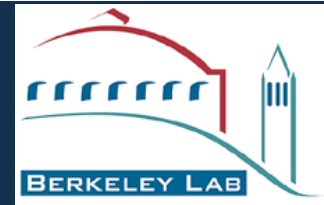


Performing parallel parameter scans on Hopper at NERSC

Robert Ryne

LBL

Sept 10, 2012

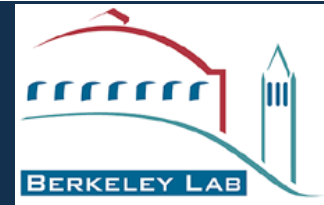


Bringing High Performance Computing (HPC) to MAP D&S



- Incorporating HPC techniques is an integral part of our D&S plans
- A new account for MAP has been set up at NERSC
- A new project directory (/project/projectdirs/map/) enables sharing of data and codes, and a place to work in addition to \$HOME and /scratch/
- Eventually we want to do parallel optimization
- We are proceeding in steps:
 - installation of serial codes ✓(partially complete; ICOOL3.0 installed)
 - parallel parameter scans using serial executables ✓
 - parallel parameter scans using parallel executables ✓
 - parallelization of serial codes such as ICOOL
 - parallel optimization using serial executables
 - parallel optimization using parallel executables

We now have the capability to perform parallel parameter scans using serial and parallel executables



Parameter scans with a serial executable

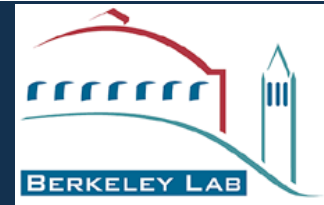


1. Login to Hopper
2. Prepare your input files as usual
3. Create a new file called "scanparams.in" describing parameters to be scanned, plus names of any other input files
4. Create a PBS batch script. Here it is called `scanscript`
 - `edit scanscript` to set walltime, queue, exec code
5. Submit `scanscript`

Results will appear in separate subdirectories. A list of the parameters used for each case will be found in a file called "scanparams.out"

Note: To copy a sample version of `scanscript` :

```
cp /project/projectdirs/map/Codes/Scanparams/scanscript .
```



Parameter scans with a parallel executable



1. Login to Hopper
2. Prepare your input files as usual
3. Create a new file called "scanparams.in" describing parameters to be scanned, plus names of any other input files
4. Create a unix script. Here it is called `pscanparams`
 - edit `pscanparams` to set `walltime`, `queue`, `exec code`, `cores/exec`
5. Run `pscanparams`
 - this will create a batch script and submit it for you

Only difference compared to the serial case is that you
(1) edit a unix script instead of a PBS batch script, and
(2) also specify the # of cores per parallel executable

Results will appear in separate subdirectories. A list of the parameters used for each case will be found in a file called "scanparams.out"

Note: To copy a sample version of `pscanparams` :

```
cp /project/projectdirs/map/Codes/Scanparams/pscanparams .
```



Example with a serial executable: scanning two quantities in an ICOOL run



- Login to Hopper: `ssh -Y hopper.nersc.gov -l your_user_name`
- Prepare your input files as usual
- Create a new file called "scanparams.in" describing parameters to be scanned, plus names of any other input files*

```
for001.dat float 0.30 0.40 24 's/0.365 1 .001/##### 1 .001/'  
for001.dat float 0.01 0.03 20 's/0.02 1 .001/##### 1 .001/'
```

*note: to specify a file for which no parameters are varied, just list the name followed by /
`ptcls.in /`

- Edit PBS batch script, called `scanscript`, to set execution time, queue, and location of serial executable

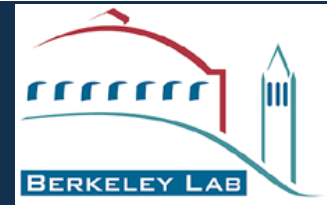
```
#PBS -q debug  
#PBS -A map  
#PBS -l mppwidth=480  
#PBS -l walltime=00:03:00  
cd $PBS_O_WORKDIR  
aprun -n 480 /project/projectdirs/map/Codes/Scanparams/scanparams.x /project/projectdirs/map/Codes/icool330/icool
```

you edit this

you edit this to point to your serial executable

edit to match above

- Submit batch script: `qsub scanscript`



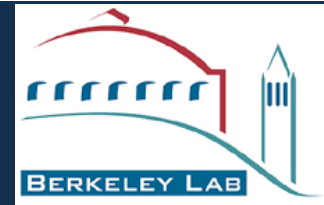
Example with a serial executable: scanning the seed in an ICOOL run



- File "scanparams.in" could look like this:

```
for001.dat integer -1 -10000 480 's/rnseed=-1/rnseed=#####/'
```

- Note: In general, besides scanning using `real` and `integer` you can also specify `octal`



Example with a parallel executable: scanning two quantities in an MLI run



- Same as for the serial case, you create a file "scanparams.in"

```
mli.in float 0.30d0 0.40d0 4 's/dr365: drift, l=0.365  slices=36/dr365: drift, l=#####  slices=36/'  
mli.in float 0.01d0 0.03d0 5 's/dr02: drift, l=0.02  slices=20/dr02: drift, l=#####  slices=20/'
```

- Instead of editing the PBS script called scanscript, you edit the beginning and end of a unix script called pscanparams

```
#!/bin/bash -l  
  
nprocs_per_job=24  
walltime='00:04:00'  
queue='premium'  
...  
...  
...  
...  
echo "aprun -n $nprocs_per_job ~/MLIjuly2012/mli.x >& std$m &" >> qscript  
...
```

you edit this

you edit this to point to your parallel executable

- Launch the job (i.e. run the unix script) by typing: `./pscanparams`