[MPICH2 with Sun Grid Engine](http://idolinux.blogspot.com/2010/04/mpich2-with-sun-grid-engine.html)

A prerequisite to this install log is a configured and validated Sun Grid Engine (SGE) 6.2 installation. Read more about that at [Deploying Sun Grid Engine on a Cluster](http://idolinux.blogspot.com/2008/09/deploying-sun-grid-engine-on-cluster.html).

MPICH2 is a high-performance and widely portable implementation of the Message Passing Interface (MPI) standard (both MPI-1 and MPI-2). The goals of MPICH2 are: (1) to provide an MPI implementation that efficiently supports different computation and communication platforms including commodity clusters (desktop systems, shared-memory systems, multicore architectures), high-speed networks (10 Gigabit Ethernet, InfiniBand, Myrinet, Quadrics) and proprietary high-end computing systems (Blue Gene, Cray, SiCortex) and (2) to enable cutting-edge research in MPI through an easy-to-extend modular framework for other derived implementations. -[www.mcs.anl.gov](http://www.mcs.anl.gov/research/projects/mpich2/)

There are multiple ways to integrate MPICH2 and SGE:

* Tight Integration of the mpd startup method
* Tight Integration of the daemonless smpd startup method
* Tight Integration of the daemon-based smpd startup method
* Tight Integration of the gforker startup method

This build log will cover the first method with mpd. Download, build an install MPICH2.

# cd /usr/global/src/  
# wget http://www.mcs.anl.gov/research/projects/mpich2/downloads/tarballs/1.2.1p1/mpich2-1.2.1p1.tar.gz  
# tar xzvf mpich2-1.2.1p1.tar.gz  
# cd mpich2-1.2.1p1  
# ./configure --prefix=/usr/global/mpich2-1.2.1p1 2>&1 |tee config.log  
# make 2>&1 |tee build.log  
# make install 2>&1 |tee install.log  
# ln -s /usr/global/mpich2-1.2.1p1 /usr/global/mpich2  
# export PATH=/usr/global/mpich2/bin:$PATH  
# export LD\_LIBRARY\_PATH=/usr/global/mpich2/lib:$LD\_LIBRARY\_PATH  
# which mpicc  
/usr/global/mpich2/bin/mpicc

Configure the queue shell.

# qconf -mq all.q  
shell /bin/bash  
shell\_start\_mode unix\_behavior

Download and install integration scripts.

# cd /usr/global/src/  
# wget http://gridengine.sunsource.net/howto/mpich2-integration/mpich2-62.tgz  
# cd /usr/global/sge/  
# tar xzvf /usr/global/src/mpich2-62.tgz  
# cd mpich2\_mpd/src  
# ./aimk  
# ./install.sh

Create the parallel execution environment template file by pasting the following into a *mpich2\_mpd.template* file:

pe\_name mpich2\_mpd  
slots 64  
user\_lists NONE  
xuser\_lists NONE  
start\_proc\_args /usr/global/sge/mpich2\_mpd/startmpich2.sh -catch\_rsh $pe\_hostfile \  
 /usr/global/mpich2  
stop\_proc\_args /usr/global/sge/mpich2\_mpd/stopmpich2.sh -catch\_rsh \  
 /usr/global/mpich2  
allocation\_rule $round\_robin  
control\_slaves TRUE  
job\_is\_first\_task FALSE  
urgency\_slots min  
accounting\_summary FALSE

Import the pe config.

# qconf -Ap mpich2\_mpd.template

Add the pe to the allowed list in your queue.

# qconf -mq all.q  
pe\_list make mpich2\_mpd

Enable killing of the processes at the end of a job.

# qconf -mconf  
execd\_params ENABLE\_ADDGRP\_KILL=TRUE

As a normal user, submit a test job.

$ mkdir ~/test  
$ cd ~/test  
$ echo 'export PATH=/usr/global/mpich2/bin:$PATH' >>~/.bashrc  
$ echo 'export LD\_LIBRARY\_PATH=/usr/global/mpich2/lib:$LD\_LIBRARY\_PATH' >>~/.bashrc  
$ . ~/.bashrc   
$ which mpicc  
/usr/global/mpich2/bin/mpicc  
$ echo 'MPD\_SECRETWORD=mr45-j9z' >~/.mpd.conf  
$ chmod 600 ~/.mpd.conf  
$ mpicc -o mpihello mpihello.c  
$ qsub mpich2\_mpd.sh

mpich2\_mpd.sh

#!/bin/bash  
# Export all environment variables  
#$ -V  
# Your job name  
#$ -N test  
# Use current working directory  
#$ -cwd  
# Join stdout and stderr  
#$ -j y  
# PARALLEL ENVIRONMENT:  
#$ -pe mpich2\_mpd 4  
export MPICH2\_ROOT=/usr/global/mpich2  
export PATH=$MPICH2\_ROOT/bin:$PATH  
export MPD\_CON\_EXT="sge\_$JOB\_ID.$SGE\_TASK\_ID"  
echo "Got $NSLOTS slots."  
# The order of arguments is important. Forst global, then local options.  
mpiexec -machinefile $TMPDIR/machines -n $NSLOTS ~/test/mpihello  
exit 0

mpihello.c

// a simple mpi test  
// compile with:  
// $ mpicc -o ~/mpi\_hello mpi\_hello.c  
#include <stdio.h>  
#include <mpi.h>  
int main (argc, argv)  
 int argc;  
 char \*argv[];  
{  
 int rank,size;  
 MPI\_Init(&argc,&argv); /\* starts MPI \*/  
 MPI\_Comm\_rank(MPI\_COMM\_WORLD,&rank); /\* get current process id \*/  
 MPI\_Comm\_size(MPI\_COMM\_WORLD,&size); /\* get number of processes \*/  
 printf("Hello world from process %d of %d\n",rank,size);  
 MPI\_Finalize();  
 return 0;  
}

Check output.

