



Installing POP Tools: Score-P, Scalasca, Cube Bernd Mohr

EU H2020 Centre of Excellence (CoE)



Grant Agreement No 824080

1 December 2018 – 30 November 2021

What you will learn



How to install the POP tools Score-P, Scalasca, and Cube

- On a HPC cluster OR
- Linux workstation/laptop

1. Simple manual installation: Do it yourself

- Download sources
- Compile and install them

2. Automated installation: Use HPC package managers

- EasyBuild
- Spack
- OpenHPC



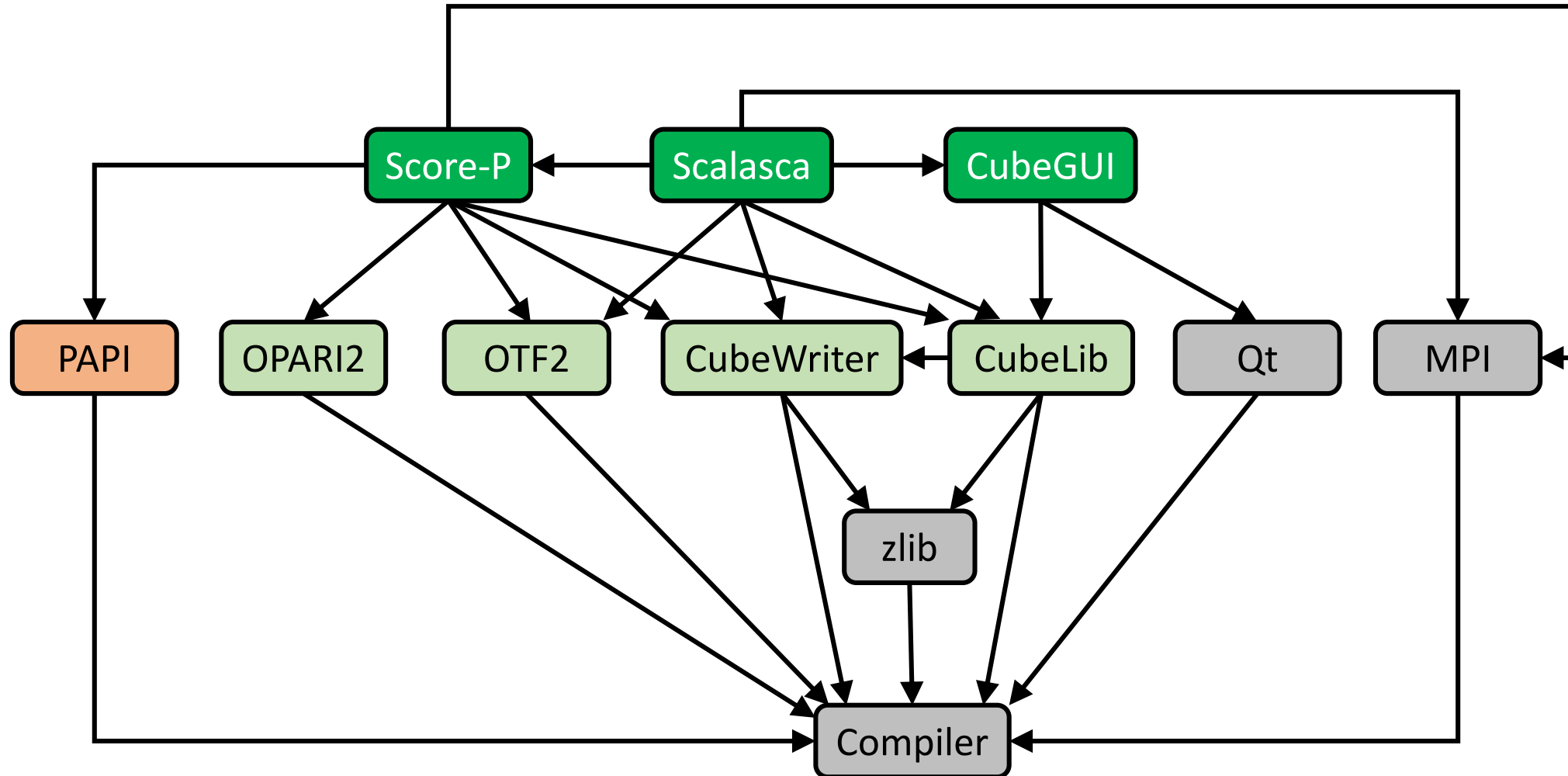
Prerequisites



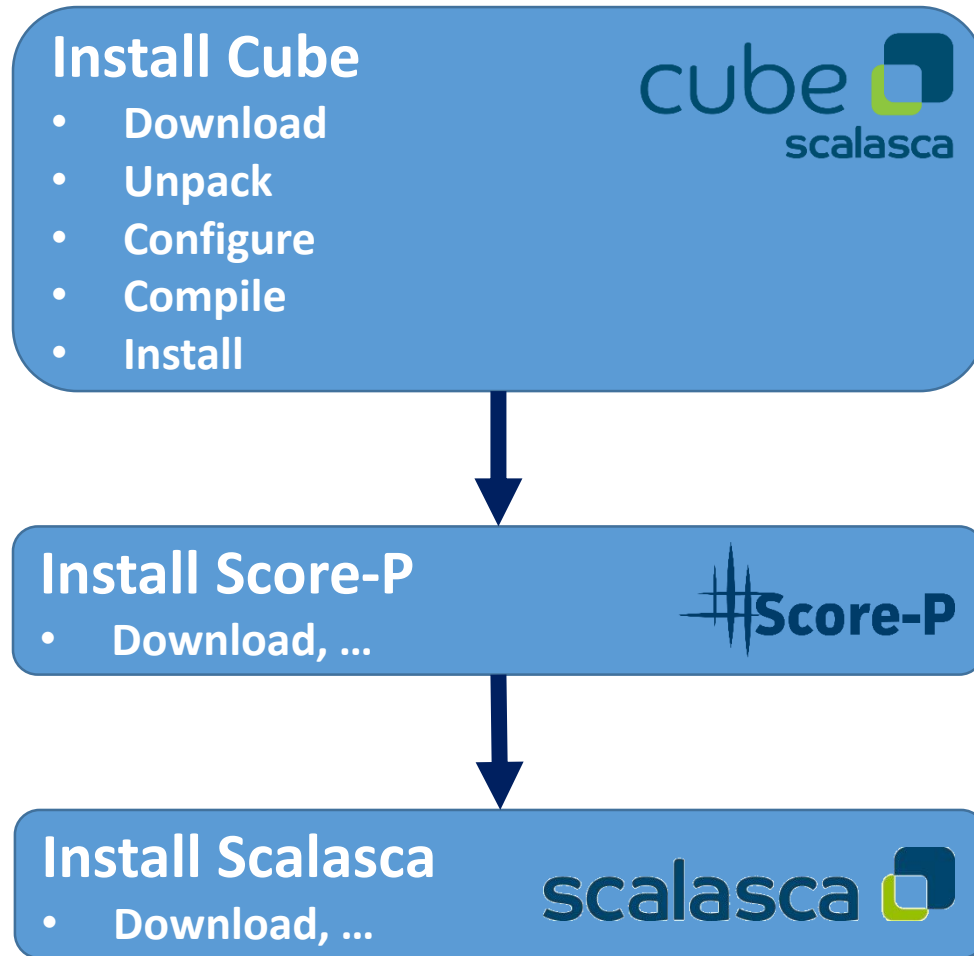
- Basic understanding of Linux shell commands
- Access to HPC cluster or Linux workstation/laptop
- **Development software**
 - Compiler suite (C, C++, Fortran)
 - GCC, Intel, IBM XL, PGI, ...
 - MPI library
 - OpenMPI, MPICH, Intel, ...
 - Cube only: QT4 ($\geq 4.6.0$) or Qt5
 - make



Package Dependencies



Manual Installation



Remote Scenario

Personal Desktop / Laptop

Remote HPC Cluster

Local Scenario

Local HPC Cluster / Desktop / Laptop

Pro Tip: Order important so already installed subcomponents can be re-used



- Download latest version
 - <http://www.scalasca.org/scalasca/software/cube-4.x/>
 - CubeBundle 4.5

- Unpack

```
% tar zxf CubeBundle-4.5.tar.gz && cd CubeBundle-4.5
```

- Configure

```
% ./configure --prefix=/opt/local/Cube-4.5
```

- Compile

```
% make
```

or
\$HOME/tools/Cube-4.5

- Install

```
% make install
```

- **Pro Tip:** Website also provides binary installers for Windows and MAC OS

Manual Installation: #Score-P



- Download latest version
 - <http://www.score-p.org> (scroll down to “Download section”)
 - Score-P 6.0

- Make sure to re-use Cube subcomponents (Local Scenario only)

```
% export PATH=/opt/local/Cube-4.5/bin:${PATH}
```

