Mat. Sci. Training More Advanced Topics

Jack Deslippe

Part 1: Building Applications at NERSC

Jack Deslippe

Applications Already Available

Did you know that NERSC offers precompiled executables for more than 100 applications?

Applications Already Available

Did you know that NERSC offers precompiled executables for more than 100 applications?

Example, Materials Science:

VASP, NAMD, LAMMPS, NWCHEM, Quantum ESPRESSO, BerkeleyGW, SIESTA, Abinit, Gamess, GROMACS, GPAW MEEP, cpmd, libxc, etsf_io, atompaw, Wiek2K, Gaussian, PARATEC, cp2k, Wannier90, Amber, Yambo, XCrysden, Q-Chem

Applications Already Available

http://www.nersc.gov/users/software/all-software-list/

Show 25 entries Search:

#	Package	Platform	Category	Version	Module	Install Date	Date Made Default
0	ABINIT	carver	applications/ material sciences	6.0.3	abinit/6.0.3	2010-05-06	
1	ABINIT	carver	applications/ material sciences	6.10.3	abinit/6.10.3	2012-01-10	
2	ABINIT	carver	applications/ material sciences	6.2.2	abinit/6.2.2	2010-08-16	
3	ABINIT	carver	applications/ material sciences	6.4.1	abinit/6.4.1	2010-11-24	2012-01-12
4	ABINIT	franklin	applications/ material sciences	5.6	abinit/5.6	2008-12-03	2008-12-03
5	ABINIT	franklin	applications/ material sciences	5.8.4	abinit/5.8.4	2009-12-06	2009-12-15
6	ABINIT	franklin	applications/ material sciences	6.10.3	abinit/6.10.3	2012-01-10	
7	ABINIT	franklin	applications/ material sciences	6.4.3	abinit/6.4.3	2011-02-22	2011-03-24
8	ABINIT	hopper	applications/ material sciences	6.10.3	abinit/6.10.3	2012-01-09	

The NERSC Software Database on the web shows all of our available pre-compiled applications for Hopper, Franklin, Carver and Euclid.

Applications Already Available

Many application pages contain compilation instructions

e.g. Abinit

Compilation Instructions

Some advanced users may be interested in tweaking the Abinit build parameters and building Abinit themselves in their own directory. In order to aid in this process, and to provide a greater degree of transparency, the build instructions for the Abinit module are listed below. The following procedure was used to build Abinit 6.8.2 on Hopper.

```
% module swap PrgEnv-pgi PrgEnv-gnu
```

```
% module swap gcc gcc/4.5.2
```

```
% module load netcdf atompaw etsf_io wannier90 libxc
```

```
% ./configure -prefix="`pwd`/.." FC=ftn CC=CC CXX=CC FCFLAGS="-O3" CFLAGS="-O3" CXXFLAGS="-O3" --with-fc-vendor=gr
```

```
% make
```

```
% make install
```

Back to Top

Available Compilers

Compilers vs. Compiler Wrappers

pgf90, pgcc, pgCC vs **ftn, cc, CC** (mpif90, mpicc, mpiCC)

The compiler wrappers are the same as the underlying compilers with the addition of flags included by default and libraries linked by default (like MPI libraries for example)

The same compiler wrapper command (.e.g. ftn) can refer to any underlying compiler available on the system (e.g. pgi, gnu, intel etc...)

Available Compilers

Compilers vs. Compiler Wrappers

pgf90, pgcc, pgCC vs **ftn, cc, CC** (mpif90, mpicc, mpiCC)

Recommended Compilers on Hopper

Available Compilers

Available Compilers Across Machines:

	Hopper	Carver	Euclid
PGI	✗	✗	✗
GNU	✓	✓	✓
Intel	✓	✓	
Pathscale	✓		
Cray	✓		

Available Compilers

Available Compilers Across Machines:

	Hopper	Carver	Euclid
PGI	✗	✗	✗
GNU	✗	✗	✗
Intel	✗	✗	
Pathscale	✗		
Cray	✗		

Hopper/Franklin Module Access:

```
% module swap PrgEnv-pgi PrgEnv-gnu
```

Carver Module Access:

```
% module swap pgi gcc
% module swap openmpi openmpi-gcc
```

Available Compilers

When to use a particular compiler?

