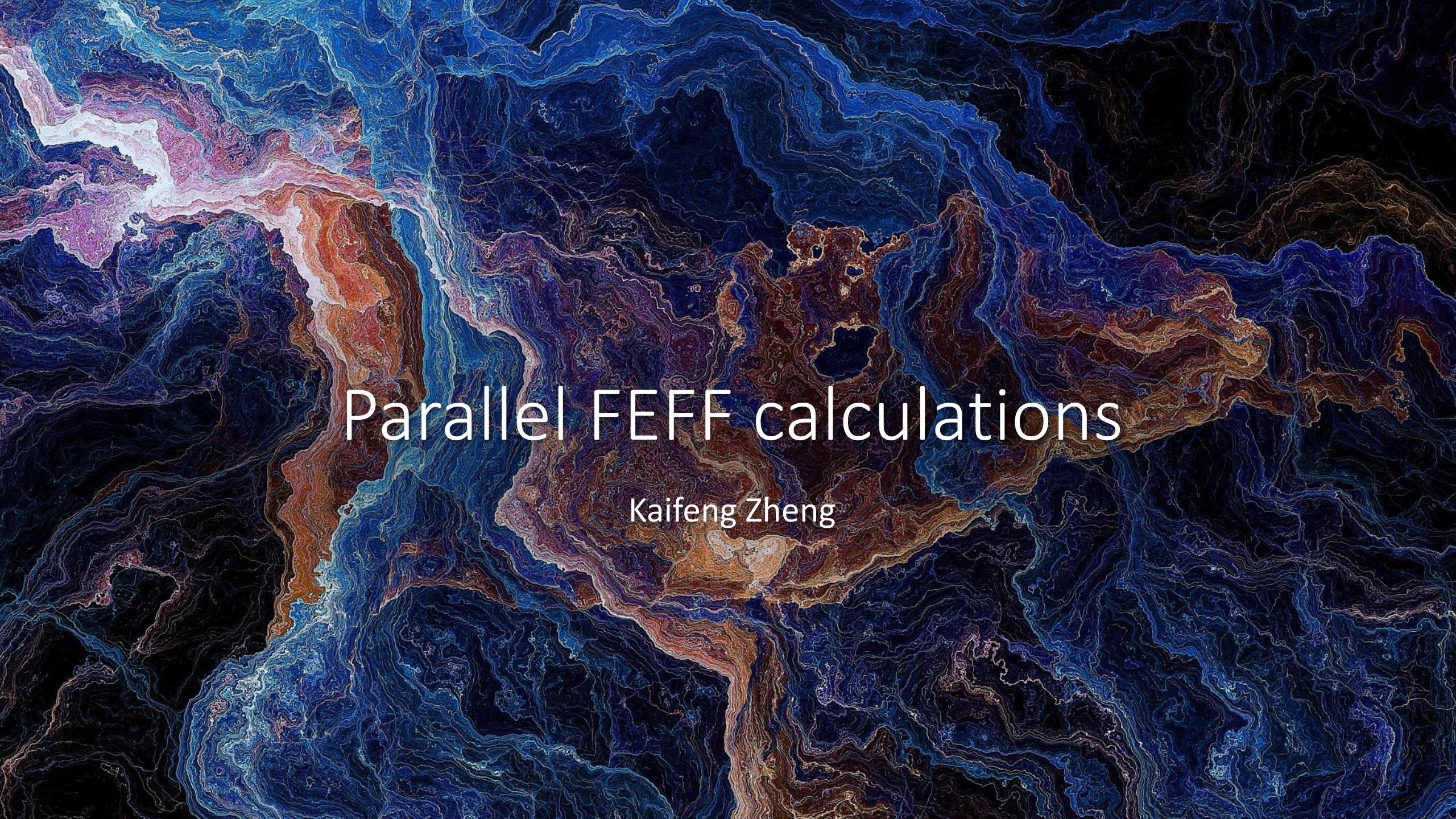
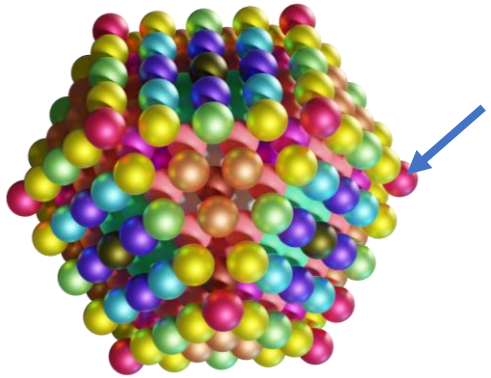


