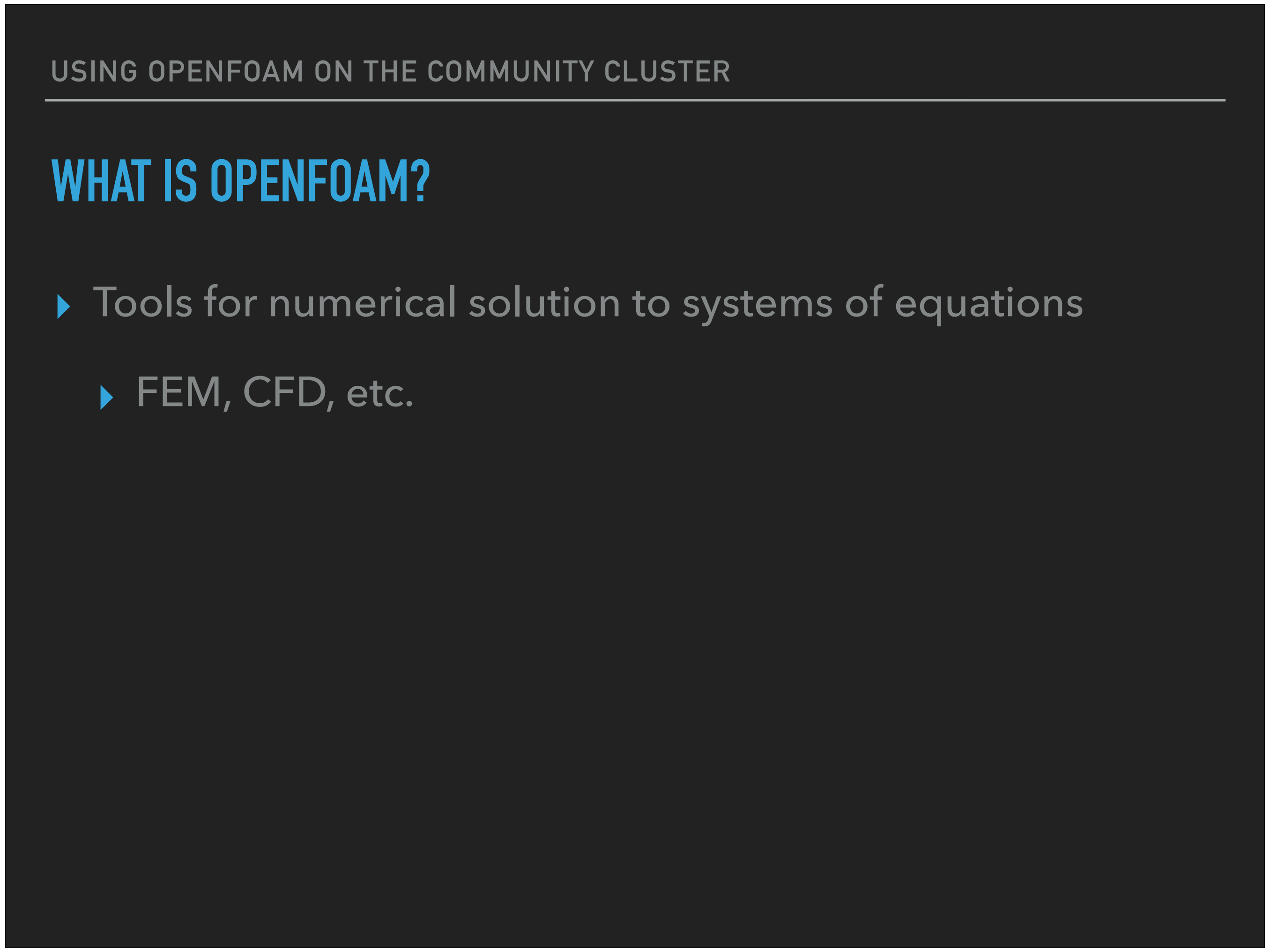
HPC INFORMATION SESSION

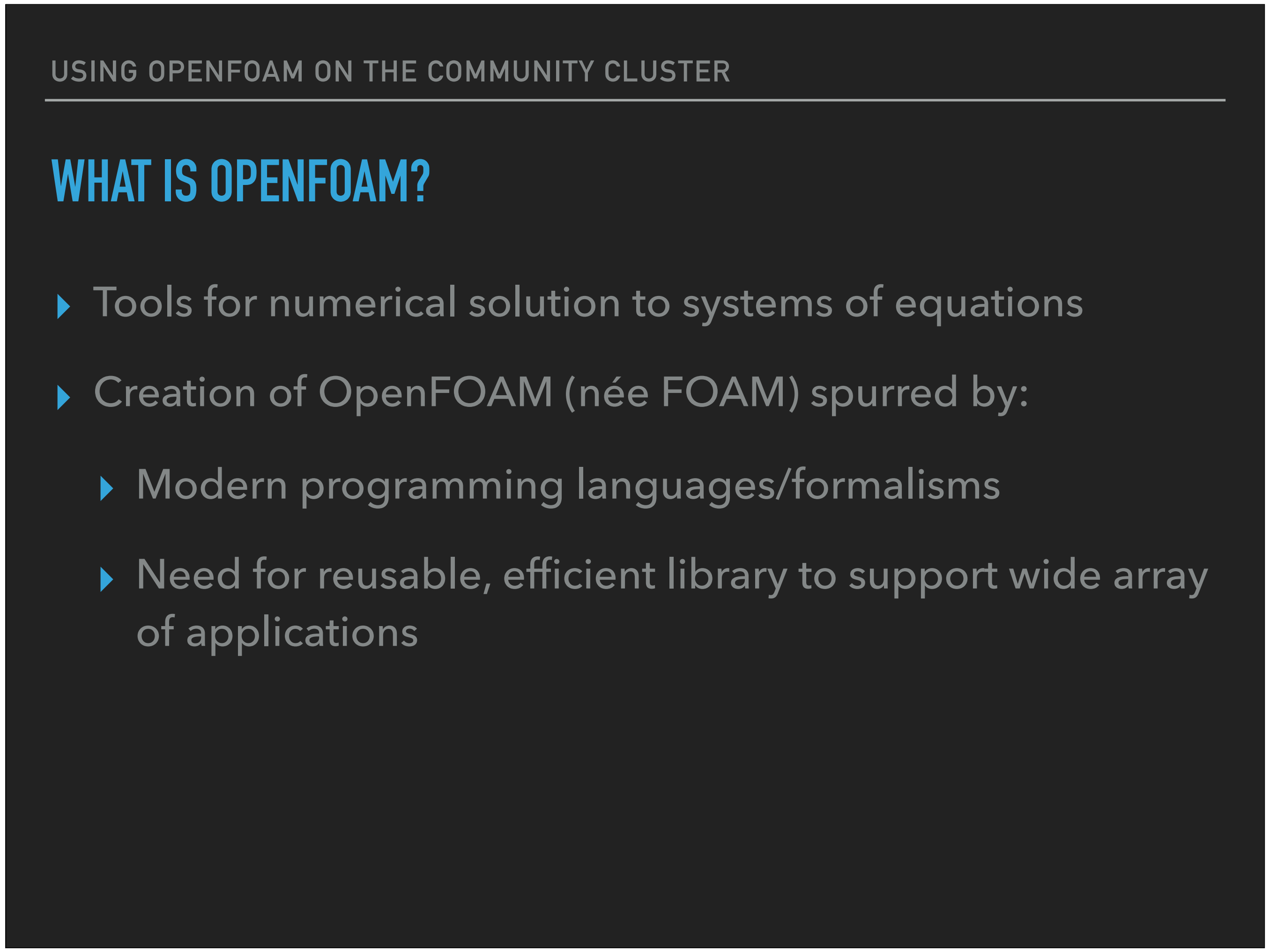
USING OPENFOAM ON THE COMMUNITY CLUSTER



WHAT IS OPENFOAM?

- Tools for numerical solution to systems of equations
  - FEM, CFD, etc.

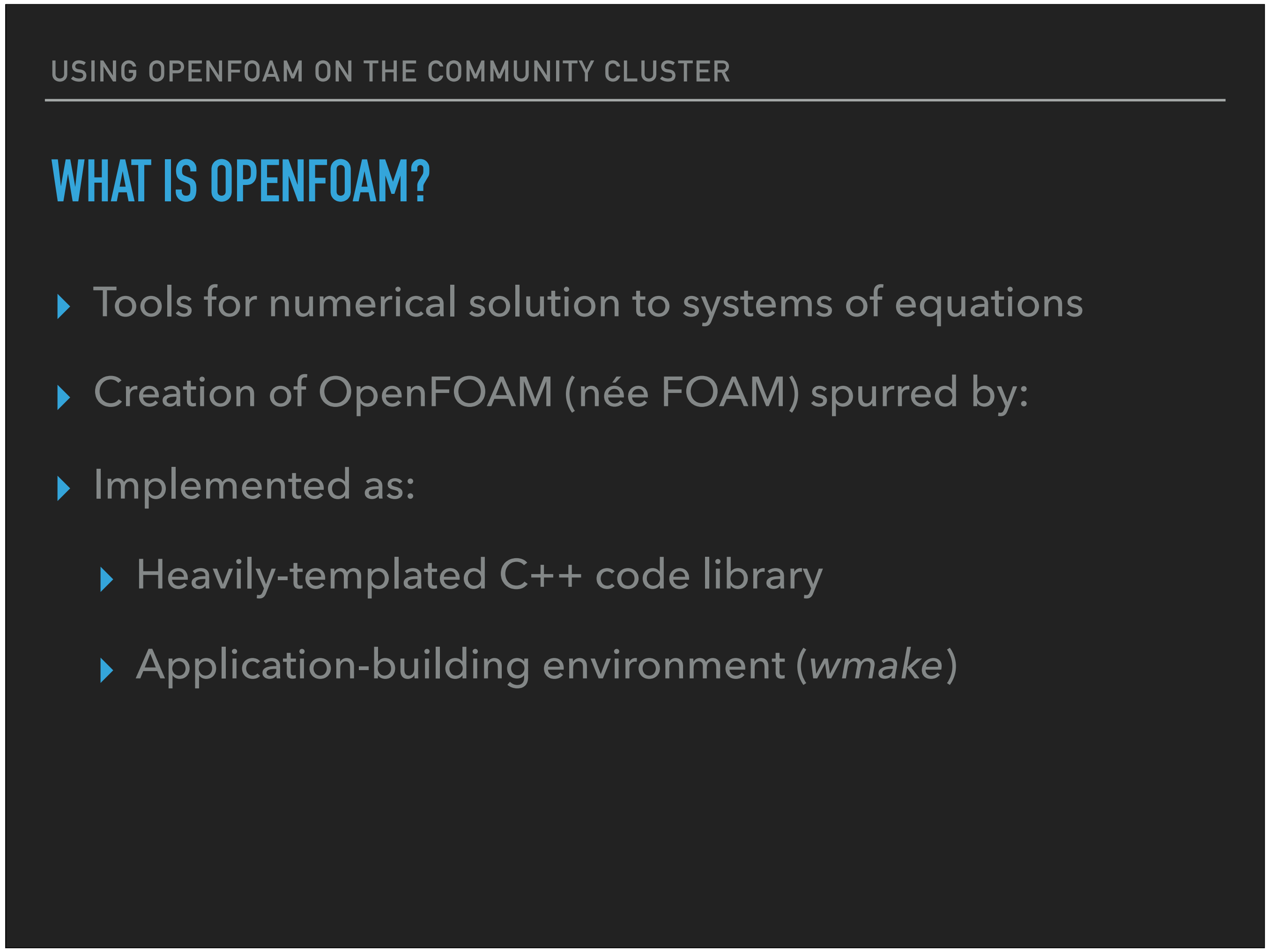
- Solve equation(s) on a grid? OpenFOAM can do that



WHAT IS OPENFOAM?

- Tools for numerical solution to systems of equations
- Creation of OpenFOAM (née FOAM) spurred by:
  - Modern programming languages/formalisms
  - Need for reusable, efficient library to support wide array of applications

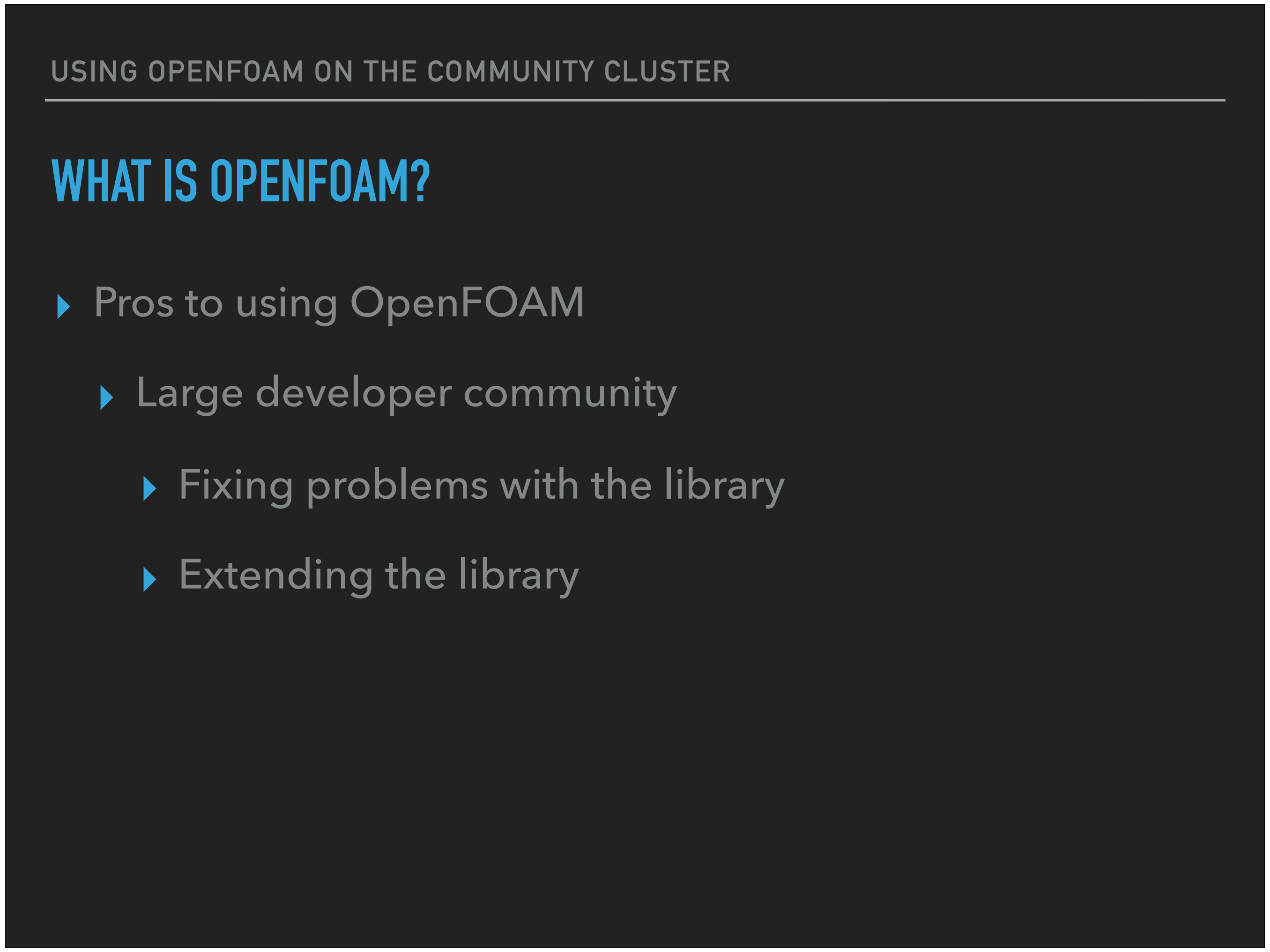
- Original author was tired of constantly writing tons of Fortran for every project
  - Who can blame him — Fortran modularity/reusability is still not great
  - Many Fortran programs still end up borrowing and adapting code versus using a library
- Paradigms like object-oriented programming, templated design, not prevalent with Fortran



WHAT IS OPENFOAM?