PGI - Default compiler. Easy to use. Good performance.

GNU - Use for best compatibility with open-source applications. Good performance on Fortran. Great C/C++ performance.

INTEL - Use for compatibility with applications + often better Fortran performance than GNU.

CRAY - Use to test performance.

PathScale - Use if you need another option to try.

Available Compilers

Useful Compiler Options:

	PGI	GNU	Intel	Cray
Optimization	-fast	-O3	-O3	-O3
OpenMP	-mp=nonuma	-fopenmp	-openmp	-Oomp
Version	-V	--version	-V	-V
Verbose	-v	-v	-v	-v
Debugging	-g -C	-g -fbounds-check	-g -warn all -CB	-g -R bs

- The "-v" option displays the complete link and compile line unwrapped from ftn

Available Compilers

Tips for using compilers:

- Use the compiler wrappers for parallel programs:
  hopper/franklin: **ftn, cc, CC** not **gcc, gfortran, pgf90 ...**
  carver: **mpif90, mpicc, mpiCC**

Available Compilers

Tips for using compilers:

- Use the compiler wrappers for parallel programs:
  hopper/franklin: **ftn, cc, CC** not **gcc, gfortran, pgf90 ...**
  carver: **mpif90, mpicc, mpiCC**

- ftn links libraries (blas etc...) by default. mpif90 does not.

Available Compilers

Tips for using compilers:

- Use the compiler wrappers for parallel programs:
  hopper/franklin: **ftn, cc, CC** not **gcc, gfortran, pgf90 ...**
  carver: **mpif90, mpicc, mpiCC**

- ftn links libraries (blas etc...) by default. mpif90 does not.

- Hopper statically link by default, but Carver dynamically links by default.

Available Compilers

Tips for using compilers:

To show libraries linked automatically with ftn

```
% ftn -v test.f90 -o test.x
...
--start-group -lscicpp_gnu -lsci_gnu_mp -lstdc++ -l gfortran -l m -lmpichf90_gnu -lmpich_gnu -
lmpichf90_gnu -lmpl -lsma -lxpmem -ldmapp -lugni -lpmi -Wl,--as-needed -lalpslli -lalpsutil -Wl,--no-as-
needed -ludreg -u pthread_mutex_destroy -u pthread_create -lpthread -Wl,--end-group -lgomp -lpthread -l
gfortran -l m
Using built-in specs.
...
```

To show which library "dgemm" is used from:

```
% ftn -Wl,-ydgemm_  test.f90 -o test.x
/scratch/scratchdirs/jdeslip/cc8ImteH.o: reference to dgemm_
/opt/xt-libsci/11.0.03/gnu/46/mc12/lib/libsci_gnu_mp.a(dgemm.o): definition of dgemm_
```

Available Libraries

Modules

Software libraries at NERSC are managed in modules.

Modules add and remove executables and libraries from your $PATH and $LD_LIBRARY_PATH as well as define environment variables.

They are used by doing "module load" command

e.g. "% module load fftw"

Available Libraries

Most math and science libraries are available

e.g. Carver:

```
% module avail fftw

--------------------------- /usr/common/usg/Modules/modulefiles ---------------------------
fftw/2.1.5(default) fftw/3.2.2          fftw-Intel/2.1.5    fftw-gnu/2.1.5
fftw/2.1.5-gnu      fftw/3.2.2-gnu      fftw-Intel/3.2.2    fftw-gnu/3.2.2
```

```
% module show fftw/2.1.5

setenv              FFTW_ROOT /usr/common/usg/fftw/2.1.5
setenv              FFTW_LIBDIR /usr/common/usg/fftw/2.1.5/lib
setenv              FFTW_INC /usr/common/usg/fftw/2.1.5/include
setenv              FFTW -I/usr/common/usg/fftw/2.1.5/include -L/usr/common/usg/fftw/2.1.5/lib
prepend-path        LD_LIBRARY_PATH /usr/common/usg/fftw/2.1.5/lib
prepend-path        INFOPATH /usr/common/usg/fftw/2.1.5/info
conflict        fftw/3.2.2
```

