DHT and DOLR Peer to Peer systems

Andrea Marin

Università Ca' Foscari di Venezia
Dipartimento di Informatica
Corso di Sistemi Distribuiti

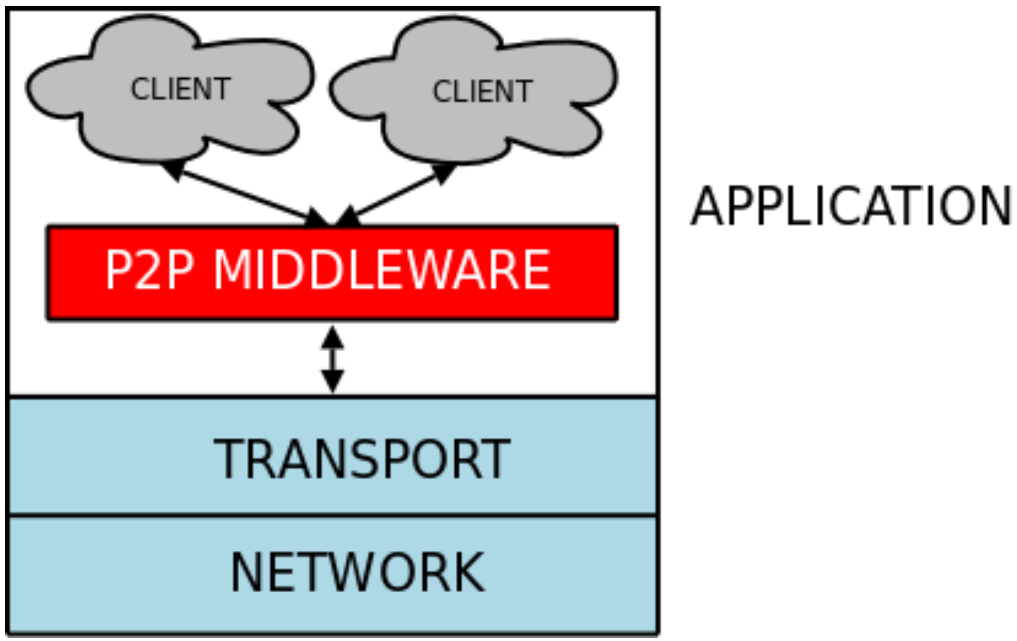
2009

Presentation outline

Routing overlays

Definition (Routing overlay)

The routing overlay is the algorithm of the P2P middleware that locates the nodes and the objects.



Salient features

- Objects are placed and relocated to any node without the client involvement
- A node can access a resource by routing the request through a sequence of nodes
- If multiple copies of a resource are stored, the routing overlay stores the location of any available replica
- A request is routed to the nearest node that provides a replica of the required resource
- Nodes and Objects are identified by GUIDs (*opaque identifiers*)

Routing overlay tasks

A client is expected to be able to...

1. specify a GUID and an operation to the routing overlay. This sends the request to a live node with a replica of the required resource
2. provide a new resource after computing its GUID
3. remove a resource
4. join or leave (maybe because of a failure) the network

Distributed hash tables: Main idea

- Distributed hash tables (DHT) are a family of algorithms
- Every object and node have a GUID
  1. GUID are usually computed with hash functions (e.g. SHA-1)
  2. Nodes with the same GUID are searched in the network to avoid name clashes
- When a client requires to publish a resource, it has to...
  1. compute the GUID
  2. ask the routing overlay to publish it
- When the routing overlay is asked to publish a resource, it...
  1. stores the resource in the node whose GUID is closest to that of the resource
  2. stores r replicas of the resources in the r nodes whose GUIDs are closest to that of the resource. r is the replication factor

Basic programming interface for DHT

DHT API

- put(*GUID*, *data*)
  - Publish an object with *GUID*. The *data* is stored in all the nodes responsible for a replica
- remove(*GUID*)
  - Remove all the replicas of the object whose GUID is *GUID*
- get(*GUID*)
  - The data associated with the object *GUID* are retrived

Computing the GUID distance

- Different approaches are defined, with pro and cons
- prefix routing: usage of a binary mask that selects an increasing number of hexadecimal digits from the destination GUID after each hop. Used by Patry.
- Numerical difference: used by CAN
- Exclusive OR: used by Kademlia

How to retrieve a resource GUID?

- GUID are not human readable
- Indexes of resource descriptions - GUIDs may be stored in a distributed way among the P2P network
- In practice these indices are often stored in web pages (like for BitTorrent)

Using DHT for file sharing

- DHTs may be used for file sharing applications (e.g. Kademlia with Emule)
- GUIDs are hashes of the shared resources and the associated data is the IP address of a client sharing that resource
  - When a node shares a file, it computes the GUID and then stores its own IP address in the r nearest nodes (to the GUID)

Distributed object location and routing (DOLR): Main idea

- Objects can be stored anywhere
- DOLR layer maintains a mapping between GUIDs and the addresses of the nodes at which replicas of the objects are located
- DOLR layer routes the requests to the nearest available replica
- Note that location of the replicas are decided outside the routing layer

Basic programming interface for DOLR

DOLR API

- publish(*GUID*)
  - Publish an object with *GUID*.
- unpublish(*GUID*)
  - Makes the object whose GUID is *GUID* inaccessible
- sendToObj(*msg*, *GUID*, *[n]*)
  - Sent a message *msg* to n replicas of object whose GUID is *GUID*