SGE/Mesa Installation Notes

Version 1.0
Kenneth Perrine, Pacific Northwest National Laboratory
June 20, 2001

SGE/Mesa is written to be built on the IBM SP and Linux with the latest version of Mesa. As of this writing, Mesa version 3.4.2 had been used in building SGE/Mesa 1.0.

These notes describe how to build SGE/Mesa for the IBM SP. No tests have been done in building SGE/Mesa for Linux, but it is expected that many parts of this procedure will work for Linux platforms as well. This document will be updated when a successful Linux build has been performed and documented.

Requirements:

Mesa 3.4.2 sources (demos may not be necessary)
MPI
PThreads
SGE X11 libraries
SGE tunneling libraries

Step 1: Installing Mesa 3.4.2

a. Download the current release of Mesa from http://www.mesa3d.org. The "MesaLib" archive will be necessary, but it will not be necessary to get "MesaDemos" as demos are included with the SGE/Mesa archive.

b. Unpack the Mesa files. Unpacking should place all files into a directory (such as "Mesa-3.4.2").

c. It is not known (due to lack of testing) whether or not it is best to run "configure". All tests have been done without running "configure", as Mesa can be built by explicitly specifying the target platform. So, don't run "configure" without trying the stuff below first.

d. Modifications must be made to the Mesa config. makefile before it is built. Open the "Make-config" file for editing (located in the unpacked root, "Mesa-3.4.2/"):

   1. Find the section(s) which apply to the platform that Mesa is to be compiled on. For example, you'll find "aix:" and "aix-sl:" for AIX static library and AIX shared library compilations, respectively.

   2. Change the compiler in the "CC=" lines to the desired compiler, such as "xlc_r". It is best to use a compiler script which includes reentrant libraries since advanced SGE/Mesa options use multiple threads.

   3. In "CFLAGS=" add the term "-DPTHREADS". This will enable multithreaded support in Mesa.

   4. Also in "CFLAGS=" add desired optimization. On the SP, the "-O4" optimization works well because generated code takes advantage of various processor features. Note that for testing purposes it may be best to forego

optimizations and use "-g" to include debug information and perform a fast compile. Also note that on AIX, it has been observed that creation of executables utilizing an optimized, static library takes a long time. It may be best to use "-g" with static libraries for initial testing and "-O4" for shared libraries. It has been observed at PNNL on the SP that about three times the performance of "-g" can be achieved with "-O4".

5. Another item to add in "CFLAGS=" is a path to the SGE X11 include files. At PNNL, this looks like:

   ```
   -I/u/phoch/X11R6.4/include
   ```

   If there are paths to regular X11 include directories, then remove those paths.

6. Finally, in the "APP_LIB_DEPS=" lines, precede the "-l" callouts with "-L" followed by the path to the SGE X11 libraries. This hopefully overrides the default inclusion of the regular, non-SGE X11 libraries. At PNNL, this looks like:

   ```
   -L/u/phoch/X11R6.4/lib
   ```

   If there are paths to regular X11 libraries, then remove those paths.

e. Open the "src/config.h" file, look for the definitions of "MAX_WIDTH" and "MAX_HEIGHT" and change them to the maximum display size that the SGE is configured for. For example, this would be 3840 and 2048 for the Bertha display.

f. If you intend to build a shared library on the SP, a couple of changes must be made to the "bin/mklib.aix" file. Note that comparable changes may be required on Linux if a shared library is to be built.

   1. Under the remark "# Make the shared lib file", change the "cc" to the string: "${CC}".
   2. Precede the "-l" library callouts with the string: "${APP_LIB_DEPS}"

g. Now, go to the root directory of the unpacked files (such as "Mesa-3.4.2") and enter "make" followed by the desired platform, which is the label in the "Make-config" under which modifications were made in Step d. Example:

   ```
   make aix
   ```
   or
   ```
   make aix-sl
   ```

   For the first try, it may be more desirable to build static libraries; these are not as prone to library path problems. The disadvantage of building static libraries is that the resulting executables are large, and they take a horrendously long time to compile if optimizations are enabled. Shared libraries result in smaller executables and fast compilations.

