Mixing Architectures

This page describes one convenient method to allow mixing of different machine types within the `mpifxcorr` distributed calculation. Also given is some general advice that should be required for any installation. This page is useful if, say, you want to correlate on a mixed 32/64 bit environment. This document assumes that all machines NSF mount at a minimum the script used to set the environment.

1. Compilation of OpenMPI

This guide assumes one is using openmpi as the message passing interface. The instructions below may be specific to openmpi, but other implementations may have similar requirements.

To allow mixed architectures, openmpi must be configured with the `–enable-heterogeneous` flag. For example, you might issue the following command before compiling openmpi:

```
./configure --prefix=${DIFXROOT} --enable-heterogeneous
```

2. Setup scripts

It is extremely handy to have a startup script that, upon shell invocation, sets the environment properly, possibly in an architecture-dependent manner. What NRAO has found convenient is to have a single setup script for all architectures for a particular version of DiFX. This script has conditionals, as shown in the example below, that cause the correct behavior in multiple cases:

```
#!/bin/bash

DIFX_PROMPT_EXTRA=""
if [[ `uname --hardware-platform` = "x86_64" ]]; then
  export DIFXBITS=64
  export IPPROOT=/home/swc/difx/intel/ipp/6.1.2.051/em64t
  export DIFX_PREFIX=/home/swc/DiFX-trunk-64
  export DIFX_PROMPT_EXTRA="${DIFX_PROMPT_EXTRA}-64"
else
  export DIFXBITS=32
  export IPPROOT=/home/swc/difx/intel/ipp/6.1.2.051/ia32
  if [[ x`pkg-config --modversion streamstor` < "x9.0" ]]; then
    export DIFX_PREFIX=/home/swc/DiFX-trunk
  else
    export DIFX_PREFIX=/home/swc/DiFX-trunk-SDK9
    DIFX_PROMPT_EXTRA="${DIFX_PROMPT_EXTRA} SDK9"
  fi
fi
.
# lots of lines skipped here
.
export DIFXROOT=${DIFX_PREFIX}
export PS1="\u@\h $DIFX_VERSION$DIFX_PROMPT_EXTRA \W> "
```

The above example will set up the environment for three different cases:

1. 32-bit OS with Streamstor SDK 8 for nativemk5
2. 32-bit OS with Streamstor SDK 9 for nativemk5
3. 64-bit OS without nativemk5 support

3. Building multiple binaries

A separate binary should then be compiled on a representative machine of each architecture. Make sure the setup script is properly run before compilation.

4. (Optional) Out-of-source building

The gnu autotools allow building of a module in a way that no binary object code is written into the source directory tree but is instead stored in another directory tree. The primary benefit of this is when making changes to the source tree recompilation for each architecture can proceed simply without duplicating changes to the source. This method is suggested by the gnu folks for compilation of `gcc`; more complete documentation on this methodology can be found at the `gcc` installation web page.

The install script (`install-difx`) currently does not have support for this mode of installation. A new script (`build-difx`) is being developed to provide this support and is described here.