AlphaFold Workshop

Lesson 28

Revision 1

Petr Kulhánek

kulhanek@chemi.muni.cz

Laboratory of Computational Chemistry
National Centre for Biomolecular Research
Faculty of Science
Masaryk University
Kamenice 5
CZ-62500 Brno

Infinity
Collections

Infinity Summary

Software management:

- **site** activation of logical computing resources
- **module** software activation/deactivation

Job management:

- **pqueues** overview of queues from the batch system
- **pnodes** overview of compute nodes available
- **pqstat** overview of all jobs submitted into the batch system
- **pjobs** overview of user jobs submitted into the batch system
- **psubmit** submit a job into the batch system
- **pinfo** job information
- **pgo** log into a compute node with a running job
- **psync** manual data synchronization
- **pcollection** collection management

Collections vs Job Arrays

Job Arrays: multiple instances of the same job

- cannot be altered
- difficult to detected failed jobs
- very difficult to resubmit the failed jobs

Collections: group of jobs

- jobs can be added/removed at any time
- easy to detect failed jobs and resubmit them
- easy to manipulate with the entire collection
- advanced statistics

Preparing collections, I

1. **Open a collection**

collection name

```
$ pcollection c1 open
```

if the collection does not exist, a new collection with the given name is created

2. **Submit jobs**

use only one possibility,
it disables manual confirmation of the job submission

```
$ pconfirmsubmit YES

$ cd job_dir1
$ psubmit -y queue job_name resources
$ cd ..

$ cd job_dir2
$ psubmit -y queue job_name resources
$ cd ..
```

multiple jobs can be submitted
jobs will be in the prepared state (P)

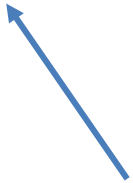
3. **Close the collection**

```
$ pcollection c1 close
```

Preparing collections, II

1. **Prepare a collection (Open, Submit jobs, and Close at once)**

```
$ pcollection c1 prepare queue job_name resources
```



the same syntax as the psubmit command

The command will recursively scan the current directory for the occurrence of the *job_name* script. For each occurrence, it enters the directory and submits the job with the provided specifications.

Limitations:

- It is currently impossible to exclude some directories from the recursive search, for example, with the job templates.
- Job directories with runtime files are ignored.

Manipulating with collections

Once the collection is prepared, the following actions are available:

Action	Description
submit	Submit all jobs in the prepared states and all jobs which were not finished yet. This action can be executed many times until all jobs are finished.
info	Print summary about jobs in the collection.
stat	Print statistics about jobs in the collection. This is equivalent to pinfo -r.
kill	Kill all queued and running jobs in the collections. Once jobs are killed, they cannot be resumed.

Syntax:

```
$ pcollection c1 [action]
```

The default action is "info".

Example

```
[kulhanek@perian 02.clients]$ pcollection c1

# Collection name : c1
# Collection path : perian.grid.cesnet.cz:/storage/brno12-
cerit/home/redshift/Projects/2022/01.DNA-BP/02.ol21/02.C-TCXGA/04.aGsT/08.abf-
repair/01.simple/02.clients
# Collection site : metavo
# Collection ID   : 36f22bf6-5b35-00f0-f715-05446fd17a04

# CID  ST    Job Name          Queue      NCPUs NGPUs NNods Last change/Duration  Metrics  Comp
# ---- -- -------------- --------------- ----- ----- ----- -------------------- --------- ----
    1 R  precycleJob     gpu                1     1     1           0d 00:23:19   43/  50  84%
    2 R  precycleJob     gpu                1     1     1           0d 02:19:29   41/  50  80%
    3 R  precycleJob     gpu                1     1     1           0d 01:16:22   43/  50  84%
    4 R  precycleJob     gpu                1     1     1           0d 00:28:59   36/  50  70%
    5 R  precycleJob     gpu                1     1     1           0d 01:43:42   42/  50  82%
    6 F  precycleJob     gpu                1     1     1  2023-02-22 17:30:56    6/  50  12%
    7 R  precycleJob     gpu                1     1     1           0d 01:42:02   44/  50  86%
    8 R  precycleJob     gpu                1     1     1           0d 01:55:52   43/  50  84%
    9 R  precycleJob     gpu                1     1     1           0d 02:36:37   45/  50  88%
   10 FE precycleJob     gpu                1     1     1  2023-02-23 05:44:01   39/  50  78%
# ------------------------------------------------------------------------------------------
# Waiting:    0 (  0%) | Processing:    8 ( 80%) | Finished: 372 ( 74%) | Total: 500
# Requires (re)submission:    2 ( 20%) | Inconsistent:    0 (  0%)
```

Example

- Collections can also be shown with the pjobs command (**option -g**)

```
[kulhanek@perian 02.clients]$ pjobs -u redshift -g
#
# Site name     : metavo
# Batch servers ...
# -> * M meta-pbs.metacentrum.cz
# ->   C elixir-pbs.elixir-czech.cz
# ->   E cerit-pbs.cerit-sc.cz
#

#                                                                                                  Util[%]
# ST    Job ID        User        Job Title         Queue      NCPUs NGPUs NNods          Times      CPU Mem Exit
# -- ------------ ------------ -------------- -------------- ----- ----- ----- ------------------------ --- --- ----
   R    14520591M redshift     run_server     oven               1     0     1  0d 20:08:40/20d 00:00:00   1  38
[  8] /storage/brno12-cerit/home/redshift/Projects/2022/01.DNA-BP/02.ol21/02.C-TCXGA/04.aGsT/08.abf-repair/01.simple/02.clients
>  R    14528975M redshift     precycleJob+040 gpu                1     1     1  0d 02:38:33/ 0d 03:00:00  98   5
>  R    14528980M redshift     precycleJob+036 gpu                1     1     1  0d 02:21:25/ 0d 03:00:00  98   6
>  R    14528994M redshift     precycleJob+039 gpu                1     1     1  0d 01:57:48/ 0d 03:00:00  97   6
>  R    14529049M redshift     precycleJob+039 gpu                1     1     1  0d 01:45:38/ 0d 03:00:00  98   4
>  R    14529050M redshift     precycleJob+041 gpu                1     1     1  0d 01:43:58/ 0d 03:00:00  98   4
>  R    14529064M redshift     precycleJob+041 gpu                1     1     1  0d 01:18:18/ 0d 03:00:00  97   4
>  R    14529213M redshift     precycleJob+036 gpu                1     1     1  0d 00:30:53/ 0d 03:00:00  93   3
   R    14529239M redshift     precycleJob+043 gpu                1     1     1  0d 00:25:23/ 0d 03:00:00  94   3

# Queued:          0 Requested NCPUs:       0 NGPUs:        0
# Running:         9 Allocated NCPUs:       9 NGPUs:        8
# Total(QR):       9         Finished:      0 Others:       0
```