READEX Tool Suite Build Guide

April 10, 2018

Contents

1 About different compilers

During the development of the READEX tool suite it turned out that it is not obvious to use the same compiler for everything. Please be aware that there are some incompatibilities between different compilers. Specially for C++ this is a problem, as different compilers may have different ABI versions, which will lead to linking errors. Moreover, a Score-P version compiled for GCC will not work with the Intel compiler and vice versa.

Therefore, it is really important that you are using the same compiler for the entire READEX tool suite as well as the applications on which you are going to use tool suite. Again, please use the same compiler for Score-P, PTF, the RRL, the PCPs, the ATP library and your application.

2 Score-P

This section outlines how to build Score-P for READEX.

2.1 Requirements

The build procedure for the READEX version of PTF requires the following tools to be already installed:

- GCC (G++ and GFortran) version 6.3.0 or Intel compiler version 2017.2.174.
- PAPI version 5.5.1.
- Bison version 3.0.4.

2.2 Download

Please download the version of Score-P for READEX from the following location and unpack it:

```
http://www.readex.eu/index.php/dissemination/software/ScoreP.tar.gz
tar -xzvf ScoreP.tar.gz
```

2.3 Preparing the Score-P directory

Please prepare the Score-P build directory as follows:

```
cd ScoreP
mkdir build
cd build
```

2.4 Configuring and installing Score-P

You may use the following naming scheme for the "-prefix" argument:

```
<Desired path for Score-P installation>/scorep/scorep_<svn revision number>_<mpi version>_<
     compiler verison>

<svn revision number>:  for example, 11271
<mpi version>:              for example, intelmpi2017.2.174
<compiler version>:     for example, intel2017.2.174
```

To run configure please do:

```
../configure          '--prefix=<Desired␣path␣for␣Score-P␣installation>/scorep/scorep_<
     scorep_revision_number>_<mpi_version>_<compiler_version>/' \
                      '--enable-backend-test-runs' \
                      '--with-nocross-compiler-suite=<gcc|intel>' \
                      '--with-mpi=<bullxmpi|intel3|...>' \
                      '--with-libcudart=<path␣to␣CUDA␣installation>' \
                      '--with-pdt=<path␣to␣PDT␣bin>' \
                      '--with-papi-header=<path␣to␣PAPI␣include>' \
                      '--with-papi-lib=<path␣to␣PAPI␣lib>' \
                      '--with-libbfd=no' \
                      '--disable-silent-rules' \
                      '--without-gui' \
                      '--enable-static' \
                      '--enable-shared' \
                      '--enable-debug'\
                      'CFLAGS=␣-g␣-O3␣-fno-omit-frame-pointer' \
                      'CXXFLAGS=␣-g␣-O3␣-fno-omit-frame-pointer'\
make -j 24
make install
```

For more details on installing Score-P, refer to Section 2.1 in `https://silc.zih.tu-dresden.de/scorep-current.pdf`.

3 PTF

This section outlines how to build the Periscope Tuning Framework (PTF) for READEX.

3.1 Requirements

The build procedure for the READEX version of PTF requires the following tools to be already installed:

- Score-P extension for READEX as described in Section 2.
- GCC (G++ and GFortran) version 6.3.0 or Intel compiler version 2017.2.174.
- PAPI version 5.5.1.
- Boost version 1.62.0.
- Cereal version 1.2.1.
- Bison version 3.0.4.
- Python version 3.6 or higher.
- Ace version 6.3.3.
- Flex version 2.5.39.
- Score-P developer tools containing patched Libtool version 2.4.6, Autoconf version 2.69, Automake version 1.13.4, Doxygen version 1.8.10 and M4 version 1.4.16.

Please make sure that the Score-P version is also compiled with the same compiler as the one used for PTF.

3.2 Download

Please download the READEX branch of PTF and Score-P development tools from the following locations, and unpack them:

```
http://www.readex.eu/index.php/dissemination/software/PTF.tar.gz
tar -xzvf PTF.tar.gz

http://www.readex.eu/index.php/dissemination/software/scorep-dev-06.tar.gz
tar -xzvf scorep-dev-06.tar.gz
```

3.3 Preparing the PTF directory

Please bootstrap PTF as follows:

```
cd PTF
./bootstrap
mkdir build
cd build
```

3.4 Configuring and installing PTF

Add the paths to the libraries and executables of the Score-P developer tools to `PATH` and `LD_LIBRARY_PATH` environment variables:

```
export PATH=<path to Score-P developer tools>/bin:$PATH
export LD_LIBRARY_PATH=<path to Score-P developer tools>/lib:$LD_LIBRARY_PATH
```

You may use the following naming scheme for "–prefix":

```
<Desired path for PTF installation>/ptf/ptf-<day of build>-readex-<compiler>-slurm_starter
    -<mpi version>-with-scorep/

<day of build>          build date in the form YYYY-MM-DD
<compiler>              for example: intel2017.2.174
<mpi version>           for example: intelmpi2017.2.174
```

