Tapestry and OceanStore

Lidan Wang

Feb 22, 2007

Outline

- Tapestry
  - A P2P overlay routing infrastructure
  - Motivated by the OceanStore project

- OceanStore
  - An architecture for global-scale persistent storage
  - A world-wide file system
  - Developed at the Univ. of California, Berkeley. ASPLOS 2000

Tapestry

- A P2P overlay routing infrastructure
- Location-independent routing to nearby copies
- Stable behavior and performance
- Key-id space
- Supports four API's
- Routing and Object Location
- Dynamic node algorithms
- Interesting features: network distance considerations; location pointers

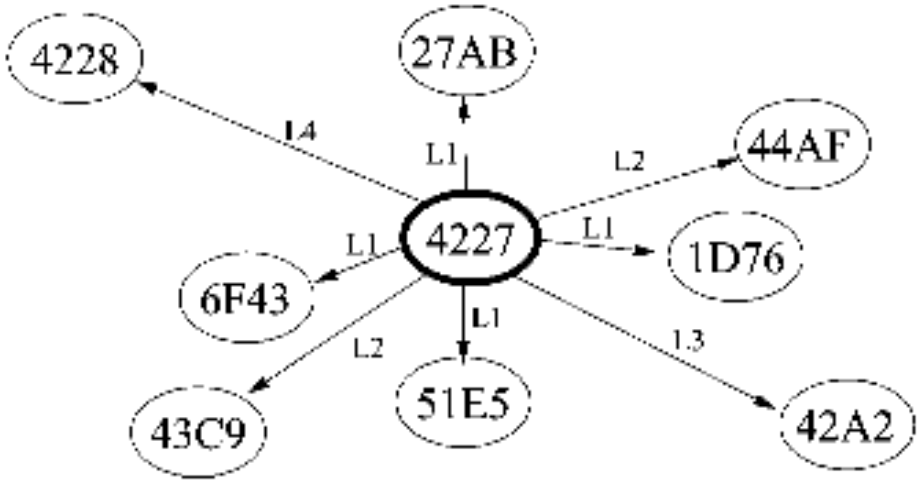
Tapestry - DOLR Networking API

- Key-id space (2^160). A sequence of digits
- nodeIDs selected uniformly at random
- More than one node can be hosted by one physical host
- GUIDs (application specific-endpoints)
- Four-part DOLR networking API:
  - PUBLISHOBJECTS (O_G, Aid)
  - UNPUBLISHOBJECT (O_G, Aid)
  - ROUTEToOBJECT (O_G, Aid)
  - ROUTEToNODE (N, Aid, Exact)

Tapestry - Routing and Object Location

- Node's neighbors at level "i"
  - Match prefix for i-1 digits
  - All possible variations for the ith digit

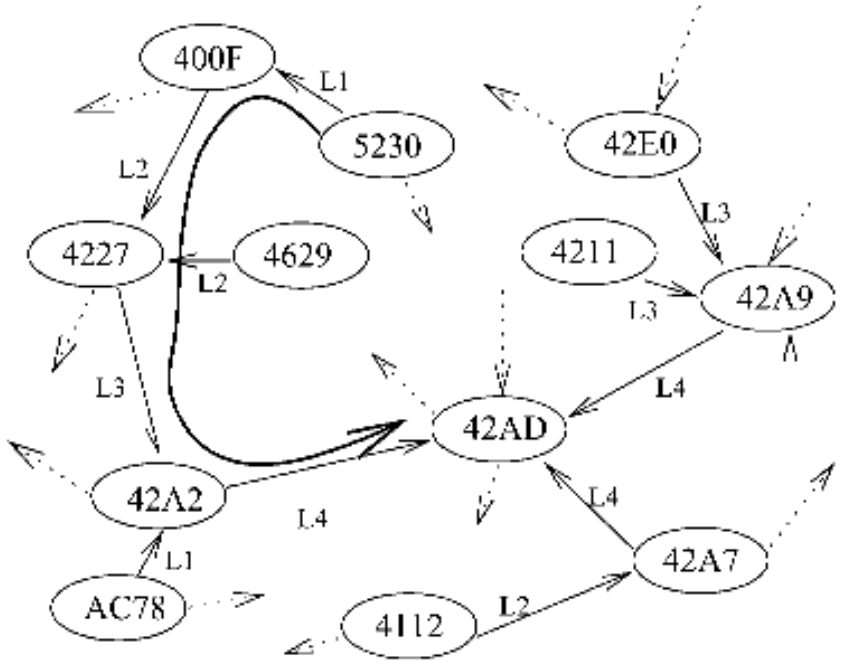
- For example, node 4227:



