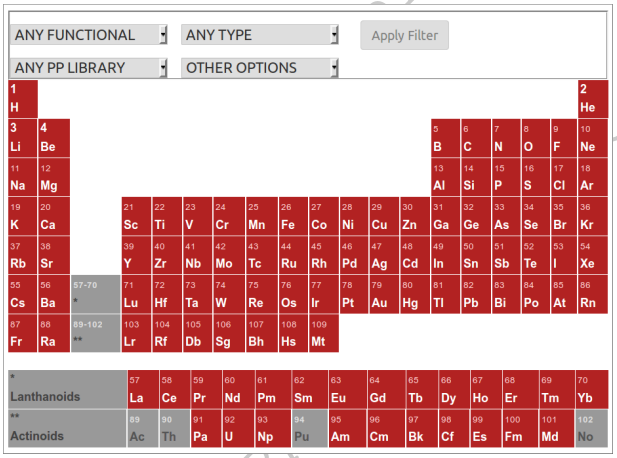
Test run

Now we want to prepare a very simple job and execute it on the cluster. The goal of this operation is to make sure that everything runs smoothly.
We will consider a simple total energy calculation for silicon in the diamond structure. In order to proceed we first need a 'pseudopotential'. Pseudopotentials will be introduced formally in Lecture 2.2, for now it suffices to know that for each atom we will need one pseudopotential. The QE pseudopotential libraries can be found at http://www.quantum-espresso.org/pseudopotentials. By clicking on this link we find the following periodic table



Now we can click on silicon and we will see a list of available pseudopotentials. In principle we could download the entire library once and for all, but for now let us proceed on a case-by-case basis. In this example we will use the pseudopotential labelled Si.pz-vbc.UPF. By hovering on this link with the mouse we can copy/paste the web link, and we can use it to download the file directly on the HPC cluster:

```
$ cd ..
$ mkdir tutorial-1.1 ; cd tutorial-1.1
$ wget http://www.quantum-espresso.org/wp-content/uploads/upf_files/Si.pz-vbc.UPF
```

Now we should have the pseudopotential file inside the directory `tutorial-1.1`. We can see inside this plain text file by using the command `more`:

```
$ more Si.pz-vbc.UPF
```

```
<PP_INFO>
Generated using unknown code
Author: von Barth and Car   Generation date: before 1984
Info: Si LDA 3s2 3p2 VonBarth-Car, l=2 local
    0        The Pseudo was generated with a Non-Relativistic Calculation
  0.00000000000E+00    Local Potential cutoff radius
nl pn  l   occ               Rcut            Rcut US             E pseu
3S  0  0  2.00      0.00000000000      0.00000000000      0.00000000000
3P  0  1  2.00      0.00000000000      0.00000000000      0.00000000000
</PP_INFO>

<PP_HEADER>
   0                  Version Number
  Si                  Element
   NC                 Norm - Conserving pseudopotential
    F                 Nonlinear Core Correction
 SLA  PZ   NOGX NOGC   PZ   Exchange-Correlation functional
    4.00000000000     Z valence
    0.00000000000     Total energy
  0.0000000  0.0000000 Suggested cutoff for wfc and rho
    1                 Max angular momentum component
  431                 Number of points in mesh
    2    2            Number of Wavefunctions, Number of Projectors
 Wavefunctions        nl  l   occ
                      3S  0  2.00
                      3P  1  2.00
</PP_HEADER>

<PP_MESH>
  <PP_R>
  1.30825992062E-03  1.34137867819E-03  1.37533584110E-03  1.41015263368E-03
  1.44585081756E-03  1.48245270526E-03  1.51998117417E-03  1.55845968079E-03
...
```


For simplicity we also copy the executable inside the current working directory (this is not standard practice but it makes things easier to understand the first time):

```
$ cp ../espresso-5.4.0/bin/pw.x ./
```

At this point we have the executable `pw.x` and the pseudopotential for silicon `Si.pz-vbc.UPF`. We are missing the input file for the executable, and the job submission script for the HPC cluster.
We can create the simplest possible input file, `silicon-1.in`, as follows:

```
$ cat << EOF > silicon-1.in
&control
 calculation = 'scf',
 prefix = 'silicon',
 pseudo_dir = './',
 outdir = './'
/
&system
 ibrav = 2,
 celldm(1) = 10.28,
 nat = 2,
 ntyp = 1,
 ecutwfc = 18.0,
/
```

```
&electrons
 conv_thr = 1.0d-8
/
ATOMIC_SPECIES
 Si  28.086  Si.pz-vbc.UPF
ATOMIC_POSITIONS
 Si 0.00 0.00 0.00
 Si 0.25 0.25 0.25
K_POINTS automatic
 4 4 4 1 1 1
EOF
```