Common Problems 1

Quantum ESPRESSO on Hopper:

```
% cat README
...
Quick installation instructions for the impatient:
   ./configure [options]
   make all
...
```

Common Problems 1

Quantum ESPRESSO on Hopper:

```
% cat README
...
Quick installation instructions for the impatient:
   ./configure [options]
   make all
...
```

That seems easy enough...

```
% ./configure ... (Success!)
% make all ... come back in 20 minutes ... (Success!)

% aprun -n 5 ~/PresentationDir/espresso-4.3.2/bin/pw.x -in ./in

    Program PWSCF v.4.3.2      starts on 25Jan2012 at 15:19:44
    Program PWSCF v.4.3.2      starts on 25Jan2012 at 15:19:44
    Program PWSCF v.4.3.2      starts on 25Jan2012 at 15:19:44
    Program PWSCF v.4.3.2      starts on 25Jan2012 at 15:19:44
    Program PWSCF v.4.3.2      starts on 25Jan2012 at 15:19:44
    This program is part of the open-source Quantum ESPRESSO suite
    This program is part of the open-source Quantum ESPRESSO suite
     .....
```

The output looks weird and repeated....

Common Problems 1

Solution

Use the compiler wrappers, ftn, cc, CC. They can often be specified in configure:

./configure FC=ftn CC=CC CXX=CC or in make.sys file

```
% cat make.sys
.....
DFLAGS        =  -D__PGI -D__ACML
FDFLAGS       = $(DFLAGS)
MPIF90        = pgf90
#F90          = pgf90
CC            = pgcc
F77           = pgf77


.....
```

```
% cat make.sys_fixed
.....
DFLAGS        =  -D__PGI -D__ACML -D__MPI
FDFLAGS       = $(DFLAGS)
MPIF90        = ftn
#F90          = ftn
CC            = cc
F77           = ftn


.....
```

Common Problems 2

BerkeleyGW on Hopper

```
% cat arch.mk
...
FCPP    = /usr/bin/cpp -ansi
F90free = ftn -Mfree
LINK    = ftn

LAPACKLIB =
FFTWLIB      =

FOPTS   = -fast
FNOOPTS = $(FOPTS)
...
```

```
% make

/global/u2/j/jdeslip/PresentationDir/BGW_2.4.x/Xi0/../Common/fftw.f90:270: undefined reference to
`fftwnd_f77_create_plan_'
/global/u2/j/jdeslip/PresentationDir/BGW_2.4.x/Xi0/../Common/fftw.f90:270: undefined reference to
`fftwnd_f77_create_plan_'
/global/u2/j/jdeslip/PresentationDir/BGW_2.4.x/Xi0/../Common/fftw.f90:285: undefined reference to `fftwnd_f77_one_'
/global/u2/j/jdeslip/PresentationDir/BGW_2.4.x/Xi0/../Common/fftw.f90:287: undefined reference to `fftwnd_f77_one_'
```

Common Problems 2

Solution

"undefined reference" errors usually mean you are missing a library at link time. In this case, we are missing the fftw library.