Taken from Tapestry: A Resilient Global-Scale Overlay for Service Deployment

Tapestry - Routing and Object Location (cont.)

- Path of a message: messages are forwarded progressively closer to the destination node in the ID space
- Example: routing a message from node 5230 to node 42AD

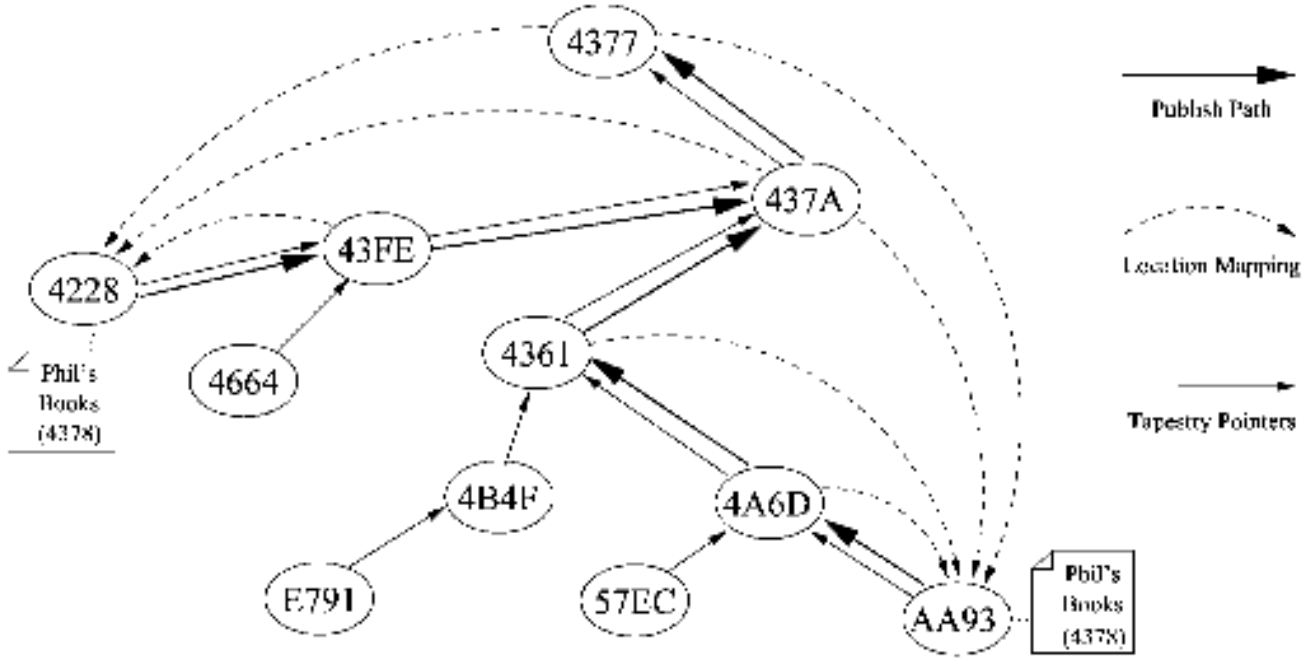


Taken from Tapestry: A Resilient Global-Scale Overlay for Service Deployment

Tapestry - Routing and Object Location (cont.)

