An overview of current computational facilities at our CREI

Sergey Matveev [1,2,3]

[1]Skolkovo Institute of Science and Technology

[2]Lomonosov Moscow State University

[3]Institute of Numerical Mathematics of Russian Academy of Science

October 31, 2015

Skoltech

Skolkovo Institute of Science and Technology

Outline of the talk

- List of hardware and software
- Performance benchmarks
- How to use all this stuff?

Our hardware

- Central Skoltech cluster
  1. Host address is **cluster.skoltech.ru**; accessible via SSH, Putty etc
  2. 18 Nodes with 16 CPU-cores and 62 Gb RAM per node.
  3. 8 Tb available HDD protected with **RAID0+1**-system
  4. Served by IT-dept
- Our CREI mini-server
  1. Host address is **10.30.100.28**; accessible via SSH, Putty etc
  2. Regular Linux workstation (4-cores CPU with 16 Gb RAM and 1 Tb HDD)
  3. Fully administrated by CREI-members

Our software

- Academic licences for use of **Gaussian**, **VASP**, **Mopac**
- Free software: **LAMMPS**, **Tinker**, **Primme** (compiled in collaboration with Anatoly Dymarsky)
- Our own open-source package **MolMod**

CLuster benchmaking

Table : Intel MKL LinPack benchmarks on numerical solution of linear systems using LU-factorization

Size of the system	N processor cores	Performance(Gflops)
16 000	16	214.85
	32	295.35
	64	441.53
	128	528.84
32 000	32	427.77
	64	631.85
	128	967.81
64 000	64	829.04
	128	1336.1

`https://software.intel.com/en-us/articles/intel-math-kernel-library-linpack-download`