Note that ftn links lapack/blas equivalents automatically.

```
% cat arch.mk
...
FCPP    = /usr/bin/cpp -ansi
F90free = ftn -Mfree
LINK    = ftn

LAPACKLIB =
FFTWLIB      =

FOPTS   = -fast
FNOOPTS = $(FOPTS)
...
```

```
% module load fftw
% cat arch.mk -fixed
...
FCPP    = /usr/bin/cpp -ansi
F90free = ftn -Mfree
LINK    = ftn

LAPACKLIB =
FFTWLIB      = -L$(FFTW_DIR) -ldfftw

FOPTS   = -fast
FNOOPTS = $(FOPTS)
...
```

Common Problems 3

BerkeleyGW on Carver

```
% module load fftw mkl
% cat arch.mk
...
F90free = mpif90 -Mfree
LINK    = mpif90
FOPTS   = -fast

FFTWPATH     = $(FFTW_ROOT)
FFTWLIB      = -L$(FFTW_LIBDIR) -ldfftw
FFTWINCLUDE  = $(FFTW_INC)

LAPACKLIB    = $(MKL)
SCALAPACKLIB = -L$(MKL_LIBDIR) -lmkl_scalapack_lp64 -lmkl_blacs_openmpi_lp64
...

% make (SUCCESS!)
```

Common Problems 3

BerkeleyGW on Carver

```
% module load fftw mkl
% cat arch.mk
...
F90free = mpif90 -Mfree
LINK    = mpif90
FOPTS   = -fast

FFTWPATH     = $(FFTW_ROOT)
FFTWLIB      = -L$(FFTW_LIBDIR) -ldfftw
FFTWINCLUDE  = $(FFTW_INC)

LAPACKLIB    = $(MKL)
SCALAPACKLIB = -L$(MKL_LIBDIR) -lmkl_scalapack_lp64 -lmkl_blacs_openmpi_lp64
...

% make (SUCCESS!)
```

(later that day...)

```
% mpirun -np 2 xi0.cplx.x
xi0.cplx.x: error while loading shared libraries: libmkl_scalapack_lp64.so: cannot open shared object file: No such file
xi0.cplx.x: error while loading shared libraries: libmkl_scalapack_lp64.so: cannot open shared object file: No such file
```

Common Problems 3

Solution

On carver, we link against shared object files. These need to be present at runtime.

These need to be in your $LD_LIBRARY_PATH

```
% module load mkl fftw
```

(or)

```
% export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/common/usg/mkl/10.2.2.025/lib/em64t:/usr/common/usg/fftw/2.1.5/lib
```

Common Problems 4

etsf_io on Hopper

```
% module swap PrgEnv-pgi PrgEnv-gnu
% module load netcdf
% ./configure --prefix="`pwd`/../" CC=cc CXX=cc FC=ftn F77=ftn FCFLAGS="-O3" F77FLAGS="-O3" CFLAGS="-O3" CXXFLAGS="-O3" --with-netcdf-module-path="$CRAY_NETCDF_DIR/gnu/45/include" --enable-fortran

checking for module extension for compiler 'gcc'... mod
checking for ranlib... ranlib
checking for ar... ar
checking for /opt/cray/netcdf/4.1.3/gnu/45/include/netcdf.mod... yes
checking for netcdf library... no
Action: install NetCDF and set the library link path with --with-netcdf-ldflags.
configure: error: "No 'NetCDF' library found."
```

But... This worked for me one month ago just fine!?!?!?

Common Problems 4

Solution

To debug ./configure errors. Look at generated config.log.

```
% cat config.log
....
configure:3866: ftn -o conftest -O3 -I/opt/cray/netcdf/4.1.3/gnu/45/include   conftest.f90  -lnetcdf >&5
conftest.f90:3.12:

  use netcdf
           1
Fatal Error: Wrong module version '4' (expected '6') for file 'netcdf.mod' opened at (1)
configure:3872: $? = 1
configure: failed program was:
|
| program main
|   use netcdf
|   integer :: s, ncid
|   s = nf90_open(path = "", mode = NF90_NOWRITE, ncid = ncid)
| end program main
....
```

ftn --version shows gcc 4.6, but netcdf module was compiled with gcc 4.5.

