CSE 535 - Asynchronous Systems
Project Proposal
Improving and Evaluating Pastry implementations in DistAlgo

Team:-
Akhil Bhutani (110898687)
Nikhil Navadiya (112046312)
Vivek Kumar Sah (112071655)

Background:

Emerging applications such as data warehousing, multimedia content distribution etc are growing at 3x to 5x per year. Such applications require scalable, highly-available and cost-effective storage systems. To solve this problem DHT(Distributed Hash Table) is used and Pastry is one of the the routing implementations of DHT.

Pastry is a scalable, distributed object location and routing substrate for wide-area peer-to-peer applications (Antony Rowstron et al.). Pastry performs application-level routing and object location in a potentially very large overlay network of nodes connected via the Internet and can be used to support a variety of peer-to-peer applications, including global data storage, data sharing, group communication and naming. And with Pastry's ability to realize the scalability and fault tolerance it is widely suited for peer-to-peer application.

Problem Description:

There are various implementations of Pastry present in different languages on the web but we couldn't find implementation of Pastry in DistAlgo. So we plan to implement Pastry in DistAlgo as per (Antony Rowstron et al.) and try to come with optimized implementation by comparing, improving and evaluating other implementations. Our goal is to find and report errors in the previous implementations and implement a bug-free optimized version in DistAlgo.

INPUT	OUTPUT
Pastry implementations in various languages including DistAlgo and as described in (Antony Rowstron et al.) We will have a PastryNode class with the following parameters :- ● routingTable ● leafSet ● neighborhoodSet ● ipAddress ● Nodeid ● Coordinates ● B, by default 4 ● keySize, by default 128 The PastryNode class represents the attributes for every pastry node process mentioned in the paper.	Best Pastry Implementation in DistAlgo with improvised version of (Antony Rowstron et al.). This implementation will closely follow the paper presented by Antony and Peter in 2001. Some of the correctness / performance test will be carried out as part of the implementation to make sure that the paper is closely followed. Some of the correctness metrics include: 1. Reachability - If src and dst node is available, then the message should be routed among them. If this happens for a fixed number of trials, we can safely assume correctness property holds. 2. Hop count - Expected number of hop count in a Pastry network of N nodes should be $\mathbf{log_{2^b}N}$. Some of the performance metrics include: 1. Routing performance - Average number of hops with / without failures. 2. Probabilistic evaluation for number of hops. 3. Average initialization time for Pastry network. 4. Average node join time for Pastry network. 5. Average lookup time with / without failures.

Motivation:

The primary motivation of this project is to understand the Pastry implementation [1] and implement an optimized version for the same in DistAlgo. The goal is closely follow the implementation by Antony and Peter as the current implementations are not closely followed and deviate from the actual paper in one way or the other. We also intend to compare our implementation using various comparison and performance metrics described in the paper and test performance using metrics like running times, average routing hops etc. We chose DistAlgo as programming language as all other implementations are coded in programming languages like C, C++, Java, Python etc. We weren't able to find an implementation for DistAlgo. Also, as

DistAlgo is a language for distributed systems, we feel it will do justice to implement the algorithm presented in the paper [1] in DistAlgo. Also, based on perusing through current implementations, we don't feel that any implementation closely follows the paper and hence, we feel there is scope for improvement.

Task Summary:

- Understand Pastry implementation from paper. [1]
- Understand Pastry implementations found in web. [2][3][4][5][6]
- Implement an optimized Pastry algorithm in DistAlgo, closely following the paper. [1]
- Verify correctness, measure performance and evaluate our implementation based on metrics like node initialization time, time taken to route join request, time taken for lookup etc.

  SubTasks:
  - Read the actual Pastry implementation paper thoroughly. [1] {Akhil Bhutani, Nikhil Navadiya, Vivek Kumar Sah}
  - Understand Pastry implementation in Java [2][4] {Nikhil Navadiya}
  - Understand Pastry implementation in Python [5][6] {Vivek Kumar Sah}
  - Understand Pastry implementation in C++ [3] {Akhil Bhutani}
  - Design and implement an optimized Pastry algorithm in DistAlgo. {Akhil Bhutani, Nikhil Navadiya, Vivek Kumar Sah}
  - Evaluate algorithm using Correctness and Performance metrics as described above.

State of Art:

We searched for some implementation and found the following implementation in various languages:

- https://github.com/ctebbe/P2P-Pastry-Implementation [Java]
- https://github.com/hamersaw/BasicPastry/tree/master/src/main/java/com/hamersaw/basic_pastry [Java]
- https://github.com/ngrj93/Pastry [C++]
- https://github.com/kapilthakkar72/Pastry-and-Chord-DHT [Python]
- https://github.com/MuxZeroNet/pastry [Python]

Professor mentioned another implementation which was implemented by her student Ken Koch in 2012. The source for the implementation is:

- https://github.com/unicomputing/pastry-distalgo-2012-Ken-Koch

We found some real-time applications of Pastry algorithm.

SCRIBE: SCRIBE is a generic, scalable and efficient group communication and event notification system. It provides application level multicast and anycast. Scribe is efficient,

self-organizing, flexible, highly scalable and supports highly dynamic groups. It is built on top of Pastry, a generic, scalable, self-organizing substrate for peer-to-peer applications.

PAST: PAST is a large-scale, peer-to-peer archival storage utility that provides scalability, availability, security and cooperative resource sharing. Files in PAST are immutable and can be shared at the discretion of their owner. PAST is built on top of Pastry, a generic, scalable and efficient substrate for peer-to-peer applications.

SQUIRREL: SQUIRREL is a fully decentralized, peer-to-peer cooperative web cache, based on the idea of enabling web browsers on desktop machines to share their local caches. This results in an efficient, highly scalable, self-organizing and fault-tolerant web cache without need for dedicated hardware or the corresponding administration cost. It exhibits performance comparable to a centralized web cache in terms of hit ratio, bandwidth usage and latency, while while imposing negligible overhead on the participating nodes.

SplitStream: SplitStream is a high-bandwidth content streaming/distribution system that is built upon Pastry.

POST: POST is a generic messaging infrastructure that is built upon Pastry. It is being used to support services like secure email (ePOST), secure instant messaging (imPOST), and collaborative applications like shared calendars, notes and whiteboards without the need for dedicated servers.

Scrivener: Scrivener is an architecture to enforce fair sharing of resources in peer-to-peer systems. It provides economic incentives for nodes to follow the rules, resulting in a system with increased resistance to selfish participants.

Project Plan:

Below is our weekly project goals:
Week 1: Study Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems paper by Antony Rowstron and Peter Druschel. [1]
Week 2: Design and develop modules for Pastry implementation in DistAlgo. Carry out necessary optimizations wherever possible.
Week 3: Test, evaluate and compare our implementation with the paper. [1]
Week 4: Optimize code, generate performance matrix, write report and prepare presentation.

References:

1. https://link.springer.com/chapter/10.1007/3-540-45518-3_18
2. https://github.com/ctebbe/P2P-Pastry-Implementation
3. https://github.com/ngrj93/Pastry

4. https://github.com/hamersaw/BasicPastry/tree/master/src/main/java/com/hamersaw/basic_pastry
5. https://github.com/kapilthakkar72/Pastry-and-Chord-DHT
6. https://github.com/MuxZeroNet/pastry
7. https://drive.google.com/file/d/0B0MWH8ngLAIFTzFna1lOTTF4UlE/view
8. https://sites.google.com/site/distalgo/