C2115 Practical Introduction to Supercomputing

8th Lesson

Petr Kulhánek, Jakub Štěpán

kulhanek@chemi.muni.cz

National Centre for Biomolecular Research, Faculty of Science
Masaryk University, Kotlářská 2, CZ-61137 Brno

INVESTMENTS IN EDUCATION DEVELOPMENT

CZ.1.07/2.2.00/15.0233

Contents

- **Infinity**
  job, command overview, aliases
- **Application submitting**
  sander, pmemd, gaussian, parallel runs
- **Exercise**
  parallel runs efficiency of sander, pmemd, gaussian applications

Infinity

- https://lcc.ncbr.muni.cz/whitezone/development/infinity/
- Job
- Command overview
- Aliases

Command overview

Software management:

- site — logical resources activation
- module — software activation/deactivation

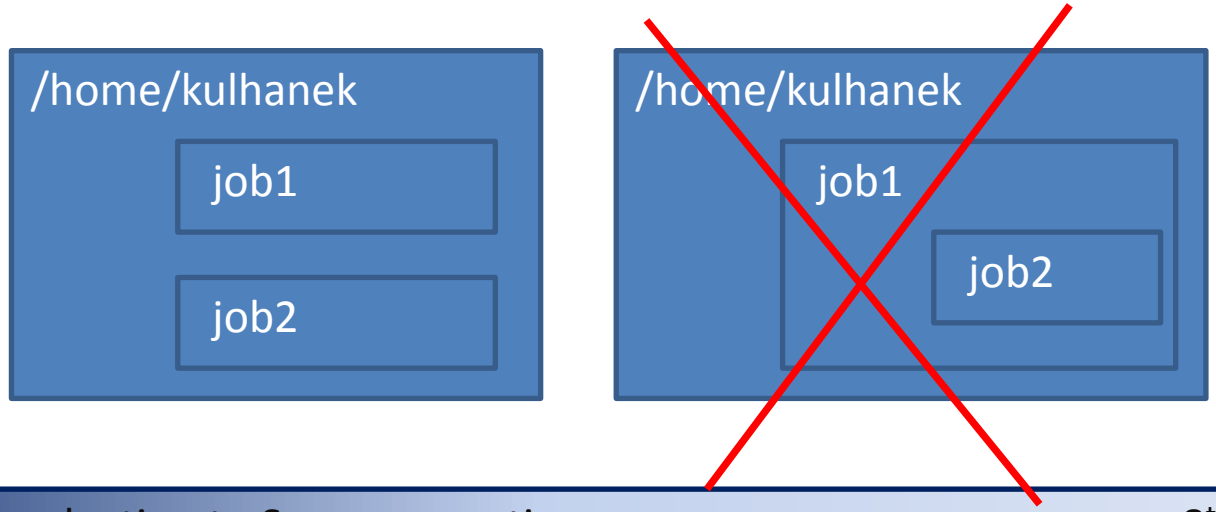
Job management:

- pqueues — batch system queues accessible to user list
- pnodes — computational nodes accessible to user
- pqstat — list of all jobs in batch system
- pjobs — list of current user jobs in batch system
- psubmit — submitting job to batch system
- pinfo — job information
- pgo — connection to job main node
- paliases — aliases definition

Job

Job has to keep following conditions:

- Each job is submitted in separate directory
- All input data has to be placed in job directory
- Job directories must not be nested
- Job run is controlled by script (or input file for automatically detected jobs)
- Job script has to be in bash
- Job script must not use absolute paths, all paths are relative to job directory



Job script

Script may be introduced by standard bash interpreter **bash** or special interpreter **infinity-env**, that ensures that script is not run except in batch system. Latter approach prevents damage/overwrite/deleting of already calculated data by unintended script start.

```
#!/bin/bash

# script itself
```

```
#!/usr/bin/env infinity-env

# script itself
```

Job submit

Job is submitted in **job directory** by command **psubmit**.

```
psubmit destination job [resources] [syncmode]
```

destination (where) is:

- queue_name
- machine_name@queue_name

job is:

- Job script name
- Input file in case of automatically recognized jobs

resources is description of requested job resources, if none given, then 1 CPU is default

syncmode defines way of data transfer mode between job directory and calculation node, default mode is "sync"

Job run monitoring

Job run may be monitored by command **pinfo**, this is run either from job directory or in computational node work directory. Other possibilities are commands **pjobs** and **pqstat**.

If job is in running mode on computational node, then command **pgo** can be used to access the node and change to job work directory there.

Management files

In job directory there are management files created upon job submission to batch system and its run and finalization. List of files and its contents follows:

- *.info control file with information about job state
- *.infex actual script (wrapper), that is run by batch system
- *.infout standard output of *.infex script, **in case of non-standard job termination contains useful informations**
- *.nodes list of computational nodes dedicated to job
- *.gpus list of GPU cards dedicated to job
- *.key job unique identifier
- ***.stdout standard output of job script**

Running applications

- sander
- pmemd
- gaussian
- Parallel running

sander

sander program is dedicated to molecular modeling. Detail information is given in: http://ambermd.org

```
#!/bin/bash

# activate module amber containing applications
# sander a pmemd
module add amber

# applications run
sander -O -i prod.in -p topology.parm7 \
                     -c input.rst7
```

pmemd

pmemd program is dedicated to molecular modeling. Detail information is given in: http://ambermd.org

```
#!/bin/bash

# activate module amber containing applications
# sander a pmemd
module add amber

# spusteni aplikace
pmemd -O -i prod.in -p topology.parm7 \
                    -c input.rst7
```

sander/pmemd

Simulation length:

Simulation length (calculation) is defined by keyword (**nstlim**) given in file prod.in, that is equal to number of integration steps.

