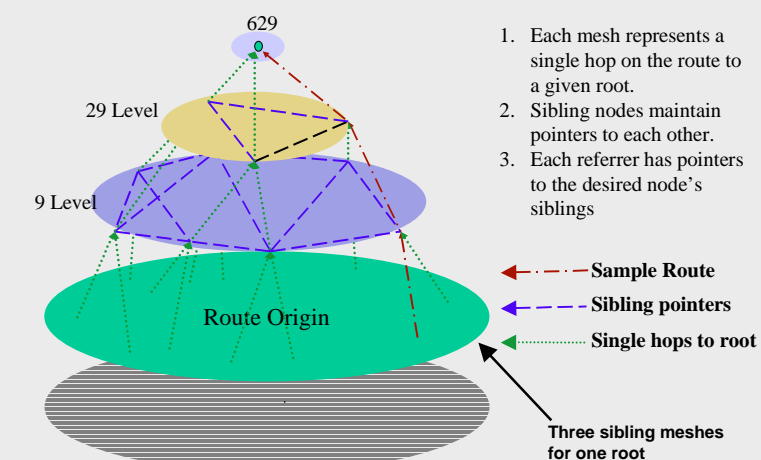
A wide-area location/routing layer based on Plaxton, with numerous enhancements.

Structural Additions

1. Logical “sibling mesh” for nodes sharing suffix
2. Several alternates in addition to each route pointer
3. Referrer list (backpointers)

Fault Handling

1. Fast fault detection
 1. Local heartbeats between neighbors, TTL=1; optionally piggyback queries to reduce traffic
 2. Neighbors propagate negative heartbeat
2. Fault repair
 1. Use alternate pointers to access sibling mesh
 2. Use mesh to circumvent faulty links
3. Fast recovery
 1. Second-chance algorithm give downed nodes time to recover before removing references
 2. Probabilistic use of query traffic as probes
 3. Invalid flag removed when node recovers



Availability

1. Incoming object IDs hashed using multiple salts and inserted as independent objects
2. Queries/inserts parallelized for redundancy
3. Potential dynamic “split” of queries at bottlenecks

Security

1. One-way hash of IDs prevent targeted DoS attacks
2. Use of backpointers actively isolate malicious nodes

Internal Multicast

1. Routing to multiple recipients reaps benefits: One copy per distinct suffix digit
2. Branch factor limited to b (base of IDs used)

Load-balancing

1. Insert arbitrary nodes can divert router load
2. Insert well-defined NodeIDs to migrate load for object pointer storage from existing nodes

Self-optimization/repair

1. Running queries embed route state (ID, latency)
2. Non-optimal routes detected during traversal; previous nodes informed via update message

Related work

1. Globe (vrije Universiteit)
2. Geographic Routing (Rutgers)
3. Content-Addressable Networks (AT&T / ACIRI)
4. GLS/Grid (MIT)

Building block: Plaxton Trees

Naming:

- Nodes and objects have bit-sequence identifiers
- Objects map deterministically to a *root node*

Routing: via Local Neighbor Maps

- Nodes maintain nearest neighbor per route-level maps
- Messages match ≥ 1 addl. target ID suffix digit per hop

Location: indexing via backpointers

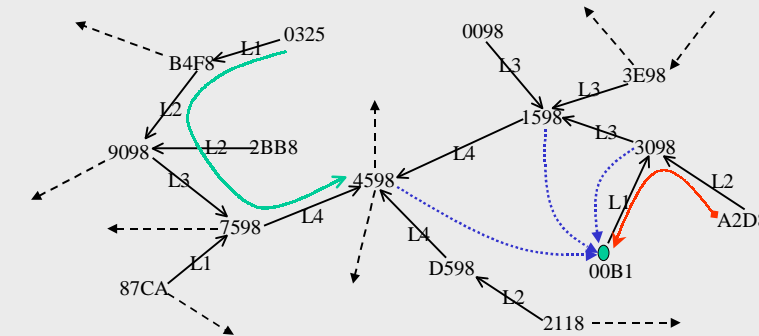
- Backpointers inserted at every hop from location to *root node*
- Searches route to *root node*, stop when pointer found

Benefits

- # of hops per route $\leq \log_b N$, $N = \#$ of nodes in system
- Exploit locality: searches rarely go to root node
- Decentralized scaling

Weaknesses

- Root nodes = single points of failure
- Vulnerable to Denial of Service attacks
- Topology changes have high cost



Tapestry Applications

- WAN-scale data dissemination
 - One to many (multimedia broadcast); Many to few (data aggregation); Many to many (large scale sensor networks)
- Decentralized PKI

Ongoing Work

- Theoretical analysis of algorithms' impact on performance
- Verification of analytical results via large-scale simulations
- Support for mobility: roaming data and clients
- Link to link MAC authentication
- Denial of Service Benchmarks