UBDA Platform

Example for R with OpenMPI

User Guide

Version 2.0

15 Dec 2020

Revision History

Version	Date	Prepared By	Summary of Changes
1.0	Jul 16, 2018		Initial release
2.0	Dec 14, 2020		Update of sample program and software version

Table of Contents

1. Introduction

This document is shown a R example by using OpenMPI method running on the UBDA platform.

Note: User should first register a user account through UBDA website at: https://www.polyu.edu.hk/pfs/index.php/177729 to access the UBDA Platform

2. Perform the test

2.1 Login to ubdaplatform.polyu.edu.hk via SSH

2.2 Create the testing directory

```
$ mkdir -p $HOME/rmpi_104c
$ cd $HOME/rmpi_104c
```

2.3 Prepare the R program (Filename: rmpi_104.R)

```
$ cp /ubda/apps/examples/R/openmpi/rmpi_104c/* ./
```

2.4 Create local R environement (rmpi)

```
$ module purge
$ module load openmpi-4.0.1-gcc-8.3.0
$ module load anaconda3-2020.02
$ conda create --name rmpi
$ source activate rmpi
$ conda install r-essentials r-base
$ R
>> install.packages("Rmpi",
                    repos = "https://cloud.r-project.org",
                    configure.args =
                    c("--with-Rmpi-
include=/ubda/apps/openmpi/openmpi-4.0.1-gcc-8.3.0/include/",
                      "--with-Rmpi-
libpath=/ubda/apps/openmpi/openmpi-4.0.1-gcc-8.3.0/lib/",
                      "--with-Rmpi-type=OPENMPI"))
>> install.packages("foreach")
>> install.packages("doSNOW")
>> quit()
$
```

2.5 Prepare the job script file.

For select OpenMPI, use the following (Filename: rmpi_104.pbs)

```
#!/bin/sh
#
#PBS -N rmpi_104c
#PBS -q q2s01
#PBS -l nodes=4:ppn=26
#PBS -l walltime=36:00:00
#PBS -l pvmem=4gb
#PBS -j oe
#PBS -V
#
R_FILE="./rmpi_104c.R"
#
module load openmpi-4.0.1-gcc-8.3.0
module load anaconda3-2020.02
source activate rmpi

cd $PBS_O_WORKDIR
echo "Process will start at : "
date
echo
"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++"
## #run compute
mpirun -np 1 --mca blt self,openib --mca mpi_warn_on_fork 0
--mca btl_openib_allow_ib 1 R CMD BATCH --no-save --no-
restore $R_FILE

echo
"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++"
echo "Process end at : "
date

module purge
```

Please remind to change the values for your application.

```
#PBS -N rmpi_104c {your job name}
#PBS -l nodes=4:ppn=26 {your requested resource;nodes and
processors per node}
#PBS -l pvmem=4gb {memory per process, total 416GB across 4 nodes}
#PBS -q q2s01 {the job queue}
```

Example files can be found at:

/ubda/apps/examples/R/openmpi/rmpi_104c/

rmpi_104c.R
rmpi_104c.pbs

3 Job submission

3.1 Submit the script (***rmpi_104c.pbs***) to job queue.
A job ID number will be returned.

```
$ qsub rmpi_104c.pbs
39684.ubda-mgt01
```

3.2 Enquiry the submitted job status.

```
$ qstat -na

ubda-mgt01:
                                                                                  Req'd
Req'd       Elap
Job ID                  Username    Queue    Jobname          SessID  NDS   TSK   Memory
Time    S   Time
----------------------- ----------- -------- ---------------- ------ ----- ------ ------
--- --------- - ---------
39684.ubda-mgt01        ubdaev3     q2s01    rmpi_104c        176957     4    104
4gb  36:00:00 R  00:00:07
   ubda-d047/0-25+ubda-d046/0-25+ubda-d045/0-25+ubda-d044/0-25
```