- Guarantees that any existing node can be reached in at most $\log_b N$ logical hops
- Surrogate routing: when a digit cannot be matched, Tapestry looks for a “close” digit in the routing table.
- Object location and publication:
  - A server S publishes an object O by routing a publish message toward OR
  - Each node along the publication path stores a pointer mapping instead of a copy
  - Nodes store location mappings for object replicas in sorted order of network latency
  - Locate object O by routing a message to OR, check if location mapping exists along the way

Tapestry - Example: Object Location and Publication



Taken from Tapestry: A Resilient Global-Scale Overlay for Service Deployment

Tapestry - Node Additions and Deletions

- When a new node joins the network:
  - Need-to-know nodes are notified of N
  - N might become the new root for existing objects, so references to those objects must be moved to N to main object availability.
  - Construct a near optimal routing table for N
  - Nodes near N may use N in their routing tables as an optimization

- Voluntary Node Deletion
  - Notified nodes need to have replacement nodes for each routing level
  - Notified nodes send republish traffic to both N and its replacement

- Involuntary Node Deletion
  - Build redundancy into routing tables to improve object availability

OceanStore - A true data utility

- Infrastructure is comprised of untrusted servers
- Data protected through redundancy and cryptographic techniques
- Uniform and highly-available access to information; separation of information from location
- Servers geographically distributed and exploit caching close to clients
- Data can be cached anywhere, anytime, and can flow freely
  - Nomadic data (an extreme consequence of separating information from location)
  - Promiscuous caching: trade off consistency for availability; continuous introspective monitoring to discover data closer to users

OceanStore - Persistent Objects

- Named by a globally unique id, GUID
- Objects are replicated and stored on multiple servers
- Provide ways to locate a replica for an object
- Objects modified through updates

OceanStore - Access Control

- Reader restriction:
  - Encrypt all data that is not public and distribute the key to users with read permission
  - Revoke read permission: the owner has to delete replicas or re-encrypt them with the new key

- Writer restriction:
  - Writes are signed; checked against an access control list (ACL)
  - An ACL entry specifies the privilege granted and the signing key, but not the explicit identity of the privileged users.

OceanStore - Distributed Routing

- Location independent addressing, using GUID
- Routing layer on top of IP, it is a two phase process:
  - Use a distributed data structure to route from node to node until a destination node is discovered (finding data location)
  - Route the message directly to the destination (routing)
- Benefits due to combining data location and routing:
  - Aggregate resources of many nodes help route a particular message. Limit the power of compromised nodes to deny service to a client.
  - Messages route directly to destination, avoiding the round-trips that a separate data location and routing process would incur
  - Infrastructure has up-to-date location information

OceanStore - Distributed Routing (cont.)

- Two-tiered approach for routing: a fast, probabilistic algorithm back up by a slower, reliable method
- Attenuated bloom filters
- Wide-scale distributed data location: just like Tapestry's

OceanStore - Updates

- Objects are modified via updates (versioning system) but data is not overwritten

- Lists of predicates associated with actions. If a predicate evaluates to true, the corresponding action is applied to the data object.
  - E.g. <road condition is bad?>, <no school today>

OceanStore - Introspection

- Used in cluster recognition and replica management
- Monitor system behavior

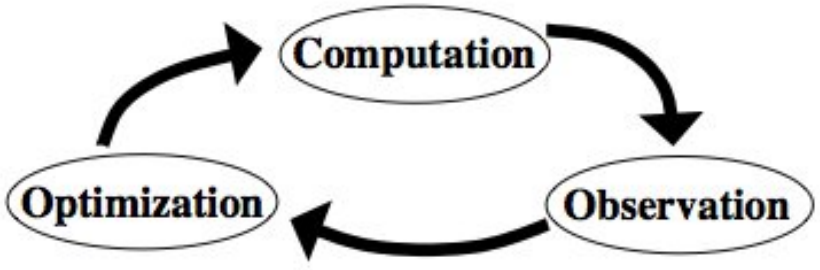


Figure 7: The Cycle of Introspection

Taken from OceanStore: An Architecture for Global-Scale Persistent Storage