Homework 9: akka Actors

CSYE 7215: Fall 2018

The objective of this project is to develop an akka program for counting the percentage of vowels in English text files submitted to the agent system, estimating the percentage of vowels in the "next" file submitted to the program, and computing the average accumulated error for each of the estimators. The counting and estimating will be performed by different agents – one Counter and two Estimators. First a file is submitted to all three agents. The estimating agents estimate (predict) the percentage of vowels in the "next" file; the steps (or files) are indexed by t. The estimation is based on the feedback received from the Counter agent and on an internal parameter, g.

For this project you can assume, for simplicity, that the vowels are: A, E, I, O, U and Y.

Estimation algorithm:

- Compute the values of C(t) and S, as specified below:
- C(t) – estimator's percentage at step t
- U(t) – feedback from Counter at step t
- g – the "learning" parameter; the value should be between 0 and 1.

1. C'(t) = C(t) + U(t) – actual percentage at step t
2. C(t+1) = g C(t) + (1 – g) C'(t) - updated percentage; to be used in step (t+1)
3. S = [U(0) + U(1) + … + U(n)]/n – average value of estimator's errors over n steps

Counter algorithm:

1. Count the number of vowels, V(t), in the file, t.
2. Count all letters, L(t), in the file.
3. Compute (actual) percentage of vowels in the file: P(t) = V(t) / L(t).
4. Read C(t) from each Estimator.
5. Compute feedback: U(t) = P(t) – C(t).
6. Send feedback to the Estimators.

You will be given a collection of text files that will need to be installed on your computer. Your program will create a "User" actor whose main goal will be to access the directory in which a number of text files are stored and get their access information, and then User will create three child actors. Name the first child actor Estimator1; the second – Estimator2; the third - Counter.

The User actor will read files, one by one, from the directory and pass the file to Estimators and Counter. The Counter will count the number of vowels in the text it receives from User and provide feedback to each of the Estimators. Estimators will compute new values of C(t+1) and will send their result to Counter, as described above. Counter will count the vowels, compute the percentage of the vowels in the file and send feedback, U, back to each of the Estimators.

The results should be sent back to the User agent. The User agent should display the results. Note that the processing is asynchronous and thus one of the Estimators may be able to finish sooner than the other. The sequence in which Estimators submitter their results for each of the steps should be visible in the outputs printed by User. However, it is not expected that one Estimator will process all of the files before the other.

The Estimators should initiate the interactions with the Counter. The first time around, the Estimators should use an initial value of C – you can pick this default.

Note that the Estimator algorithm uses the parameter g. It represents the "weight" parameter, i.e., how much the old value influences the new value of C vs. how much the old value influences C. You should experiment with the selection of this value and set it to the one that you think is best, i.e., the one that gives lower error rates – S.

In the development of this program you should use the Sequence Diagrams, as shown in the lecture slides. You can develop these diagrams in any drawing tool, e.g., PowerPoint or Visio. Or you can use a UML tool for that (all UML tools support creation of Sequence Diagrams). An example of a free UML tool is Star UML http://staruml.io/.

You are free to use an existing Java library for vowel counting; just specify explicitly where you downloaded it from and provide appropriate credits.

You will be given skeleton files which you will use for your project. You will also need to install akka on your system, as described in the lecture slides or in the akka documentation:

http://doc.akka.io/docs/akka/2.4/java.html#java-api

Submission

Submit a .zip file containing your project files to Assignments in Blackboard. The zip file you submit should have your name as the author. The zip file should also include a small document that will include the sequence diagram.

The files we gave you are just for your testing. We will test your programs on a different set of files. So please make sure that the program is not fixed on any specific set of file names; just let it read the directory and then read them. You can assume that that directory will contain any other files.

To enforce academic integrity, your code may be checked for similarity to other submissions.