# Nexus Monthly User and Developer Meeting

## Meeting 1: Using Multiple Machines and Job Bundling

Jaron T. Krogel

16 November 2018

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

# Agenda for this Meeting

- Workflow demos:
  - Running workflows across multiple machines: Diamond DFT to VMC
  - Bundling jobs on HPC environments: Diamond “equation of state”
- Q&A about Nexus features, workflow how to’s, etc. + Updates

Files located at: `nexus_training/monthly_meetings/02_181116_multiple_machines/`

# Anatomy of a Nexus Script

Invoke interpreter

```
#! /usr/bin/env python
```

Nexus imports

```
from nexus import settings,job,run_project
from nexus import generate_physical_system
from nexus import generate_pwscf
from nexus import generate_pw2qmcpack
from nexus import generate_qmcpack,vmc
```

User settings

```
settings(
    pseudo_dir    = './pseudopotentials',
    status_only   = 0,
    generate_only = 0,
    sleep         = 3,
    machine       = 'ws16'
    )
```

Supercell definition

```
dia16 = generate_physical_system(
    units  = 'A',
    axes   = [[ 1.785,  1.785,  0.   ],
              [ 0.   ,  1.785,  1.785],
              [ 1.785,  0.   ,  1.785]],
    elem   = ['C','C'],
    pos    = [[ 0.    , 0.    , 0.    ],
              [ 0.8925, 0.8925, 0.8925]],
    tiling = (2,2,2),
    kgrid  = (1,1,1),
    kshift = (0,0,0),
    C      = 4
    )
```

QE SCF

Workflow composition

```
scf = generate_pwscf(
    identifier   = 'scf',
    path         = 'diamond/scf',
    job          = job(cores=16,app='pw.x'),
    input_type   = 'generic',
    calculation  = 'scf',
    input_dft    = 'lda',
    ecutwfc      = 200,
    conv_thr     = 1e-8,
    nosym        = True,
    wf_collect   = True,
    system       = dia16,
    pseudos      = ['C.BFD.upf'],
    )
```

Orbital Conversion

```
conv = generate_pw2qmcpack(
    identifier   = 'conv',
    path         = 'diamond/scf',
    job          = job(cores=1,
                       app='pw2qmcpack.x'),
    write_psir   = False,
    dependencies = (scf,'orbitals'),
    )
```

QMCPACK VMC

Workflow composition

```
qmc = generate_qmcpack(
    identifier   = 'vmc',
    path         = 'diamond/vmc',
    job          = job(cores=16,
                       threads=4,
                       app='qmcpack'),
    input_type   = 'basic',
    system       = dia16,
    pseudos      = ['C.BFD.xml'],
    jastrows     = [],
    calculations = [
        vmc(
            walkers     =   1,
            warmupsteps =  20,
            blocks      = 200,
            steps       =  10,
            substeps    =   2,
            timestep    =  .4
            )
        ],
    dependencies = (conv,'orbitals'),
    )
```

