

University of Wolverhampton

School of Mathematics and Computer Science

5CS022 Distribute and Cloud Systems Programming

Workshop 4 The Akka Framework Part 2

Overview

This workshop continues the exploration of the Akka Framework from the previous workshop.

Part 1 – Minimal sample program

1. Download the sample project

Go to the Canvas topic for 5CS022 Distributed and Cloud Systems Programming, and download either the zip file “akka-hello(Eclipse).zip” if you are using Eclipse, or “akka-hello(IntelliJ).zip” if you are using JetBrains IntelliJ.

2. Import the project into Eclipse/IntelliJ

Use the steps in the previous workshop to import the "akka-hello" project into your chosen IDE.

3. Build and run the program

Make sure that the minimal Akka program builds and runs correctly. If it does, you should see output similar to :

```
> Task :Main.main()
```

```
[2021-03-02 10:33:02,796] [INFO] [akka.event.slf4j.Slf4jLogger]  
[default-akka.actor.default-dispatcher-5] [] - Slf4jLogger started
```

Press ENTER to end the program.

```
[2021-03-02 10:33:03,008] [INFO] [akka.actor.ActorSystemImpl]  
[default-akka.actor.default-dispatcher-6] [ActorSystemImpl(akka://default)] - Creating  
Actor A
```

```
[2021-03-02 10:33:03,022] [INFO] [com.example.ActorA]  
[default-akka.actor.default-dispatcher-5] [akka://default/user/$a] - Actor A received  
Message A : 'Hello!' from 'Actor[akka://default/user/$a#508038483]'
```

This minimal program will form the basis of all of the programs that you will work with for this workshop.

Part 2

1. The Actor class ActorA in the sample Akka program currently responds to only one message object – MessageA. Modify the createReceive() method so that it will also respond to any other message and print out the message on the standard output.
2. Every message in Akka is a Java object. However, not all messages need to have a custom Java class created for them. Convert the sample program so that it can respond to messages that contain the primitive data types such as byte, short, int, long, float, double, boolean, and char, without having to create custom Java

classes.

3. In standard multithreading programs (for example Pthread programs), shared-resource contention (e.g. global variables) can be an issue and requires the use of mutexes and critical sections. Demonstrate that with Akka Actor, this is not an issue, by creating a "Counter" Actor class to keep track of a global counter, and lots of instances(20) of "ActorA" objects to send messages to "Counter" to increment the global counter.
4. Create 2 Akka Actor classes "ActorA" and "ActorB" to demonstrate the Akka API `setReceiveTimeout()`. ActorA will generate a random integer number from 1 to 5 in a loop for 100 times, and send this integer as a message to ActorB, which would then call `Thread.sleep()` that many seconds. Set the receive timeout to 2 seconds, and when the timeout triggers, send ActorB a `stop()` message, and then create a new instance of ActorB to process the next number.

Assessed Task

1. Create 3 Akka Actor classes called "Producer", "Supervisor" and "Worker". The "Producer" will generate 1000 random long integer numbers between 10000 and 100000. The "Producer" will send each number as a message to "Supervisor".

At start-up, the Supervisor will create 10 "Worker" Actors. When the "Supervisor" receives a number from the "Producer", it will use the API `forward()` to forward that message to one of the "Worker" actors, in a round-robin fashion.

The "Worker" actor will determine if the number in the message is a prime number. If it is a prime number, it will then send a string/text message to the "Producer", saying that "The number XXX is a prime number." And the Producer will print out the message on the standard output.

When the 1000 numbers have been produced and checked, the "Producer" actor will terminate the Actor system.

Export your project as a Zip file (either IntelliJ or Eclipse) and submit this program as "**workshoptask4.zip**" as part of your final portfolio submission. You can also upload it to the formative submission point for formative feedback.