Compiling software on the user profile

Incompact3d

Following instructions from https://github.com/xcompact3d/Incompact3d

salloc

git clone https://github.com/xcompact3d/Incompact3d
cd Incompact3d/
git checkout v3.0
module load mpi/openmpi-x86_64
make

once compiled
cd examples/Cylinder
pwd
/homes/<user>/Incompact3d/examples/Cylinder

mpirun -n 1 ~/Incompact3d/xcompact3d

```
 Xcompact3d is run with the default file -->input.i3d
 ============================================================
 ======================Xcompact3D============================
 ===Copyright (c) 2018 Eric Lamballais and Sylvain Laizet===
 ===Modified by Felipe Schuch and Ricardo Frantz============
 ===Modified by Paul Bartholomew, Georgios Deskos and=======
 ===Sylvain Laizet -- 2018- ================================
 ============================================================
 Git version        : v3.0-0-g9015765

 Simulating cylinder
 (lx,ly,lz)=   20.000000000000000        12.000000000000000
6.0000000000000000
 (nx,ny,nz)=         257         128          64
 (dx,dy,dz)=   7.8125000000000000E-002   9.3750000000000000E-002
9.3750000000000000E-002
 (nx*ny*nz)=     2105344
 (p_row,p_col)=           0           0

 Numerical precision: Double
 Boundary condition : (nclx1 ,nclxn )=(2,2)
                      (ncly1 ,nclyn )=(0,0)
                      (nclz1 ,nclzn )=(0,0)
 High and low speed : u1=  1.00 and u2=  1.00
 Reynolds number Re :    300.00000000
 Gravity vector     : (gx, gy, gz)=(     0.00000000,     0.00000000,
0.00000000)
 Time step dt       :      0.00250000
```

```
 Spatial scheme     :      0.69477383
 Temporal scheme    : Adams-bashforth 3
 Scalar             : off
 Immersed boundary  : on with Lagrangian Poly

 In auto-tuning mode......
 factors:            1
 processor grid           1  by            1  time=
8.6869291961193085E-002
 the best processor grid is probably            1  by            1
 Initializing variables...
 Using the hyperviscous operator with (nu_0/nu,c_nu) = (
4.0000000000000000      ,  0.44000000000000000      )
 Using the hyperviscous operator with (nu_0/nu,c_nu) = (
4.0000000000000000      ,  0.44000000000000000      )
 Using the hyperviscous operator with (nu_0/nu,c_nu) = (
4.0000000000000000      ,  0.44000000000000000      )
 Generating the geometry!
     step 1
     step 2
     step 3
--------------------------------------------------------------------------
mpirun noticed that process rank 0 with PID 16346 on node ava01 exited on
signal 9 (Killed).
--------------------------------------------------------------------------
```

OpenFoam7

infos https://openfoam.org/download/7-source/

salloc -n 16 -p big

mkdir OpenFOAM
cd OpenFOAM/
wget -O - http://dl.openfoam.org/source/7 | tar xvz
wget -O - http://dl.openfoam.org/third-party/7 | tar xvz
mv OpenFOAM-7-version-7 OpenFOAM-7
mv ThirdParty-7-version-7 ThirdParty-7

https://openfoam.org/download/source/software-for-compilation/
module load mpi/openmpi-x86_64
source $HOME/OpenFOAM/OpenFOAM-7/etc/bashrc

https://openfoam.org/download/source/third-party-software/
cd ThirdParty-7
./Allwmake -j 16

https://www.paraview.org/download/
wget
“https://www.paraview.org/paraview-downloads/download.php?submit=Download&version=v5.6&typ

```
e=binary&os=Linux&downloadFile=ParaView-5.6.0-MPI-Linux-64bit.tar.gz" -O ParaView.tar.gz
gunzip ParaView.tar.gz
tar -xf ParaView.tar
rm ParaView.tar
wmRefresh

cd ..
cd OpenFOAM-7
./Allwmake -j 16

# Getting started
mkdir -p $FOAM_RUN
cd $FOAM_RUN
cp -r $FOAM_TUTORIALS/incompressible/simpleFoam/pitzDaily .
cd pitzDaily
blockMesh
simpleFoam
# more at http://cfd.direct/openfoam/user-guide
```