Parallel FEFF calculations

Kaifeng Zheng

The motivation

- Heavy calculation for large particles(XANES and EXAFS)



Inequivalent site

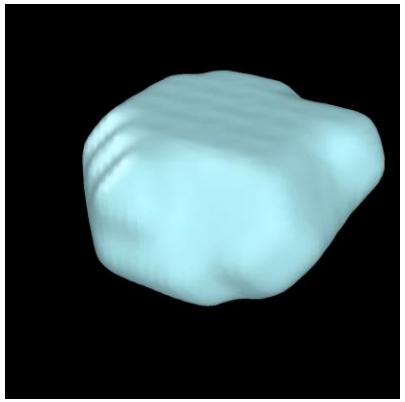
3000 147-atom particles contain 441,000 atoms

Each calculation uses roughly 1 min(single cpu).

3000 147-atom particles contain 441,000 atoms

441,000 atoms need 306.25 days to finish all calculations!!!

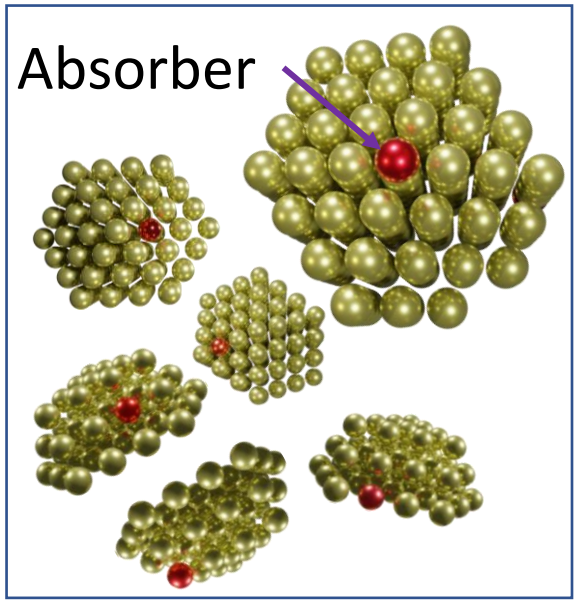
Solution: make the calculation parallel!



Multi-tasking FEFF calculations

- Successful example
  - 55000 FEFF calculations for 55-atom particles need 32hrs to finish, which means 1 min finish 28 jobs!