Cluster benchmarking

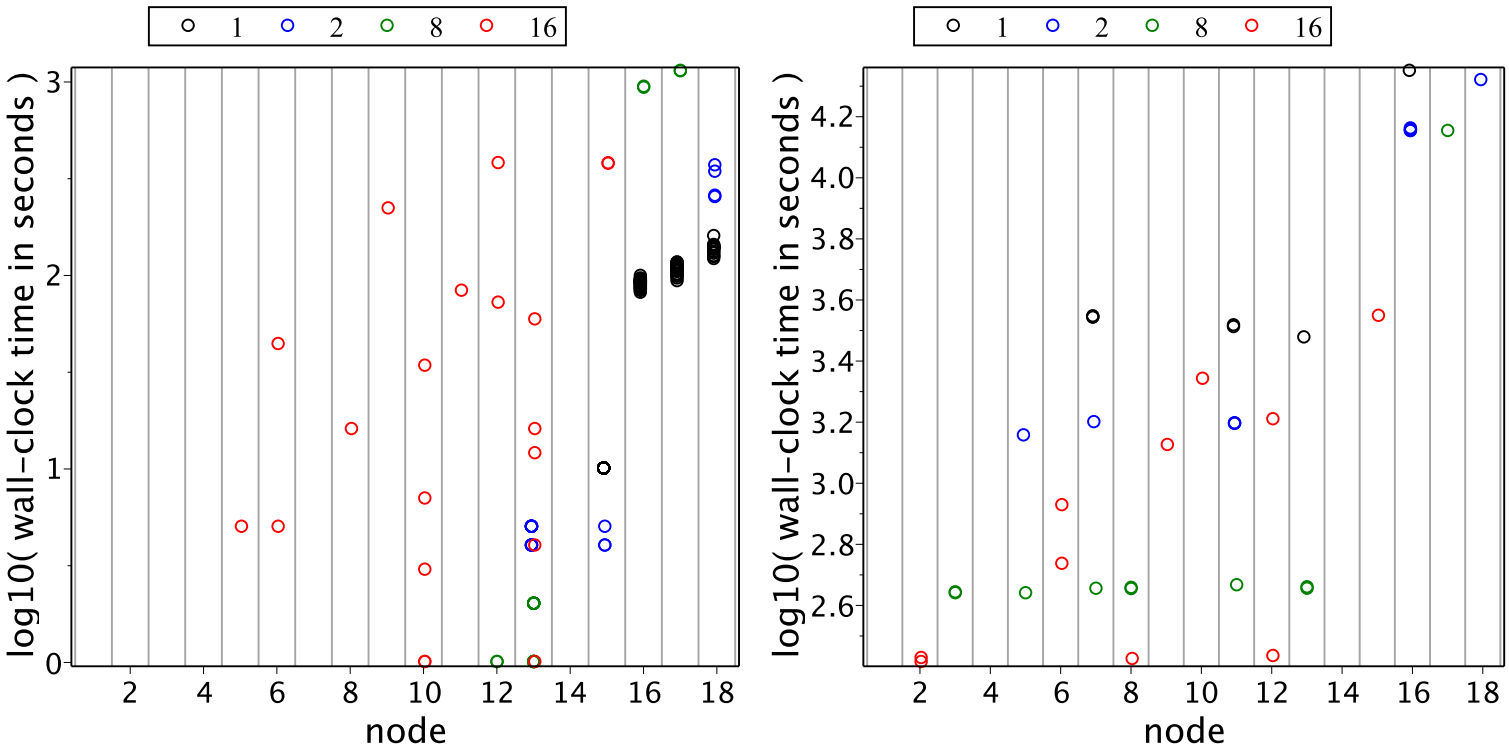


Figure : Test of LAMMPS performance with 2 000 and 64 000 atoms

How to use cluster scheduler

At our cluster we have a standard **Torque PBS** scheduler with the following resource per user constrains:

- 500 Gb HDD-space
- 20 simultaneously submitted jobs
- 128 CPU-cores per 1 job

The most common PBS commands are:

- **qstat** – get current cluster usage info
- **qsub** – submits some job
- **qdel** – cancels some job

How to use cluster scheduler; Example

- Trivial example of interactive PBS job submission:
  echo "sleep 10; echo helloworld" | qsub -N Helloworld -l nodes=1:ppn=1:pmem=2Gb
  As a result you will get files:
  Helloworld.e**JOB-NUMBER** – standard error stream
  Helloworld.o**JOB-NUMBER** – standard output stream
- Trivial example of PBS-script
  submit.qs
  #!/bin/bash
  #PBS -l nodes=8:ppn=4:mem=2Gb
  #PBS -l walltime=40:00
  echo helloworld
- qsub submit.qs

Example: calculations in MOPAC

Sergey Matveev, Andriy Zhugayevych

```
> with(MolMod):
```

Set the workfolder and method

```
> sys,method,PG:= "biphenyl", "PM7", "D2";
  fld:=cat(sys,"/");
```

Build molecular model

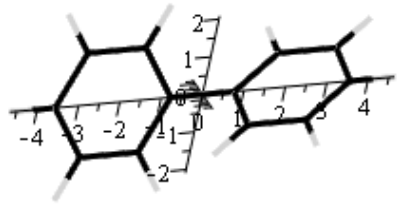
```
> CC,CC1,CH,phi:= 1.40, 1.53, 1.09, 30;
```

Build fundamental domain

```
> A:=UnfoldBySymmetry([["C",<CC,0,0>],["H",<CC+CH,0,0>]],"C6"):
  FD:=remove(v->v[2][1]<0 or v[2][2]<0,map(v->[v[1],RotationM(-
  phi*Pi/360,<1.,0,0>).(v[2]+<CC+CC1/2,0,0>)],A));
  plots[display]([plotMol(FD)],scaling=constrained);
```

Build molecule

```
> Atoms:=UnfoldBySymmetry(FD,PG):
  plots[display]([plotMol(Atoms)],scaling=constrained);
```



Save it

```
> WriteXYZ(cat(fld,"model.xyz"),Atoms,cat("model of ",sys),
  overwrite);
```

Optimize geometry

Read initial geometry

```
> Atoms:=ReadXYZ(cat(fld,"model"),printout):
model of biphenyl //22
```

Write MOPAC input

```
> WriteInput(cat(fld,method,"_em"), "mop", method, "em", Atoms)
  ;
```

Run the following file from OS

```
> printf(cat(fld,method,"_em.mop"));
  OpenFolder(cat(fld,method,"_em.mop"));
biphenyl/PM7_em.mop
```

Clean-up

```
> CleanUp(cat(fld,method,"_em.mop"));
```

Read and save optimized geometry

```
> Atoms_opt := ReadAtoms(cat(fld,method,"_em")):
```

```
WriteXYZ(cat(fld,method,".xyz"),Atoms_opt,cat(sys," optimized
by ",method),overwrite);
```

Symmetrize atoms if needed and save symmetrized geometry

```
> Atoms_rot:=CanonicOrientation(Atoms_opt):
  Atoms_sym:=SymmetrizeAtoms(Atoms_rot,PG,printout);
  WriteXYZ(cat(fld,method,".xyz"),Atoms_opt,cat(sys," optimized
  by ",method,", symmetrized by ",PG),overwrite):
 #  symmetry     MSD     distances
 2   2^1 [0,0,1] 0.017   [22,21] 0.005
 3   2^1 [1,0,0] 0.019   [14,14] 0.006
 4   2^1 [0,1,0] 0.008   [14,13] 0.003
```

Scan dihedral

Read initial geometry

```
> Atoms:=ReadXYZ(cat(fld,method),printout):
biphenyl optimized by PM7, symmetrized by D2 //22
```