Summary of Good Practices

- Use developer recommended compiler and compiler options
- Use compiler wrappers
- Test your application against lower optimization levels and included tests
- Use system provided libraries

Part 2: Scripting for Advanced Workflows

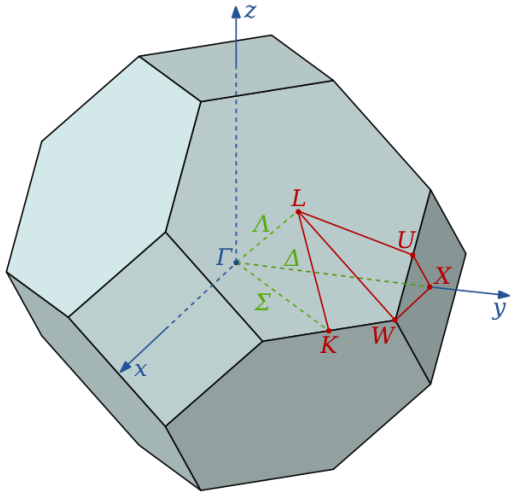
Jack Deslippe

Scripts for Multiple MPI Jobs

Many applications have a trivially parallelizable layer.

-This layer be exploited by multiple runs of the same executable with slightly different input.

For example: Material science codes are often trivially parallelizable over k-points



Scripts for Multiple MPI Jobs


```
#!/bin/bash -l

njobs=32

for (( i = 0; i < $njobs ; i++  ))
do

    echo "starting $i"

    mkdir job_$i
    cd job_$i

    # Application Specific Input Modifications
    cp -f ../input ../CD ../*DAT .
    shift=$(echo "($i) / ($njobs)" | bc -l)
    echo "k_grid_shift 0.0 0.0 $shift" >> input

    sed 's/^.*-N.*$/\#PBS -N job-'$i'/' ../qscript_template > tmp
    mv tmp qscript_$i
    qsub qscript_$i

    cd ..

done
```

PARATEC requires the following input files:

input - contains job parameters, including k-points
CD, POT.DAT - Other application input files

```
% cat qscript_template

#PBS -q regular
#PBS -l mppwidth=120
#PBS -l walltime=10:00:00
#PBS -N paratec-template
#PBS -j eo

cd $PBS_O_WORKDIR
aprun -n 120 paratec.x
```

Scripts for Multiple MPI Jobs

You can pack your multiple jobs into one larger calculation.

(Hopper Only:
See Extra Slides for Carver Instructions)

```
% cat qscript_template

#PBS -q regular
#PBS -l mppwidth=120
#PBS -l walltime=10:00:00
#PBS -j eo

cd $PBS_O_WORKDIR
```

```
#!/bin/bash -l

njobs=32
nprocs_per_job=24
let total_procs=($njobs * $nprocs_per_job)

sed 's/^.*width.*$/\#PBS -l mppwidth='$total_procs'/' qscript_template > tmp
mv tmp qscript

for (( i = 0; i < $njobs ; i++  ))
do
   mkdir job_$i
   cd job_$i

   # Application Specific Input Modifications
   cp -f ../input ../CD ../*DAT .
   shift=$(echo "($i) / ($njobs)" | bc -l)
   echo "k_grid_shift 0.0 0.0 $shift" >> input
   cd ..

   echo "cd job_$i" >> qscript
   echo "aprun -n $nprocs_per_job paratec.x &" >> qscript
   echo "cd .." >> qscript
done

echo "wait" >> qscript
qsub qscript
```

Scripts for Multiple MPI Jobs

You
mu
larg

(Ho
See
Ca

te > tmp

```
% cat qscript

#PBS -q regular
#PBS -l mppwidth=768
#PBS -l walltime=10:00:00
#PBS -j eo

cd $PBS_O_WORKDIR
cd job_0
aprun -n 24 paratec.x &
cd ..
cd job_1
aprun -n 24 paratec.x &
cd ..
-------
cd job_30
aprun -n 24 paratec.x &
cd ..
cd job_31
aprun -n 24 paratec.x &
cd ..
wait
```

Scripts for Multiple MPI Jobs

Job Arrays (Carver Only):

