Using CPL on ARCHER2

- Self-install
  - Default environment
  - Installing cpl-library. CPL_APP_OPENFOAM, CPL_APP_LAMMPS-DEV
- Central-install via modules
  - Replace 30 lines with 4
- Example of simple coupling
  - Setup and batch script
- Example of coupling OpenFOAM with dummy MD app
  - Setup and batch script
- Shared vs distinct MPI Communicators

Installing cpl-library

- Users first load the default environment

```
module load openfoam/com/v2106
module load gcc/10.3.0
module load cray-python
module load cray-fftw
```

- Then clone the cpl-library git repository, and install

```
cd /work/y23/y23/gavincpl
git clone https://github.com/Crompulence/cpl-library.git
cd cpl-library
make PLATFORM=ARCHER2
source SOURCEME.sh
```

Installing CPL_APP_OPENFOAM

- Then clone the CPL_APP_OPENFOAM git repository, and install

```
cd /work/y23/y23/gavincpl
git clone https://github.com/Crompulence/CPL_APP_OPENFOAM.git
cd CPL_APP_OPENFOAM
source SOURCEME.sh
cd src;ln -s CPLPstream_v2106 CPLPstream;cd ..
make pstream
make cpltestfoam
make cpltestsocketfoam
make cplinterfoam
make cplinterfoamhardtphasechange
```

Installing CPL_APP_LAMMPS-DEV

- Then clone the LAMMPS-DEV git repository, and install

```
cd /work/y23/y23/gavincpl
git clone https://github.com/Crompulence/CPL_APP_LAMMPS-DEV.git
git clone -b stable https://github.com/lammps/lammps.git mylammps
cd CPL_APP_LAMMPS-DEV
echo "/work/y23/y23/gavincpl/mylammps" > CODE_INST_DIR
source SOURCEME.sh
cd config; ./enable-packages.sh make; cd ..
make patch-lammps
make CC=CC LINK=CC
```

Using the centrally installed versions via modules

```
module load other-software
module load cpl-openfoam/2106
module load cpl-lammps
source $FOAM_CPL_APP/SOURCEME.sh
```

- Quicker and easier to employ,
  - Example input files may be altered
  - but source code and executables cannot be changed

- More details on this and how to run all the examples using either the modules or your own version
  - https://www.cpl-library.org/docs/Running_on_ARCHER2.pdf

Dummy CFD coupled with dummy MD using modules

```
module load other-software
module load cpl-openfoam/2106
module load cpl-lammps
source $FOAM_CPL_APP/SOURCEME.sh

cd /work/y23/y23/gavincpl
cp -r $CPL_APP/examples/minimal_send_recv_mocks .
cd minimal_send_recv_mocks
cplc++ minimal_MD.cpp -o MD
cplc++ minimal_CFD.cpp -o CFD
sbatch script.bat
```

Script.bat example

```
#!/bin/bash
#SBATCH --job-name=my_cpl_test
#SBATCH --time=0:10:0
#SBATCH --exclusive
#SBATCH --export=none
#SBATCH --account=y23
#SBATCH --partition=standard
#SBATCH --qos=standard
#SBATCH --nodes=2
export OMP_NUM_THREADS=1
module load other-software
module load cpl-openfoam/2106
module load cpl-lammps
source $FOAM_CPL_APP/SOURCEME.sh
SHARED_ARGS="--distribution=block:block -hint=nomultithread“
srun ${SHARED_ARGS} --het-group=0 --nodes=1 --tasks-per-node=1 MD :
--het-group=1 --nodes=1 --tasks-per-node=1 CFD
```

OpenFOAM coupled with dummy MD using modules

```
module load other-software
module load cpl-openfoam/2106
module load cpl-lammps
source $FOAM_CPL_APP/SOURCEME.sh

cd /work/y23/y23/gavincpl
cp -r $FOAM_CPL_APP/examples/CPLTestSocketFoam .
cd CPLTestSocketFoam
cplc++ minimal_MD.cpp -o MD
sbatch script.bat
```

Script.bat example

```
#!/bin/bash
#SBATCH --job-name=my_cpl_test
#SBATCH --time=0:10:0
#SBATCH --exclusive
#SBATCH --export=none
#SBATCH --account=y23
#SBATCH --partition=standard
#SBATCH --qos=standard
#SBATCH --nodes=2
export OMP_NUM_THREADS=1
module load other-software
module load cpl-openfoam/2106
module load cpl-lammps
source $FOAM_CPL_APP/SOURCEME.sh
blockMesh
decomposePar -force
SHARED_ARGS="--distribution=block:block -hint=nomultithread"
srun ${SHARED_ARGS} --het-group=0 --nodes=1 --tasks-per-node=2 MD :
--het-group=1 --nodes=1 --tasks-per-node=2 CPLTestSocketFoam -parallel
```

Shared vs distinct MPI communicators

- Coupling two applications can involve
  - Two **distinct** MPI_Comm_world communicators
  - One **share** MPI_Comm_world communicator
  - Both are available using CPL on ARCHER2
- Coupling codes on ARCHER2
  - Multiple “heterogeneous jobs”,
  - Each job (application)
    - resides inside its “het-group”.
    - must have at least one node each
  - Distinct: “client/server model”
  - https://docs.archer2.ac.uk/user-guide/scheduler/#heterogeneous-jobs

- Thank you for your attention
- Questions at the end