- Tools for numerical solution to systems of equations
- Creation of OpenFOAM (née FOAM) spurred by:
- Implemented as:
  - Heavily-templated C++ code library
  - Application-building environment (*wmake*)

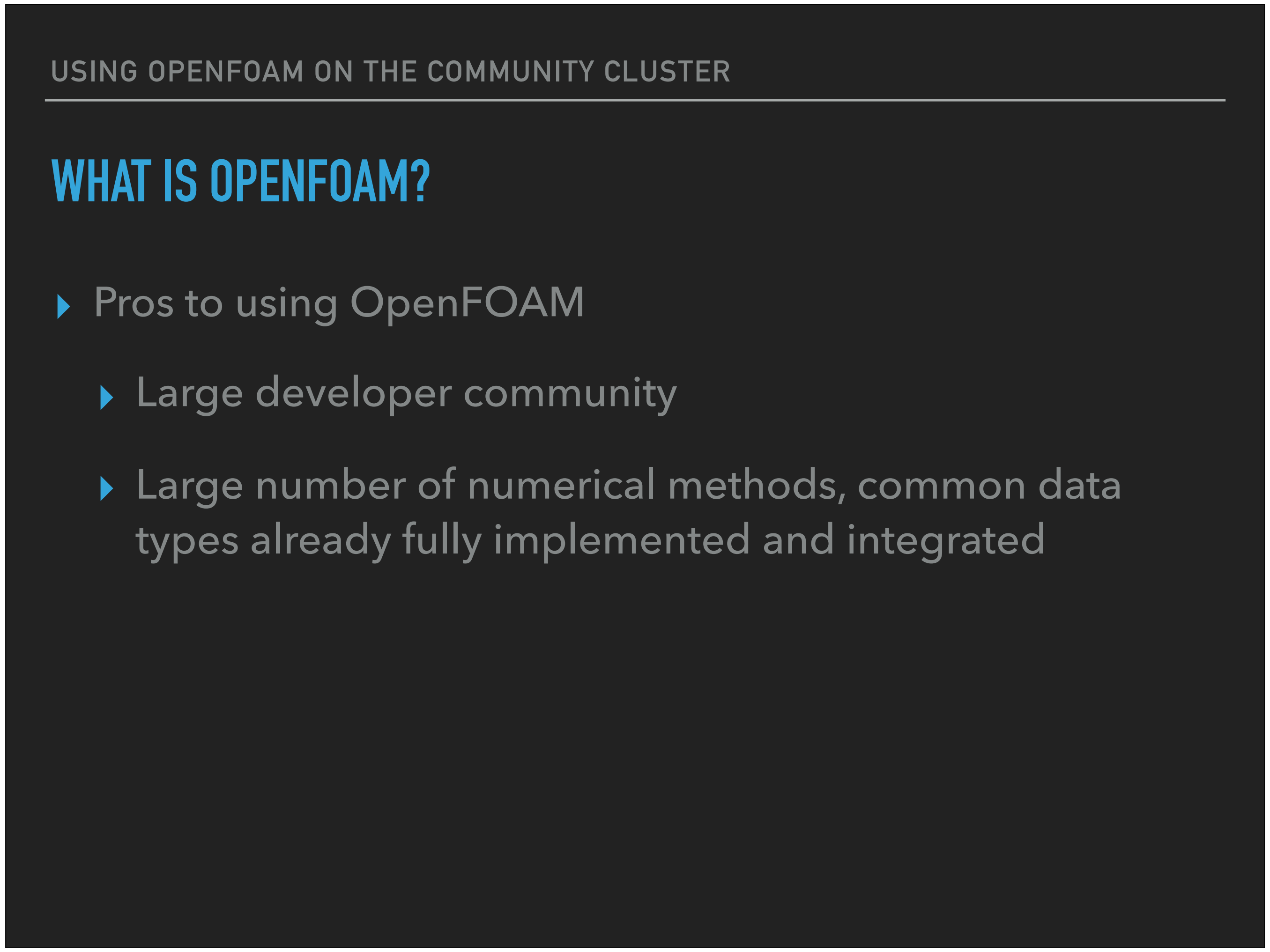
- Original author was tired of constantly writing tons of Fortran for every project
  - Who can blame him — Fortran modularity/reusability is still not great
  - Many Fortran programs still end up borrowing and adapting code versus using a library
- Paradigms like object-oriented programming, templated design, not prevalent with Fortran



WHAT IS OPENFOAM?

- Pros to using OpenFOAM
  - Large developer community
    - Fixing problems with the library
    - Extending the library

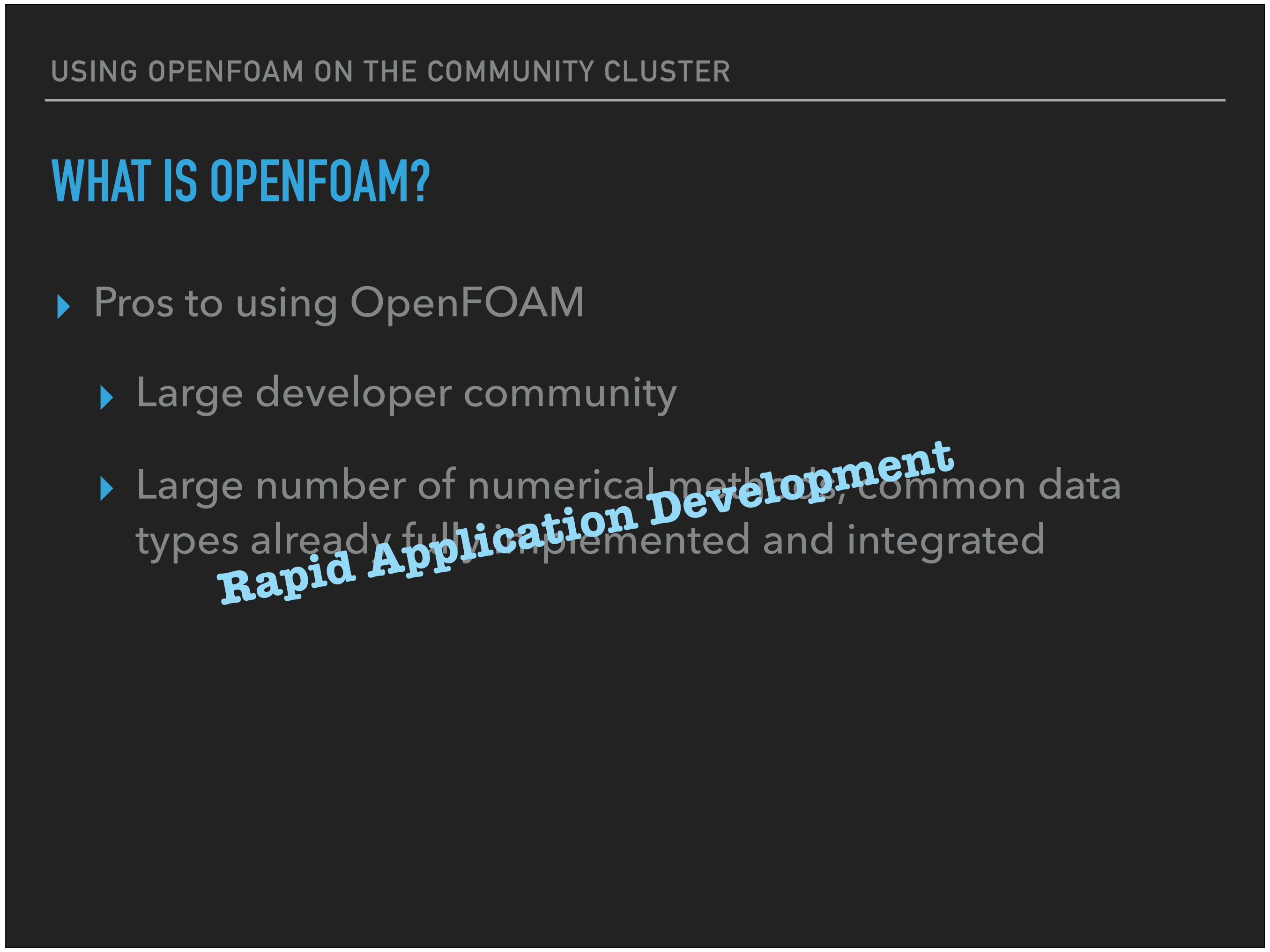
- The amount of code in each OpenFOAM release is astounding; easy to imagine there must be bugs present
  - Complexity isn't a bug, but making OpenFOAM easier to use DOES address a problem
- Addition of new methods or data types is ongoing
  - Necessary given the goal of OpenFOAM: a wide-coverage tool set



WHAT IS OPENFOAM?

- Pros to using OpenFOAM
  - Large developer community
  - Large number of numerical methods, common data types already fully implemented and integrated

- General change in paradigm for scientific computing over the last decade:
  - BEFORE: model the problem, outline the solution, translate the solution to Fortran
  - TODAY: avoid reinventing the wheel each time by building atop existing tools
  - What has facilitated that change: faster CPUs, more memory, new blood (younger scientists)
- Goal of rapid application development: faster turnaround from idea to working software

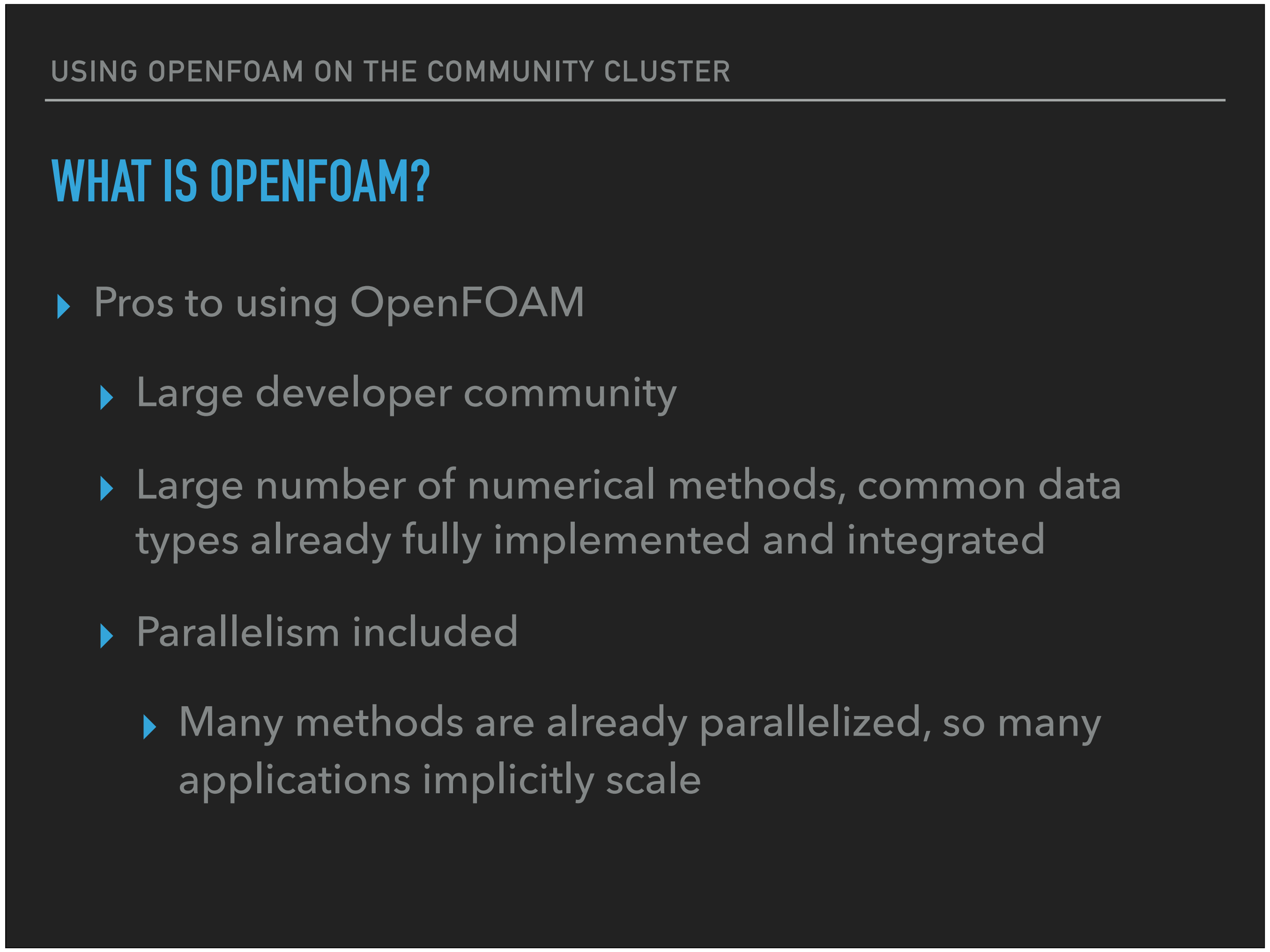


WHAT IS OPENFOAM?

