

Javanaise Project

Boukris Walid

Bouri Labib

Javanaise is a service handling distributed shared objects on top of Java/RMI. It aims at improving performance by caching local duplicates of the shared objects. It has a three-layered architecture: Javanaise clients talk with their local servers, which in turn exchange information (mostly locking information) with a global coordinator.

In order to run a Javanaise-enabled application, you need a Java Virtual Machine in the machine where you want to run a Javanaise Client, Server and Coordinator. The machine that hosts the Javanaise Coordinator also needs a Java RMI Registry running locally.

This tutorial assumes you know how to compile and run Java programs in the command line, or in an IDE such as IntelliJ or Eclipse. We used IntelliJ to develop and test our implementation of Javanaise.

How to run it ?

You have to run the coordinator (JvnCoordImpl.java) first, and then the client (Irc.java) to test the javanaise 1.

You have to run the coordinator (JvnCoordImpl.java) first, and then the client (IrcProxy.java) to test the javanaise 2.

We also made a test who stresses using a loop of threads that read and write 250 times, and the test works perfectly without error.