As Mesa is building, there is a possibility that an error will result at file "src/config.c". On PNNL machines, the term "var" is already defined in a system include file. Changing all instances of "var" to "evar" in file "src/config.c" fixes this problem.

Note that the build procedure should place files into the unpacked directory's "lib" directory.

Step 2: Unpacking and Configuring SGE/Mesa

a. Unpack the "sgemesa.tar" file. Everything should unpack into an "sgemesa1.0" directory.

b. Open the "Make-config" file and make modifications as directed below. NOTE that all relative directories (those starting with "..") must be relative to one child directory from where "Make-config" exists. So, for all directories which would start with "../" from where "Make-config" sits, use: "../../".

   1. For "MESADIR" identify the directory in which the Mesa distribution exists. Example: "../../Mesa-3.4.2".
   2. For "SGE" identify the directory in which the SGE files exists. Example: "/u/phoch/sge2"
   3. For "SGEX" identify the directory where the SGE X11 installation exists. Example: "/u/phoch/X11R6.4"
   4. For "CFLAGS" specify the desired optimization for the compiler. When building the "src" directory, faster performance may be achieved with "-O4". For debugging, specify "-g" instead. If you are debugging any code which is not in the "src" directory, you can build the "src" directory with "-O4" but then change this flag to "-g" for building executable files. This will retain optimizations in SGE/Mesa but allow for debugging within user-application source code.
   5. In "EXPT", specify "sgekern" for normal SGE operation or "nullhal" for use with the SGE simulator. This has not been tested with SGE "GBE" (gigabit Ethernet) libraries, so this may not be applicable to "GBE". (See Step 8 below for clarification.)
   6. If in Linux a shared-library build of Mesa results in ".so" files being created, then the libraries listed in "GL_LIB", and "GLU_LIB" must be changed to ".so" extensions. Most likely, the "SGEMESA_LIB" and "SGEGLUT_LIB" settings would remain with ".a" extensions because these are statically-built.
   7. The "APP_LIB_DEPS" setting may be modified if a build fails.
   8. The "SGEPARLIBS" line forms together many items expressed above into a line which passed to the compiler/linker having to do with SGE libraries. A remarked-out line is included for use with SGE "GBE" libraries (gigabit Ethernet), but has not been tested. See the SGE samples makefile for examples in linking with the SGE "GBE" libraries. If you are using "GBE" and not SP TBS links, then remark out the TBS line and un-remark the "GBE" line.
   9. At the bottom of the file, set the "CC" setting to your preferred compiler. On the SP, this should be "mpcc_r" which creates applications which are to run in the parallel environment and utilizes reentrant libraries. In Linux, this would be different.

Step 3: Building SGE/Mesa and Demos

The makefile system isn't nicely integrated yet. To build SGE/Mesa, a series of makefiles must be executed individually, in the correct order.

a. Start in the sgemesa1.0 directory.

b. Build the SGE/Mesa library:

```
cd src
make
cd ..
```

c. Build SGE/GLUT:

```
cd sgeglut
make
cd ..
```

d. Now, build the "unchanged" demos. These are demo files copied directly from the Mesa "demos" directory. A couple have undergone minor changes as required to work with SGE/GLUT as described in the SGE/Mesa Reference. These demonstrations transparently utilize the SGE through the use of SGE/GLUT. They also (mostly) demonstrate accelerated rendering performance when run in parallel. There are a couple bugs which haven't been ironed out, described in the "Bugs/Improvements" section in the Reference.

```
cd demos
make
cd ..
```

Any problems in including libraries will show up in this step, as executable files are attempted to be created. For ideas on fixing problems, see "Troubleshooting".

e. Finally, build the SGE/Mesa demos. These utilize multiple threads to render simultaneous frames.

```
cd sgedemos
make
cd ..
```

Troubleshooting

Here are some ideas on what to do if some step doesn't work:

In building demos, I get error messages saying that various X11 extensions aren't found.