- Pros to using OpenFOAM
  - Large developer community
  - Large number of numerical methods, common data types already fully implemented and integrated

Rapid Application Development

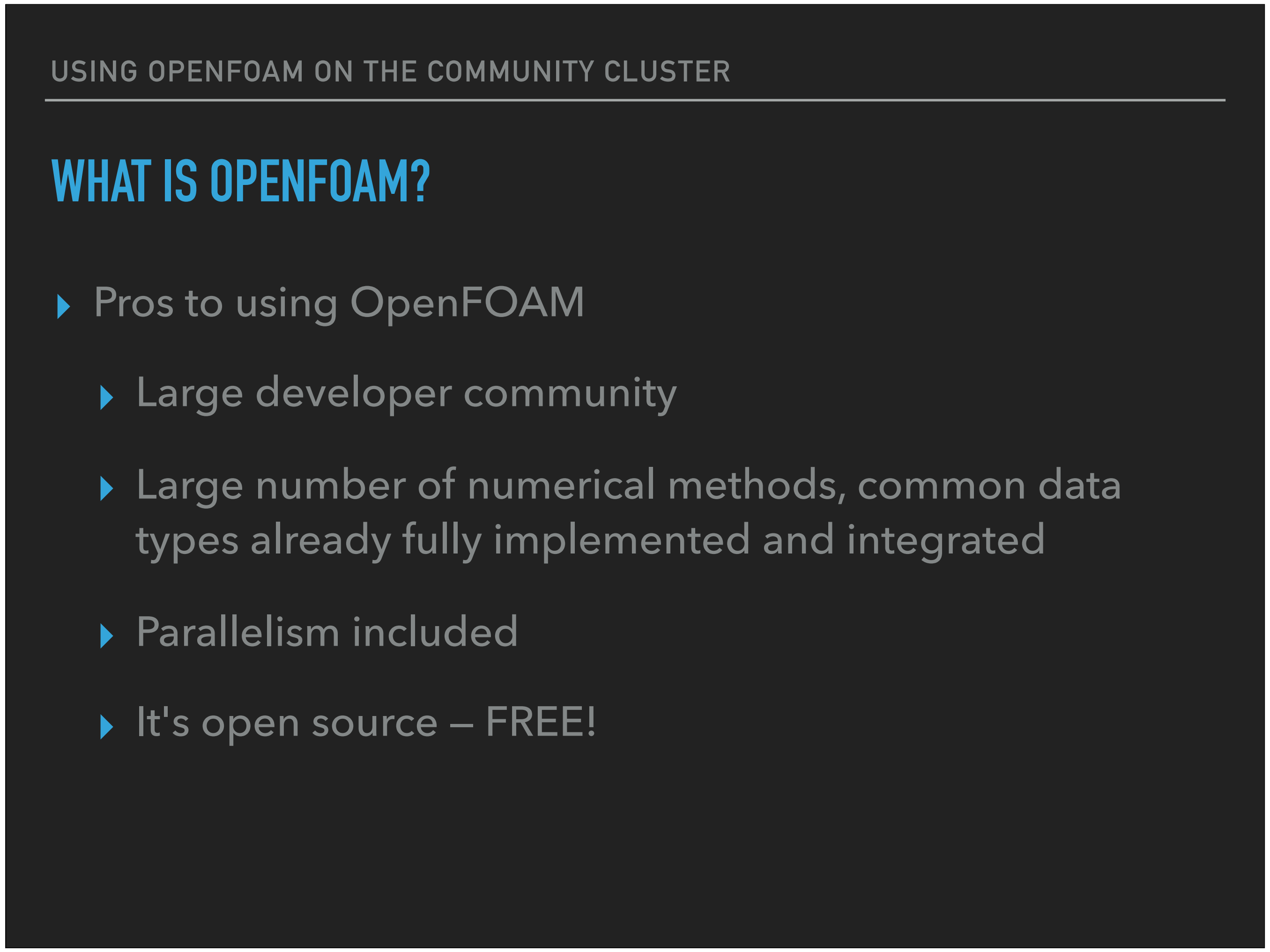
- General change in paradigm for scientific computing over the last decade:
  - BEFORE: model the problem, outline the solution, translate the solution to Fortran
  - TODAY: avoid reinventing the wheel each time by building atop existing tools
  - What has facilitated that change: faster CPUs, more memory, new blood (younger scientists)
- Goal of rapid application development: faster turnaround from idea to working software



WHAT IS OPENFOAM?

- Pros to using OpenFOAM
  - Large developer community
  - Large number of numerical methods, common data types already fully implemented and integrated
  - Parallelism included
    - Many methods are already parallelized, so many applications implicitly scale

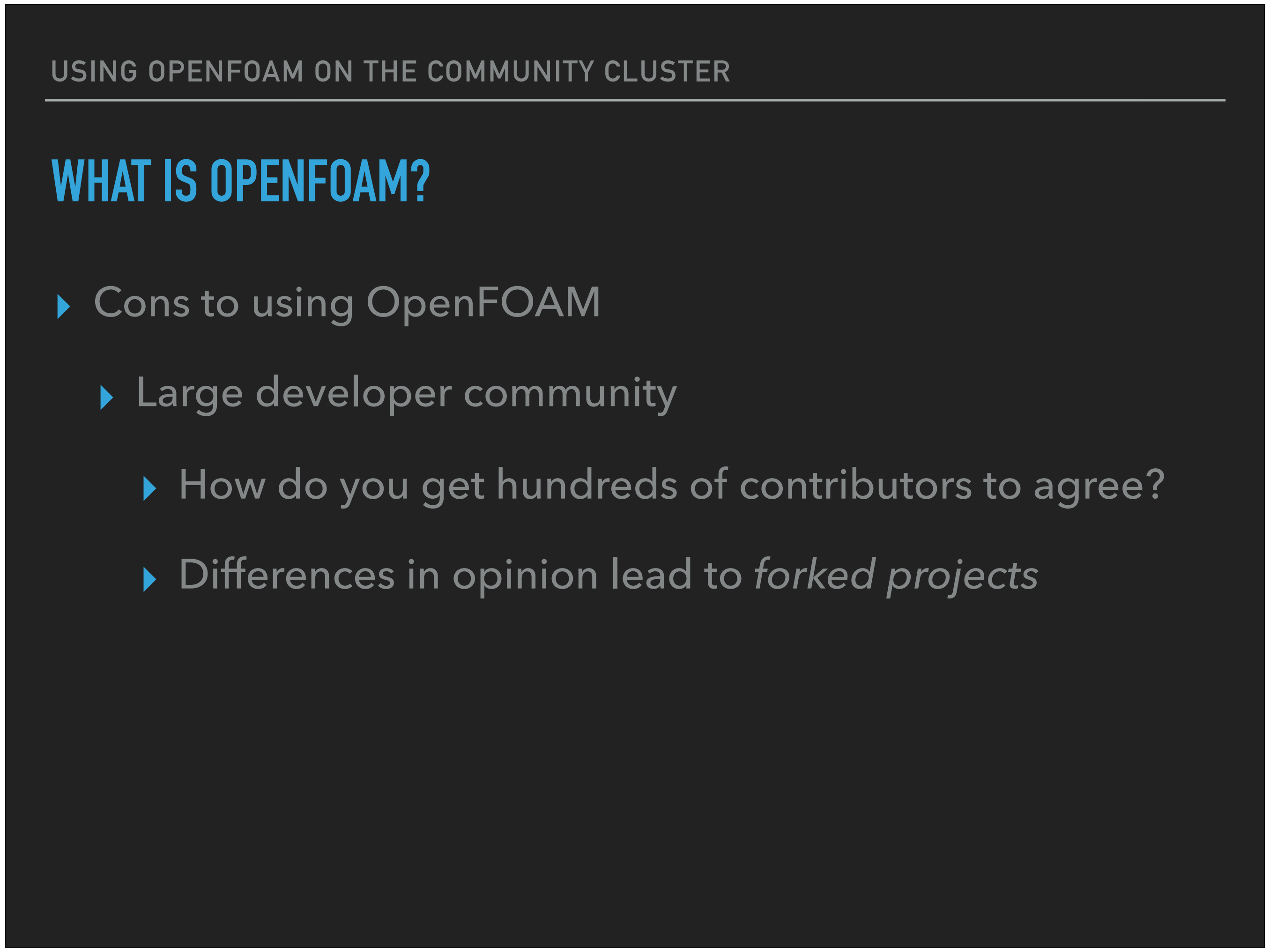
- MPI parallelism dominates modern supercomputing systems
  - OpenFOAM built atop Open MPI



WHAT IS OPENFOAM?

- Pros to using OpenFOAM
  - Large developer community
  - Large number of numerical methods, common data types already fully implemented and integrated
  - Parallelism included
  - It's open source – FREE!

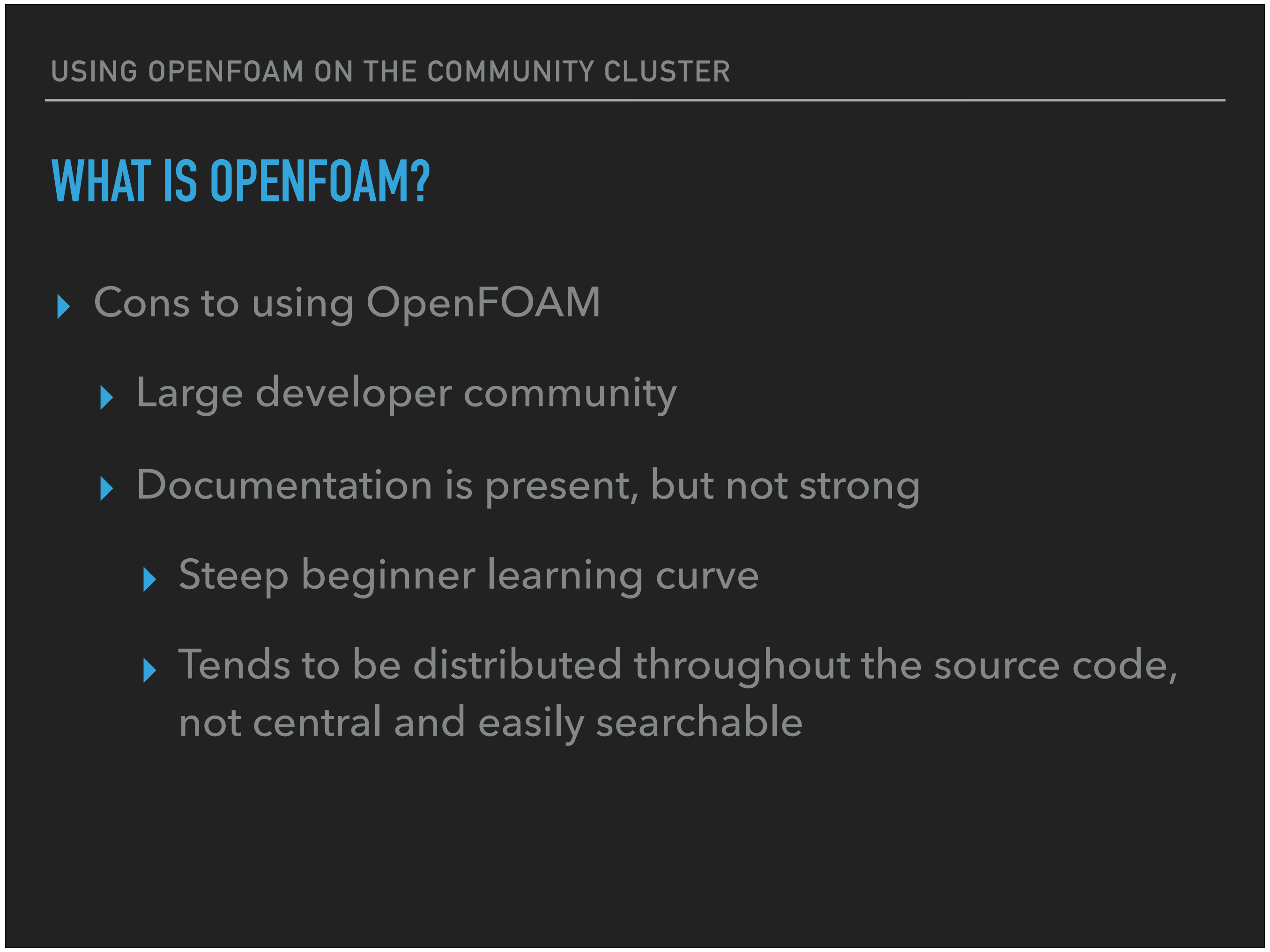
- Paid support options exist, as well



WHAT IS OPENFOAM?

- Cons to using OpenFOAM
  - Large developer community
    - How do you get hundreds of contributors to agree?
    - Differences in opinion lead to *forked projects*

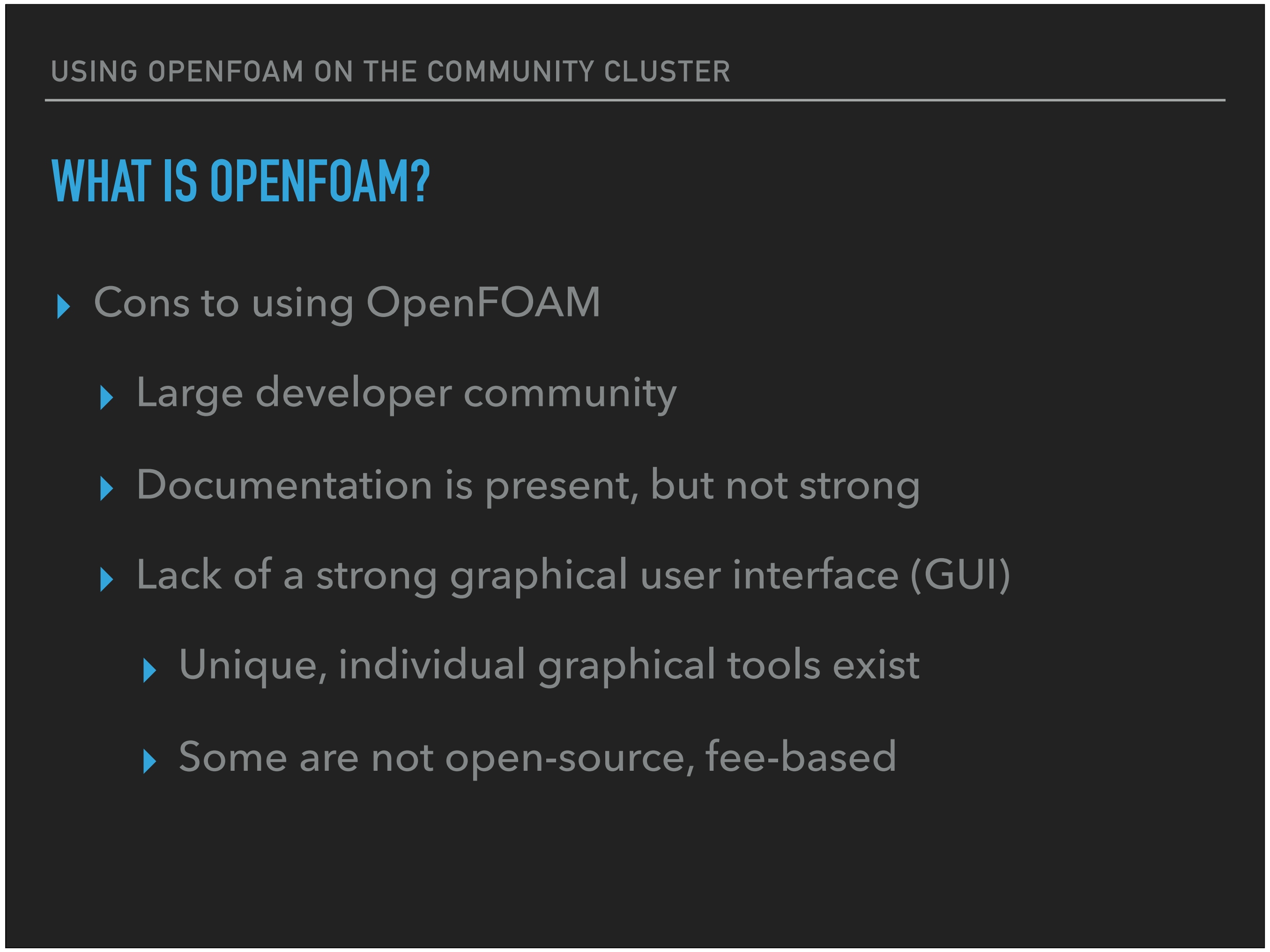
- Having a large developer community can also be detrimental
  - Additions by committee: can become mired in debate/politics, delayed implementation
  - Addition by contribution: can lead to multiple disparate implementations



WHAT IS OPENFOAM?

