

## Debugging Practical

### Introduction

This practical aims to give you hands on experience using the DDT debugger on ARCHER. Log on to ARCHER using the guest account ID and password provided. Change to your work directory in the /work filesystem. Use `wget` to obtain the archive file for the exercise:

```
archer:~> wget http://tinyurl.com/archer030914/Exercises/Debug.tar.gz
```

Unpack the archive using the `tar` command:

```
archer:~> tar -xzvf Debug.tar.gz
```

This should produce a directory called `Debug`, which contains a `ddt` sub-directory. In that sub-directories are C and FORTRAN version of a broken MPI code that we will use DDT to debug and fix. Choose one of the source code directories and move into it. Compile the code using the `compile` script as follows:

```
archer:~> ./compile
```

You will need to modify the `compile` script to include the `-g` flag to ensure that the debugger can properly link the executable to the source code when debugging.

The code will work with all the compilers on ARCHER.

### DDT debugging

The recommended method for using DDT on ARCHER is to install the remote DDT client on your machines and use that to connect to ARCHER and run your jobs. Follow the instructions on the ARCHER website, in the Best Practice Guide

(<http://www.archer.ac.uk/documentation/best-practice-guide/debug.php#sec-7.4>) to install and setup the DDT remote client. Because we are using a queue reservation for the training sessions we need to use a slightly different *Submission template file* for the setup, you should use `/home/y07/y07/cse/allinea/templates/archer_phase1_reservation.qtf` instead of the normal `/home/y07/y07/cse/allinea/templates/archer_phase1.qtf` (this will allow use to specify the reservation ID as the queue on ARCHER).

Once you have installed the client and set it up to connect to ARCHER and submit jobs you should choose the **Run** option from the DDT client and configure it to run the executable you

have compiled. Remember that you will need to provide a budget code for the run (for these accounts the code is y14) and because we are using a reservation you will need to specify this as the queue. Also, this application has been built to require 8 processes, so you will need to set the `Number of processes` and `Processes per Node` to 8.

Now you can click *Submit* and the job will be submitted to the batch system. You will then get a screen showing the state of the job in the queue. Once the job runs `ddt` will open showing the source code and ready to run your program.

To run the program click the play icon (the little green triangle on the top left of the window). Run the program and identify where there is a problem (this MPI program has a fault which causes it not to work properly) and see if you can see what the problem is. Once the program is running you can pause it (using the yellow pause button next to the play button) to see where the program is. When you think you've identified the problem you can quit `ddt` by selecting *End session* from the File menu. `ddt` remembers your setup so you can close and open `ddt` without having to re-configure the program for debugging.

Try setting breakpoints in `ddt` and running the program. Experiment with some of the other `ddt` functionality. For a full guide of what `ddt` can do see the user guide at <http://www.allinea.com/products/support>.

## **Additional Exercises**

As well as using DDT you could try using the STAT debugging functionality on ARCHER to identify the deadlock problem with the code you have been given. Have a look at slide 20 of the tools lecture at <http://www.tinyurl.com/archer030914/Slides/L07-Tools.pdf>, follow the method outlined for using STAT and see if it identifies where the problem is for you.