

COP5615- Distributed Operating System Principles
Fall 2023
Programming Assignment #3
Gossip Algorithm

Deadline: Thursday, November 30th, 2023

LATE submission will not be accepted for grading!

How to submit: Please complete the group submission via CANVAS system.

Introduction

Please read ahead on Chapter 6 of textbook (up to Section 6.4.4) for the Gossip type algorithm.

Gossip type algorithms can be used for both **group communication and aggregate computation**. The goal of this project is to determine the convergence of such algorithms through a simulator based on actors written in F# using the Akka framework. Since actors are fully asynchronous, the particular type of Gossip implemented is the so-called *Asynchronous Gossip*.

Gossip Algorithm for information propagation

The Gossip algorithm involves the following:

- **Starting:** A participant(actor) told/sent a rumor (fact) by the main process
- **Step:** Each actor selects a random *neighbor* and tells it the rumor.
- **Termination:** Each actor keeps track of rumors and how many times it has heard the rumor. It stops transmitting once it has heard the rumor 10 times (10 is arbitrary, you can select other values).

Push-Sum algorithm for sum computation

- **State:** Each actor A_i maintains two quantities: s and w . Initially, $s = x_i = i$ (that is actor number i has value i , play with other distribution if you so desire) and $w = 1$
- **Starting:** Ask one of the actors to start from the main process.
- **Receive:** Messages sent and received are pairs of the form (s, w) . Upon receiving, an actor should add the received pair to its own corresponding values. Upon receiving, each actor selects a random neighbor and sends it a message.
- **Send:** When sending a message to another actor, half of s and w is kept by the sending actor, and half is placed in the message.
- **Sum Estimate:** At any given moment of time, the sum estimate is s/w where s and w are the current values of an actor.
- **Termination:** If an actor's ratio s/w did not change more than 10^{-10} in 3 consecutive rounds the actor terminates. **WARNING:** the values s and w independently never converge, only the ratio does.

Topologies: The actual network topology plays a critical role in the dissemination speed of Gossip protocols. As part of this project, you have to experiment with various topologies. The topology determines who is considered a neighbor in the above algorithms.

- **Full Network:** Every actor is a neighbor of all other actors. That is, every actor can talk directly to any other actor.
- **2D Grid:** Actors form a 2D grid. The actors can only talk to the grid neighbors
- **Line:** Actors are arranged in a line. Each actor has only 2 neighbors (one left and one right, unless you are the first or last actor).
- **Imperfect 3D Grid:** Grid arrangement but one random other neighbor is selected from the list of all actors (8+1 neighbors).

Requirements:

Input: The input provided (as a command line to your project3) will be of the form:

dotnet run numNodes topology algorithm

eg :dotnet run 20 line gossip

Where numNodes is the number of actors involved (for 2D-based topologies you can round up until you get a square), topology is one of full, 2D, line, imp3D, the algorithm is one of gossip, push-sum.

Output: Print the amount of time it took to achieve convergence of the algorithm.

Actor modeling: In this project, you have to use exclusively the actor facility in F#. (projects that do not use multiple actors or use any other form of parallelism will receive no credit).

README File

In the README file you have to include the following material:

- Team members
- What is working
- How to run your program?
- What is the largest network you managed to deal with for each type of topology and algorithm?
- For each type of topology and algorithm, draw the dependency of convergence time as a function of the size of the network.
- You can overlap different topologies on the same graph, i.e. you can draw 4 curves, one for each topology, and produce only 2 graphs for the two algorithms.

Submission Guidelines

1. The project folder should be called Gossip. It should contain a Program.fs file which would be the entry point of your program and some other configuration files which are included when the project is created.
2. Do not include any executable files in your submission.
3. Include the report in .pdf or .txt format. Name it report.pdf/txt.
4. Zip all your files into a packet: Team_ID.zip
5. Upload the zip packet as attachment in CANVAS before deadline.

Your project structure should be like this:

- **Team_ID.zip**
 - **Gossip**
 - **Program.fs**
 - **Gossip.fsx**
 - **<Other configuration file and folders>**
 - **Report.pdf**

Grading Criteria

Correct Implementation / Outputs	80%
Report	15%
Readability / Comments / Code structure	5%
Total	100%