- Cons to using OpenFOAM
  - Large developer community
  - Documentation is present, but not strong
    - Steep beginner learning curve
    - Tends to be distributed throughout the source code, not central and easily searchable

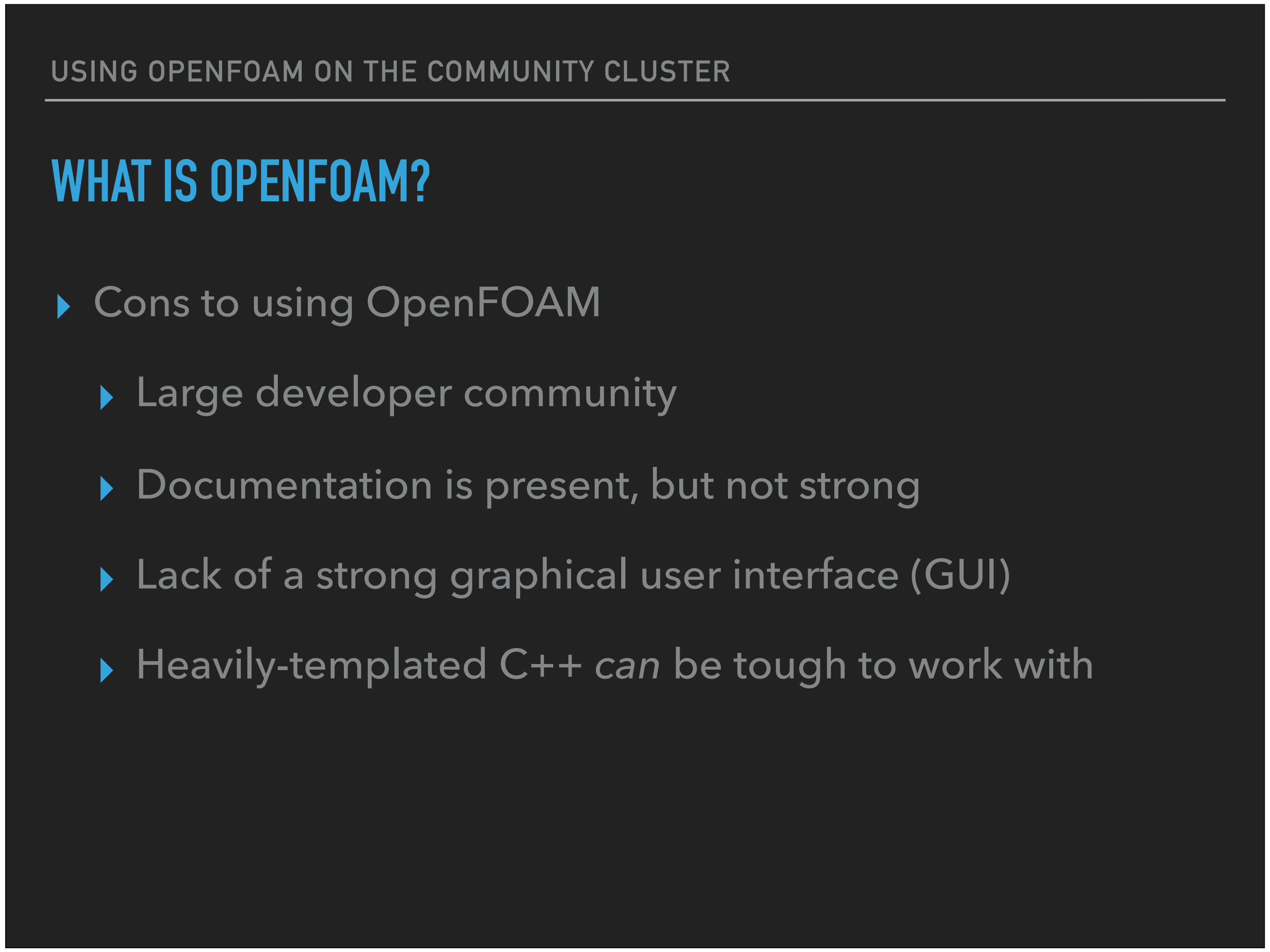
- Steep learning curve to
  - getting familiar with the OpenFOAM ecosystem
  - learning how the components of OpenFOAM fit together and are used
- …means that it could take just as long to become fluent in OpenFOAM as it could to just write your project yourself!



WHAT IS OPENFOAM?

- Cons to using OpenFOAM
    - Large developer community
    - Documentation is present, but not strong
    - Lack of a strong graphical user interface (GUI)
        - Unique, individual graphical tools exist
        - Some are not open-source, fee-based

- By their nature, numerical problems require graphical post-processing and visualization
- Without built-in tools for this part of the problem, OpenFOAM is a less cohesive solution



WHAT IS OPENFOAM?

- Cons to using OpenFOAM
  - Large developer community
  - Documentation is present, but not strong
  - Lack of a strong graphical user interface (GUI)
  - Heavily-templated C++ *can* be tough to work with

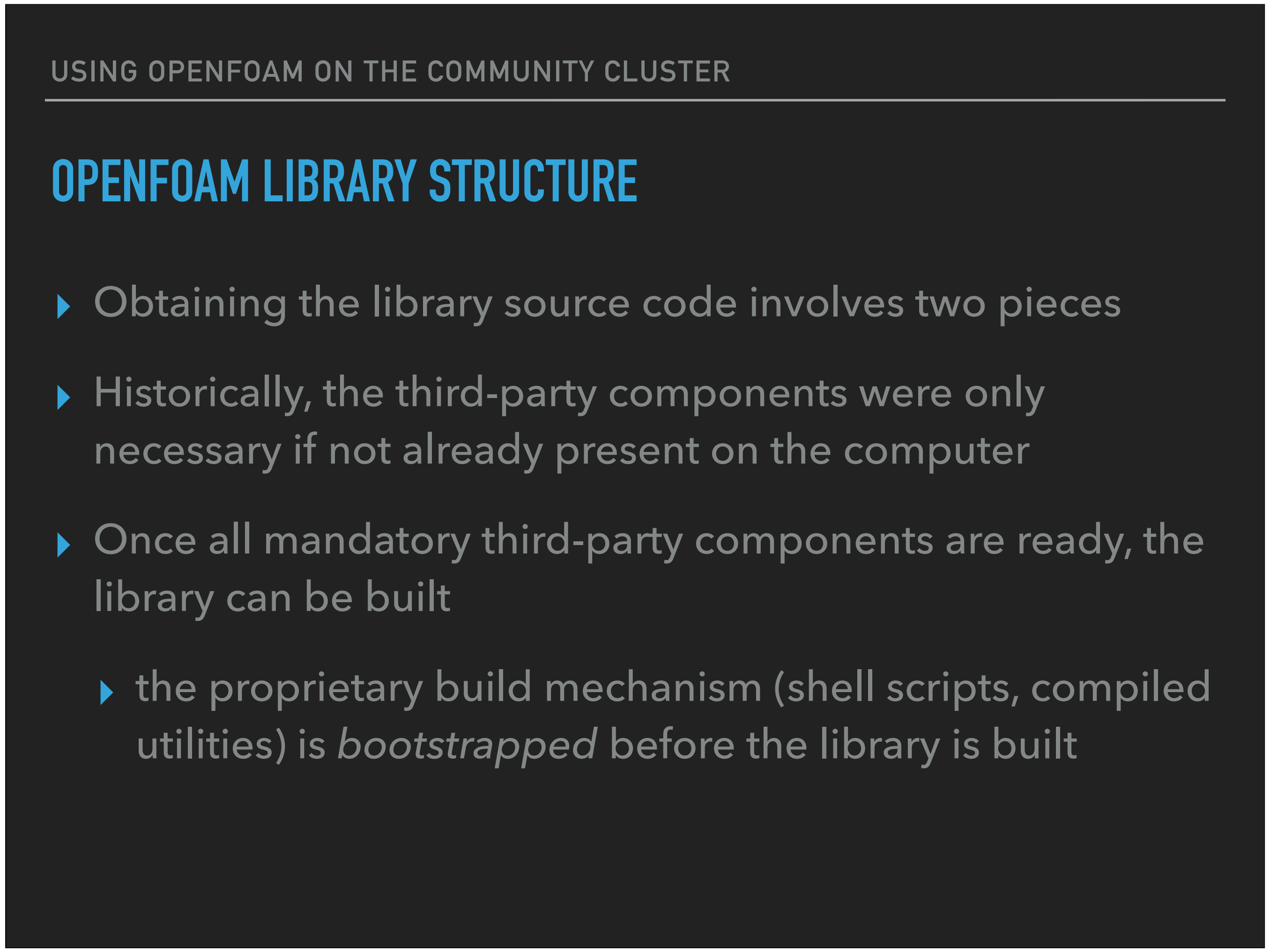
- Like any programming language, C++ isn't always fun, awesome, easy, simple, powerful, add your own adjectives
  - Strong static typing in C++ leads to fragile class hierarchy and frequent rebuilds
  - While templating does reduce the amount of explicit code that must be written, it can lead to inefficiency (how *smart* can the compiler be??), abuse (it's not always the best solution), and difficult-to-debug code (ever seen a C++ compiler dump a multi-page backtrace of heavily-templated source for a simple logic error??)

OPENFOAM LIBRARY STRUCTURE

- Obtaining the library source code involves two pieces
  - OpenFOAM: the source code for all methods, data types, and tools in the library
  - ThirdParty: external open source code that underpins some of OpenFOAM's key functionalities
    - E.g. Open MPI for distributed computing, BOOST for core C++ data types, CGAL for coordinate systems

OPENFOAM LIBRARY STRUCTURE

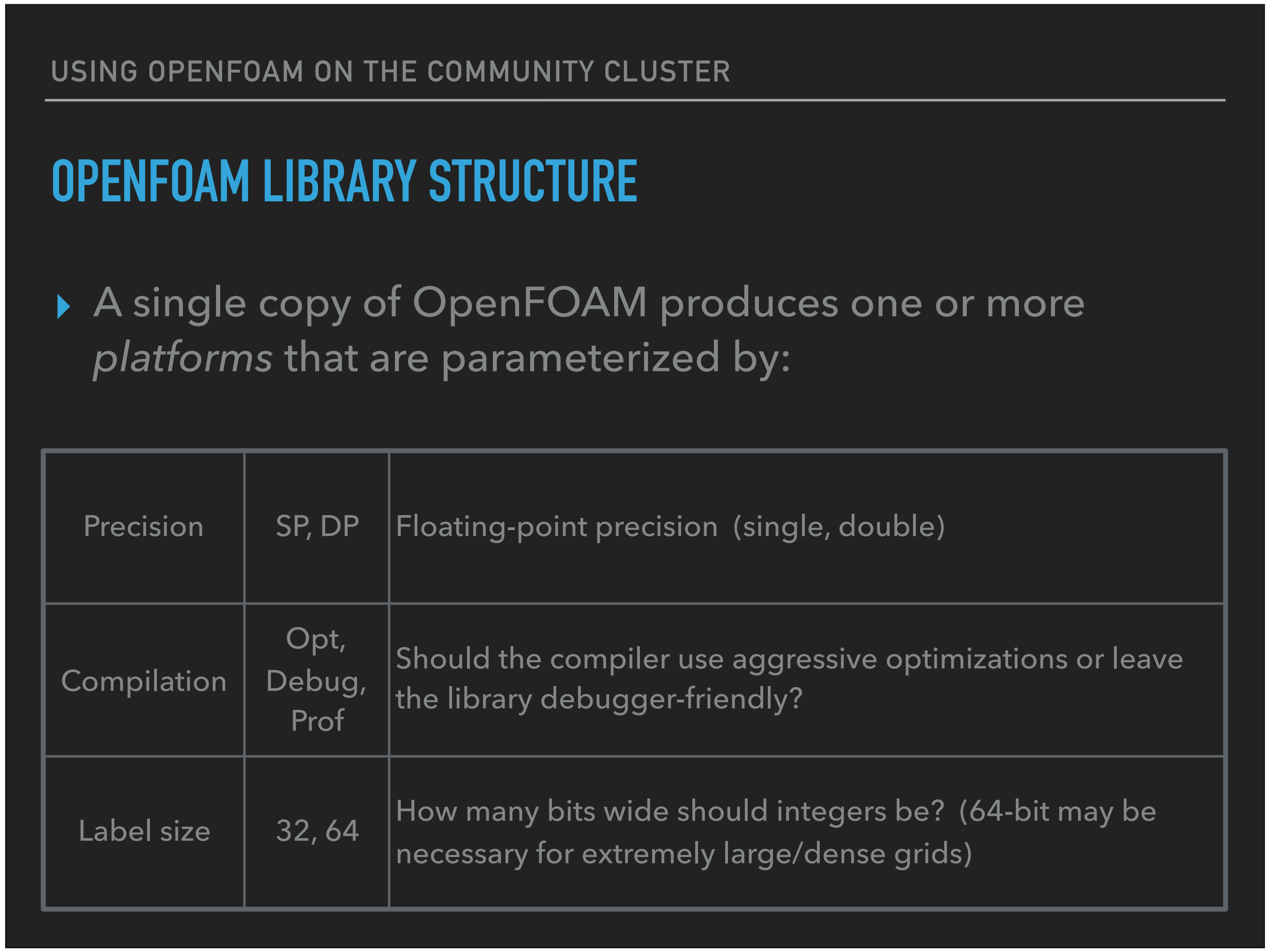
- Obtaining the library source code involves two pieces
- Historically, the third-party components were only necessary if not already present on the computer
    - Latest releases seem to have such strict compatibility guidelines that it is easiest to just use the OpenFOAM bundle
    - Must build these pieces FIRST



OPENFOAM LIBRARY STRUCTURE

- Obtaining the library source code involves two pieces
- Historically, the third-party components were only necessary if not already present on the computer
- Once all mandatory third-party components are ready, the library can be built
  - the proprietary build mechanism (shell scripts, compiled utilities) is *bootstrapped* before the library is built

- Not only has OpenFOAM implemented its own build system, it essentially has its own environment management system present, as well
  - See wmake/ for the build tools
  - See etc/ for the environment tools

