Running ATK

.

Running an ATK job

First, you need to load the module (/content/module-utility)for the ATK package. A few versions of ATK may be available, all managed through the modules; use **module avail atk** to see what versions are available. The sought version can be specified when loading the module, *e.g.*

```
module load atk/12.8.2
```

If no version is specified, the default version is loaded. (The default is normally the latest version available.) When loaded, the module for **atk** also loads the appropriate module for **mpich2**.

Then, you need to prepare a submission script for the ATK job. Assume the calculations to run are programmed in the **python** file **atkTest.py**, which are to be carried out using **atkpython**. The submission script for this job should look something like this (Torque script on the left, Slurm script on the right):

```
#!/bin/bash

#PBS -V
#PBS -l nodes=2:ppn=16
#PBS -l walltime=01:00:00
#PBS -N atkTest

cd $PBS_O_WORKDIR

. atk.sh

atkpython_run atkTest
```

```
#!/bin/bash

#SBATCH --nodes=2
#SBATCH --ntasks-per-node=16
#SBATCH --time=01:00:00
#SBATCH --job-name=atkTest

. atk.sh

atkpython_run atkTest
```

The setup for the parallel run is prepared automatically by the script **atk.sh**, which works with both the Torquen and Slurm environment. Starting from the number of nodes requested (**nodes=2** in this case), this script sets

- the number of parallel MPI processes that each node runs and
- the number of parallel threads that each process uses.

Following the recommendations from QuantumWise, the setup uses a number of processes per node equal to the number of cores available (16, currently). This means there is a one to one correspondence between the MPI processes running on a node and the number of cores available on that node. For some time, QuantumWise recommended the number of MPI processes be equal to the number of CPU sockets available per node, thus leaving each process multithread on the CPU cores. They have however reviewed this recommendation, and the general wisdom now is to fully populate the nodes with single-threaded MPI processes. (Our own tests on the ARC clusters have confirmed this finding.) The default setting reflects that.

If users with to change the default values, they can do so by setting a couple of variables in the submission script. For example, a job can be set to use 2 MPI process per node (one process per CPU socket) instead of the default 1 by adding

```
export NUM_PROCS_PER_NODE=2
export NUM_THREADS_PER_PROCESS=8
```

in the submission script just before the line sourcing the script **atk.sh**. The total number of MPI processes in the job depends on the **nodes** request in the PBS headers but is independent of the **ppn** value. Also, the total number of threads, equal to the number of processes multiplied by the number of threads per process, is always set to equal the total number of cores available per job.

The parallel launch is managed using the function **atkpython_run** (also defined by **atk.sh**), which launches **atkpython** through **mpiexec**. The command line argument to this function is the name of the job, *i.e.* the name of the **python** script without the **.py** extension. The standard output from the ATK run goes to a file with the same name and the extension **.out**; in the above example, output goes to **atkTest.out**.

Finally, supposing the above example submission script is called **atkTestRun.sh**, the job is sent to the execution queue with the Torque command (left) or the Slurm command (right)

```
qsub atkTestRun.sh
```

```
sbatch atkTestRun.sh
```

Example Job

We have provided an example job environment (based on the scripts above) which you can use to familiarise yourself with the batch system or modify for your own jobs. To extract it:

```
cd $DATA
tar xf /system/software/examples/scripts/atk.tgz
```

This will create a directory named **atk-example**, to run the example job:

```
cd atk-example
qsub atk_torque.sh
```

```
cd atk-example
sbatch atk_slurm.sh
```

Querying ATK License Server

The ATK license server can be queried from your laptop, desktop or departmental machine with the command

```
lmxendutil -licstat -host grant.oerc.ox.ac.uk -port 19876 -network
```

lmxendutil is a FlexNet (formerly known as FLEXlm) tool that comes with the ATK license server software.