Arrays of separate jobs that can be controlled by a single mother job.

```
% cat qscript

#PBS -q regular
#PBS -l nodes=4
#PBS -l walltime=10:00:00
#PBS -N myjob
#PBS -j eo

cd $PBS_O_WORKDIR
cd job_$PBS_ARRAYID
mpirun -np 32  paratec.x
```

```
#!/bin/bash -l

njobs=32
let njobs_minus_one=($njobs - 1)

for (( i = 0; i < $njobs ; i++  ))
do

    echo "starting $i"

    mkdir job_$i
    cd job_$i

    # Application specific section
    cp -f ../input ../CD ../*DAT .
    shift=$(echo "($i) / ($njobs)" | bc -l)
    echo "k_grid_shift 0.0 0.0 $shift" >> input

    cd ..

done

qsub -t 0-$njobs_minus_one qscript
```

Scripts for Multiple MPI Jobs

Job Arrays (Carver Only)

Arr
tha
sin

```
% qstat -u jdeslip

1793543[].cvrsvc    jdeslip  reg_med  myjob          --    20  20    --  00:10 Q   --

% qstat -t -u jdeslip

1793543[0].cvrsv    jdeslip  reg_med  myjob-0        --    20  20    --  00:10 Q   --
1793543[1].cvrsv    jdeslip  reg_med  myjob-1        --    20  20    --  00:10 Q   --
1793543[2].cvrsv    jdeslip  reg_med  myjob-2        --    20  20    --  00:10 Q   --
1793543[3].cvrsv    jdeslip  reg_med  myjob-3        --    20  20    --  00:10 Q   --
1793543[4].cvrsv    jdeslip  reg_med  myjob-4        --    20  20    --  00:10 Q   --
1793543[5].cvrsv    jdeslip  reg_med  myjob-5        --    20  20    --  00:10 Q   --
...

% qdel 1793543[5]

% qdel 1793543[]
```

```
mpirun -np 32  paratec.x
```

```
qsub -t 0-$njobs_minus_one qscript
```

Chaining Jobs

Each job starts only after completion of previous job

```
% cat qscript_template

#PBS -q regular
#PBS -l mppwidth=120
#PBS -l walltime=10:00:00
#PBS -N myjob
#PBS -j eo

cd $PBS_O_WORKDIR
aprun -n 120 paratec.x
```

```
#!/bin/bash -l

njobs=32

for (( i = 0; i < $njobs ; i++  ))
do
   mkdir job_$i
   cd job_$i

   #Application specific section
   cp -f ../input ../CD ../*DAT .
   shift=$(echo "($i) / ($njobs)" | bc -l)
   echo "k_grid_shift 0.0 0.0 $shift" >> input

   sed 's/^.*-N.*$/\#PBS -N myjob-'$i'/' ../qscript_template > qscript_$i

   if [ -n $last_jobID ]
   then
      jobID=`qsub -W depend=afterok:$last_jobID qscript_$i`
   else
      jobID=`qsub qscript_$i`
   fi

   last_jobID=${jobID%%.*}
   #last_jobID="${last_jobID}.sdb@sdb" # uncomment this line on hopper
   cd ..
done
```

Scripts for Multiple Serial Jobs

Carver:

A serial queue exists. You can submit multiple serial jobs to this queue.

Hopper:

No serial queue exists. If you submit a serial job with "aprun -n 1 command" you will be charged for a whole node.

aprun is limited so that each "aprun -n 1 command &" in a batch script will require a new empty node.

Scripts for Multiple Serial Jobs

Fortran MPI Job Wrapper.