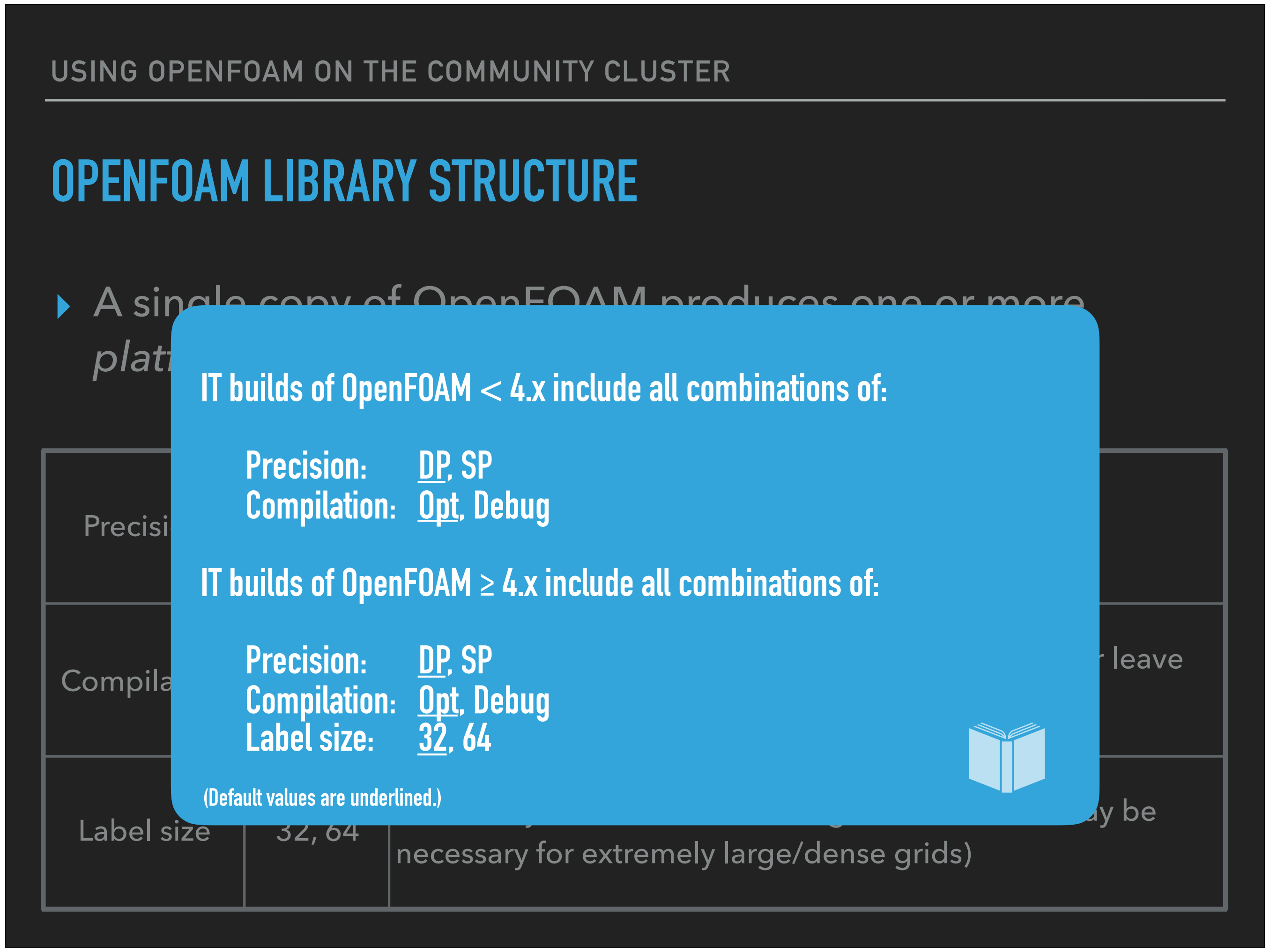


OPENFOAM LIBRARY STRUCTURE

- A single copy of OpenFOAM produces one or more *platforms* that are parameterized by:

Precision	SP, DP	Floating-point precision (single, double)
Compilation	Opt, Debug, Prof	Should the compiler use aggressive optimizations or leave the library debugger-friendly?
Label size	32, 64	How many bits wide should integers be? (64-bit may be necessary for extremely large/dense grids)

- LS-DYNA distributes their program in variants according to float-point precision
  - A very "Fortran-reminiscent" way of doing it, which is odd considering the library is pure C++
- When developing your OpenFOAM application using the "Debug" library is advisable
- Label size was added in version 4, defaults to 32
- IT builds all combos of SP/DP, Opt/Debug, and 32/64 (8 library variants)



USING OPENFOAM ON THE COMMUNITY CLUSTER

OPENFOAM LIBRARY STRUCTURE

- A single copy of OpenFOAM produces one or more *plat*

Precisi		
Compila		r leave
Label size	32, 64	ay be necessary for extremely large/dense grids)

IT builds of OpenFOAM < 4.x include all combinations of:

Precision: DP, SP
Compilation: Opt, Debug

IT builds of OpenFOAM ≥ 4.x include all combinations of:

Precision: DP, SP
Compilation: Opt, Debug
Label size: 32, 64

(Default values are underlined.)

- LS-DYNA distributes their program in variants according to float-point precision
  - A very "Fortran-reminiscent" way of doing it, which is odd considering the library is pure C++
- When developing your OpenFOAM application using the "Debug" library is advisable
- Label size was added in version 4, defaults to 32
- IT builds all combos of SP/DP, Opt/Debug, and 32/64 (8 library variants)

CONFIGURING YOUR ENVIRONMENT

- General procedure in the OpenFOAM environment management system:
  - Set environment variables that indicate directory containing OpenFOAM, platform parameters, directory of the project you are working on
  - *Source* a script into your shell that uses those environment variables to set the rest of the OpenFOAM environment variables accordingly

CONFIGURING YOUR ENVIRONMENT

- General procedure in the OpenFOAM environment management system:
- On community clusters, VALET handles some of this for you
    - Sets OpenFOAM directory, some immutable platform parameters
    - Sets an environment variable pointing to the script you must source to complete the setup

CONFIGURING YOUR ENVIRONMENT

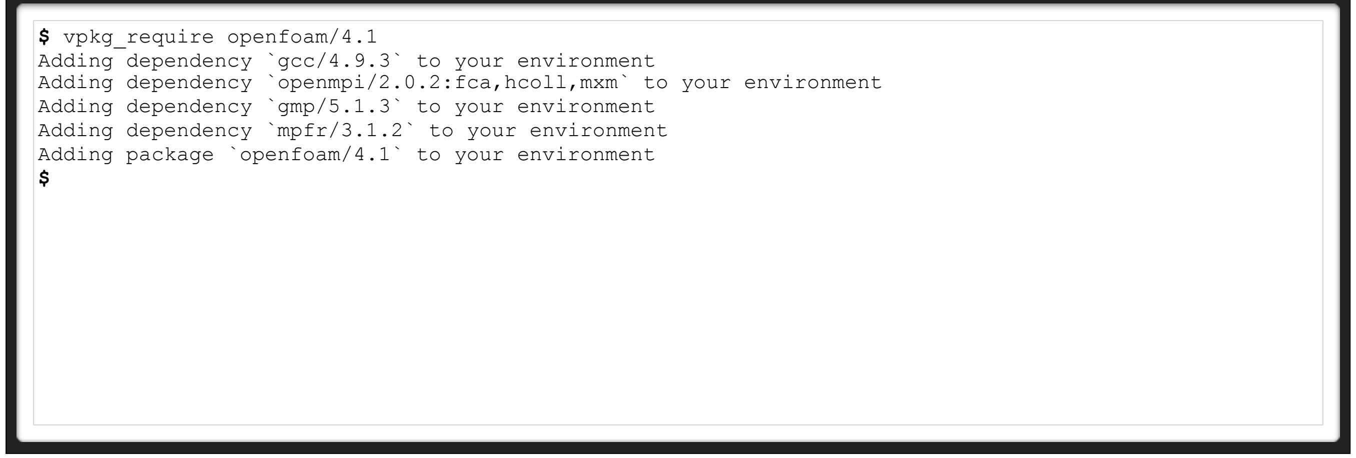
- General procedure in the OpenFOAM environment management system:
- On community clusters, VALET handles some of this for you
  - Se orm
    pa
  - Se you
    m

Before the OpenFOAM environment setup script is called, the directory containing your OpenFOAM project MUST be set. So the OpenFOAM VALET package does not source the script itself: it gets the environment ready for you (or your application's VALET package) to do so!

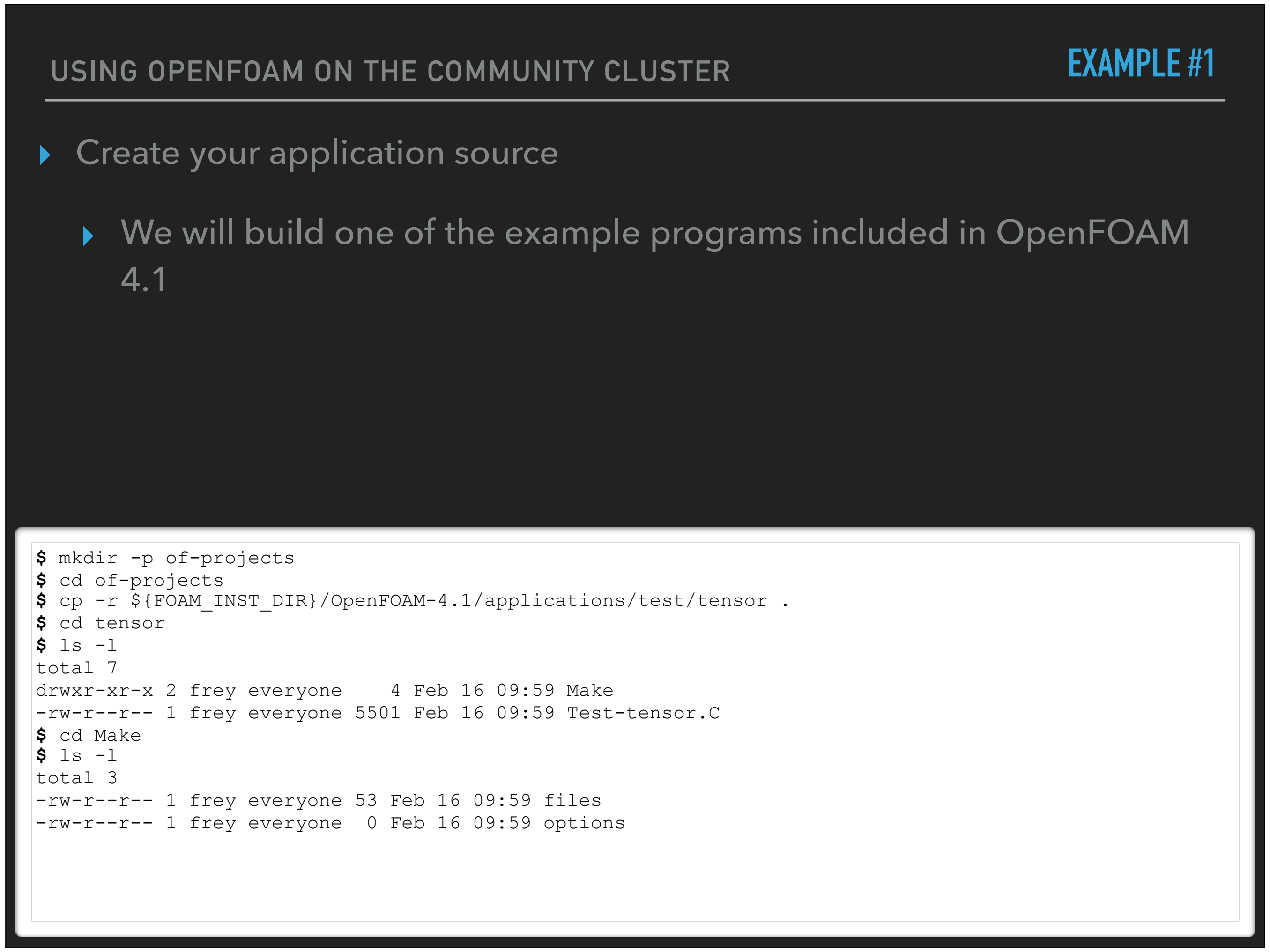
USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #1

- Login to Farber, then load OpenFOAM 4.1 into your shell



```
$ vpkg_require openfoam/4.1
Adding dependency `gcc/4.9.3` to your environment
Adding dependency `openmpi/2.0.2:fca,hcoll,mxm` to your environment
Adding dependency `gmp/5.1.3` to your environment
Adding dependency `mpfr/3.1.2` to your environment
Adding package `openfoam/4.1` to your environment
$
```



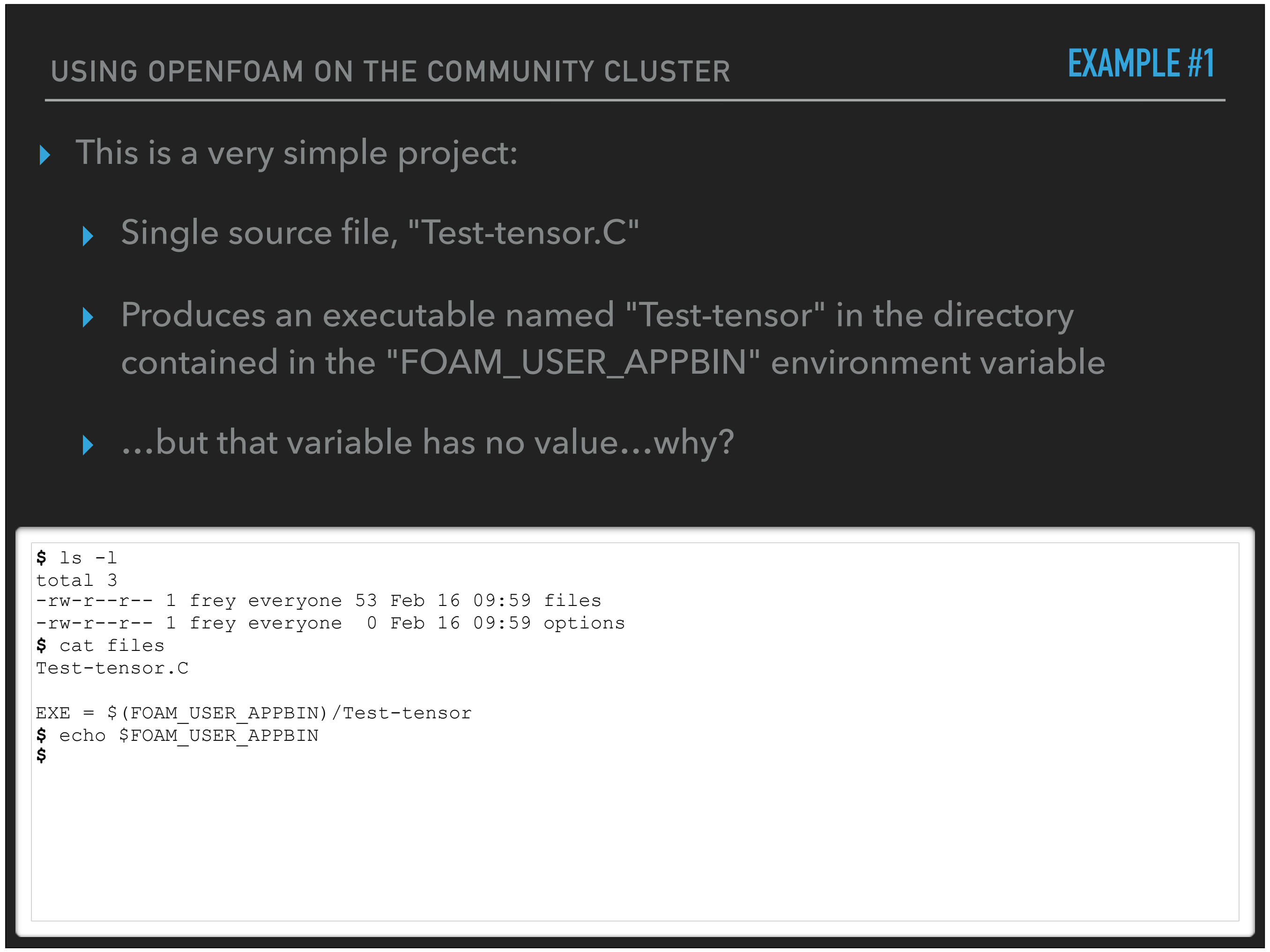
USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #1

