Exercise session 3: Distributed Shared Memory

13 April 2004

The goal of this exercise is to implement a Distributed Shared Memory (DSM) using the Java RMI and threading library.

Suppose there are four processes each running on a different physical node (machine) as in the figure below. The processes are called Server 0, Server 1, Server 2 and Server 3.

In our distributed application which consists of the four processes “Server 0”, “Server 1”, “Server 2”, and “Server 3”, we have three shared variables of type integer which are stored in the array variable called variables. Each process has a replica of the
shared array variable variables. Each variable should be written and read at most by one process. To guarantee this behavior one can use for each variable exactly one token. As we have three variables we need altogether three tokens in our distributed application. The tokens are always passed around to the next neighbor process. For example if process “Server 0” has the token it can read or write the variable whose token it has, and after some time it passes the token to its neighbor process “Server 1”, which itself after reading and/or writing the appropriate variable passes it to the next process and so one. It is important to note that a process can only read or write a variable if and only if it has the appropriate token.

Using the Java RMI and the threading library write a distributed shared memory application which consists of exactly four processes and three variables as sketched above.