For the job submission script we can create job-1.pbs:

```
$ cat << EOF > job-1.pbs
#!/bin/bash -l
#SBATCH --reservation=Paradim
#SBATCH --job-name=job-1
#SBATCH --time=00:30:00
#SBATCH --partition=parallel
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=12
#SBATCH --mem-per-cpu=2000MB
mpirun -np 12 pw.x -npool 4 < silicon-1.in > silicon-1.out
EOF
```

In the current directory we should now see the following:

```
$ ls
job-1.pbs  pw.x  silicon-1.in  Si.pz-vbc.UPF
```

Eventually we are ready to submit our first job to the queue. We issue:

```
$ qsub job-1.pbs
```

We can check the status of this job in the queue using the command qstat. If we do not remember our username we can find this information using:

```
$ whoami
fgiusti1.temp@jhu.edu
```

```
$ qstat -u fgiusti1.temp@jhu.edu
login-node04.cm.cluster:
                                                            Req'd  Req'd      Elap
Job id          Username Queue    Name       SessID NDS TSK Memory Time Use S Time
--------------- -------- -------- ---------- ------ --- --- ------ -------- - -----
7415791         fgiusti1 shared   job-1      --     1   12  --     00:30    Q 00:00
```

If the cluster is too busy we can alternatively try to run within an **interactive** session:

```
$ interact-paradim -r Paradim -p parallel -n 12 -c 1 -t 60 -m 24G
```

This command opens a session where we will be able to execute `pw.x` directly from the command line, ie without a job submission script.
Here we are requesting 12 CPUs, since `-n 12` is the number of tasks, and `-c 1` is the number of cores per task. We are asking for a session of 60 min (`-t 60`) and with a total memory of 24 GB (`-m 24G`).
If we enter an interactive session, then we can simply run `pw.x` by issuing:

```
$ mpirun -n 12 pw.x -npool 4 < silicon-1.in > silicon-1.out
```

This job will only take less than a second to complete. The output file is `silicon-1.out` and should look like the following:

```
$ more silicon-1.out
     Program PWSCF v.5.4.0 starts on  8Jul2016 at  8:27:35

     This program is part of the open-source Quantum ESPRESSO suite
     for quantum simulation of materials; please cite
         "P. Giannozzi et al., J. Phys.:Condens. Matter 21 395502 (2009);
          URL http://www.quantum-espresso.org",
     in publications or presentations arising from this work. More details at
     http://www.quantum-espresso.org/quote

     Parallel version (MPI), running on    12 processors
     K-points division:     npool     =       4
     R & G space division:  proc/nbgrp/npool/nimage =       3
     Waiting for input...
     Reading input from standard input
...
     PWSCF        :     0.11s CPU         0.18s WALL


   This run was terminated on:   8:27:35   8Jul2016

=------------------------------------------------------------------------------=
   JOB DONE.
=------------------------------------------------------------------------------=
```


Generating new jobs

Throughout this school we will prepare input files and job submission scripts by modifying the files `silicon-1.in` and `job-1.pbs`.
Instead of using the `cat` command as we did in the previous section, we first create two new files by just copying the previous ones:

```
$ cp silicon-1.in silicon-2.in
$ cp job-1.pbs job-2.pbs
```

Now we use `vi` to modify the newly-created files:

```
$ vi silicon-2.in
```

This will open the file inside the current terminal window. The rules for using `vi` are simple:

1 We move around using ↑ ↓ → ←

2 In order to change the text we press i and modify as we wish

3 When we are done making changes we press ESC

4 We write the modified file and exit by typing : w q

As an example we now change the parameter `ecutwfc` from 18 Ry to 30 Ry. This parameter represents the planewaves kinetic energy cutoff and will be described in Lecture 2.2. We can also change the Brillouin zone sampling from `4 4 4 1 1 1` to something more accurate, say `8 8 8 1 1 1`.
In the case of `job-2.pbs` we proceed similarly. We first change the input and output filenames into `silicon-2.in` and `silicon-2.out`. We also change the number of CPUs to be used from 4 (`-np 4`) to 12 (`-np 12`).
We submit this job as we did earlier:

```
$ qsub job-2.pbs
```

During the hands-on session we will explore in detail various calculation parameters and runtime options.

Documentation

A comprehensive description of the input variables accepted by `pw.x` can be found here:

http://www.quantum-espresso.org/wp-content/uploads/Doc/INPUT_PW.html

During the next few days we will also need documentation for the code `ph.x` (phonons) and `pp.x` (post-processing). These can be found here:

pp.x: http://www.quantum-espresso.org/wp-content/uploads/Doc/INPUT_PH.html

pp.x: http://www.quantum-espresso.org/wp-content/uploads/Doc/INPUT_PP.html

Information about the job scheduling system of MARCC can be found at:
http://www.marcc.jhu.edu/getting-started/running-jobs

In order to find out which 'queues' or 'partitions' are available we can use the command

```
$ sinfo -s
```