Name: Prashanth Mallyampatti
Unity ID: pmallya
Student ID: 200250501

CSC724- Advanced Distributed Systems
Paper Review

Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems

Summary:

This paper presents PASTRY, a scalable, distributed P2P location and routing substrate. Earlier P2P applications used centralized database such as Napster. Later attempts to decentralize it came up by storing keys in each node for communication. Challenge was to come up with an efficient algorithm for the data lookups and routing. Pastry is one such (like Chord) substrate providing the required functionalities.

In PASTRY, every node has a unique 128-bit nodeId, which is assigned randomly when a node joins. The existing ids are uniformly distributed. Each Pastry node maintains a leaf set, a routing table, and a neighborhood set. Numerically closer nodes are maintained by the leaf set, using the prefixes from nodeId routing table organizes the routing information, wherein, both these sets handle message routing and the neighborhood set contains the IP addresses and ids that are physically close to the node under consideration and maintains locality properties. During routing, a node will forward a message to a node whose identifier is at least one digit longer than the current node's data key, else message is forwarded with to the node with same prefix but whose identifier is numerically closer the data key.

Like Chord, Pastry also guarantees that the number of hops is not more than log N. The routing algorithm checks for the key which is the leaf set, if found, forwards the message to the node which has numeric identifier value closest to the key. If not found, it is sent to the closest entry in the routing table. For the purpose of better network proximity (locality properties), Pastry updates its routing table based on the information from the nodes in the neighborhood set. The authors also discuss about how to efficiently and dynamically maintain the state of the node when a new node arrives/departs/fails. Experiment results shows that Pastry has an acceptable deviation from the optimal routing, with an average of 4 hops in a network of 10,000 nodes to reach the destination.

Strong Points:

1) Flexibility factor is greatly improved upon by not imposing rigid restrictions on the identifiers, and also adjusting the nodes in the routing table which accounts for network locality.
2) Failures, new node arrivals are handled well- Pastry maintains a near optimal routing even after failure of 10% of the nodes.
3) Pastry allows applications to easily integrate it into their code base.
4) About malicious hosts paper suggests possible solutions to them, whereas Chord doesn't
5) Through replication and decentralization, Pastry ensures fault resiliency.

Weak Points:

1) The assumption of nodes being uniformly distributed is never the case, they tend to cluster, so most of the locality claim is lost.
2) Formal proofs (like Chord) for routing performance has not been discussed, which I feel if done would have proved (such as proximity performance) the claims to be correct.
3) The chance of new node having the same nodeId as existing node, imposes: same nodeId being assigned for more than one node, and routing algorithm not able to see the new node as its considered as current node (due to same nodeId), thus no node to forward the message problems.