Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems by Antony Rowstron and Peter Druschel

CSC 724 Paper review - Vaibhav Singh, vsingh7

January 22, 2019

1 Summary

The paper presents Pastry, a distributed object location and routing protocol for peer-to-peer applications. The routing algorithm for Pastry takes into account network locality; it seeks to minimize the distance messages travel using proximity metrices like the number of IP routing hops.
The paper then shows Pastry to be completely decentralized, scalable and self-organizing, and is able to react automatically to arrival and departure of nodes. Finally, the paper provides experimental results which show Pastry prototype implementation to scale to an emulated network of 100,000 nodes.

2 Description

Each node in a Pastry network has a unique nodeid. When a message is sent to a node, the node routes the message to the node with nodeid closest to the id of the message. The number of routing steps required for this is O(logN).

The nodeids in each Pastry network form a ring, from 0 to 2pow(128) - 1. Each node is assumed to be provided an id uniformly across the ring, to ensure nodes having similar ids are diverse in geography, jurisdiction, ownership etc. Each node maintains a routing table, a neighbourhood set and a leaf set. In the routing table, each node maintains the IP addresses of one of many nodes whose nodeid have the appropriate prefix.

When a message hits a Pastry node, the node checks if the key falls within the range of nodeids in its leaf set (set of L nodes which have the closest nodeid to that node), if yes, the message is forwarded directly to the node which has its key most closely matching the message's key. If the key is not covered by leaf set, the node then tries to find a node which shares a common prefix with the key by at least one more digit. If in the rare case the appropriate entry is empty, the node passes the message to a node which matches the message id by atleast one more bit than itself.

The smaller the number of bits which match with the message, the more the number of candidates which can be used to route the message to, and as pastry prefers nodes with the smallest IP routing hops, it is generally better to fetch a node which has a small number of matching bits in its nodeid.

3 Strong Points

Pastry accounts for network locality, i.e. it seeks to minimize the distance messages travel by calculating IP routing hops.
Pastry uses consistent hashing to generate nodeids, and hence, is automatically load balanced.

4 Weak Points

A malicious or failed node (close to the data) could make queries to be repeated several times by the client, and tolerating this behaviour has not been covered in the paper.
If certain nodes are unreachable to certain other nodes, the pastry ring could get partitioned to two or several isolated rings, and this behaviour could persist even after the nodes start talking to each other after some time.

5 Improvement

O(logN) lookup cost can be optimized further