Distributed Systems Project, Spring 2015 – Assignment 3

Assignment

In this assignment, you are given 1024 nodes. Each node has a unique ID (e.g. 0, 1, 2, 3 … 1023) and maintains a routing table of certain size in order to communicate with each other.

You are required to organize the given nodes in a certain way to construct a communication overlay. You can freely choose way you like to organize them, but there are some constraints on the design of the overlay (see Grading Section). You need to design the overlay, analyze its communication complexity, and experimentally verify the complexity on the Ukko cluster.

Requirements

1. The overlay must be able to deliver a message correctly between any two nodes (e.g. deliver a message from node x to node y).
2. Node **must only** use the routing table it maintains for communication. Broadcast is not allowed. Hard-coding everything in the code is not allowed.
3. Implement the overlay and run it on the Ukko cluster.
4. Experimentally determine values for maximum routing table size and the average number of hops to deliver a message between any two nodes.

Documentation

In the documentation, you are required to report the following things:

1. Describe **how** you organize the nodes, and **why** you choose that way. If you used any references, cite the references in the documents.
2. Calculate the average hops to deliver a message between any two nodes in your overlay, from both experimental and theoretical perspective.
3. Calculate the minimum, maximum and average size of the routing table of all the nodes in the overlay.
4. Analyze the pros and cons of your design, and what is best context it can be applied to.
5. Present results of your experimental verification and compare them with your calculations

Grading

Suppose the average number hops for delivering a message between any two nodes in your overlay is H and the maximum routing table over all the nodes is R. The metrics we use to evaluate your solution are as follows:

1. Quality and functionality of the code.
2. Quality of the documentation.
3. The product H * R. The product must be below 387 to pass the assignment (for your 1024 node overlay). Higher values result in a failed grade. Values below 387 do not affect your grade.

Guidelines

Groups of up to 3 people are allowed. Every group must return their own implementation. You can of course discuss any problems you encounter with other groups, but sharing code is not allowed and if found, will be considered as plagiarism.

If you work in a group, every group member **MUST** turn in a short individual report describing what their contribution to the project was and describe how to work in the group was organized.

You are free to choose any programming language, but we recommend using a higher level language, e.g., Ruby or Python, even if you have to learn the language from scratch during the assignment.

You can either design your own overlay or pick an existing overlay, but if you use existing overlays, make sure they meet the criteria for the product H*R.

You can assume that all nodes start at the same time and are aware of each other. You may need to implement some tweaks to guarantee the simultaneous start. Fault tolerance against crashes is not required, but obviously frequent crashes will make experimental evaluation hard or impossible.

Deliverables

Program source code with documentation. The document should meet the requirements in the Documentation Section.

Timeline

The assignment is due on March 8th at 20:00. No extensions will be given.

Return

Return your code and documentation by email to Liang.Wang@cs.helsinki.fi as one tar-archive. Please indicate clearly the names and student IDs of every group member in every source code file.

The individual reports about group work should be sent individually to both Liang.Wang@cs.helsinki.fi and Jussi.Kangasharju@helsinki.fi. They must not be included with the code.