Job status field name	Explanation	Example
JOB ID	Unique Job id.	39684.ubda-mgt01
Job name	Name for the job allocation	rmpi_104c
Queue name	Name of the job queue that the job has assigned.	q2s01
Username	Your NetID	ubdaev3
S	Job current status. Q = queued R = Job is executing C = Job was completed	Job is executing
ELAP TIME	Time for the job executed	00:00:07
Req'd Time	The maximum execution time	36:00:00
NDS	Total number of nodes assigned	4
TSK	Total number of cores assigned	104

4 How to check the result

4.1 Check for any error messages for your job with the JobID

```
$ more rmpi_104c.o39684
Process will start at :
Mon Dec 14 17:07:55 HKT 2020
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
--------------------------------------------------------------------------------
--------
MPI_ABORT was invoked on rank 0 in communicator <Unknown>
with errorcode 0.

NOTE: invoking MPI_ABORT causes Open MPI to kill all MPI
processes.
You may or may not see output from other processes, depending on
exactly when Open MPI kills them.
--------------------------------------------------------------------------------
--------
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Process end at :
Mon Dec 14 17:08:16 HKT 2020
WARNING: $PATH does not agree with $PATH_modshare counter. The
following directories' usage counters were adjusted to match. Note
that
this may mean that module unloading may not work correctly.
 /ubda/apps/anaconda/anaconda3-2020.02//bin
```

4.2 Check for result

```
$ more rmpi_104c.Rout
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-conda_cos6-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
> hello.world <- function(i) {
```

```
+    sprintf('Hello from loop iteration %d running on rank %d on
node %s',
+        i, mpi.comm.rank(), Sys.info()[c("nodename")]);
+ }
>
> ## Use snow backed by Rmpi
> library(Rmpi)                                    # for mpi.*
> library(foreach)
> library(doSNOW)
Loading required package: iterators
Loading required package: snow
>
> ## Total No. to Workers e.g 52 processors, 1 master & nsalve
should be n-1=51
> ## Get the Total No. of Processor from ENV.
> nslaves <- as.numeric(Sys.getenv(c("PBS_NP")))-1
> nslaves
[1] 103
>
> cl <- makeCluster(nslaves, type="MPI")        # Create cluster
object with the given number of slaves
    103 slaves are spawned successfully. 0 failed.
> registerDoSNOW(cl)                            # Register the
cluster object with foreach
>
> start <- Sys.time()
> max_loop <- 300
> output.lines <- foreach(i = (1:max_loop)) %dopar% {
+    hello.world(i)
+ }
> cat(unlist(output.lines), sep='\n')
Hello from loop iteration 1 running on rank 1 on node ubda-d047
Hello from loop iteration 2 running on rank 2 on node ubda-d047
Hello from loop iteration 3 running on rank 3 on node ubda-d047
Hello from loop iteration 4 running on rank 4 on node ubda-d047
Hello from loop iteration 5 running on rank 5 on node ubda-d047
Hello from loop iteration 6 running on rank 6 on node ubda-d047
Hello from loop iteration 7 running on rank 7 on node ubda-d047
Hello from loop iteration 8 running on rank 8 on node ubda-d047
Hello from loop iteration 9 running on rank 9 on node ubda-d047
Hello from loop iteration 10 running on rank 10 on node ubda-d047
Hello from loop iteration 11 running on rank 11 on node ubda-d047
Hello from loop iteration 12 running on rank 12 on node ubda-d047
Hello from loop iteration 13 running on rank 13 on node ubda-d047
Hello from loop iteration 14 running on rank 14 on node ubda-d047
--More--(11%)
```

5 Useful Reference

- Manpage for OpenMPI - mpicc
  URL: https://www.open-mpi.org/doc/v3.0/man1/mpicc.1.php

- Manpage for OpenMPI - mpirun
  URL: https://www.open-mpi.org/doc/v3.0/man1/mpirun.1.php

- Manpage for R
  URL: https://cran.r-project.org/manuals.html