NARVAL course – Actors in Ada

Check Narval environment on your cloud host

- Does **/etc/hosts** contain your cloud host IP instead of 127.0.1.1?
- Does **narval.conf** contain your cloud host IP ?
- Is your ssh well configured? You don't need to provide a password to launch these commands:
  - `ssh localhosts`
  - `ssh fnarval`
  - `ssh your_IP`

Exercise 1: Producer and Consumer in Ada

Compile Ada Actors

- Get files in your home
  - `cd`
  - `wget https://forge.in2p3.fr/attachments/download/1290/tp_ada_actors.tar.gz`
  - `tar xvzf narval.ada.actors.tgz`
- Compile tp_producer and tp_consumer
  - `cd tp_ada_actors`
  - `make actors`
  - `make install`
    (will put some scripts in $NARVAL_SCRIPTS and configuration in $NARVAL_CONF)
- You will see some screen prints for each buffer sent from producer Actor (the first data in each data block gets incremented)
- You can open narval-actors-actives-consumers-tp.adb and narval-actors-actives-producers-tp.adb files to look at buffer_handling functions, to see how data are created in the producer and used in the consumer.

Execute Narval with Ada Actor

- *narval_launch* has to run

- Create a new script in $NARVAL_SCRIPTS or execute in a narval_shell (narval_shell –end_point http://localhost:$AWS_PORT):
  - `launch mysubsystem localhost`
  - `set configuration_file tp_simple.xml mysubsystem`
  - `set action configure mysubsystems`
  - `set action load mysubsystem`
  - `set action initialise mysubsystem`
  - `set action start mysubsystem`

- Stop yours actors with:
  - `set action stop mysubsystem`
  - `set action reset_com mysubsystem`
  - `set action unload mysubsystem`
  - `set action unconfigure mysubsystem`
  - `set action finish mysubsystem`

Modifying Ada Actor

- Create a second output buffer for the producer into the Buffer_Handling method and fill the second output buffer in the same way as the first one.
- Modify your topology file in the narval_conf directory, adding the new output buffer to the producer actor and a second instance of the consumer which receives as input the second buffer of the producer actor.
- Modify the producer to send even numbers to the first consumer and odd numbers to the second one.
- Use the On_stop function to reinitialise the values in both producer outputs. When you have your system in the running state, play with stop and start states.

Exercise 2: Adding a filter in Ada

Execute Narval with Ada filter

- Create a new script in $NARVAL_SCRIPTS or execute in a narval_shell (narval_shell –end_point http://locahost:$AWS_PORT):
  - `launch mysubsystem localhost`
  - `set configuration_file tp_filter.xml mysubsystem`
  - `set action configure mysubsystem`

- `set action load mysubsystem`
- `set action initialise mysubsystem`
- `set action start mysubsystem`

- You will see your job (or not)...
- Stop yours actors with:
  - `set action stop mysubsystem`
  - `set action reset_com mysubsystem`
  - `set action unload mysubsystem`
  - `set action unconfigure mysubsystem`
  - `set action finish mysubsystem`

Adding parameters

- You can add a new parameter from any of this types: Boolean_Type, Log_Level_Type, Unsigned_8_Type, Unsigned_16_Type, Unsigned_32_Type, Unsigned_64_Type, Unsigned_32_Array, Natural_Type, Positive_Type, Integer_Type, Integer_Array, Float_Type, Long_Float_Type, String_Type.

  The parameter could be in 4 modes : No_Access, Read_Only, Write_Only, Read_Write.

  In the initialization function in you can add a parameter boolean named “ada_param” initialized at “true” who can be accessed in read/write with:

```
Parameter := new Parameter_Type'(Container_Kind =>
                                 String_Type,
                                 Array_Length => 0,
                                 Name => To_Unbounded_String
                                   ("ada_param"),
                                 Mode => Read_Write,
                                 Monitor => Request,
                                 Run_Parameter => False,
                                 Editor => None,
                                 String_Value =>
                                   To_Unbounded_String (""));
```

Testing the new parameter

`make && make install`

- Relaunch your Narval subsytem
- When running, try to access the parameters with narval_shell commands (or Werewolf):
- get argument...
- set argument...