COL819: Programming Assignment 1

Implement Pastry and Chord [1]

Download PDF

Logistics

1. Release date: **28 Jan 2020**
2. Due date: **29 Feb 2020**
3. Maximum marks: 100
4. Individual assignment
5. Submissions on Moodle: Code and report has to submitted.
6. Languages: Java, C, C++, and Python3.X
7. **Report must be in Latex and use vector graphics for images.**

1 Pastry [3] and Chord [2] [Total 30 Marks each]

In this part of the assignment, you need to implement/simulate a *Pastry* and a *Chord* network. Both of them have to implemented separately.
The implementation has to support some key aspects of the algorithm such as addition of nodes, deletion of nodes, and look up queries for data. Along with this, there has to be mechanisms for *addition* of key value (data) pairs in the network.

1.1 Pastry routing table [10 Marks]

Along with these mandatory aspects, your implementation should also support printing of the *routing table* of a given node, which can be identified by using the *node id*. Node id is a unique number which will be assigned to each node in the network. Along with the routing table of a node, a summary of the complete network has to be printed:

```
Total number of nodes : XX.
Total number of data elements : XX.
Total search queries : XX.
Total node add queries : XX.
Total node delete queries : XX.
Total data add queries : XX.

Routing table. ID: XX

|----------------------|
|S.No.|Target|Successor|
|----------------------|
|  1  | XX   | Node X  |
|  2  | XX   | Node X  |
|  3  | XX   | Node X  |
|  4  | XX   | Node X  |
|  5  | XX   | Node X  |
.
.
.
|----------------------|
```

1.2 Chord: Path taken by a look up request

In case of Chord, you need to print the path taken by a particular look up query. These print statements can also be used as a log to debug the Chord implementation. There should be a mechanisms to enable and disable the logs.

Look up XX : $NodeId_0 \rightarrow NodeId_1 \rightarrow NodeId_2 ... \rightarrow NodeId_i \rightarrow ... NodeId_M$

where, $NodeId_0$ was the first node in the path, and the path consists of M nodes.

1.3 Evaluation [20 Marks]

You need to evaluate the performance of your network for different number of nodes in the network. There will be three configurations, with number of nodes as 100, 500 and 1000. For each configuration you need to perform 1 million of random search queries, after populating the network with 10,000 data points. You need to report following for each of the configurations:

- Average number of hops for search queries.
- A histogram showing the distribution of the hops required.

- Operation: **Delete half of the nodes from the network, randomly.**
- Average number of hops for search queries in this reduced set.
- A histogram showing the distribution of the hops required in this reduced set.

2 Report [Total 40 Marks]

There will be a single report, which must be in PDF format.

2.1 Experiments [15 Marks]

The report should contain:

- Experiment setting (number of nodes, search queries etc.)
- Graphs with clear labels, x-axis, y-axis and caption for the graph.
- Few lines about the behavior observed.
- Justification of the trends observed.
- The routing table of a single node of your choice for Pastry, and Path taken by 10 requests (of your choice) for Chord.

This has to be done for both, *Pastry* and *Chord*.

2.2 Performance comparison [15 Marks]

The report should also contain a performance comparison of both the algorithms, and few lines about this.

2.3 How to run the code [10 Marks]

The report should also contain details on how to run the code, along with few screenshots.

General guidance

- Please stick to basic packages during implementation.
- If you are not sure if a particular package is allowed, ask on Piazza.
- Grading will be done based on the report. There might be a demo, if required. So please ensure that the submitted code executed on your machine correctly.
- We will run **MOSS** on the submissions. Anyone found with a copied code, either from Internet or from another student, will be dealt with as per the class policy.

References

[1] Col 819: Distributed systems. http://www.cse.iitd.ac.in/~srsarangi/courses/col_819_2020/index.html. (Accessed on 01/27/2020).

[2] paper-ton.dvi. https://pdos.csail.mit.edu/papers/ton:chord/paper-ton.pdf. (Accessed on 01/27/2020).

[3] pastry.pdf. http://rowstron.azurewebsites.net/PAST/pastry.pdf. (Accessed on 01/28/2020).

[4] Selected dht algorithms: Chord and pastry. https://www.cs.tut.fi/kurssit/ELT-53206/lecture03.pdf. (Accessed on 01/28/2020).