Developing parallel applications using Clean and the Clean-CORBA interface

Introduction

Clean is a pure, lazy function programming language developed at the University of Nijmegen in the Netherlands. In this report we examine the possiblity of writing SPMD (Single Program Multiple Data) style parallel programs in Clean, using the CORBA middleware standard for communication instead of the usual PVM/MPI based communication. We will use MICO as our CORBA implementation, and a Clean-CORBA interface package developed by the author as an interface between the two systems.

The example problem

We will use a rather simple problem as an example: the computation of PI by approximation of an integral. The relevant equation is as follows:

$$\int_0^1 \frac{4}{1+x^2}\,dx = \pi$$

The approximation of this integral can be computed using the following Clean code:

```
pi_approx n = sum [4 / (1 + (squared ((i - 1 + 0.5) / n)))
/ n | i <- [1..n]]
```

Here, "n" is the number of intervalls used for the approximation. The expression above is easily paralellized: if we have m processors, each processor will compute n/m members of the sum, then sends it to a "master" process which will compute the sum and prints out the result.

The solution

As described in the previous section, the solution will use m+1 processes, each running on a separate machine. There is one process, called the master process which will start the slave processes, waits for their results then computes the final results. The master process will be a CORBA server process and the slave processes will be CORBA clients. The parameters for the slave processes will be given on the command line instead of being sent in CORBA requests, because the current Clean-CORBA interface does not permit a Clean program to act as a CORBA client and a server at the same time.
In order to communicate, the clients and the server need to know each others identity. In this simple example, this can be accomplished by supplying the identity of the server (the so-called IOR - Interoperable Object Reference) on the command line of the clients. In a more complex situation, the CORBA Naming Service could be used.

Finally, here is the IDL Interface offered by the server to the clients:

```
interface PIComputerMaster {
    void store_result(int client_index, double client_result);
};
```

Budapest, 2001. Nov. 27

Zoltan Varga