Notes on the SNOW/Rmpi R packages with OpenMPI and Sun Grid Engine

Last updated: 6/2/2008 4:43PM EDT

We informally discuss the basic set up of the R Rmpi and SNOW packages with OpenMPI and the Sun Grid Engine (SGE) job manager. Some knowledge of R, MPI and SGE is assumed. Elemetary examples are provided to verify basic operation. The flow of this document is linear. The sections should be completed in the order presented.

The following software environment was used to prepare the notes:

- OS: Ubuntu GNU/Linux 7.10, kernel 2.6.22-14-generic (x86_64, SMP) on a small cluster of four hosts with shared NFS /home directories.
- Compiler: gcc version 4.2.1 (Ubuntu 4.2.1-5ubuntu4) (gfortran)
- R: R version 2.7.0 (2008-04-22)
- Grid Engine: ge-6.1u4 (N1 Grid Engine 6)
- OpenMPI: 1.2.6
- Rmpi: Rmpi_0.5-5.tar.gz
- SNOW: snow_0.2-9.tar.gz

We assume that the test cluster hosts are named n1, n2, n3, and n4.

Contents

Conventions

<angle bracketed terms> are placeholders for non-optional parameters that must be filled in based on context

[square bracketed terms] are placeholders for optional parameters

We generally assume a Bourne shell (sh, bash).

All displayed instructions may be carried out by a non-root user. Root privileges are invoked by sudo when required. Note that SGE installation requires root privileges at some point. The other software packages do not require special privilege and may be installed, e.g., in a user home directory.

Software References

- Sun Grid Engine is available from http://gridengine.sunsource.net
- R, Rmpi, and SNOW may be found at http://cran.r-project.org/
- SNOW home page http://www.stat.uiowa.edu/~luke/R/cluster/cluster.html
- OpenMPI is available from http://www.open-mpi.org/

Install R

```
wget http://cran.r-project.org/src/base/R-2/R-2.7.0.ta
tar xf R-2.7.0.tar.gz
cd R-2.7.0
./configure && make
# Alternatively, run ./configure --prefix=<R 2.7.0 ins                    a
# If make completes without error, continue with:
sudo make install
# Otherwise, it is likely that additional packages are required to
# compile R. Install required packages and try again.
# Note: we can just run make install (without the sudo) if the installation
```

```
# path does not require special access.
# Place the path to the R executable, <R install path>/bin/  in your
# ~/.profile, ~/.bashrc, ~/.cshrc, or other shell start up file.
```

Alternatively, one may install a vendor-supplied R distribution (RPM, Debian package).

Test R

<XXX write me>

Install OpenMPI

We'll compile OpenMPI with explicit Sun Grid Engine support. This will allow tight integration with OpenMPI and the GE scheduler, simplifying operation.

```
wget http://www.open-mpi.org/software/ompi/v1.2/downloads/openmpi-1.2.6.tar.bz2
tar -xf openmpi-1.2.6.tar.bz2
./configure --with-sge --prefix=<openmpi installation path> && make
# Note the --with-sge option
# Upon successful make, run
sudo make install
# Note: we can just run make install (without the sudo) if the
# installation path does not require special access.
# Place the path to the OpenMPI executables, <install path>/bin/
# in your profile, .bashrc, .cshrc, or other shell start up file.
#

# Configure OpenSSH/SSH so that you can run a secure shell
# across cluster hosts n1--n4 without passwords:

ssh-keygen -t rsa -f ~/.ssh/id_rsa -N ""
ssh-keygen -t rsa1 -f ~/.ssh/identity -N ""
chmod 700 ~/.ssh
cd ~/.ssh
cat *.pub >> authorized_keys
chmod 600 authorized_keys
```

Ubuntu Bash shell users only:

Disable (comment out) the following lines in ~/.bashrc:

```
# If not running interactively, don't do anything
#[ -z "$PS1" ] && return
```

Now, ssh to each host in the cluster to register its host key:

```
ssh n1 uptime
ssh n2 uptime
ssh n3 uptime
ssh n4 uptime
```

Test OpenMPI

Test 1: Verify Grid Engine Support

```
# The command:
ompi_info | grep gridengine
```

```
# Should yield output similar to:
      MCA ras: gridengine (MCA v1.0, API v1.0, Component v1.2)
      MCA pls: gridengine (MCA v1.0, API v1.0, Component v1.2)
```

Test 2: A simple MPI job

```
# Enter a trivial program called vars.c:
cat << EOF > vars.c
#include<stdio.h>
#include<stdlib.h>
int main ()
{
    system ("export");
    return 0;
}
EOF

# Now, try it out:
mpicc -o vars vars.c
mpirun -host n1,n2,n3,n4 -np 4 vars
```

The terminal should display the exported shell variables for four processes, including all the defined OMPI variables. Adjust your host names and number of hosts accordingly. Run mpicc and mpirun without arguments for additional help, or consult the OpenMPI documentation.

