



POLITECNICO
MILANO 1863

Actor Model – Akka Evaluation Lab

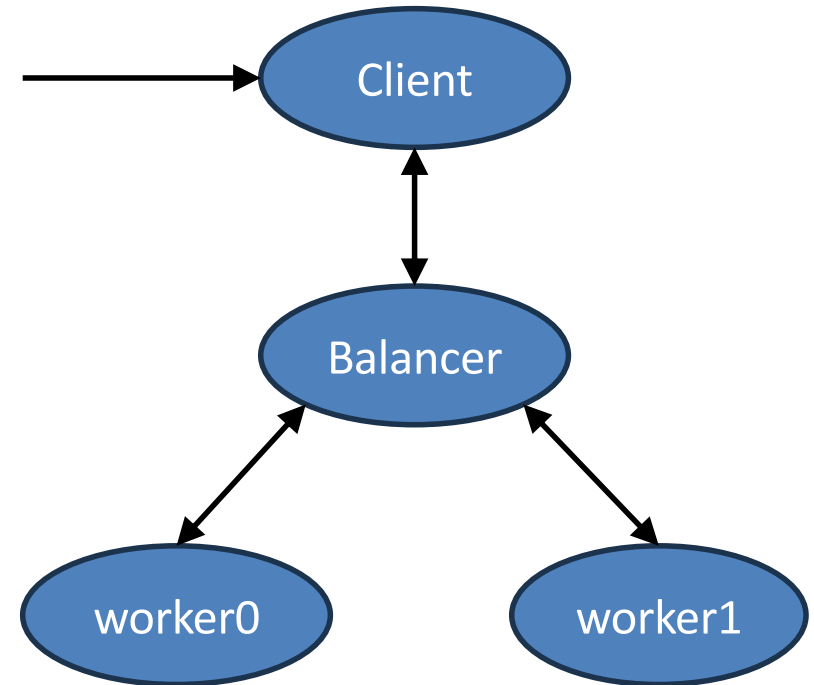
Luca Mottola

`luca.mottola@polimi.it`

`http://mottola.faculty.polimi.it`

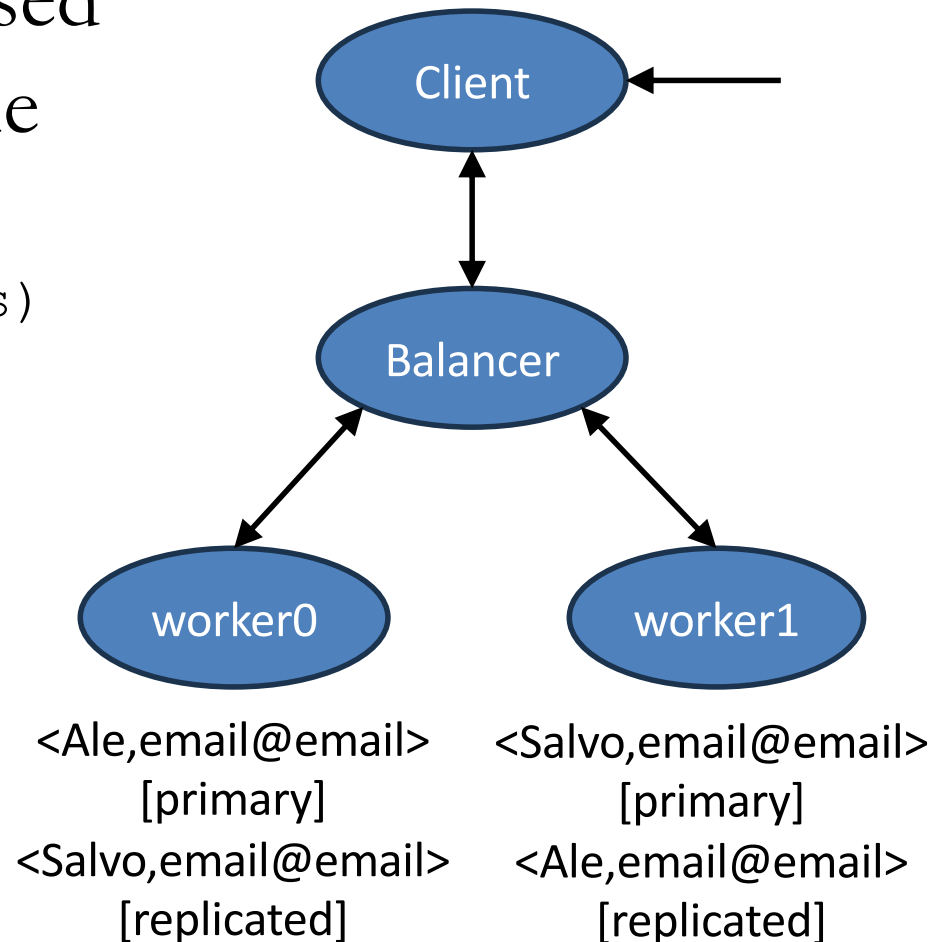
Overview (1/2)

- You are to implement a **replicated** Address Book
- Four actors are involved
 - A **client**: adds entries to the address book and retrieves emails given the name
 - A **balancer**: distributes the entries among the two workers so that one is a primary copy and the other is the replicated copy
 - Two **workers**: each one stores its primary entries and the replicated ones, on behalf of the other worker



Overview (2/2)

- Replication is implemented at the Balancer actor, based on the initial of the name
 - Check
`splitByInitial(String s)`
in the template code



Regular Operation

- The Client actor issues **PutMsg** to add entries and **GetMsg** to retrieve an email given a name
- Upon receiving a PutMsg, the Balancer tells the primary and replicated copies of each entry to store data accordingly
- Upon receiving a GetMsg, the Balancer asks the primary copy for the email and returns it to the Client actor

Resting and Resuming

- When a worker receives a **RestMsg**, it stops responding to queries and silently drops them
- If the primary copy is currently not responding to queries, the balancer tries with the replicated copy
- If both copies are resting, the Client is notified with a **TimeoutMsg** instead of a regular reply
- When a worker receives a **ResumeMsg**, it starts working normally again

Requirements

- Query/reply interactions between the Client and the Balancer should use the **Ask pattern**
 - All other interactions are free to define
- The Balancer **must not store** any information on the state (resting or working) of the workers

Code

- In the assignment, you find
 - A definition of basic message types
 - You are free to **extend** the message definition
 - You can of course define more message types, if needed
 - Templates for **all actors involved**
 - Place the debug output **where appropriate**
 - **Do not change or remove the debug output!**
 - A template for a **test main method**
 - Uncomment all lines starting with `client`, `worker0`, and `worker1`
 - This needs to be completed!!
 - It cannot run as it is!!
 - It must run when you submit
 - Do **not** change the package name of the template code!
 - Do **not** change the console output when present!
 - You cannot change the template code **unless stated explicitly!**

Rules and Guidelines

- Proper use of actor abstractions is the **key grading criteria**
- Complete the README.md file with
 - Your group identifier
 - From the group registration document
 - Name of each group member
 - A 200-word (max) description of the message flows in your solution
 - What actor talks to what other actor using what message, when, ...
- Create and submit a single zip file with the entire code of your project
 - Name of the file: akka-groupXX.zip
 - XX is the group identifier from the group registration document
 - Submit by the user corresponding to the contact email specified in the group registration document