Algorithms and Programming IV
DHT-based P2P-Systems (20-3)

Summer Term 2020 | 29.06.2020
Claudia Müller-Birn, Barry Linnert
Peer-to-Peer

- Resources are shared between the peers.
- Resources can be accessed directly from other peers.
- Peer is provider and requester (servent concept).

### Unstructured Overlay
- Centralized P2P
- Pure P2P

### Structured Overlay
- DHT-based

- All features of P2P included
- Any terminal entity can be removed without loss of functionality
- No central entities
- Connections in the overlay are fixed
- Example: Paestry, Chord, CAN
Peer-to-peer systems

PASTRY
Introducing Pastry

• P2P overlay that is using Dynamic Hash Tables (DHT) with prefix-based routing with both peer ID and object ID.

• Prefix routing narrows the search for the next node along the route by applying a binary mask that selects an increasing number of hexadecimal digits from the destination GUID after each hop.

• It is originally developed Microsoft and Rice Uni but a free version (FreePastry) exists that is a prototypical implementation of Pastry. The latter is mostly used by scientific community.

• Similar algorithms are Chord and CAN.
Introducing Pastry (cont.)

• Any computer connected to the Internet and running PASTRY node software can be a PASTRY node.

• Application specific security polices may be applied.

• Each node is identified by a unique 128 bit node identifier (NodeId).
  − The node identifier is assumed to be generated randomly
  − Each NodId in is assumed to have the same probability of being chosen
  − Node with similar NodId may be geographically far
Mode of Operation of a Distributed Hash Table

- Every node stores and maintains part of the hash table.
- Every object/resource has a (hash) key which is stored at the node responsible for its key.

**lookup(key)** -> node or data directly

Lookup where it is stored
Distributed Hash Table: Steps of Operation

1. Mapping of nodes and data same address space
   - Peers and content are addressed using flat identifiers (IDs)
   - Common address space for data and nodes
   - Nodes are responsible for data in certain parts of the address space
   - Association of data to nodes may change since nodes may disappear

2. Storing / Looking up data in the DHT
   - “Look-up” for data = routing to the responsible node
   - Responsible node not necessarily known in advance
   - Deterministic statement about availability of data
Sketch of the Routing Algorithm

- Assume we want to find the node in the PASTRY network with the NodeId closest to a given key.

Routing Idea

In each routing step, a node normally forwards the message to a node whose NodeId shares with the key a prefix that is at least one digit longer than than the key shares with the present node. If such a node is not known, the message is forwarded to a node that shares the same prefix of the actual node but its NodeId is numerically closer to the key.
First Four Rows of a Pastry Routing Table

<table>
<thead>
<tr>
<th></th>
<th>GUID prefixes and corresponding nodehandles n</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F</td>
</tr>
<tr>
<td></td>
<td>n   n   n   n   n   n   n   n   n   n   n   n   n   n   n   n</td>
</tr>
<tr>
<td>1</td>
<td>60  61  62  63  64  65  66  67  68  69  6A  6B  6C  6D  6E  6F</td>
</tr>
<tr>
<td></td>
<td>n   n   n   n   n   n   n   n   n   n   n   n   n   n   n   n</td>
</tr>
<tr>
<td>2</td>
<td>650 651 652 653 654 655 656 657 658 659 65A 65B 65C 65D 65E 65F</td>
</tr>
<tr>
<td></td>
<td>n   n   n   n   n   n   n   n   n   n   n   n   n   n   n   n</td>
</tr>
<tr>
<td>3</td>
<td>65A0 65A1 65A2 65A3 65A4 65A5 65A6 65A7 65A8 65A9 65AA 65AB 65AC 65AD 65AE 65AF</td>
</tr>
<tr>
<td></td>
<td>n   n   n   n   n   n   n   n   n   n   n   n   n   n   n   n</td>
</tr>
</tbody>
</table>
Pastry Routing Example

Source NodeID 65A1FC

Destination NodeID D46A1C

0 FFFFF....F (2^{128})
Peer-to-Peer

- Resources are shared between the peers.
- Resources can be accessed directly from other peers.
- Peer is provider and requester (servent concept).

Unstructured P2P

- Centralized P2P
- Pure P2P
- Hybrid P2P

Structured P2P

- DHT-based
  - Example: Pastry, Chord, CAN

1st Generation
- Example: Napster
- Example: Gnutella 0.4, Freenet

2nd Generation
- Example: Gnutella 0.6
References


• Smruti R. Sarangi: Distributed Hash Tables Pastry (http://www.cse.iitd.ernet.in/~srsarangi/csl860/docs/pastry-lec.pdf)

Routing in Unstructured Overlay Networks: Flooding vs. Random Walk