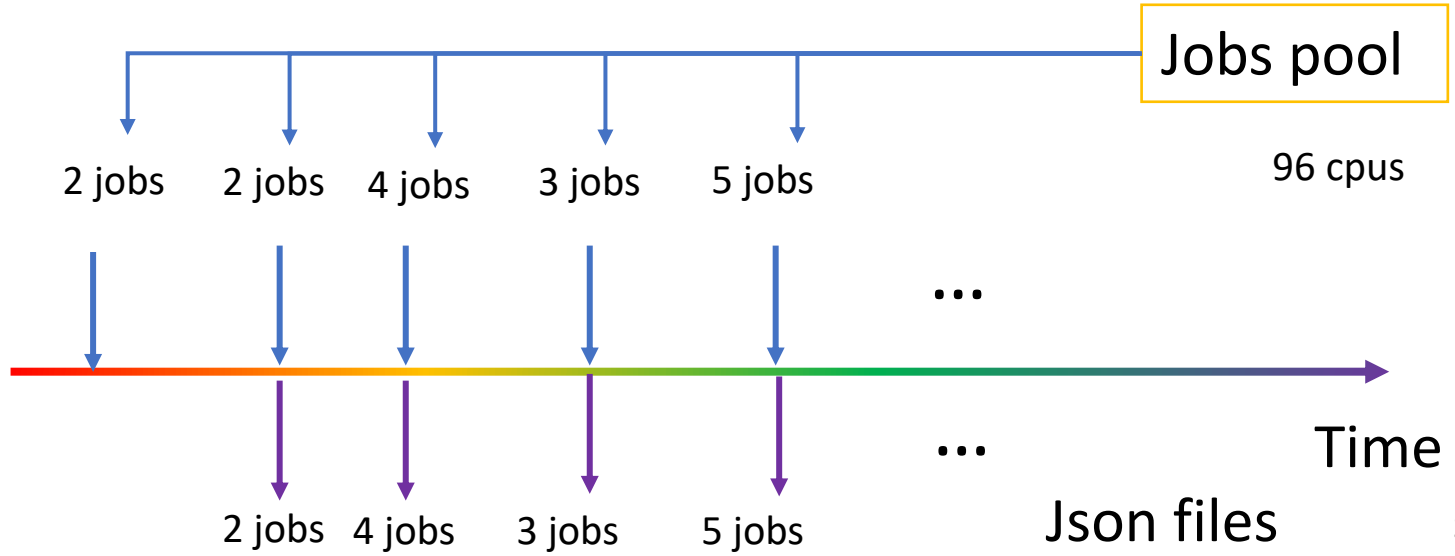
Concurrent.futures



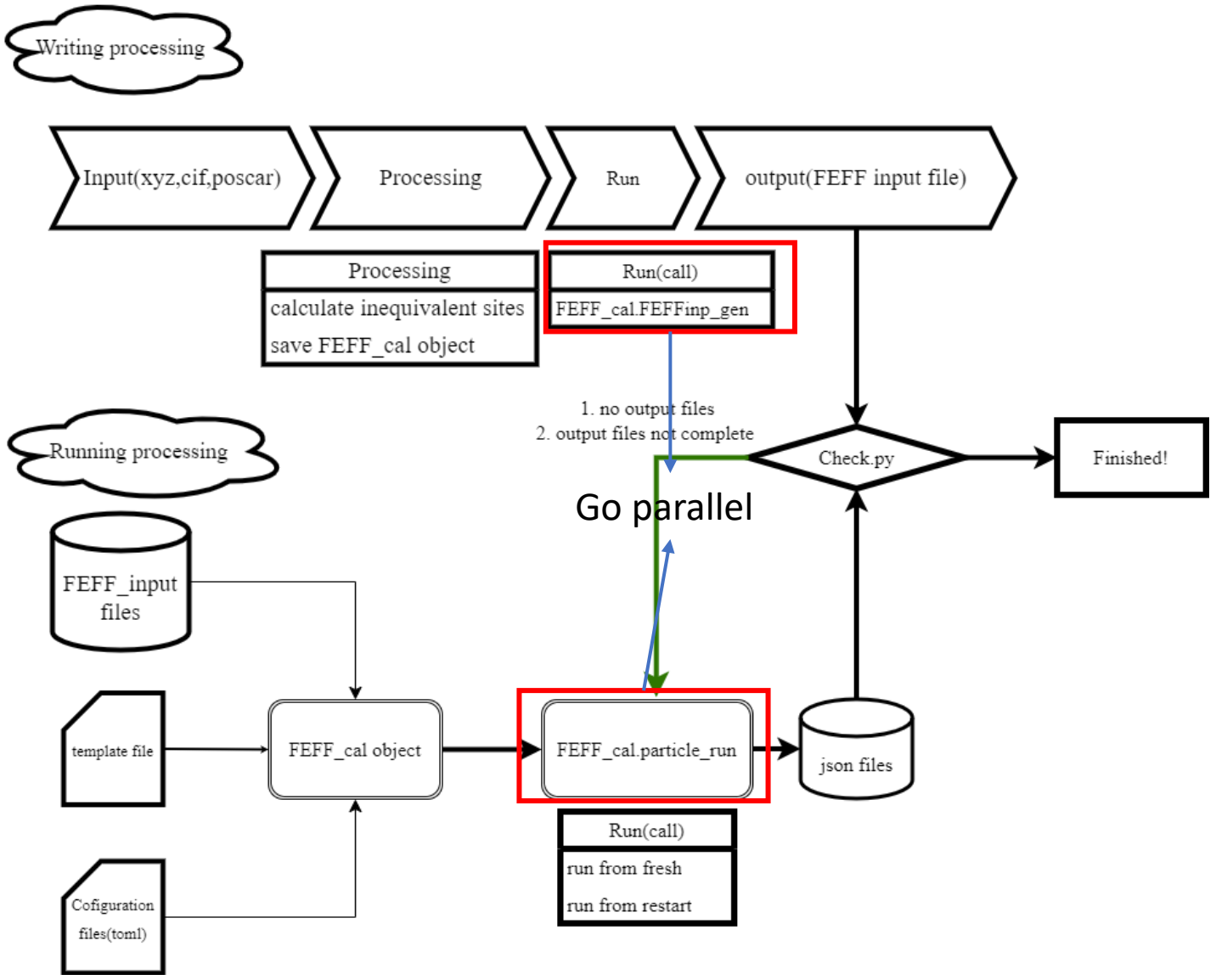
- Each particle has inequivalent sites
- Particles with different shape