Select the dihedral manualy...

```
> lsi:=[8,12,11,7];
```

$$lsi := [8, 12, 11, 7] \tag{3.1}$$

... or automatically

```
> C:=ConnectAtoms(Atoms,output="Vector"):
  F:=CyclesAndTrees(GraphTheory[Graph](ConnectAtoms(Atoms,
  output="Matrix"))):
  for p from 1 to nops(F) do printf("%2d %a\n",p,F[p]) end;
  i,j:=F[1][2][1],F[2][2][1];
  lsi:=[ remove(`=`,C[i],j)[1], i, j, remove(`=`,C[j],i)[1] ];
```

Measure the old dihedral

```
> dihedral(seq(Atoms[i][2], i = lsi));
```

$$122.563253 \tag{3.2}$$

Scan

```
> list_of_jobs := []:
  for _angle from 0 by 3 to 180 do
     Atoms2 := ChangeDihedral(Atoms, lsi, _angle);
     f:=cat(method, "_scan", round(_angle));
     list_of_jobs :=[op(list_of_jobs), f];
     WriteInput(cat(fld,f), "mop", method, "sp", Atoms2)
  end do:
  WriteScript(cat(fld,"run.bat"), "mop", list_of_jobs,
  overwrite,printout=[3,4]):
```

Run the batch file from OS

```
> OpenFolder(cat(fld,"run.bat"));
```

Clean-up

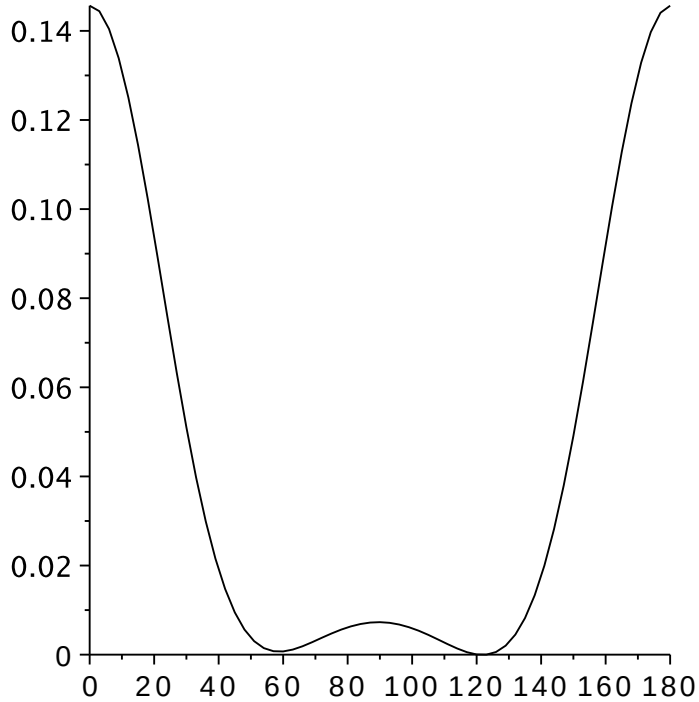
```
> for f in list_of_jobs do CleanUp(cat(fld,f,".mop")) end;
  fremove(cat(fld,"run.bat"));
```

Read and plot PES

```
> Energies:=[seq([sscanf(f,cat(method,"_scan%d"))[1],ReadEnergy
  (cat(fld,f))],f=list_of_jobs)]:
  E0:=min(map2(op,2,Energies));
  Energies:=[seq([v[1],v[2]-E0],v=Energies)];
```

$$E0 := -1608.384640 \tag{3.3}$$

```
> plot(Energies);
```



- **Molecular dynamics**

Further directions

- Construct CREI-wide minicluster in order to be less dependent to IT-dept emergencies
- Enlarge and upgrade the MolMod toolset
- Help and instruct users: Andriy, Artem, Stanislav ...
- Calculate more data

- The possible guidelines on PBS scheduler are available at:
  `http://cluster2.inm.ras.ru/torque.html` – in Russian

  `http://www.rcc.uh.edu/hpc-docs/`

  `49-using-torque-to-submit-and-monitor-jobs.html`
- Brief guides and examples on use of packages from Andriy
  `http://zhugayevych.me/soft/index.htm`
- MolMod package
  `http://zhugayevych.me/maple/MolMod/index.htm`
- LAMMPS

  `http://lammps.sandia.gov/`

- Gaussian

  `http://www.gaussian.com/`
- VASP

  `https://www.vasp.at/`
- MOPAC

  `http://openmopac.net/`
- Tinker

  `http://dasher.wustl.edu/tinker/`
- Primme

  `http://www.cs.wm.edu/~andreas/software/`

Thank you for your attention!