
# DAC Tools

Running analysis programs

epcc

# 1. Aims

The aim of this exercise is to show you how to run a variety of parallel programs on the DAC. Hopefully this will give you a framework to help you port and run your own applications.

# 2. Introduction

You should already have run serial versions of the image sharpening exercise on both ARCHER and the DAC. We will investigate the following:

1. Compiling and running parallel applications (MPI and OpenMP) on ARCHER;
2. Running the same parallel applications on the DAC;
3. Running parallel python programs;
4. Performing data visualisation with paraview.

# 3. Exercises

Although it is a very simple application, the image sharpening code is an exemplar of a parallel program that reads an input data set, processes it and writes some output. Using multiple cores on the DAC nodes is key to getting good performance.

## MPI version

1. On ARCHER, build the parallel MPI program using "make" and run a batch job by submitting "sharpen.pbs"
2. Change the number of parallel processes by editing the script, and note how performance scales with the number of processes.
3. Run the same codes on the DAC; note you will need to edit the name of the compiler in the Makefile, and use the new script "dacsharpen.pbs"; a python version is also available.
4. Does the code benefit from hyperthreading on the DAC, i.e. does the performance increase when using more than the physical number of cores?
5. Is there any performance difference between the low and high memory nodes?

## OpenMP version

Repeat the same study for the OpenMP version of the sharpen code.

## Visualisation

The "heasdq.tar" archive contains a file called "headsq.vti" where the individual MRI scans (each slice stored as a separate PNG) have been combined into a single file for paraview.

Launch paraview following the instructions given in the lecture. Once it is running, you should be able to visualise the file using the following options:

- File -> Open -> headsq.vti (then click Apply)
- Filters -> Alphabetical -> Contour (then click Apply)

See if you can get a paraview client GUI to connect to a pvserver on the compute nodes. The instructions to run the pvserver are on the slides. Before doing this, you should launch the GUI and do the following:

- File -> Connect -> Add Server
- Set the port to 11112 and set "Server Type" to "Client / Server (reverse connection)"

Having a separate client and server introduces overheads, so you would probably need a very large dataset to see the performance benefits of the parallel version of paraview. However, setting the "Representation" in the "Display (GeometryRepresentation)" tab to "3D Glyphs" seems to require a lot of processing, and I have seen this operation go faster in parallel.