- Create your application source
  - We will build one of the example programs included in OpenFOAM 4.1

```
$ mkdir -p of-projects
$ cd of-projects
$ cp -r ${FOAM_INST_DIR}/OpenFOAM-4.1/applications/test/tensor .
$ cd tensor
$ ls -l
total 7
drwxr-xr-x 2 frey everyone    4 Feb 16 09:59 Make
-rw-r--r-- 1 frey everyone 5501 Feb 16 09:59 Test-tensor.C
$ cd Make
$ ls -l
total 3
-rw-r--r-- 1 frey everyone 53 Feb 16 09:59 files
-rw-r--r-- 1 frey everyone  0 Feb 16 09:59 options
```

- The project includes a C++ source file and a "Make" directory



USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #1

- This is a very simple project:
  - Single source file, "Test-tensor.C"
  - Produces an executable named "Test-tensor" in the directory contained in the "FOAM_USER_APPBIN" environment variable
  - ...but that variable has no value...why?

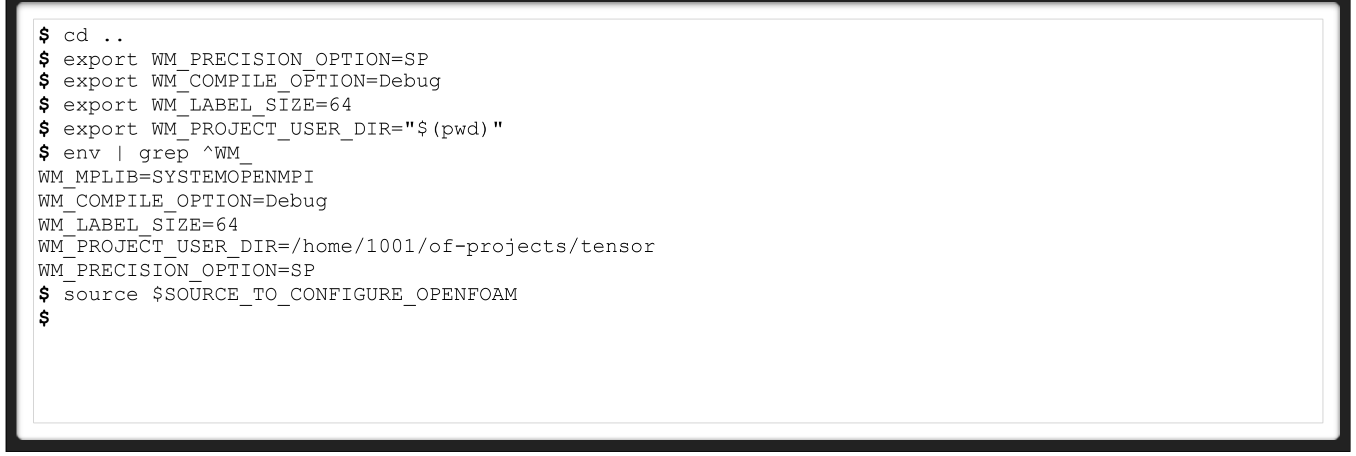
```
$ ls -l
total 3
-rw-r--r-- 1 frey everyone 53 Feb 16 09:59 files
-rw-r--r-- 1 frey everyone  0 Feb 16 09:59 options
$ cat files
Test-tensor.C

EXE = $(FOAM_USER_APPBIN)/Test-tensor
$ echo $FOAM_USER_APPBIN
$
```

- The "Make" directory contains files that tell the OpenFOAM build system what to build, using what components, and with what options

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #1

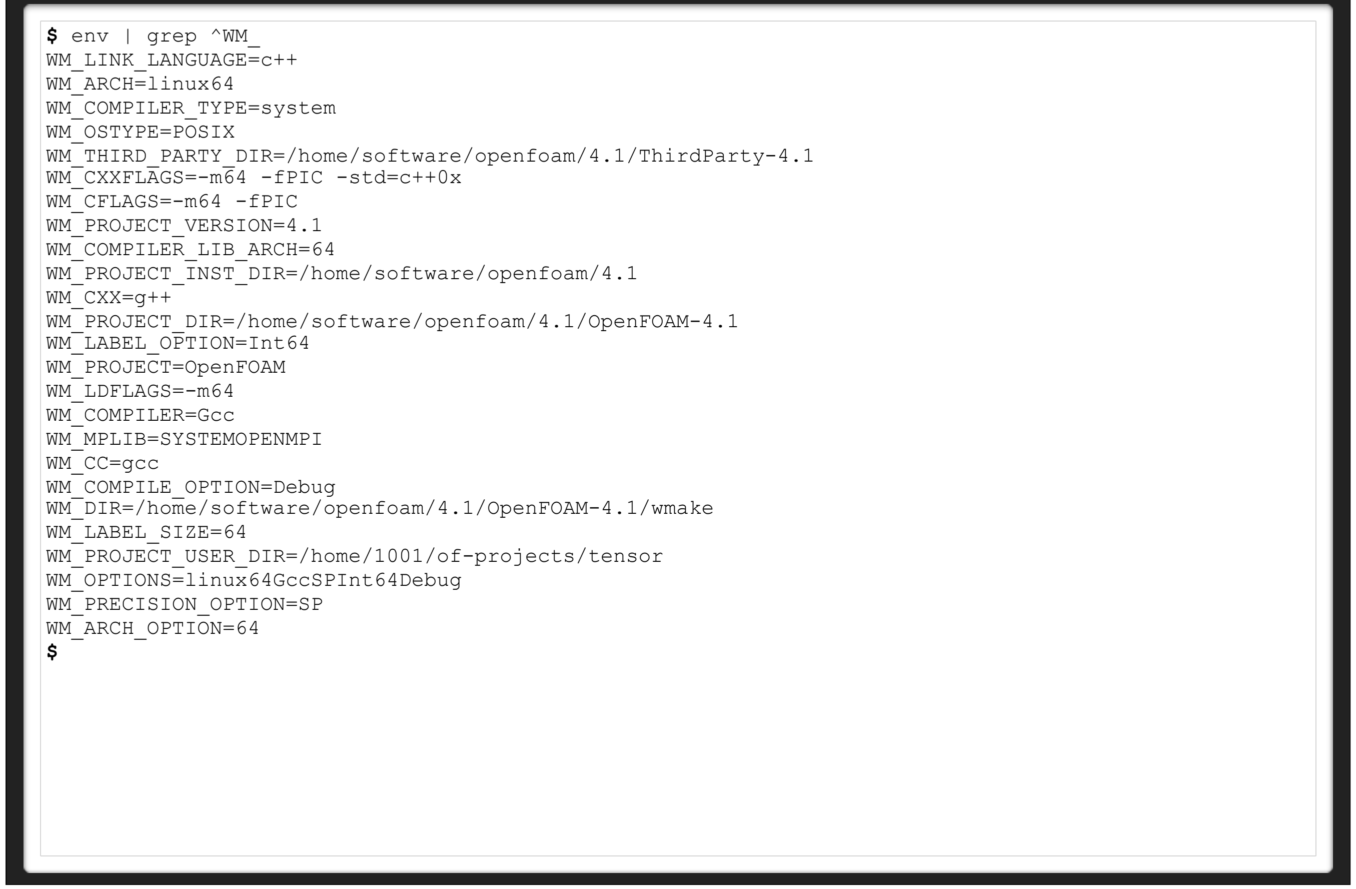
- Need to set platform parameters and complete the environment setup!
  - We're in the "Make" directory, go back one level
  - Let's do single-precision, debug, with 64-bit labels
  - Don't forget to set the project directory, too



```
$ cd ..
$ export WM_PRECISION_OPTION=SP
$ export WM_COMPILE_OPTION=Debug
$ export WM_LABEL_SIZE=64
$ export WM_PROJECT_USER_DIR="$(pwd)"
$ env | grep ^WM_
WM_MPLIB=SYSTEMOPENMPI
WM_COMPILE_OPTION=Debug
WM_LABEL_SIZE=64
WM_PROJECT_USER_DIR=/home/1001/of-projects/tensor
WM_PRECISION_OPTION=SP
$ source $SOURCE_TO_CONFIGURE_OPENFOAM
$
```

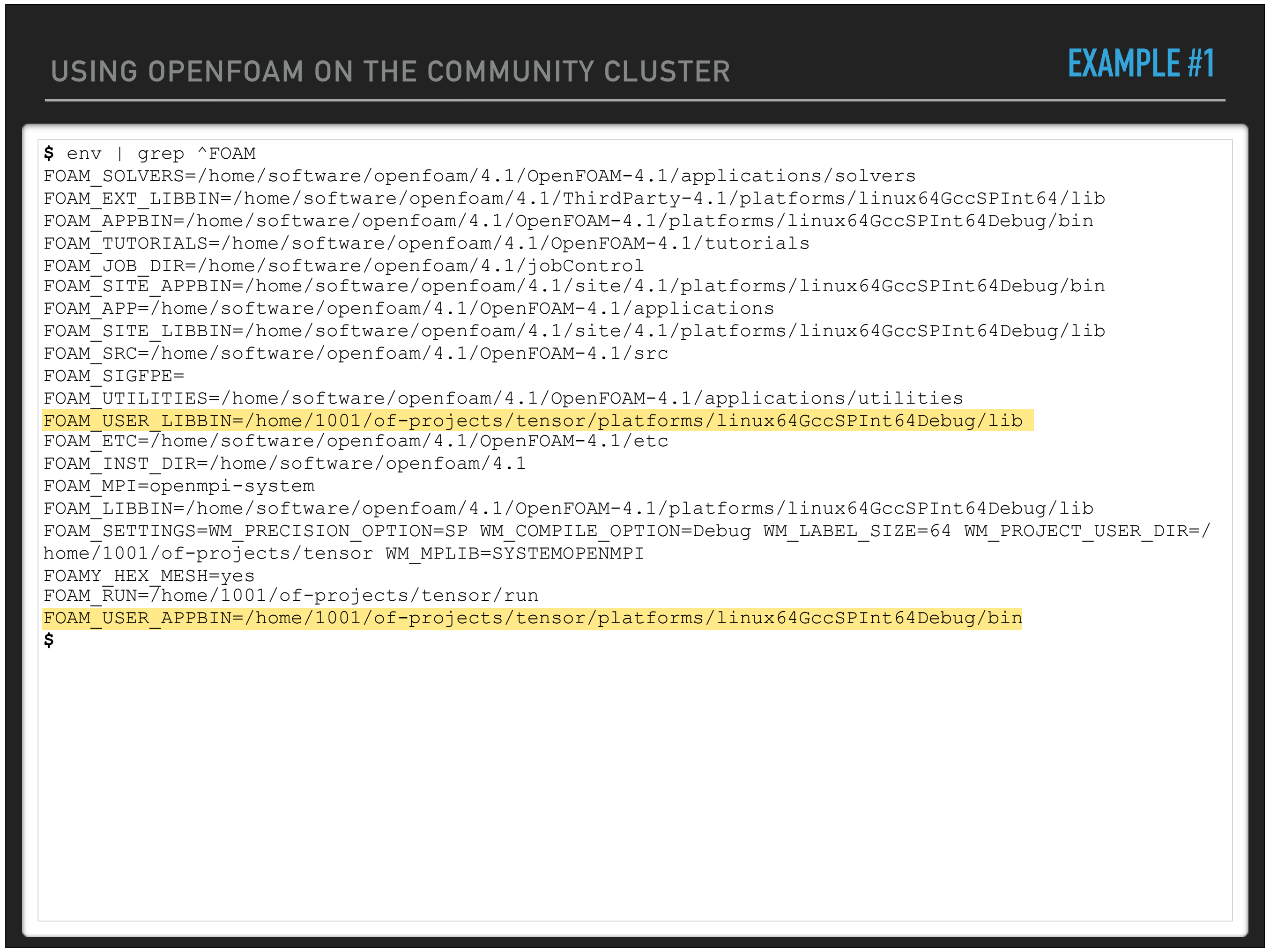
USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #1



```
$ env | grep ^WM_
WM_LINK_LANGUAGE=c++
WM_ARCH=linux64
WM_COMPILER_TYPE=system
WM_OSTYPE=POSIX
WM_THIRD_PARTY_DIR=/home/software/openfoam/4.1/ThirdParty-4.1
WM_CXXFLAGS=-m64 -fPIC -std=c++0x
WM_CFLAGS=-m64 -fPIC
WM_PROJECT_VERSION=4.1
WM_COMPILER_LIB_ARCH=64
WM_PROJECT_INST_DIR=/home/software/openfoam/4.1
WM_CXX=g++
WM_PROJECT_DIR=/home/software/openfoam/4.1/OpenFOAM-4.1
WM_LABEL_OPTION=Int64
WM_PROJECT=OpenFOAM
WM_LDFLAGS=-m64
WM_COMPILER=Gcc
WM_MPLIB=SYSTEMOPENMPI
WM_CC=gcc
WM_COMPILE_OPTION=Debug
WM_DIR=/home/software/openfoam/4.1/OpenFOAM-4.1/wmake
WM_LABEL_SIZE=64
WM_PROJECT_USER_DIR=/home/1001/of-projects/tensor
WM_OPTIONS=linux64GccSPInt64Debug
WM_PRECISION_OPTION=SP
WM_ARCH_OPTION=64
$
```