$ cat test.po4479   
-catch\_rsh /var/spool/sge/node06/active\_jobs/4479.1/pe\_hostfile /usr/global/mpich2  
node06:1  
node11:1  
node10:1  
node08:1  
startmpich2.sh: check for local mpd daemon (1 of 10)  
/usr/global/sge/bin/lx24-amd64/qrsh -inherit -V node06 /usr/global/mpich2/bin/mpd  
startmpich2.sh: check for local mpd daemon (2 of 10)  
startmpich2.sh: check for mpd daemons (1 of 10)  
/usr/global/sge/bin/lx24-amd64/qrsh -inherit -V node11 /usr/global/mpich2/bin/mpd -h node06 -p 51413 -n  
/usr/global/sge/bin/lx24-amd64/qrsh -inherit -V node10 /usr/global/mpich2/bin/mpd -h node06 -p 51413 -n  
/usr/global/sge/bin/lx24-amd64/qrsh -inherit -V node08 /usr/global/mpich2/bin/mpd -h node06 -p 51413 -n  
startmpich2.sh: check for mpd daemons (2 of 10)  
startmpich2.sh: got all 4 of 4 nodes

$ cat test.o4479   
Got 4 slots.  
Hello World from Node 0.  
Hello World from Node 2.  
Hello World from Node 3.  
Hello World from Node 1.

$ ssh node06 ps -e f -o pid,ppid,pgrp,command --cols=120 |less  
...  
 3538 1 3538 /usr/global/sge/bin/lx24-amd64/sge\_execd  
24471 3538 24471 \\_ sge\_shepherd-4479 -bg  
24568 24471 24568 | \\_ /bin/bash /var/spool/sge/node06/job\_scripts/4479  
24569 24568 24568 | \\_ python2.4 /usr/global/mpich2/bin/mpiexec -machinefile /tmp/4479.1.all.q/machines -n 4 /hom  
24517 3538 24517 \\_ sge\_shepherd-4479 -bg  
24518 24517 24518 \\_ /usr/global/sge/utilbin/lx24-amd64/qrsh\_starter /var/spool/sge/node06/active\_jobs/4479.1/1.nod  
24528 24518 24528 \\_ python2.4 /usr/global/mpich2/bin/mpd  
24570 24528 24570 \\_ python2.4 /usr/global/mpich2/bin/mpd  
24571 24570 24571 \\_ /home/myuser/test/mpihello  
...  
24509 1 24472 /usr/global/sge/bin/lx24-amd64/qrsh -inherit -V node06 /usr/global/mpich2/bin/mpd  
24532 1 24472 /usr/global/sge/bin/lx24-amd64/qrsh -inherit -V node11 /usr/global/mpich2/bin/mpd -h node06 -p 51413 -  
24534 1 24472 /usr/global/sge/bin/lx24-amd64/qrsh -inherit -V node10 /usr/global/mpich2/bin/mpd -h node06 -p 51413 -  
24537 1 24472 /usr/global/sge/bin/lx24-amd64/qrsh -inherit -V node08 /usr/global/mpich2/bin/mpd -h node06 -p 51413 -  
...

$ ssh node11 ps -e f -o pid,ppid,pgrp,command --cols=120 |less  
...  
 3681 1 3681 /usr/global/sge/bin/lx24-amd64/sge\_execd  
20881 3681 20881 \\_ sge\_shepherd-4479 -bg  
20882 20881 20882 \\_ /usr/global/sge/utilbin/lx24-amd64/qrsh\_starter /var/spool/sge/node11/active\_jobs/4479.1/1.nod  
20889 20882 20889 \\_ python2.4 /usr/global/mpich2/bin/mpd -h node06 -p 51413 -n  
20890 20889 20890 \\_ python2.4 /usr/global/mpich2/bin/mpd -h node06 -p 51413 -n  
20891 20890 20891 \\_ /home/myuser/test/mpihello  
...

Test MPICH2 without SGE.

$ ssh node01  
$ /usr/global/mpich2/bin/mpdboot -f /etc/machine.list -n 4  
$ /usr/global/mpich2/bin/mpdtrace  
$ /usr/global/mpich2/bin/mpdringtest  
$ /usr/global/mpich2/bin/mpiexec -n 4 hostname  
$ /usr/global/mpich2/bin/mpiexec -n 4 /bin/sh -c "hostname; ping -c1 node01; whoami"  
$ /usr/global/mpich2/bin/mpiexec -n 4 ~/test/mpihello  
$ /usr/global/mpich2/bin/mpdallexit