- Calculate Averaged spectrum for each particle after the calculation

Accelerate FEFF calculations by running multiple jobs simultaneously.



The workflow of calculations



FEFF_cal object

functions

FEFFinput_gen: write FEFF input files

Particle_run: run FEFF calculations

Notes

1. We can specific absorber site or calculate sites based on symmetry
2. The input files can be xyz, cif, poscar
3. The code is easy to transfer to another project(XANES, EXAFS)
4. Template file contains the FEFF input header

Toolbox

1. average(output: csv)
2. Check.py(delete error calculations)

The backdraw for using concurrent.futures

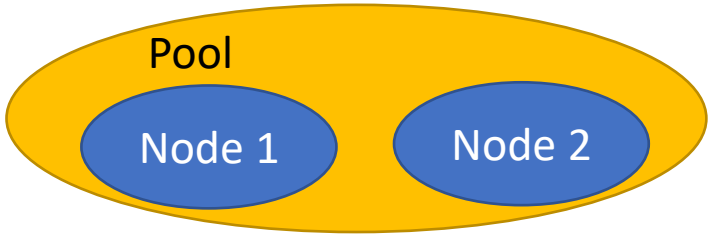
- It cannot run on multiple nodes.
- restrict us to run FEFF calculation on one 96 core node

Submit multiple slurm jobs

Queue	Default run time	Max run time	Max # of nodes	Min # of nodes	Max # of simultaneous jobs per user
debug-28core	1 hour	1 hour	8	n/a	n/a
extended-24core	8 hours	7 days	2	n/a	4
extended-28core	8 hours	7 days	2	n/a	6
gpu	1 hour	8 hours	2	n/a	2
gpu-long	8 hours	48 hours	1	n/a	2
gpu-large	1 hour	8 hours	4	n/a	1
p100	1 hour	24 hours	1	n/a	1
v100	1 hour	24 hours	1	n/a	1
large-24core	4 hours	8 hours	60	24	1
large-28core	4 hours	8 hours	80	24	1
long-24core	8 hours	48 hours	8	n/a	6
long-28core	8 hours	48 hours	8	n/a	6
medium-24core	4 hours	12 hours	24	8	2
medium-28core	4 hours	12 hours	24	8	2
short-24core	1 hour	4 hours	12	n/a	8
short-28core	1 hour	4 hours	12	n/a	8

Queue	Default run time	Max run time	Max # of nodes	Min # of nodes	Max # of simultaneous jobs per user
extended-40core	8 hours	7 days	2	n/a	3
extended-96core	8 hours	7 days	2	n/a	3
large-40core	4 hours	8 hours	50	16	1
large-96core	4 hours	8 hours	38	16	1
long-40core	8 hours	48 hours	6	n/a	3
long-96core	8 hours	48 hours	6	n/a	3
medium-40core	4 hours	12 hours	16	6	1
medium-96core	4 hours	12 hours	16	6	1
short-40core	1 hour	4 hours	8	n/a	4
short-96core	1 hour	4 hours	8	n/a	4
a100	1 hour	8 hours	2	n/a	2
a100-long	8 hours	48 hours	1	n/a	2
a100-large	1 hour	8 hours	4	n/a	1