- After completing the environment setup, many more WM_ variables are present

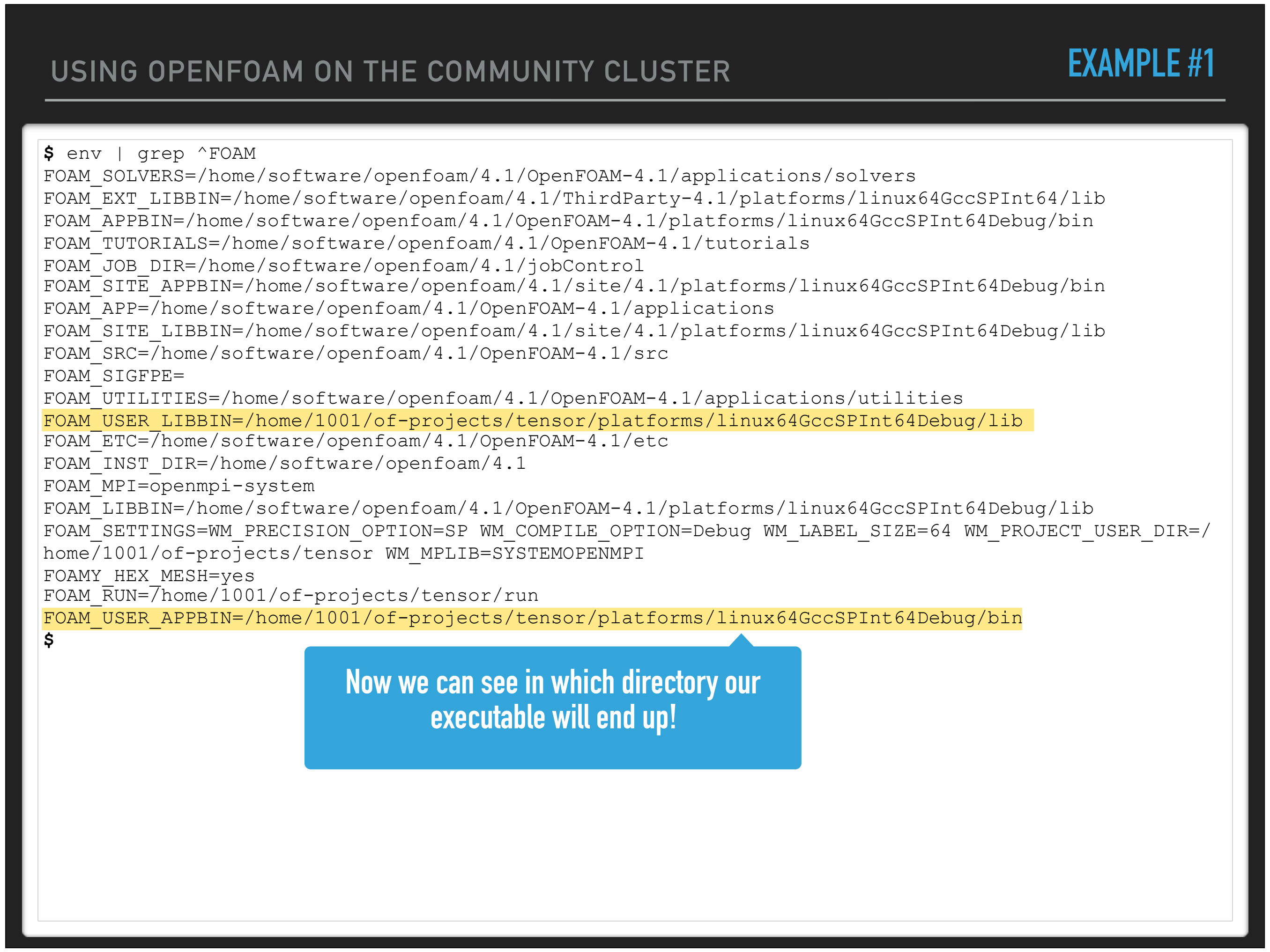


USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #1

```
$ env | grep ^FOAM
FOAM_SOLVERS=/home/software/openfoam/4.1/OpenFOAM-4.1/applications/solvers
FOAM_EXT_LIBBIN=/home/software/openfoam/4.1/ThirdParty-4.1/platforms/linux64GccSPInt64/lib
FOAM_APPBIN=/home/software/openfoam/4.1/OpenFOAM-4.1/platforms/linux64GccSPInt64Debug/bin
FOAM_TUTORIALS=/home/software/openfoam/4.1/OpenFOAM-4.1/tutorials
FOAM_JOB_DIR=/home/software/openfoam/4.1/jobControl
FOAM_SITE_APPBIN=/home/software/openfoam/4.1/site/4.1/platforms/linux64GccSPInt64Debug/bin
FOAM_APP=/home/software/openfoam/4.1/OpenFOAM-4.1/applications
FOAM_SITE_LIBBIN=/home/software/openfoam/4.1/site/4.1/platforms/linux64GccSPInt64Debug/lib
FOAM_SRC=/home/software/openfoam/4.1/OpenFOAM-4.1/src
FOAM_SIGFPE=
FOAM_UTILITIES=/home/software/openfoam/4.1/OpenFOAM-4.1/applications/utilities
FOAM_USER_LIBBIN=/home/1001/of-projects/tensor/platforms/linux64GccSPInt64Debug/lib
FOAM_ETC=/home/software/openfoam/4.1/OpenFOAM-4.1/etc
FOAM_INST_DIR=/home/software/openfoam/4.1
FOAM_MPI=openmpi-system
FOAM_LIBBIN=/home/software/openfoam/4.1/OpenFOAM-4.1/platforms/linux64GccSPInt64Debug/lib
FOAM_SETTINGS=WM_PRECISION_OPTION=SP WM_COMPILE_OPTION=Debug WM_LABEL_SIZE=64 WM_PROJECT_USER_DIR=/
home/1001/of-projects/tensor WM_MPLIB=SYSTEMOPENMPI
FOAMY_HEX_MESH=yes
FOAM_RUN=/home/1001/of-projects/tensor/run
FOAM_USER_APPBIN=/home/1001/of-projects/tensor/platforms/linux64GccSPInt64Debug/bin
$
```

- …quite a few FOAM_ variables, too



USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #1

```
$ env | grep ^FOAM
FOAM_SOLVERS=/home/software/openfoam/4.1/OpenFOAM-4.1/applications/solvers
FOAM_EXT_LIBBIN=/home/software/openfoam/4.1/ThirdParty-4.1/platforms/linux64GccSPInt64/lib
FOAM_APPBIN=/home/software/openfoam/4.1/OpenFOAM-4.1/platforms/linux64GccSPInt64Debug/bin
FOAM_TUTORIALS=/home/software/openfoam/4.1/OpenFOAM-4.1/tutorials
FOAM_JOB_DIR=/home/software/openfoam/4.1/jobControl
FOAM_SITE_APPBIN=/home/software/openfoam/4.1/site/4.1/platforms/linux64GccSPInt64Debug/bin
FOAM_APP=/home/software/openfoam/4.1/OpenFOAM-4.1/applications
FOAM_SITE_LIBBIN=/home/software/openfoam/4.1/site/4.1/platforms/linux64GccSPInt64Debug/lib
FOAM_SRC=/home/software/openfoam/4.1/OpenFOAM-4.1/src
FOAM_SIGFPE=
FOAM_UTILITIES=/home/software/openfoam/4.1/OpenFOAM-4.1/applications/utilities
FOAM_USER_LIBBIN=/home/1001/of-projects/tensor/platforms/linux64GccSPInt64Debug/lib
FOAM_ETC=/home/software/openfoam/4.1/OpenFOAM-4.1/etc
FOAM_INST_DIR=/home/software/openfoam/4.1
FOAM_MPI=openmpi-system
FOAM_LIBBIN=/home/software/openfoam/4.1/OpenFOAM-4.1/platforms/linux64GccSPInt64Debug/lib
FOAM_SETTINGS=WM_PRECISION_OPTION=SP WM_COMPILE_OPTION=Debug WM_LABEL_SIZE=64 WM_PROJECT_USER_DIR=/
home/1001/of-projects/tensor WM_MPLIB=SYSTEMOPENMPI
FOAMY_HEX_MESH=yes
FOAM_RUN=/home/1001/of-projects/tensor/run
FOAM_USER_APPBIN=/home/1001/of-projects/tensor/platforms/linux64GccSPInt64Debug/bin
$
```

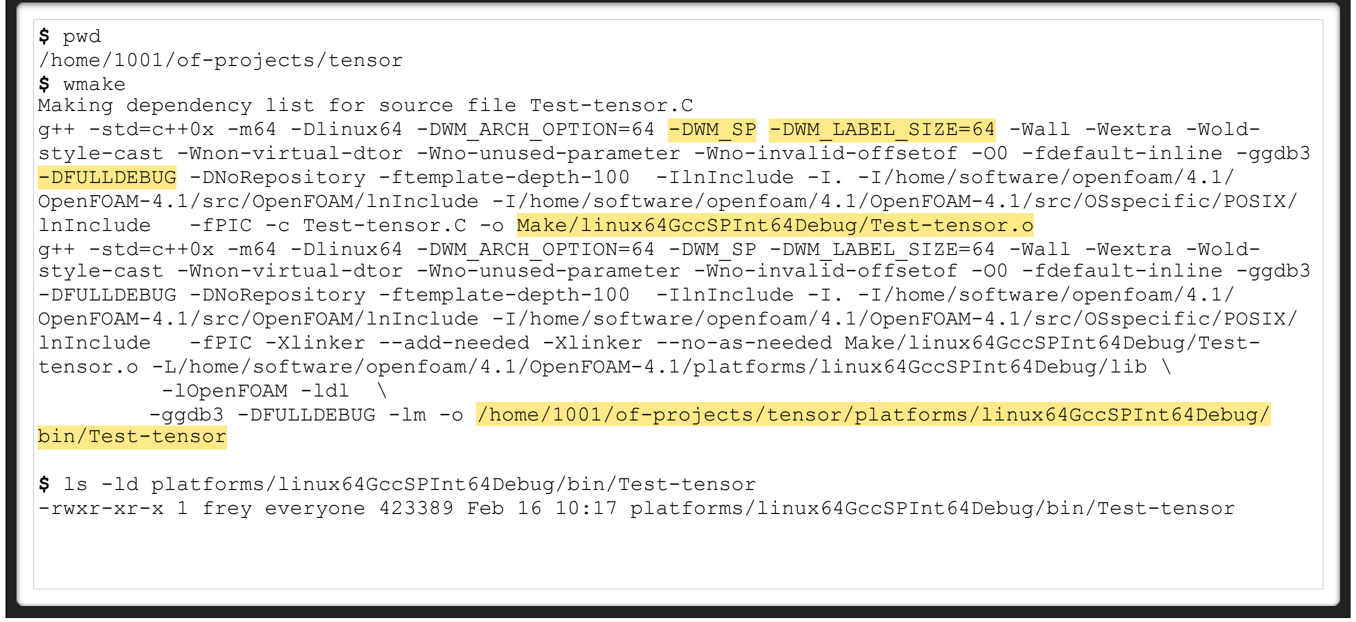
Now we can see in which directory our executable will end up!

- …quite a few FOAM_ variables, too

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #1

- Let's build the program
  - Rather than the Unixy "make" command, use the OpenFOAM "wmake" command

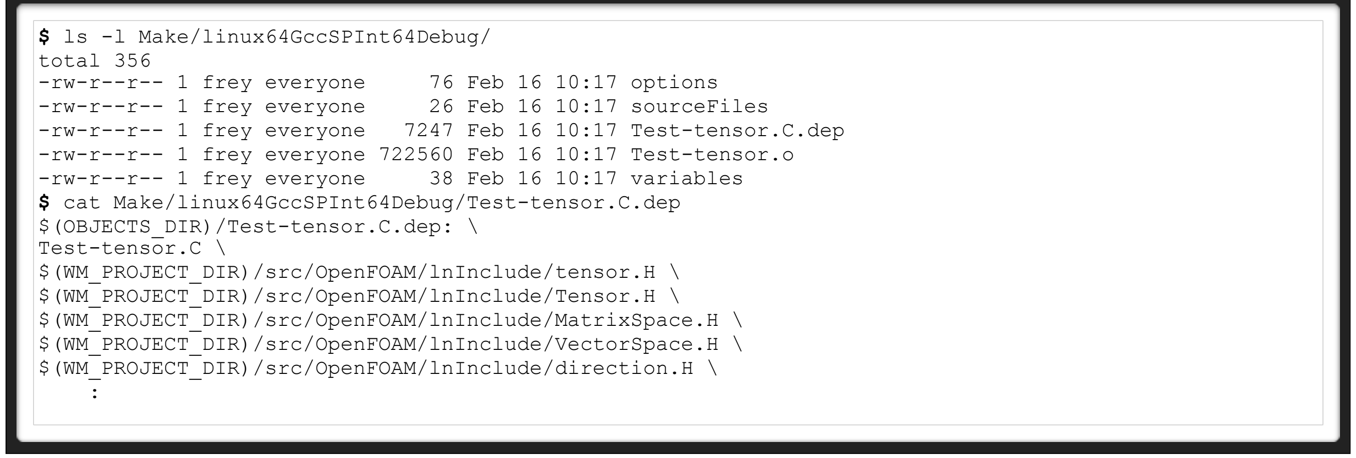


```
$ pwd
/home/1001/of-projects/tensor
$ wmake
Making dependency list for source file Test-tensor.C
g++ -std=c++0x -m64 -Dlinux64 -DWM_ARCH_OPTION=64 -DWM_SP -DWM_LABEL_SIZE=64 -Wall -Wextra -Wold-
style-cast -Wnon-virtual-dtor -Wno-unused-parameter -Wno-invalid-offsetof -O0 -fdefault-inline -ggdb3
-DFULLDEBUG -DNoRepository -ftemplate-depth-100  -IlnInclude -I. -I/home/software/openfoam/4.1/
OpenFOAM-4.1/src/OpenFOAM/lnInclude -I/home/software/openfoam/4.1/OpenFOAM-4.1/src/OSspecific/POSIX/
lnInclude   -fPIC -c Test-tensor.C -o Make/linux64GccSPInt64Debug/Test-tensor.o
g++ -std=c++0x -m64 -Dlinux64 -DWM_ARCH_OPTION=64 -DWM_SP -DWM_LABEL_SIZE=64 -Wall -Wextra -Wold-
style-cast -Wnon-virtual-dtor -Wno-unused-parameter -Wno-invalid-offsetof -O0 -fdefault-inline -ggdb3
-DFULLDEBUG -DNoRepository -ftemplate-depth-100  -IlnInclude -I. -I/home/software/openfoam/4.1/
OpenFOAM-4.1/src/OpenFOAM/lnInclude -I/home/software/openfoam/4.1/OpenFOAM-4.1/src/OSspecific/POSIX/
lnInclude   -fPIC -Xlinker --add-needed -Xlinker --no-as-needed Make/linux64GccSPInt64Debug/Test-
tensor.o -L/home/software/openfoam/4.1/OpenFOAM-4.1/platforms/linux64GccSPInt64Debug/lib \
         -lOpenFOAM -ldl  \
        -ggdb3 -DFULLDEBUG -lm -o /home/1001/of-projects/tensor/platforms/linux64GccSPInt64Debug/
bin/Test-tensor

$ ls -ld platforms/linux64GccSPInt64Debug/bin/Test-tensor
-rwxr-xr-x 1 frey everyone 423389 Feb 16 10:17 platforms/linux64GccSPInt64Debug/bin/Test-tensor
```