Simulation results are in files:

mdout

mdinfo <-- contains statistics information, for example how many ns per day is program able to calculate

mdcrd

restrt

sander/pmemd – parallel run

Parallel run differs only in submission to batch system by command **psubmit**. **Rest of input remains same!** (same input file, job script).

```
$ psubmit short test_sander ncpus=1
```

May be omitted

***.stdout**

```
.....
Module build: amber:12.0:x86_64:single
.....
```

Calculation node:

```
S %CPU %MEM     TIME+  COMMAND
R  100  0.6   1:13.37 sander
R    0  0.0   0:00.01 top
```

```
$ psubmit short test_sander ncpus=2
```

***.stdout**

```
.....
Module build: amber:12.0:x86_64:para
.....
```

Calculation node:

```
%CPU %MEM     TIME+  COMMAND
 100  1.6   0:40.41 sander.MPI
  99  1.7   0:40.60 sander.MPI
```

Exercise

Input data:
/home/kulhanek/Data/2115/data/sander/small

1. Run job on 1CPU on cluster wolf.
2. Run job on 2CPU on cluster wolf.

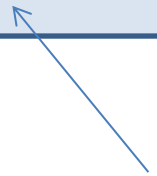
gaussian

gaussian program is dedicated to quantum-chemical calculations. Detail information are to be found at: http://www.gaussian.com

```
#!/bin/bash

# activate modeule gaussian
module add gaussian

# run application
g09 input
```



Input file **input.com** no extension

gaussian

Calculation length:
Calculation length is limited by maximum optimisation steps number (**MaxCycle** in input file).

Calculation result is in file:
input.log

gaussian – parallel run

Parallel run has to be set in resource definition for command psubmit **and input file for program gaussian (input.com) as well**.

input.com (first line)

```
%NProcShared=1
```

```
$ psubmit short test_gaussian ncpus=1
```

May be omitted

Calculation node:

```
S %CPU %MEM     TIME+  COMMAND
R  100  1.2  1:01.25 l502.exe
S    0  0.1  1:38.57 pbs_mom
```

input.com (first line)

```
%NProcShared=4
```

Identical number!!!

```
$ psubmit short test_gaussian ncpus=4
```

Calculation node:

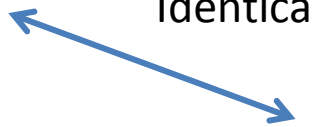
```
S %CPU %MEM     TIME+  COMMAND
R  399  1.1  0:49.38 l502.exe
S    0  0.1  0:00.90 init
```

gaussian – parallel run, II

Auto detection may be used by gaussian calculation submiting.

No auto detection:

input.com (first line)
%NProcShared=4



Identical number!!!

```
$ psubmit short test_gaussian ncpus=4
```

With auto detection:

input.com (does not need to contain
%NProcShared=4

```
$ psubmit short input.com ncpus=4
```

Exercise

Input data:
/home/kulhanek/Data/2115/data/gaussian

1. Run job with 1CPU on cluster wolf.
2. Run job with 2CPU on cluster wolf.

Exercise

- parallel runs efficiency of
  - sander
  - pmemd
  - gaussian

Exercise LVII.1

Exercise goal is to determine scaling of sander, pmemd, gaussian applicationon cluster SOKAR in range of CPU number 1, 2, 4, 8, 16, 32 a 64. For each CPU number select simulation length appropriate for particular CPU number. Then state theoretical simulation length, real speedup, and real CPU usage percent. Plot graph with real speedup as function of CPU number. Compare plot with ideal scaling graph.

Do testing calculation on cluster WOLF. Final calculations has to be done on cluster SOKAR. Make sure, that you have ssh keys setup done on SOKAR.
There are two nodes available with 64 CPU in queue **long**. Nodes has property **c2115**.

Request 64 CPU for all jobs during submit, proper CPU number will be selected "lowered" to desired value in script.

Input data are on cluster WOLF in directories:
/home/kulhanek/Data/2115/data /pmemd/medium/
/home/kulhanek/Data/2115/data /pmemd/small/
/home/kulhanek/Data/2115/data /pmemd/big/
/home/kulhanek/Data/2115/data /sander/medium/
/home/kulhanek/Data/2115/data /sander/small/
/home/kulhanek/Data/2115/data /sander/big/
/home/kulhanek/Data/2115/data /gaussian/

CPU number adjusting for testing

```
.....

INF_NCPU=N
AMS_NCPU=$INF_NCPU
# activate module amber containing applications
# sander a pmemd
module add amber
....
```

N = desired CPU number

```
$ psubmit long test_sander ncpus=64,props=c2115
```

Always 64

---

input.com (first line)

```
%NProcShared=N
```

N = desired CPU number

```
$ psubmit long test_gaussian ncpus=64,props=c2115
```

Always 64

Exercise LVII.2

According to manual on MetaCentra web pages run job in gaussianu, take input file from previous exercise.