Beyond concurrent.futures




```
from mpi4py import MPI
from mpipool import MPIExecutor
```

comm=MPI.COMM_WORLD
rank=comm.Get_rank() ← Get CPU tag
pro_size=comm.Get_size() ← Get Pool size
Name=MPI.Get_processor_name() ← Get node name

mpi4py MPI for python package

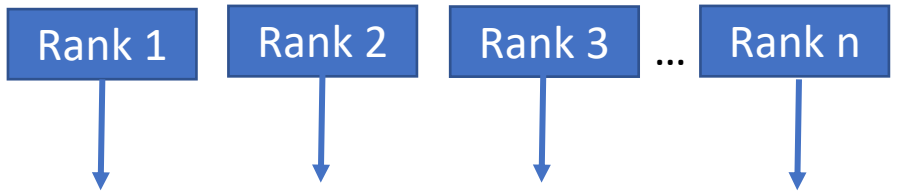
mpipool Offers MPI based parrel execution of tasks through implementations of Python's standard library interfaces such as multiprocessing and concurrent.futrues

Some concepts for parallel computing:

mpirun -n 240 python -m mpi4py FEFF_run_v3.py –w/-r

Rank:

Host: Rank 0 → Inequivalent sites calculations, read files…



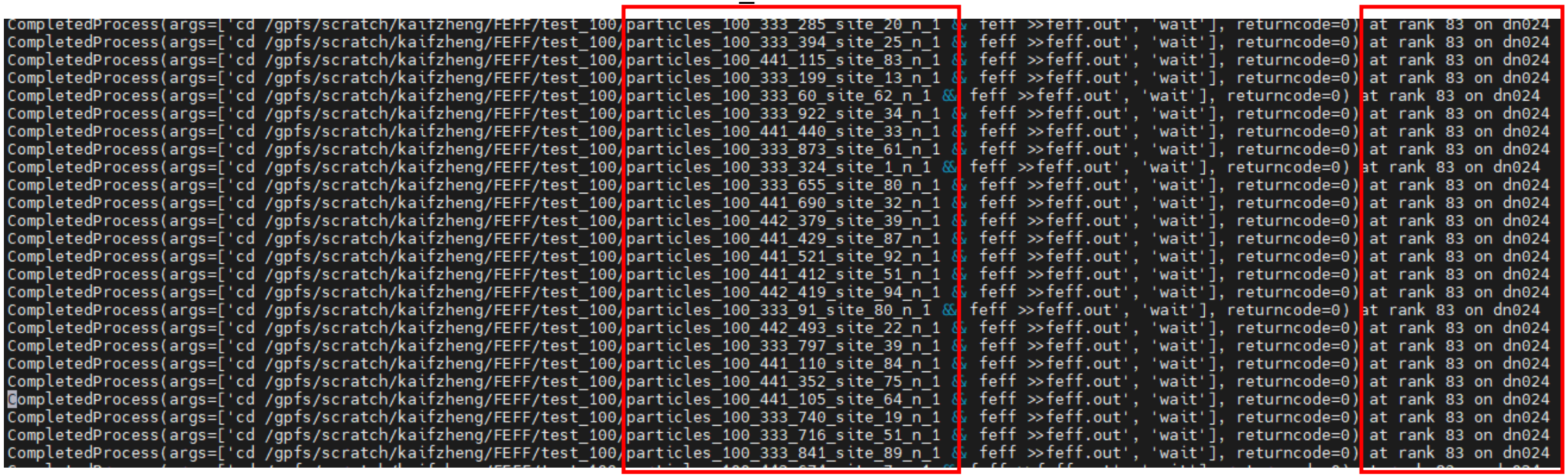
From rank=0:
comm.send(data,dest=i,tag=i)
To rank=i:
Comm.recv(source=0,tag=rank)

Writing and running processes…

Some results

Particle_site name

CPU name I'm running on



Finished 60465 sites on 100-atom particles with in 24 hr on 240 cpus(6 40core partition). Total:299686 needs 5 days

- Old result
  - 55000 FEFF calculations for 55-atom particles need 32hrs to finish, which means 1 min finish 28 jobs!

