FAF.PTR16.1 Spring 2023
Project 1: Stream Processing with Actors

Handed out:	Monday, March 6, 2023
Due:	Friday, April 14, 2023

General Requirements

Compared to the previous Project, all weeks for this one aim to build upon the same application. The goal is to finish the Project with a more or less functional stream processing system.

Since you will be working on a complex application, each presentation will now require you present 2 diagrams: a Message Flow Diagram and a Supervision Tree Diagram. The Message Flow Diagram describes the message exchange between actors of your system whereas the Supervision Tree Diagram analyzes the monitor structures of your application.

Every task you work on should be easily verifiable. Your system should provide logs about starting / stopping actors, auto-scaling / load balancing workers and printing processed tweets on screen.

P1W1

Minimal Task Initialize a VCS repository for your project.

Minimal Task Write an actor that would read SSE streams. The SSE streams for this lab are available on Docker Hub at `alexburlacu/rtp-server`, courtesy of our beloved FAFer Alex Burlacu.

Minimal Task Create an actor that would print on the screen the tweets it receives from the SSE Reader. You can only print the text of the tweet to save on screen space.

Main Task Create a second Reader actor that will consume the second stream provided by the Docker image. Send the tweets to the same Printer actor.

Main Task Continue your Printer actor. Simulate some load on the actor by sleeping every time a tweet is received. Suggested time of sleep – 5ms to 50ms. Consider using Poisson distribution. Sleep values / distribution parameters need to be parameterizable.

Bonus Task Create an actor that would print out every 5 seconds the most popular hashtag in the last 5 seconds. Consider adding other analytics about the stream.

P1W2

Minimal Task Create a Worker Pool to substitute the Printer actor from previous week. The pool will contain 3 copies of the Printer actor which will be supervised by a Pool Supervisor. Use the one-for-one restart policy.

Minimal Task Create an actor that would mediate the tasks being sent to the Worker Pool. Any tweet that this actor receives will be sent to the Worker Pool in a Round Robin fashion. Direct the Reader actor to sent it's tweets to this actor.

Main Task Continue your Worker actor. Occasionally, the SSE events will contain a "kill message". Change the actor to crash when such a message is received. Of course, this should trigger the supervisor to restart the crashed actor.

Bonus Task Continue your Load Balancer actor. Modify the actor to implement the "Least connected" algorithm for load balancing (or other interesting algorithm). Refer to this article by Tony Allen.

Good Luck!