

## Amber12-bf7 for PRIMERGY RX300

### Webpage

<http://ambermd.org/>

### Version

Amber 12 bugfix 7 + AmberTools 12 bugfix 5

### Tools for Compiling

- Intel Compiler 11.1.080
- Intel MPI 4.0.2.003

### Necessary Files for Compiling

- Amber12.tar.bz2
- AmberTools12.tar.bz2
- Patches in <http://ambermd.org/bugfixes12.html>
- Patches in <http://ambermd.org/bugfixesat.html>

### Preparation of Wrapper Commands for Compiler

Put the following wrapper commands in `/home/users/$USER/compiler/intel-11.1`.

#### mpicc

```
#!/usr/bin/perl
#
use strict;
use warnings;
my @extra;
my $compiler = "/opt/intel/Compiler/11.1/080/bin/intel64/icc";
my $cflag = 0;
my $oflag = 0;;
for (my $i = 0; $i < scalar(@ARGV); ++$i) {
    if ($ARGV[$i] eq "-c") {
        $cflag = 1;
    } elsif ($ARGV[$i] eq "-o") {
        $oflag = 1;
    }
}
}
if ($cflag) {
    @extra = ("-I/opt/intel/impi/4.0.2.003/intel64/include");
} elsif ($oflag) {
    @extra = (
        "-ldl",
        "-I/opt/intel/impi/4.0.2.003/intel64/include",
        "-L/opt/intel/impi/4.0.2.003/intel64/lib",
        "-Xlinker", "-rpath",
        "/opt/intel/impi/4.0.2.003/intel64/lib",
        "-Xlinker", "-rpath",
        "/opt/intel/mpi-rt/4.0.2",
        "-Xlinker", "-rpath",
        "/opt/intel/Compiler/11.1/080/lib/intel64",
        "-Xlinker", "-rpath",
        "/opt/intel/Compiler/11.1/080/mkl/lib/em64t",
        "-lmpi", "-lmpigf", "-lmpigi", "-lpthread", "-lrt");
}
system($compiler, @extra, @ARGV);
```

#### mpif90

```

#!/usr/bin/perl
#
use strict;
use warnings;
my @extra;
my $compiler = "/opt/intel/Compiler/11.1/080/bin/intel64/fort";
my $cflag = 0;
my $oflag = 0;;
for (my $i = 0; $i < scalar(@ARGV); ++$i) {
    if ($ARGV[$i] eq "-c") {
        $cflag = 1;
    } elsif ($ARGV[$i] eq "-o") {
        $oflag = 1;
    }
}
if ($cflag) {
    @extra = ("-I/opt/intel/impi/4.0.2.003/intel64/include");
} elsif ($oflag) {
    @extra = (
        "-ldl",
        "-I/opt/intel/impi/4.0.2.003/intel64/include",
        "-L/opt/intel/impi/4.0.2.003/intel64/lib",
        "-Xlinker", "-rpath",
        "/opt/intel/impi/4.0.2.003/intel64/lib",
        "-Xlinker", "-rpath",
        "/opt/intel/mpi-rt/4.0.2",
        "-Xlinker", "-rpath",
        "/opt/intel/Compiler/11.1/080/lib/intel64",
        "-Xlinker", "-rpath",
        "/opt/intel/Compiler/11.1/080/mkl/lib/em64t",
        "-Impi", "-Impigf", "-Impigi", "-lpthread", "-lrt");
}
system($compiler, @extra, @ARGV);

```

icc

```

#!/usr/bin/perl
#
use strict;
use warnings;
my @extra;
my $compiler = "/opt/intel/Compiler/11.1/080/bin/intel64/icc";
for (my $i = 0; $i < scalar(@ARGV); ++$i) {
    if ($ARGV[$i] eq "-o") {
        @extra = ("-Wl,-rpath,/opt/intel/Compiler/11.1/080/lib/intel64",
            "-Wl,-rpath,/opt/intel/Compiler/11.1/080/mkl/lib/em64t");
    }
}
system($compiler, @extra, @ARGV);