Configurations

```
#specific directory
template_dir = "template.inp"
scratch = "/gpfs/scratch//FEFF"
name='test_200'

#specific the method for writting FEFF inputfiles
particle='particle' #atom: specific site, particle: finding the inequivalent sites
CA = "Pt" #absorber atom type
radius = 100 #How large of the size of particle for this calculation

cutoff = 9 #radius for calculating symmetry using distance matrix method.
file_type="*.xyz" #specific the format of input file(we can also use cif, POSCAR)

#calculation specification
mode = "seq_multi"#the method to use(seq_seq, seq_multi,multi_multi and multi_seq)
cores = 1#number of cpu to use(unused)
tasks = 1#number of tasks go parallel(unused)
site = [19] #site should greater than or equal to 1(use it if you don't want to calculate inequivalent sites)
restart=false


#run SCF_test to check rSCF and rFMS
SCF_test=false
#rSCF=[5,7,8,9,10,15]
#rFMS=[3,5,7,8,9,10,12,15,17,20]
#rSCF=[3,4]
#rFMS=[3,4]


####future####
average=false
```

Specific the running directory

Note:
1st: sequential(seq)/parallel(multi)
2nd:sequential(seq)/parallel(multi)
for FEFF calculation

Do we get the unlimited speed to calculate FEFF?

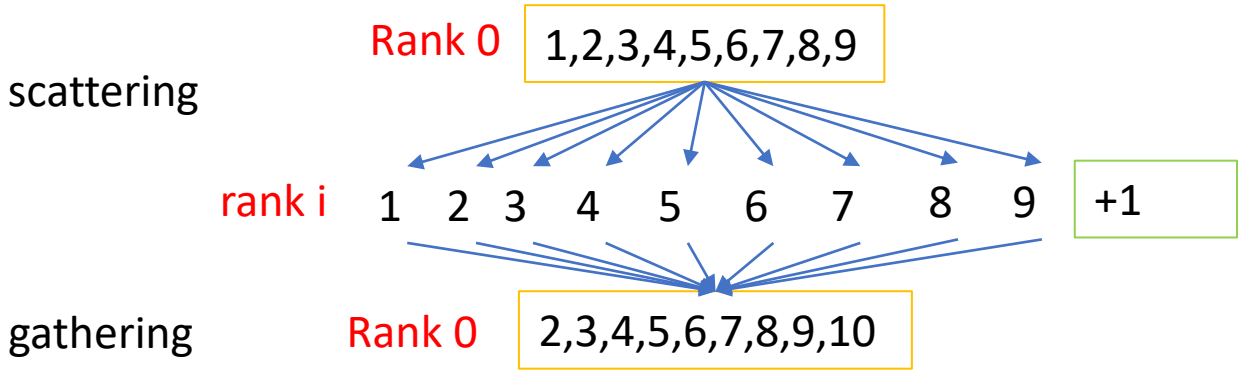
No!

The communication between nodes are slow. I tried to parallel on 1000 cpus, it was not work. It takes forever to start.

For writing process, I used two methods to calculate inequivalent sites:point group(support in Pymatgen(very slow for large particle)), distance matrix(consider a partial structure within a radius of a particle to get inequivalent sites(fast but not accurate))

Future:

1. Update the codes by using parallel features(scattering, broadcasting, gathering…)
2. Optimize our algorithms for writing and running(I/O operations are very slow).



The functionality of my codes

- 1. calculate site-specific or particle-averaged spectrum
- 2. two symmetry calculations are available(point-group and distance matrix)
- 3. we can restart the calculation from the checkpoint.
- 4. there is a SCF-test calculation for FEFF parameters available in the code.
- 5. it can read from xyz, cif, and POSCAR.

My codes:

https://github.com/kaifengZheng/FEFF_package.git

Special thanks to :

Ryuichi Shimogawa

Mehmet Topsakal