

# ARCHER Scripts

Tomas Lazauskas  
t.lazauskas@ucl.ac.uk  
University College London  
lazauskas.net

June 1, 2017

## Abstract

The help file of the set of scripts to execute jobs on ARCHER  
(UK's National Supercomputing Service)

## 1 JobArray.bash

A job array style submission script to execute multiple FHI-aims calculations.

An example is provided in the examples directory (Examples/JobArray).

### 1.1 Requirements

- jobs.txt (file): a namelist of simulation directories written line by line.
- prepared FHI-aims simulations (directories): must contain control.in and geometry.in files inside.
- JobArray.bash (file): The main job array script.

### 1.2 Preparation

- Preparing the directories and files:  
Copy the main JobArray.bash file to the dedicated directory on Archer.

Save the jobs.txt file in the same directory containing the list of names of the simulation directories. Copy the prepared simulation directories matching the job.txt file.

- Setting up the Archer job options:  
Modify the JobArray.bash accordingly:
  - number of nodes (line 3)  
e.g. `#PBS -l select=4`  
will request 4 nodes (96 cores) which will be used for each job in the job array.
  - Project code (line 5)  
e.g. `#PBS -A projectX`  
the project code against which the job will be charged is set as "projectX".
  - Stride (line 27) e.g. `STRIDE=2`  
it means that two simulations will be carried out per job, this is very helpful if runtime of an average simulation is quite short.
  - number of cores per FHI-aims simulation (line 35)  
e.g. `aprun -n 96 aims > FHIaims${i}.out &`  
each FHI-aims simulation will use 96 cores (4 nodes).

### 1.3 Execution

Simply by qsub'ing the JobArray

## 2 JobMultiple.bash

A job submission script to execute multiple FHI-aims calculations using the same settings (i.e. the same control.in file) - multiple aprun commands approach.

An example is provided in the examples directory (Examples/JobMultiple).

### 2.1 Requirements

- input (directory): A directory where the system files and control files have to be saved.

- input/control.in (file): A standard control.in FHI-aims input file where the basis set and the computational parameters are set. The control file must be saved in the input subdirectory
- input/\*.xyz (files): A set of xyz type structure files which will be optimised/evaluated with FHI-aims. At the moment the script only works with nanocluster type systems. If you would like to use it for periodic systems, please do not hesitate to contact me and I will make the changes.
- JobMultiple.bash: (file) The main job script which will pre-process the xyz files, prepare the directories and execute FHI-aims simulations. The results will be saved in the output directory.

## 2.2 Preparation

- Preparing directories and files:  
Copy the main JobMultiple.bash file and create a new directory "input" in the dedicated directory on Archer. Save all the structure that you would like to optimise/evaluate with FHI-aims in the new input directory. Save the control.in file in the input directory.
- Setting up the Archer job options:  
Modify the JobMultiple.bash accordingly:
  - number of nodes (line 5)  
e.g. #PBS -l select=5  
will request 5 nodes (120 cores).
  - Project code (line 9)  
e.g. #PBS -A projectX  
the project code against which the job will be charged is set as "projectX".
  - number of cores per FHI-aims simulation (line 80)  
e.g. aprun -n 24 aims > FHIaims\${i}.out &  
each FHI-aims simulation will use 24 cores (1 node) and 5 simulations will be run at same time. It is important to make sure that the total requested number of cores is divisible by the number of cores per simulation to ensure the efficient use of the computational resources.

## **2.3 Execution**

Simply by qsub'ing the JobMultiple.bash file