```

icpc

```
#!/usr/bin/perl
#
use strict;
use warnings;
my @extra;
my $compiler = "/opt/intel/Compiler/11.1/080/bin/intel64/icpc";
for (my $i = 0; $i < scalar(@ARGV); ++$i) {
    if ($ARGV[$i] eq "-o") {
        @extra = ("-WI,-rpath,/opt/intel/Compiler/11.1/080/lib/intel64",
            "-WI,-rpath,/opt/intel/Compiler/11.1/080/mkl/lib/em64t");
    }
}
system($compiler, @extra, @ARGV);
```

## ifort

```
#!/usr/bin/perl
#
use strict;
use warnings;
my @extra;
my $compiler = "/opt/intel/Compiler/11.1/080/bin/intel64/ifort";
for (my $i = 0; $i < scalar(@ARGV); ++$i) {
    if ($ARGV[$i] eq "-o") {
        @extra = ("-WI,-rpath,/opt/intel/Compiler/11.1/080/lib/intel64",
            "-WI,-rpath,/opt/intel/Compiler/11.1/080/mkl/lib/em64t");
    }
}
system($compiler, @extra, @ARGV);
```

## Procedure of Compiling

```
#!/bin/csh -f
#PBS -l select=ncpus=16:mpiprocs=16:ompthreads=1:jobtype=gpu
#PBS -l walltime=72:00:00
#PBS -j oe
cd ${PBS_O_WORKDIR}
umask 022
# Set working directory.
set work="/work/users/${USER}"
setenv AMBERHOME "$work/amber12"
setenv CUDA_HOME /usr/local/cuda
# Delete old files.
if (-d $AMBERHOME) then
    mv $AMBERHOME $work/amber12-erase
    rm -rf $work/amber12-erase &
endif
#
cd $work
bunzip2 -c /home/users/${USER}/build/amber12/AmberTools12.tar.bz2 | tar xf -
bunzip2 -c /home/users/${USER}/build/amber12/Amber12.tar.bz2 | tar xf -
cd $AMBERHOME
foreach i (/home/users/${USER}/build/amber12/patches/Amber*/*)
    patch -p0 < $i
end
#
echo "[GPU (SPDP) serial edition]"
./configure --no-updates -cuda gnu
make -j 16 install
make test.cuda
make clean
echo "[GPU (SPDP) parallel edition]"
./configure --no-updates -mpi -cuda gnu
```

```
./configure --no-updates -mpi -cuda gnu
make -j 16 install
make clean
```

```
echo "[GPU (SPSP) serial edition]"
```

```
./configure --no-updates -cuda_SPSP gnu
make -j 16 install
cd $AMBERHOME/test/
./test_amber_cuda.sh SPSP
cd $AMBERHOME
make clean
```

```
echo "[GPU (SPSP) parallel edition]"
```

```
./configure --no-updates -mpi -cuda_SPSP gnu
make -j 16 install
make clean
```

```
echo "[GPU (DPDP) serial edition]"
```

```
./configure --no-updates -cuda_DPDP gnu
make -j 16 install
cd $AMBERHOME/test/
./test_amber_cuda.sh DPDP
cd $AMBERHOME
make clean
```

```
echo "[GPU (DPDP) parallel edition]"
```

```
./configure --no-updates -mpi -cuda_DPDP gnu
make -j 16 install
make clean
```

```
# To get compiler version correctly, set LANG environment value.
```

```
setenv LANG C
```

```
# SSE_TYPES environment value means option "-ax${SSE_TYPES}", so unset it.
```

```
unsetenv SSE_TYPES
```

```
# If you use intel compiler version 12,
```

```
# rism1d will be aborted and test of MPI fail.
```

```
source /opt/intel/Compiler/11.1/080/bin/iccvars.csh intel64
```

```
source /opt/intel/Compiler/11.1/080/bin/ifortvars.csh intel64
```

```
# Create symbolic link for mpicc and mpif90 to become mpiicc and mpiifort respectively.
```

```
# Add option "-Wl,-rpath,/opt/intel..." in icc, icpc and ifort.
```

```
set path=(/home/users/$USER/compiler/intel-11.1 $path)
```

```
#
```

```
echo "[CPU serial edition]"
```

```
./configure --no-updates intel
make -j 16 install
make test
make clean
```

```
echo "[CPU parallel edition]"
```

```
./configure --no-updates -mpi intel
make -j 16 install
make test
make clean
```