

# Parallel simulations and using clusters

## Fluidity Training Course 2014

Michael Lange

AMCG, Imperial College London

6th November 2014

## Parallel Simulations

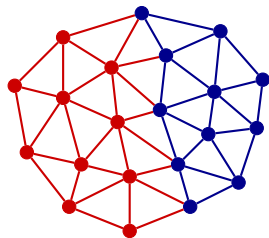
Large simulations require more than a laptop

- ▶ Fluidity runs in parallel via MPI
- ▶ Optional: OpenMP threading for hybrid runs
  - ▶ Configuration option `--enable-openmp`
- ▶ Fluidity runs on various clusters
  - ▶ Imperial College: CX1, CX2 (Intel)
  - ▶ UK national supercomputer: Archer (Cray CX30)
  - ▶ ANU/NCI: Raijin (Fujitsu Primergy)

## Parallel Simulations

How to run Fluidity in parallel:

- ▶ No changes to the .flml required!
  - ▶ Parallel compatible preconditioner (not eisenstat)
  - ▶ Optional: Remove fields from output (.stat and .vtu)
- ▶ Need to decompose the mesh
  - ▶ Tools provided:  
`flrecomp`, `fldecomp`
  - ▶ Build both with: `make fltools`



## Mesh decomposition

Mesh decomposition before the run:

- ▶ `mpiexec -n 4 flredecomp -i 1 -o 4 serial parallel`
  - ▶ Needs to be run in parallel!
  - ▶ Decomposes mesh named in `serial.flml`
  - ▶ Creates new parallel setup: `parallel.flml`
- ▶ Now run this on 4 processors:  
`mpiexec -n 4 fluidity parallel.flml`

## Mesh decomposition

Mesh decomposition with checkpoints:

- ▶ Switch on checkpointing!
  - ▶ `/io/checkpointing` in `.flml`
- ▶ `mpiexec -n 8 flredecomp -i 4 -o 8 par_4 par_8`
  - ▶ Needs to be run on `max(-i, -o)` processes
  - ▶ Enables re-starting on more/fewer processes
- ▶ Run as before:  
`mpiexec -n 8 fluidity par_8.flml`

## Running Fluidity on clusters

- ▶ Set up simulation on local machine!
- ▶ Install Fluidity on cluster
  - ▶ CX1/CX2: module load fluidity
  - ▶ Archer: instructions here
- ▶ Create .pbs submission script
  - ▶ Submit with qsub
- ▶ Copy results back to local machine
  - ▶ Visualisation as before

## CX1

```
#!/bin/bash
#PBS -N pile
#PBS -l walltime=0:20:00
# Resource requirements
#PBS -l select=1:ncpus=8:mem=1550mb

module load fluidity

echo Working directory is $PBS_O_WORKDIR
cd $PBS_O_WORKDIR

mpiexec firedcomp -i 4 -o 8 flow_past_pile_50_checkpoint flow_past_pile_50_cp_n8

mpiexec fluidity -v2 -l flow_past_pile_50_cp_n8.flml
```

## Archer

```
#!/bin/bash —login
#PBS -l walltime=01:00:00
#PBS -N jet_sim
#PBS -l select=2
#PBS -A n01-IC1

# Load Fluidity environment
module swap PrgEnv-cray PrgEnv-fluidity

# Make sure any symbolic links are resolved to absolute path
export PBS_O_WORKDIR=$(readlink -f $PBS_O_WORKDIR)
cd $PBS_O_WORKDIR

# Workaround for python environment
export LD_LIBRARY_PATH=$ANACONDA_LIB:$LD_LIBRARY_PATH

export FLUIDITYDIR=path/to/my/fluidity/install

# Decompose the mesh
aprun -n 48 -N 24 -d 1 -j 1 $FLUIDITYDIR/bin/flredecomp -i 1 -o 48 jet_sim set_sim_48

# Run simulation
aprun -n 48 -N 24 -d 1 -j 1 $FLUIDITYDIR/bin/fluidity -l -p -v2 jet_sim_48.flml
```