openHPC
http://openhpc.community

Meeting of the Technical Steering Committee (TSC) Board

Wednesday, April 24th, 2019
11:00am ET

Meeting Logistics

- https://zoom.us/j/556149142

- United States : +1 (646) 558-8656
  - Meeting ID: 556 149 142

Antitrust Policy Notice

- Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

- Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at http://www.linuxfoundation.org/antitrust-policy. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

THE LINUX FOUNDATION

Agenda

- 1.3.7.1 release with singularity permissions fix went live April 11
- ISC'19 BoF Submission
  - BoF has been accepted
  - Scheduled for Wed, June 19th, 8:30-9:30am
- PEARC'19 tutorial
  - if you are interested in helping prepare content for the tutorial this summer, let me know (if you haven't already done so). Expect to setup additional prep meetings for that.
- 1.3.8 project area created in OBS
  - issues created for component updates
  - looks like ~38% of components have updates available
- alternate build flag support continued discussion

Alternate build flag support

- Thanks to Kevin for starting fresh PR on this topic
- Would like to continue the discussion....

- First: What do we want to achieve?
  - Primary Goal: make it easier for downstream user to rebuild ohpc packages uniformly with different compiler flags (alternate optimizations, micro-architecture specific flags, etc)
  - Ideally, no change to .spec file required

```
# rpmbuild --bb --define "new_CXXflag -O42" foo.spec
```

Alternate build flag support

- Regarding setting of opt flags, recall that rpmmacro exists to provide uniform settings on build host
- Obviously only applies to autotool'ed packages (majority, but certainly not all of our components)
- We use this currently in small number of .spec files (it hard-codes the install prefix)

```
# rpm --eval %configure


  CFLAGS="${CFLAGS:--O2 -g -pipe -Wall -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-strong --param=ssp-buffer\
-size=4 -grecord-gcc-switches   -m64 -mtune=generic}" ; export CFLAGS ;
  CXXFLAGS="${CXXFLAGS:--O2 -g -pipe -Wall -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-strong --param=ssp-bu\
ffer-size=4 -grecord-gcc-switches   -m64 -mtune=generic}" ; export CXXFLAGS ;
  FFLAGS="${FFLAGS:--O2 -g -pipe -Wall -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-strong --param=ssp-buffer\
-size=4 -grecord-gcc-switches   -m64 -mtune=generic -I/usr/lib64/gfortran/modules}" ; export FFLAGS ;
  FCFLAGS="${FCFLAGS:--O2 -g -pipe -Wall -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-strong --param=ssp-buff\
er-size=4 -grecord-gcc-switches   -m64 -mtune=generic -I/usr/lib64/gfortran/modules}" ; export FCFLAGS ;
  LDFLAGS="${LDFLAGS:--Wl,-z,relro }"; export LDFLAGS;
  [ "1" == 1 ] && [ "x86_64" == ppc64le ] && /usr/lib/rpm/redhat/libtool-handle-ppc64le.sh ;
  for i in $(find . -name config.guess -o -name config.sub) ; do
      [ -f /usr/lib/rpm/redhat/$(basename $i) ] && /usr/bin/rm -f $i && /usr/bin/cp -fv /usr/lib/rpm/redhat/$(basename \
$i) $i ;
  done ;
  ./configure --build=x86_64-redhat-linux-gnu --host=x86_64-redhat-linux-gnu \
  --program-prefix= \
  --disable-dependency-tracking \
  --prefix=/usr \
```

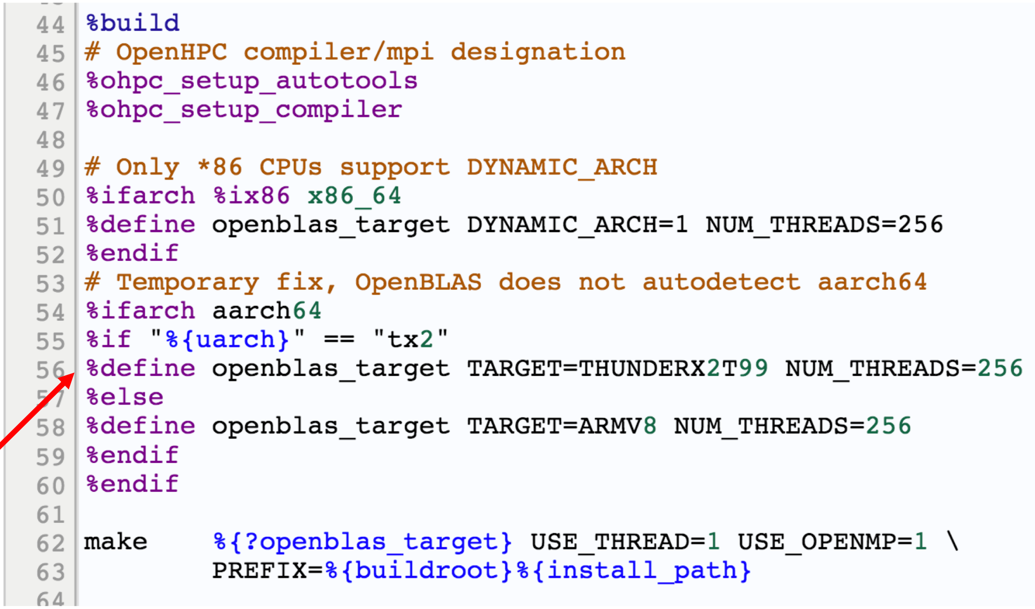
example from x86_64 host

Alternate build flag support

- For most autoconf'ed components (those installed into %{OHPC_PUB}) we call "./configure" directly
  - These autotool'd projects typically have their own default settings
- Fundamental issue I see presently is that we do not have a mechanism to apply external compiler flags settings without touching .spec file
- To resolve this, we would first need to undertake updates to apply uniform settings to each component's build system
  - vanilla make
  - autoconf/automake
  - cmake
  - bjam
  - other misc. craziness
- Defaults can be specified in our current OHPC_macros file; should then be straightforward to allow users to override

Alternate build flag support

- Note: this doesn’t solve all problems as some components have very specific settings to choose different compile options
- OpenBLAS is one of those (and the package which started us on this discussion)
- For these, might need additional changes to allow override



```
%build
# OpenHPC compiler/mpi designation
%ohpc_setup_autotools
%ohpc_setup_compiler

# Only *86 CPUs support DYNAMIC_ARCH
%ifarch %ix86 x86_64
%define openblas_target DYNAMIC_ARCH=1 NUM_THREADS=256
%endif
# Temporary fix, OpenBLAS does not autodetect aarch64
%ifarch aarch64
%if "%{uarch}" == "tx2"
%define openblas_target TARGET=THUNDERX2T99 NUM_THREADS=256
%else
%define openblas_target TARGET=ARMV8 NUM_THREADS=256
%endif
%endif

make    %{?openblas_target} USE_THREAD=1 USE_OPENMP=1 \
        PREFIX=%{buildroot}%{install_path}
```

however, would prefer to not do this, and inherit from external macro variable instead.

Alternate build flag support

- Other thoughts…
  - if we were to do this, presumably need equivalent of a CFLAGS, CXXFLAGS, FCFLAGS that has default settings in OHPC_MACROS and update all .spec to take advantage
  - say, OHPC_CFLAGS, OHPC_CXXFLAGS, OHPC_FCFLAGS
  - if putting in work to update each .spec to use external compiler settings, could also embed the flags used in resulting module file (e.g. make it visible with "module show foo")
  - would likely have to be an incremental rollout
    - could agree on conventions for 1.3.8 and include in components that are being updated for this release
    - update all .specs accordingly over time