Each MPI task executes serial "executable" in directory run.rank

```
program mwrapper
include 'mpif.h'
integer :: size, rank, ierr, nargs, i
character (len=50) :: rundir, stdoutfile, executable, args, tmp

args=''
call getarg(1,executable)
do i = 2, iargc()
  call getarg(i,tmp)
  args=trim(args)//' '//trim(tmp)
enddo

call mpi_init(ierr)
call mpi_comm_size(mpi_comm_world,size,ierr)
call mpi_comm_rank(mpi_comm_world,rank,ierr)

write(rundir,"('run.',i3.3)") rank
write(outfile,"('stdout.',i3.3)") rank

call system ('mkdir '//rundir)
call system ('cd '//rundir//';'//trim(executable)//' '//args//' >'//outfile)

call mpi_finalize(ierr)
end program mwrapper
```

Scripts for Multiple Serial Jobs

Fortran Wrapper Example Usage:

```
% ftn mwrapper.f90 -o mwrapper.x

% qsub -I -V -q interactive -l mppwidth=48

% aprun -n 48 mwrapper.x /global/homes/j/jdeslip/paratec.x arg1 arg2
```

Scripts for Multiple Serial Jobs

Fortran Wrapper Example Usage:

```
% ftn mwrapper.f90 -o mwrapper.x

% qsub -I -V -q interactive -l mppwidth=48

% aprun -n 48 mwrapper.x /global/homes/j/jdeslip/paratec.x arg1 arg2
```

-Works well on both Cray systems and Carver.
-Fortran wrapper has very little overhead.
-Can be used with threaded applications.

Scripts for Multiple Serial Jobs

Python MPI Job Wrapper

Each MPI task executes serial "executable" in directory run.rank

```
from mpi4py import MPI
from subprocess import call
import sys

exctbl = sys.argv[1]
args = ""
i=0
for arg in sys.argv:
   if (i > 1):
      arguments = args+sys.argv[i]+" "
   i += 1

comm = MPI.COMM_WORLD
rank = comm.Get_rank()

myDir = "run."+str(rank)
outfile = "out."+str(rank)

cmd = "mkdir "+myDir+" ; cd "+myDir+" ; "+exctbl+" "+args+" > "+outfile

sts = call(cmd,shell=True)

comm.Barrier()
```

Scripts for Multiple Serial Jobs

Example python script and limitations

Usage:

```
% qsub -I -V -q interactive -l mppwidth=48

% mpirun -np 48 python mwrapper.py /global/homes/j/jdeslip/paratec.x arg1 arg2
```

Scripts for Multiple Serial Jobs

Example python script and limitations

Usage:

```
% qsub -I -V -q interactive -l mppwidth=48

% mpirun -np 48 python mwrapper.py /global/homes/j/jdeslip/paratec.x arg1 arg2
```

Limitations:

- Best used on Carver.
- On Cray systems, loading python modules can be slow if using many processors.

Extra Slides

Scripts for Multiple MPI Jobs

On Carver, and other systems with mpirun instead of aprun. You need a hostfile.

```
% cat qscript_template

#PBS -q regular
#PBS -l nodes=4:ppn=8
#PBS -l walltime=10:00:00
#PBS -j eo

cd $PBS_O_WORKDIR
```

```
#!/bin/bash -l

njobs=32
nodes_per_job=4 ; PPN=8
let nprocs_per_job=($nodes_per_job * $PPN)
let tot_nodes=($njobs * $nodes_per_job)
sed 's/^.*nodes.*$/\#PBS -nodes='$tot_nodes\':ppn='$PPN'/' qscript_template > tmp
mv tmp qscript

for (( i = 0; i < $njobs ; i++  ))
do
   mkdir job_$i
   cd job_$i

   cp -f ../input ../CD ../*DAT . # Application Specific Section
   shift=$(echo "($i) / ($njobs)" | bc -l)
   echo "k_grid_shift 0.0 0.0 $shift" >> input
   cd ..

   echo "cd job_$i" >> qscript
   let lstart=${i}*${nprocs_per_job}+1
   let lend=${lstart}+${nprocs_per_job}-1
   echo "sed -n ${lstart},${lend}'p' < \$PBS_NODEFILE > nodefile" >> qscript
   echo "mpirun -np $nprocs_per_job paratec.x -hostfile nodefile &" >> qscript
   echo "cd .." >> qscript
done
echo "wait" >> qscript
qsub qscript
```