To configure and install PTF please now do:

```
../configure        '--prefix=<Desired_path_for_PTF_installation>/ptf/ptf-<day_of_build>-
    readex-<compiler>-slurm_starter-<mpi_version>-with-scorep/' \
                    --enable-developer-mode \
                    --with-starter=<slurm|superMUC> \
                    --enable-ace \
                    --enable-boost \
                    --with-boost-lib=<path to Boost lib>\
                    --enable-cube \
                    --with-cube-include=<path to Cube include> \
                    --with-cube-lib=<path to Cube lib> \
                    --enable-scorep \
                    --with-scorep-include=<path to Score-P include> \
                    --with-scorep-lib=<path to Score-P lib> \
                    --enable-cereal \
                    --with-cereal-include=<path to Cereal include>

make -j 24
make install
```

If you want to use the Intel compiler to compile PTF, please add the following to "../configure" :

```
    --with-compiler-suite=intel
```

Please be aware that there is no special cube module needed, if you are using the same compiler for Score-P and PTF. Therefore the options for "–with-cube-include" and "–with-cube-lib" are the same as for "–with-scorep-include" and "–with-scorep-lib".

For more details on installing PTF, refer to Section 2.3 in `http://periscope.in.tum.de/releases/latest/pdf/PTF_Installation_Guide.pdf`.

4 RRL

This section outlines how to build the READEX Runtime Library (RRL).

4.1 Requirements

The build procedure for RRL requires the following tools to be already installed:

- Score-P extension for READEX as described in Section 2.
- GCC version 6.3.0 or Intel compiler version 2017.2.174.
- CMake version 3.8 or higher.
- PAPI version 5.5.1.

Please make sure that the Score-P version is also compiled with the same compiler as the one used for RRL.

4.2 Download

Please download RRL from the following location and unpack it:

```
http://www.readex.eu/index.php/dissemination/software/RRL.tar.gz
tar -xzvf RRL.tar.gz
```

4.3 Preparing the RRL directory

Please switch to the RRL directory of the downloaded repository:

```
cd RRL
mkdir build
cd build
```

4.4 Configuring and installing the RRL

You may use the following naming scheme for “–DCMAKE_INSTALL_PREFIX”:

```
<Desired path for RRL installation>/readex-rrl/rrl-<day of build>-<compiler>-<mpi version>

<day of build>        build date in the form YYYY-MM-DD
<compiler>            for example: intel2017.2.174
<mpi version>         for example: intelmpi2017.2.174
```

To build the RRL, you will need x86_adapt (`https://github.com/tud-zih-energy/x86_adapt`). Please be sure to have the path to `libx86_adapt.so` in your `LD_LIBRARY_PATH`.

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<path to libx86_adapt.so>
```

Now, to build the RRL please do:

```
cmake ../ -DCMAKE_INSTALL_PREFIX=<Desired path for RRL installation>/readex-rrl/rrl-<day of
      build>-<compiler>-<mpi version> -DDISABLE_CALIBRATION=ON
make -j 24
make install
```

NOTE: You may omit x86_adapt and PAPI from the build process if you choose to use RRL without the calibration feature.

5 PCPs

This section outlines how to build the different Parameter Control Plugins (PCPs) for READEX.

5.1 Requirements

The build procedure for the PCPs requires the following tools to be already installed:

- READEX Runtime Library (RRL) as described in Section 4.
- GCC version 6.3.0 or Intel compiler version 2017.2.174.

Please make sure that the RRL version is also compiled with the same compiler as the one used for the PCPs.

5.2 Download

Please download the PCP from the following location and unpack it:

```
http://www.readex.eu/index.php/dissemination/software/PCPs.tar.gz
tar -xzvf PCPs.tar.gz
```

5.3 Configuring and installing the PCP's

You may use the following naming scheme for the argument of the build script:

```
<Desired path for PCPs installation>/parameter_control_plugins/pcp_<day of build>-<
     compiler>

<day of build>          build date in the form YYYY-MM-DD
<compiler>              for example: intel2017.2.174
```

To build the PCPs please now do:

```
cd PCPs
./build.sh <Desired path for PCPs installation>/parameter_control_plugins/pcp_<day of build
     >-<compiler>
```

6 ATP

This section outlines how to build the Application Tuning Parameter library (ATP library) for READEX.

6.1 Requirements

The build procedure for the ATP library requires the following tools to be already installed:

- READEX Runtime Library (RRL) as described in Section 4.
- GCC version 6.3.0 or Intel compiler version 2017.2.174.
- CMake version 3.1 or higher.

Please make sure that the RRL version is also compiled with the same compiler as the one used for the ATP library.

6.2 Download

Please download ATP library from the following location and unpack it:

```
http://www.readex.eu/index.php/dissemination/software/ATP.tar.gz
tar -xzvf ATP.tar.gz
```

6.3 Preparing the ATP library directory

Please do:

```
cd ATP
mkdir build
cd build
```

6.4 Configuring and installing the ATP library

You may use the following naming scheme for “–DCMAKE_INSTALL_PREFIX”:

```
<Desired path for ATP library installation>/readex-atp/atp-<day of build>-<compiler>-<mpi
     version>

<day of build>         actuall date in the form YYYY-MM-DD
<compiler>             for example: intel2017.2.174
<mpi version>          for example: intelmpi2017.2.174
```

To build the ATP please now do:

```
cmake ../ -DCMAKE_INSTALL_PREFIX=<Desired path for ATP library installation>/readex-atp/atp
     -<day of build>-<compiler>-<mpi version>
make -j 24
make install
```