Politecnico di Milano

090950 – Distributed Systems

Prof. G. Cugola and Prof. A. Margara

Projects for the A.Y. 2024-2025

Rules

1. The project is optional and, if correctly developed, contributes by increasing the final score.
2. Projects must be developed in groups composed of a minimum of two and a maximum of three students.
3. The set of projects described below are valid for this academic year only. This means that they have to be presented before the last official exam session of this academic year.
4. Students are expected to demonstrate their project using their own notebooks (at least two) connected in a LAN (wired or wireless) to show that everything works in a really distributed scenario.
5. To present their work, students are expected to use a few slides describing the software and run-time architecture of their solution.
6. Projects developed in Java cannot use networking technologies other than sockets (TCP or UDP, unicast or multicast) or RMI.
7. Students interested in doing their thesis in the area of distributed systems should contact Prof. Cugola for research projects that will substitute the course project.

Reliable queuing system

Implement a distributed and reliable queuing platform where a set of brokers collaborate to offer multiple queues to multiple clients.

Queues are persistent, append only, FIFO data structures. Multiple (not necessarily every) brokers replicate the data of queues to guarantee fault tolerance in case one of them crashes.

Clients connect to brokers to create new queues, append new data (for simplicity assume that queues store integer values) on an existing queue, or read data from a queue. Each

client is uniquely identified and the brokers are responsible for keeping track of the next data element each client should read.

Investigate and clarify the level of reliability offered by your system.

The project can be implemented as a real distributed application (for example, in Java) or it can be simulated using OmNet++.

Assumptions

- You may assume no partitions happening, while nodes (brokers) may fail (crash failures).
- Nodes running brokers holds a stable storage (the file system) that can be assumed to be reliable.