Most likely, regular X11 libraries are being used rather than the SGE X11 libraries. Ensure that paths to the SGE X11 libraries are expressed as identified in the building steps and that there are no paths to the regular X11 libraries. Check environment variables to see that there aren't paths to regular X11 libraries. Two places in the Mesa files to check are the "Make-config" file and the "bin/mklib.aix" file (which is only utilized for shared libraries).

In building Mesa, it bombs out when it gets to the "xdemos" directory.

Something in the SGE X11 libraries may be preventing this directory from being built. That's okay; the Mesa libraries should have been built by this time, and that is all that is needed for use with SGE/Mesa.

When I attempt to run an executable, I get some error saying that a shared object isn't found.
This is a generic error message that isn't very clear and makes debugging difficult. Try rebuilding Mesa as static libraries. Another thing to try is to specify full paths to each library rather than using "–L" followed by the "-l" for linking libraries.

I try to rebuild libraries but nothing seems to change
Delete libraries from the Mesa "lib" directory before rebuilding. Also, "make realclean" can be run to delete ".o" files and executables. (It doesn't delete the files in the "lib" directory for SGE/Mesa).

Compilation of the demos takes a really long time
It hasn't been figured out why this happens, but this occurs when a static Mesa library (which was compiled using optimization) is linked in making an executable. It is suspected that optimizations are being made across function calls between libraries and it gets out of hand. Two known ways to avoid this is to compile Mesa statically without optimizations, or to compile Mesa as a shared library (in which you can compile optimized).

When I run a demo, I get an empty frame window without any tunneling activity inside of it.
This may be caused by shared-library issues. Specifically, some standard X11 libraries may be intermingling with SGE X11 libraries. Check your library paths to insure standard X11 is entirely excluded. (See the first tip above). Alternatively, it could be that a library (such as the SGE/GLUT library) is instancing itself twice so that static variable scopes are not shared among other libraries. Try building everything as a static library. It is known that problems can arise if SGE/GLUT is built as a shared library, but as written the makefiles should build this as a static library.

I see problems when I try to run my application as multiple processes on one node
Multiple TBS links from one node cannot be made to the SGE. This is possible for GBE, however, but has not yet been tested at PNNL.

The "demos/teapot" demo won't compile
This utilizes a C++ header file called "shadow.h". The Makefile doesn't currently make considerations for this. It can be manually built with some work.

Some demos look strange, crash, or freeze when they are run on more than one node
There are a couple of unresolved issues as identified in the "Bugs/Imporvements" section of the SGE/Mesa Reference. One more notable one is the frustum problem which causes the perspective to rapidly grow to crazy proportions in the "Ray" and "Tunnel" demos. The SGE tunneling libraries are sent nonsense data after a couple of seconds and freeze. There is a workaround to this specific problem (which hasn't been tested but is worth trying).

When I run a demo, I get error messages saying that there are MPI-related problems or SGE-related problems.
This could be an issue independent of SGE/Mesa. Try running the SGE "stretch" demo to see if the same errors appear. If "stretch" runs, then check the makefile that "stretch" was

built with to see if the same compiler and library paths are used as what are identified in the Mesa and SGE/Mesa "Make-config" files.

SGE/Mesa Directories

The SGE/Mesa 1.0 "distribution" consists of several directories:

src – The location of the SGE/Mesa source file.

include – Contains SGE/Mesa and SGE/GLUT include files.

lib – SGE/Mesa and SGE/GLUT libraries are placed here when they are built.

sgeglut – A distribution of GLUT with modified files to automate SGE window creation, tunneling, event handling, and parallel event distribution.

sgedemos – These demonstrate advanced features of SGE/Mesa and explicitly call several SGE/Mesa functions.

demos – This is virtually a direct copy of the Mesa-3.4.2 "demos" directory with a new Makefile to automate building with the SGE/Mesa and SGE/GLUT libraries.

images, util – These directories are utilized by some demos in the "demos" directory.