Install Sun Grid Engine

C-shell

Make sure that your GNU/Linux distribution includes a c-shell (many do not). Several Grid Engine scripts require /bin/csh. If you don't have one, install the appropriate package.

Download and Install

Follow the licensing, download and installation instructions from
http://gridengine.sunsource.net/downloads/61/download.html.

Basic installation proceeds via a sequence of scripts that automate most of the configuration tasks. Installation of SGE requires root access at some point. The SGE administrator account need not be the root user, but we assume that it is for simplicity.

Post-installation configuration

We assume the existence of a default queue named all.q. If this queue does not exists, run

```
qconf -aq all.q
```

as root and add the queue. At any time, the queue attributes may be modified with the command:

```
qconf -mq all.q
```

Make sure that the all.q queue contains the line

```
hostlist  n1 n2 n3 n4
pe_lista  orte
```

where the host names n1 n2 n3 n4 indicate the compute execution host names in your cluster. We will define the “orte” parallel environment in the next step.

Set up a OpenMPI/SGE parallel environment

```
qconf -ap orte
```

Edit the environment to look like:

```
pe_name           orte
slots             16
user_lists        NONE
xuser_lists       NONE
start_proc_args   /bin/true
stop_proc_args    /bin/true
allocation_rule   $round_robin
control_slaves    TRUE
job_is_first_task FALSE
urgency_slots     min
```

The “slots” attribute should indicate the total number of processing elements.

SSH

We will submit jobs using qsub and the qrsh mechanisms. The SGE qrsh command may not work out of the box on many GNU/Linux distributions. Follow the instructions/guidance here: http://gridengine.sunsource.net/howto/qrsh_qlogin_ssh.html to enable Grid Engine to use OpenSSH/SSH as a remote shell. In particular, run

```
qconf -mconf
```

and make sure that the rlogin_daemon line is set to:

```
rlogin_daemon                /usr/sbin/sshd -i
```

Finally, make sure that the SGE binary path is set in your PATH environment variable, and set the variable SGE_ROOT to the SGE installation path. Set these values in your ~/.bashrc, ~/.cshrc or similar shell init file.

Clearing errors in a queue

Grid Engine queues may enter error states, indicated by the “E” flag in the states output field of the qstat -f command. When this occurs, you should inspect the logs for the source of the error as indicated in the documentation. A useful command to clear the error states and bring the queue back up is:

```
qmod -c '*'
```

See the Sun documentation and notes at http://gridengine.info/articles/2008/01/20/understanding-queue-error-state-e for more information.

Test Grid Engine

Log in to any node in the cluster and run

```
qstat -f
```

You should see output similar to:

```
queuename                      qtype used/tot. load_avg arch          states
---------------------------------------------------------------------------------
all.q@n1                       BIP   0/4       0.25     lx24-amd64
---------------------------------------------------------------------------------
all.q@n2                       BIP   0/4       0.00     lx24-amd64
---------------------------------------------------------------------------------
all.q@n3                       BIP   0/4       0.12     lx24-amd64
---------------------------------------------------------------------------------
all.q@n4                       BIP   0/4       0.01     lx24-amd64
```

If any nodes show an error (E), then clear it before proceeding.

Test the Grid Engine installation by running one or more of the provided example job scripts. For example:

```
qsub $SGE_ROOT/examples/jobs/simple.sh
```

The simple.sh job prints the date, sleeps for 20 seconds, prints the date again, then exists. Output and error streams for the job should appear in your home directory. Inspect the running job with qstat -f.

Testing SGE and OpenMPI

Additional information is available from: http://www.open-mpi.org/faq/?category=running

For SNOW/Rmpi use, we only require the qsub and qrsh job submission methods. In each case, we'll run the simple vars program from the OpenMPI test above.

Interactive qrsh test

This is the simpler of the two tests. It depends on proper configuration of SGE and OpenSSH as outlined above and in the references. The 'orte' parallel environment must also be defined and enabled as outlined above.

The following command interactively submits an MPI run of four vars programs (q.v. OpenMPI Test above) through the orte parallel environment:

```
qrsh -V -cwd -pe orte 4 mpirun -np 4  vars
```

The command-line parameters mean:

1. −V forward current environment variables to the jobs
2. -cwd run out of the current working directory
3. -pe orte use the orte parallel environment

4 run with four processing element slots

the remainder being the usual OpenMPI command.

The job output will return to stdout. See the qrsh documentation for more information.