- Unpack

```
% tar zxf scorep-6.0.tar.gz && cd scorep-6.0
```

- Configure

```
% ./configure --prefix=/opt/local/ScoreP-6.0
```

- Compile

```
% make
```

- Install

```
% make install
```



Manual Installation: #Score-P (2)



- **Note:** Score-P is Compiler and MPI dependent
 - Needs to be installed for every Compiler/MPI combination

- Advanced configuration

- Specify compiler suite other than GCC

```
% ./configure ... --with-nocross-compiler-suite=(ibm|intel|pgi)
```

- Specify MPI library if NOT auto-detected or more than one MPI library available

```
% ./configure ... --with-mpi=(intel3|mpich3|openmpi3|...)
```

- Specify PAPI component if NOT auto-detected (needed for POP metrics calculation)

```
% ./configure ... --with-papi-header=<path-to-papi.h>  
--with-papi-lib=<path-to-libpapi.*>
```

- See installation guide on how to configure support for additional programming models (CUDA, OpenCL, SHMEM, OpenACC)



- Download latest version
 - <https://www.scalasca.org/scalasca/software/scalasca-2.x/>
 - Scalasca 2.5

- Make sure to re-use Score-P subcomponents (e.g. OTF2)

```
% export PATH=/opt/local/ScoreP-6.0/bin:${PATH}
```

- Unpack

```
% tar xzf scalasca-2.5.tar.gz && cd scalasca-2.5
```

- Configure

```
% ./configure --prefix=/opt/local/Scalasca-2.5
```

- Compile

```
% make
```

- Install

```
% make install
```



- **Note:** Scalasca is Compiler and MPI dependent
 - Needs to be installed for every Compiler/MPI combination

- Advanced configuration

- Specify compiler suite other than GCC

```
% ./configure ... --with-nocross-compiler-suite=(ibm|intel|pgi)
```

- Specify MPI library if NOT auto-detected or more than one MPI library available

```
% ./configure ... --with-mpi=(intel3|mpich3|openmpi3|...)
```



DEMO: Show output of Score-P configure command and point out summary



```
openSUSE-Leap-15-1 x + v - □ ×
zlib compression flags :
                        -DBACKEND_CUBE_COMPRESSED=yes
Advanced memory handling :
                        no
Internal memory tracking :
                        no
Internal memory tracing : no
C99 compiler used:      gcc
C++ compiler used:      g++
Compiler flags used:    -g -O2
Compiler cxxflags used: -g -O2

OPARI2 2.0.5:
Platform:               linux (provided)
Cross compiling:        no (provided)

OTF2 2.2:
Platform:               linux (provided)
Cross compiling:        no (provided)

OTF2 (backend):
C99 compiler used:      gcc
SIONlib support:        no, missing sionconfig
Entropy sources:        clock_gettime getpid sysinfo gethostid
Python bindings support: no, missing builtins module
Running tests:          no
Parallel tests:         yes
zam310:~ [103]
```



Automated Installation



- Use IF
 - You need to install tools for multiple Compiler/MPI library combinations
 - You need to install also other HPC tools, libraries, and applications
 - You easily want to maintain and update a HPC software stack

- HPC package managers

- EasyBuild (UGhent, source)

- <https://easybuild.readthedocs.io/en/latest/>



- Spack (LLNL, source)

- <https://spack.io/>



- OpenHPC (Linux Foundation, binary)

- <https://openhpc.community/>



Automated Installation



- EasyBuild (<tool> = Scalasca | Score-P | CubeGUI)
 - Search for suitable easyconfigs: `eb -S <tool>`
 - Copy best matching easyconfig and adapt desired version and toolchain
 - Install: `eb <tool>-<version>-<toolchain>.eb`
- Spack (<tool> = scalasca | scorep | cube)
 - Install: `spack install <tool>@<version> %<compiler> ^<mpi-library>`
- OpenHPC (<tool> = scalasca | scorep)
 - Use Linux package manager (zypper, yum, ...) to install suitable RPM
 - E.g. `zypper install <tool>-<compiler>-<mpi>-<ohpcversion>-<arch>.rpm`



Where to get help



Tool	Support Email
Score-P	support@score-p.org
Scalasca, Cube	scalasca@fz-juelich.de

HPC Package Manager	Support Email List
EasyBuild	easybuild@lists.ugent.be
Spack	https://groups.google.com/d/forum/spack
OpenHPC	https://groups.io/g/openhpc-users





Performance Optimisation and Productivity

A Centre of Excellence in HPC

Contact:

 <https://www.pop-coe.eu>

 pop@bsc.es

 [@POP_HPC](https://twitter.com/POP_HPC)

 youtube.com/POPHPC