Workflow execution

```
run_project()
```

# Workflow Demo:
# Running on Multiple Machines

# Diamond Example: DFT on Workstation, QMC on Cluster

01_multiple_machines_demo/diamond.py

```
#! /usr/bin/env python

from nexus import settings,job,run_project
from nexus import generate_physical_system
from nexus import generate_pwscf
from nexus import generate_pw2qmcpack
from nexus import generate_qmcpack,vmc

settings(
    pseudo_dir    = './pseudopotentials',
    status_only   = 0,
    generate_only = 0,
    sleep         = 3,
    machine       = 'ws16',
    #machine       = 'eos',
    #account       = 'mat151',
    )

on_desktop = settings.machine=='ws16'
on_cluster = settings.machine=='eos'

if on_desktop:
    scf_job  = job(cores=16,app='pw.x')
    conv_job = job(cores=1,app='pw2qmcpack.x')
    qmc_job  = job(cores=1) # fake/placeholder job
elif on_cluster:
    scf_job  = job(nodes=1) # fake/placeholder job
    conv_job = job(nodes=1) # fake/placeholder job
    modules = '''
module load PrgEnv-intel
module load gcc
module load cray-hdf5-parallel
module load fftw
module load boost'''
    qmcpack = ‘/path/to/qmcpack-3.5.0/qmcpack_soa_real'
    qmc_job  = job(nodes   = 1,
                   threads = 4,
                   minutes = 30,
                   queue   = 'debug',
                   presub  = modules,
                   app     = qmcpack)
else:
    print 'not on desktop or cluster!'
    exit()
#end if
```

```
dia16 = generate_physical_system(
    units  = 'A',
    axes   = [[ 1.785,  1.785,  0.   ],
              [ 0.   ,  1.785,  1.785],
              [ 1.785,  0.   ,  1.785]],
    elem   = ['C','C'],
    pos    = [[ 0.    ,  0.    ,  0.    ],
              [ 0.8925,  0.8925,  0.8925]],
    tiling = (2,2,2),
    kgrid  = (1,1,1),
    kshift = (0,0,0),
    C      = 4
    )

scf = generate_pwscf(
    identifier   = 'scf',
    path         = 'diamond/scf',
    job          = scf_job,
    input_type   = 'generic',
    calculation  = 'scf',
    input_dft    = 'lda',
    ecutwfc      = 200,
    conv_thr     = 1e-8,
    nosym        = True,
    wf_collect   = True,
    system       = dia16,
    pseudos      = ['C.BFD.upf'],
    )
```

```
conv = generate_pw2qmcpack(
    identifier   = 'conv',
    path         = 'diamond/scf',
    job          = conv_job,
    write_psir   = False,
    dependencies = (scf,'orbitals'),
    )

qmc = generate_qmcpack(
    block        = not on_cluster,
    identifier   = 'vmc',
    path         = 'diamond/vmc',
    job          = qmc_job,
    input_type   = 'basic',
    system       = dia16,
    pseudos      = ['C.BFD.xml'],
    jastrows     = [],
    calculations = [
        vmc(
            walkers     =   1,
            warmupsteps =  20,
            blocks      = 200,
            steps       =  10,
            substeps    =   2,
            timestep    =  .4
            )
        ],
    dependencies = (conv,'orbitals'),
    )

run_project()
```

# Diamond Example: DFT on Workstation, QMC on Cluster

## Running the demo

```
# enter demo directory
cd 01_multiple_machines_demo/

# open demo file
emacs diamond.py&

# check graph of simulations
./diamond.py --graph_sims

# check status of simulations
./diamond.py --status_only

# run dft and conversion step
./diamond.py

# change machine from ws16 to eos
(edit machine in diamond.py)

# copy all files over to eos
rsync -av diamond.py pseudopotentials runs
jtkrogel@eos:/lustre/atlas/proj-
shared/mat151/jtkrogel/nexus_user_dev/01_multiple_machines_demo/

# login to eos
ssh -Y eos
```

```
# enter demo directory on eos
cd :/lustre/atlas/proj-
shared/mat151/jtkrogel/nexus_user_dev/01_multiple_machines_demo/

# check that run status was preserved during transfer
./diamond.py --status_only

# run qmc (let this run, switch to new eos terminal)
./diamond.py

# view submitted job in eos queue
qstat -u jtkrogel

# view files generated by nexus
ls runs/diamond/vmc/

# view submission file generated by nexus
cat runs/diamond/vmc/vmc.qsub.in

# once ./diamond.py finishes above, view status
./diamond.py --status_only

# get total energy of qmc run
qmca -q ev runs/diamond/vmc/*scalar*
```