Batch qsub test

The qsub command requires creation of a job control script. Enter the following script, called vars.sh:

```
cat << EOF > vars.sh
#!/bin/sh
# Export all environment variables
#$ -V
# Set this job name
#$ -N vars
# Use current working directory
#$ -cwd
# Join stdout and stder into one file
#$ -j y
# The mpirun command; note the lack of host names.
mpirun -np $NSLOTS vars
EOF
```

Most of the commands mirror those in the qrsh interactive example. The above script may be submitted to SGE by qsub with

```
qsub -pe orte 4 vars.sh
```

Output will not be sent to stdout, but instead to an output file labeled as <job name>.o<job id>. See the qsub documentation for more info.

Installing Rmpi

Method 1: Interactive Operation

Additional information on Rmpi is available at the project home page: http://www.stats.uwo.ca/faculty/yu/Rmpi.

Download the Rmpi package from cran and install it in the usual way:

```
wget http://cran.rakanu.com/src/contrib/Rmpi_0.5-5.tar.gz
R CMD INSTALL Rmpi_0.5-5.tar.gz --configure-args=--with-mpi=<OpenMPI install path>
```

Testing Rmpi with OpenMPI

```
R
> library("Rmpi")
> mpi.spawn.Rslaves(nslaves=2)
      2 slaves are spawned successfully. 0 failed.
      master (rank 0, comm 1) of size 3 is running on: n1
      slave1 (rank 1, comm 1) of size 3 is running on: n1
      slave2 (rank 2, comm 1) of size 3 is running on: n1
> mpi.remote.exec(rnorm(5))
        X1         X2
  1 -1.7564038 -0.3259411
  2  0.4125642 -0.3917568
  3  0.6395533 -1.5784756
  4  0.3643705 -0.9501001
  5 -2.2256307 -0.2844620
> mpi.close.Rslaves()
 [1] 1
> q()
```

Caveat

Note that all the MPI processes in the above example were started on the local host n1. This can be changed by editing the `<OpenMPI-install-path>/etc/openmpi-default-hostfile` file and listing the desired execution hosts. **NOTE HOWEVER**, that this is not compatible with SGE or other scheduler systems operation.

Method 2: Batch operation

The Rmpi package includes a `.Rprofile` script to support batch operation. The following steps illustrate batch operation with mpirun using the `.Rprofile` startup file.

```
cp Rmpi/inst/Rprofile ~/.Rprofile
```

```
mpirun -host n1,n2,n3,n4 -np 4 R --no-save -q
```

This form of startup has the advantage over interactive operation that a host list can be supplied on the command line.

Testing Rmpi with OpenMPI and SGE

Method 1: Interactive sessions

Follow the instructions for installing the `~./.Rprofile` startup script distributed with the Rmpi package as outlined in the last section. The following example uses SGE to start an interactive session with four process slots (one master and three slaves):

```
qrsh -V -pe orte 4 mpirun -np 4 /home/blewis/R/bin/R --no-save -q
```

Method 2: Batch sessions

<Documentation in process>

Installing SNOW

The R Simple Network of Workstations (SNOW) package provides an easy-to-use, high-level interface to the OpenMPI message-passing protocol (and others). SNOW can function in batch mode and interactively with OpenMPI and SGE.

```
R CMD INSTALL snow_0.2-9.tar.gz
```

Add the line

```
export PATH=$PATH:<R install path>/lib64/R/library/snow/
```

to your ~/.bashrc file (c-shell users use SETENV instead).

Testing SNOW with OpenMPI and SGE

Method 1: Interactive sessions

Even though the Grid Engine is designed largely as a batch queueing system, it is possible to launch interactive R/SNOW sessions through the scheduler with the RMPISNOW script provided by the SNOW package.

This example starts an interactive session with four process slots (one master and three slaves), allocated by SGE:

```
qrsh -V -pe orte 4 mpirun -np 4 <R install path>/lib64/R/library/snow/RMPISNOW

R version 2.7.0 (2008-04-22)
Copyright (C) 2008 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

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.

> cl <- makeCluster()
cl <- makeCluster()
> clusterCall(cl, function() Sys.info()[c("nodename","machine")])
clusterCall(cl, function() Sys.info()[c("nodename","machine")])
[[1]]
nodename  machine
  "n2" "x86_64"

[[2]]
nodename  machine
```

```
  "n3" "x86_64"

[[3]]
nodename  machine
  "n1" "x86_64"

> stopCluster(cl)
stopCluster(cl)
> q()
q()
```

XXX Standard output from all R sessions may be visible in the terminal window. A comment about this behavior appears in the RMPISNOW script. This behavior is associated with OpenMPI and will be corrected in a future release.

Note that a reference to the running cluster, cl, must be initialized with the makeCluster () command; see the SNOW documentation for more information.

Only the master R process is interactive. Note that the master slot is assigned by SGE and may not be on the launching node.

Method 2: Batch sessions

Batch jobs using SNOW may be submitted by qsub as illustrated in the following simple example.

<in process>