# Workflow Demo:
# Job Bundling for HPC Environments

# Diamond Example: Lattice Scan + Job Bundling

02_job_bundling_demo/diamond.py

```
#! /usr/bin/env python

from nexus import settings,job,run_project
from nexus import generate_physical_system
from nexus import generate_pwscf
from nexus import generate_pw2qmcpack
from nexus import generate_qmcpack,vmc
from nexus import bundle

settings(
    pseudo_dir    = './pseudopotentials’,
    status_only   = 0,
    generate_only = 0,
    sleep         = 3,
    machine       = 'ws16',
    #machine       = 'eos',
    #account       = 'mat151',
    )

on_desktop = settings.machine=='ws16'
on_cluster = settings.machine=='eos'

if on_desktop:
    scf_job  = job(cores=16,app='pw.x')
    conv_job = job(cores=1,app='pw2qmcpack.x')
    qmc_job  = job(cores=1) # fake/placeholder job
elif on_cluster:
    scf_job  = job(nodes=1) # fake/placeholder job
    conv_job = job(nodes=1) # fake/placeholder job
    modules = '''
module load PrgEnv-intel
module load gcc
module load cray-hdf5-parallel
module load fftw
module load boost'''
    qmcpack = ‘/path/to/qmcpack-3.5.0/qmcpack_soa_real'
    qmc_job  = job(nodes   = 1,
                   threads = 4,
                   minutes = 30,
                   queue   = 'debug',
                   presub  = modules,
                   app     = qmcpack)
else:
    print 'not on desktop or cluster!'
    exit()
#end if
```

```
scalings = [0.95,1.00,1.05,1.10]

qmc_runs = []

for scaling in scalings:

    dia16_struct = generate_structure(
        units  = 'A',
        axes   = [[ 1.785,  1.785,  0.   ],
                  [ 0.   ,  1.785,  1.785],
                  [ 1.785,  0.   ,  1.785]],
        elem   = ['C','C'],
        pos    = [[ 0.    ,  0.    ,  0.    ],
                  [ 0.8925,  0.8925,  0.8925]],
        tiling = (2,2,2),
        kgrid  = (1,1,1),
        kshift = (0,0,0),
        )

    dia16_struct.rescale(scaling)

    dia16 = generate_physical_system(
        structure = dia16_struct,
        C         = 4
        )

    basepath = 'diamond/scale_{0:3.2f}'.format(scaling)

    scf = generate_pwscf(
        identifier   = 'scf',
        path         = basepath+'/scf',
        job          = scf_job,
        input_type   = 'generic',
        calculation  = 'scf',
        input_dft    = 'lda',
        ecutwfc      = 200,
        conv_thr     = 1e-8,
        nosym        = True,
        wf_collect   = True,
        system       = dia16,
        pseudos      = ['C.BFD.upf'],
        )
```

```
    conv = generate_pw2qmcpack(
        identifier   = 'conv',
        path         = basepath+'/scf',
        job          = conv_job,
        write_psir   = False,
        dependencies = (scf,'orbitals'),
        )

    qmc = generate_qmcpack(
        block        = not on_cluster,
        identifier   = 'vmc',
        path         = basepath+'/vmc',
        job          = qmc_job,
        input_type   = 'basic',
        system       = dia16,
        pseudos      = ['C.BFD.xml'],
        jastrows     = [],
        calculations = [
            vmc(
                walkers     =   1,
                warmupsteps =  20,
                blocks      = 200,
                steps       =  10,
                substeps    =   2,
                timestep    =  .4
                )
            ],
        dependencies = (conv,'orbitals'),
        )

    qmc_runs.append(qmc)

#end for

if on_cluster:
    bqmc = bundle(qmc_runs)
#end if

run_project()
```

# Diamond Example: Lattice Scan + Job Bundling

## Running the demo

```
# enter demo directory
cd 02_job_bundling_demo

# open demo file
emacs diamond.py&

# check graph of simulations
#   note multiple workflows
./diamond.py --graph_sims

# check status of simulations
./diamond.py --status_only

# run dft and conversion step
./diamond.py

# change machine from ws16 to eos
(edit machine in diamond.py)

# check graph of simulations
#   note addition of bundled simulation
./diamond.py --graph_sims

# copy all files over to eos
rsync -av diamond.py pseudopotentials runs
jtkrogel@eos:/lustre/atlas/proj-
shared/mat151/jtkrogel/nexus_user_dev/02_job_bundling_demo/

# login to eos
ssh -Y eos
```

```
# enter demo directory on eos
cd :/lustre/atlas/proj-
shared/mat151/jtkrogel/nexus_user_dev/01_multiple_machines_demo/

# check that run status was preserved during transfer
#   note presence and directory of bundled simulation
./diamond.py --status_only

# run qmc (let this run, switch to new eos terminal)
./diamond.py

# view submitted job in eos queue
#   should be a single bundled job
qstat -u jtkrogel

# view files generated by nexus
#   this is only for the first directory
ls runs/diamond/scale_0.95/vmc/

# view submission file generated by nexus
#   others are made single, only bundled is submitted
cat runs/diamond/scale_0.95/vmc/bundle.qsub.in

# once ./diamond.py finishes above, view status
./diamond.py --status_only

# get total energy of qmc run
qmca -q ev runs/diamond/scale_*/vmc/*scalar*
```

# Question & Answer + Updates

# Question & Answer + Updates

- Q&A for Nexus features, workflow how to’s, code details, etc
- Updates on current development
- Order
  - NCSU, ORNL, SNL, UIUC, UPR, ANL

# Next Monthly Meeting

- Date/Time: Friday Dec. 14th 1-2pm EST
- BlueJeans link: https://bluejeans.com/190169126
- Tentative topic: TBD
- Feature demo and/or topic requests welcome!