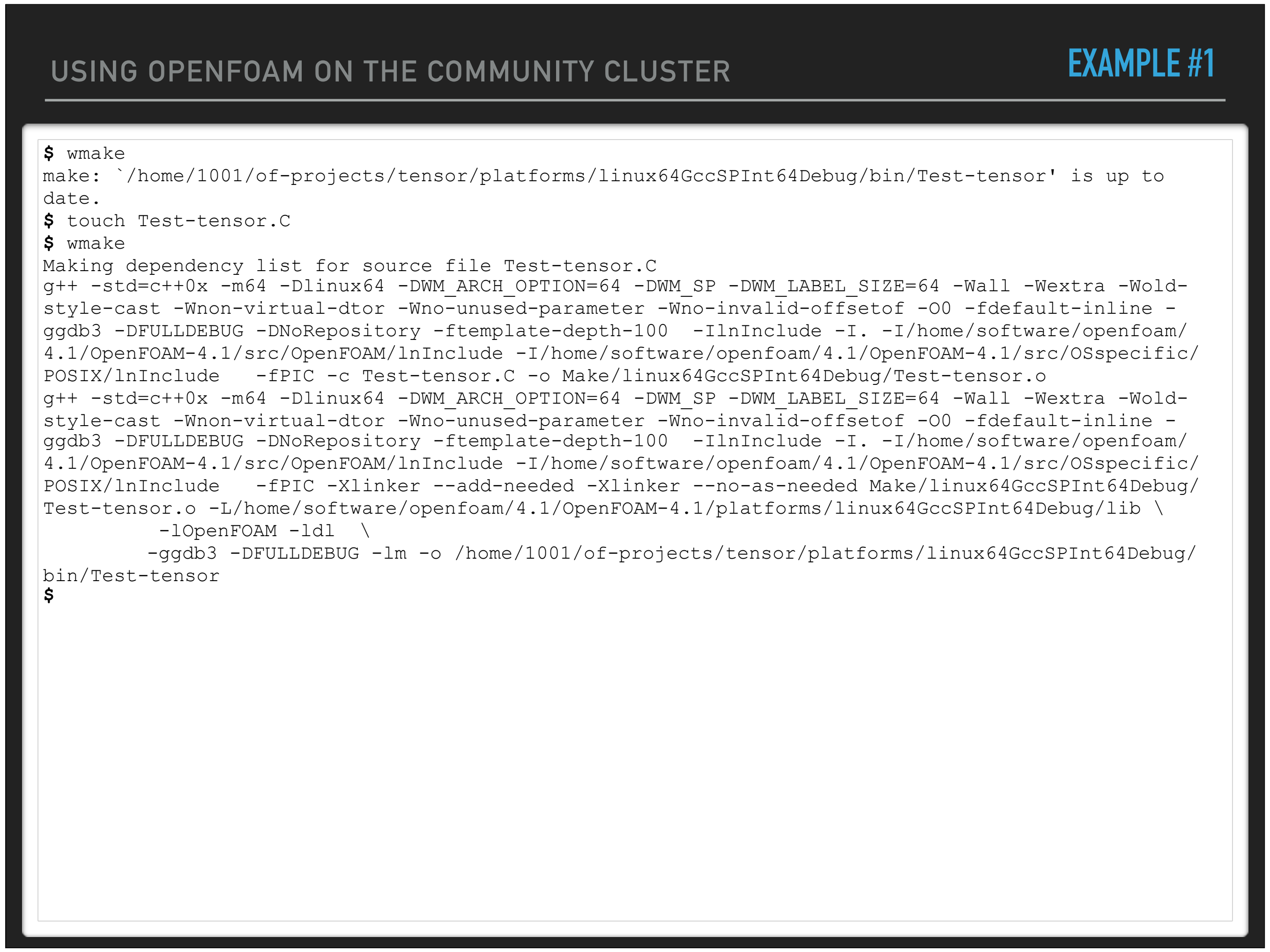
USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #1

- Let's build the program
  - All intermediate object files, etc., can be found in the "Make" directory
  - The .dep file indicates what components the source file "Test-tensor.C" *depends* on
    - If any of the files mentioned is modified, then a rebuild is required



```
$ ls -l Make/linux64GccSPInt64Debug/
total 356
-rw-r--r-- 1 frey everyone     76 Feb 16 10:17 options
-rw-r--r-- 1 frey everyone     26 Feb 16 10:17 sourceFiles
-rw-r--r-- 1 frey everyone   7247 Feb 16 10:17 Test-tensor.C.dep
-rw-r--r-- 1 frey everyone 722560 Feb 16 10:17 Test-tensor.o
-rw-r--r-- 1 frey everyone     38 Feb 16 10:17 variables
$ cat Make/linux64GccSPInt64Debug/Test-tensor.C.dep
$(OBJECTS_DIR)/Test-tensor.C.dep: \
Test-tensor.C \
$(WM_PROJECT_DIR)/src/OpenFOAM/lnInclude/tensor.H \
$(WM_PROJECT_DIR)/src/OpenFOAM/lnInclude/Tensor.H \
$(WM_PROJECT_DIR)/src/OpenFOAM/lnInclude/MatrixSpace.H \
$(WM_PROJECT_DIR)/src/OpenFOAM/lnInclude/VectorSpace.H \
$(WM_PROJECT_DIR)/src/OpenFOAM/lnInclude/direction.H \
    :
```



USING OPENFOAM ON THE COMMUNITY CLUSTER

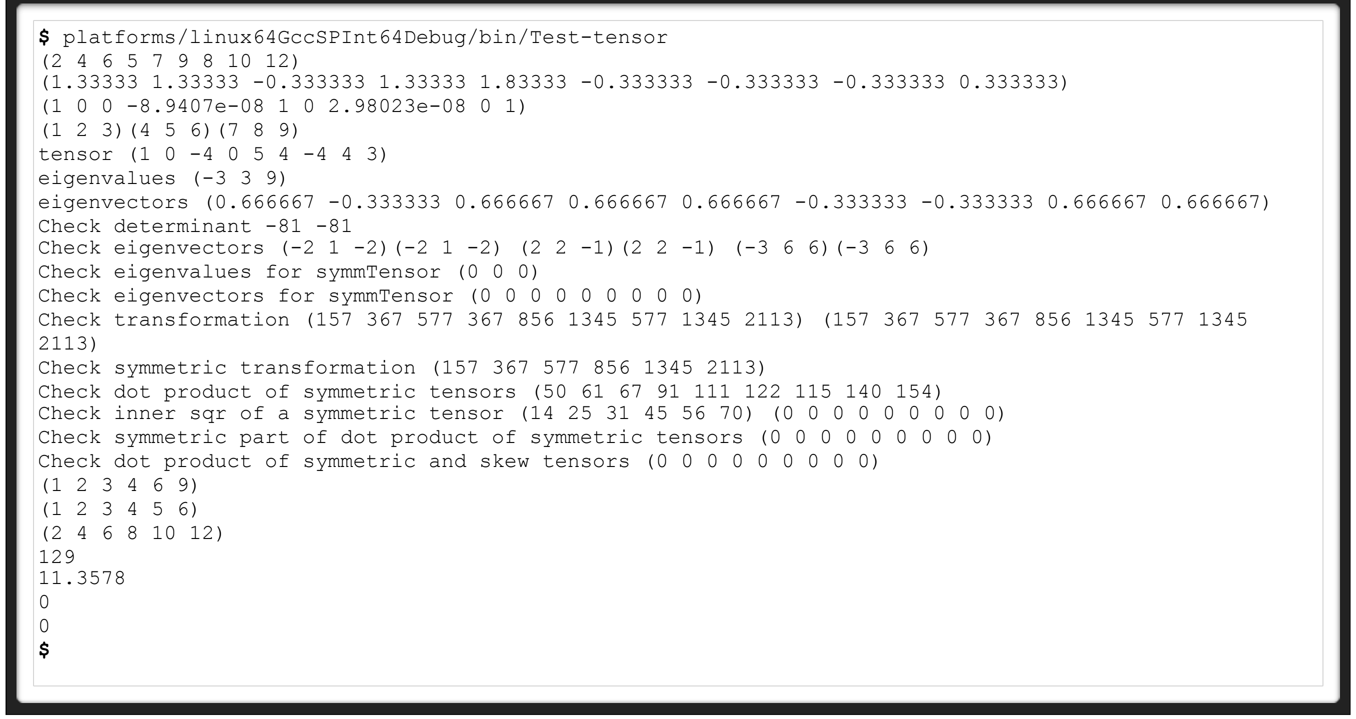
EXAMPLE #1

```
$ wmake
make: `/home/1001/of-projects/tensor/platforms/linux64GccSPInt64Debug/bin/Test-tensor' is up to
date.
$ touch Test-tensor.C
$ wmake
Making dependency list for source file Test-tensor.C
g++ -std=c++0x -m64 -Dlinux64 -DWM_ARCH_OPTION=64 -DWM_SP -DWM_LABEL_SIZE=64 -Wall -Wextra -Wold-
style-cast -Wnon-virtual-dtor -Wno-unused-parameter -Wno-invalid-offsetof -O0 -fdefault-inline -
ggdb3 -DFULLDEBUG -DNoRepository -ftemplate-depth-100  -IlnInclude -I. -I/home/software/openfoam/
4.1/OpenFOAM-4.1/src/OpenFOAM/lnInclude -I/home/software/openfoam/4.1/OpenFOAM-4.1/src/OSspecific/
POSIX/lnInclude   -fPIC -c Test-tensor.C -o Make/linux64GccSPInt64Debug/Test-tensor.o
g++ -std=c++0x -m64 -Dlinux64 -DWM_ARCH_OPTION=64 -DWM_SP -DWM_LABEL_SIZE=64 -Wall -Wextra -Wold-
style-cast -Wnon-virtual-dtor -Wno-unused-parameter -Wno-invalid-offsetof -O0 -fdefault-inline -
ggdb3 -DFULLDEBUG -DNoRepository -ftemplate-depth-100  -IlnInclude -I. -I/home/software/openfoam/
4.1/OpenFOAM-4.1/src/OpenFOAM/lnInclude -I/home/software/openfoam/4.1/OpenFOAM-4.1/src/OSspecific/
POSIX/lnInclude   -fPIC -Xlinker --add-needed -Xlinker --no-as-needed Make/linux64GccSPInt64Debug/
Test-tensor.o -L/home/software/openfoam/4.1/OpenFOAM-4.1/platforms/linux64GccSPInt64Debug/lib \
          -lOpenFOAM -ldl  \
         -ggdb3 -DFULLDEBUG -lm -o /home/1001/of-projects/tensor/platforms/linux64GccSPInt64Debug/
bin/Test-tensor
$
```

- First "wmake" doesn't do anything — none of the dependencies have been modified
- touch'ing the source file updates its modification timestamp (it's been modified!!)
- Subsequent "wmake" rebuilds the "Test-tensor.C" source file, relinks the executable

EXAMPLE #1

- Now run the program

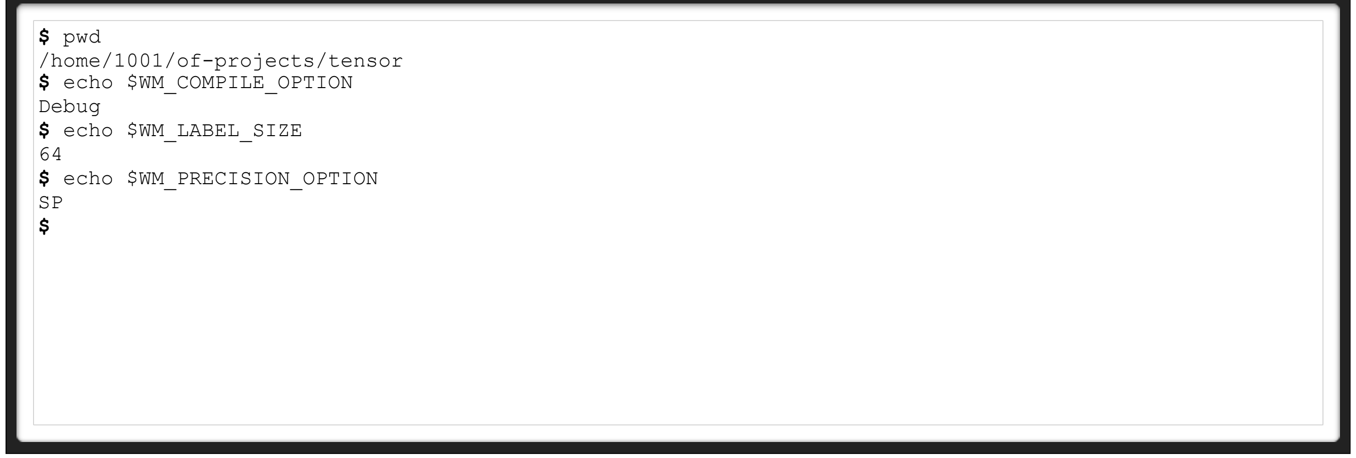


```
$ platforms/linux64GccSPInt64Debug/bin/Test-tensor
(2 4 6 5 7 9 8 10 12)
(1.33333 1.33333 -0.333333 1.33333 1.83333 -0.333333 -0.333333 -0.333333 0.333333)
(1 0 0 -8.9407e-08 1 0 2.98023e-08 0 1)
(1 2 3)(4 5 6)(7 8 9)
tensor (1 0 -4 0 5 4 -4 4 3)
eigenvalues (-3 3 9)
eigenvectors (0.666667 -0.333333 0.666667 0.666667 0.666667 -0.333333 -0.333333 0.666667 0.666667)
Check determinant -81 -81
Check eigenvectors (-2 1 -2)(-2 1 -2) (2 2 -1)(2 2 -1) (-3 6 6)(-3 6 6)
Check eigenvalues for symmTensor (0 0 0)
Check eigenvectors for symmTensor (0 0 0 0 0 0 0 0 0)
Check transformation (157 367 577 367 856 1345 577 1345 2113) (157 367 577 367 856 1345 577 1345
2113)
Check symmetric transformation (157 367 577 856 1345 2113)
Check dot product of symmetric tensors (50 61 67 91 111 122 115 140 154)
Check inner sqr of a symmetric tensor (14 25 31 45 56 70) (0 0 0 0 0 0 0 0 0)
Check symmetric part of dot product of symmetric tensors (0 0 0 0 0 0 0 0 0)
Check dot product of symmetric and skew tensors (0 0 0 0 0 0 0 0 0)
(1 2 3 4 6 9)
(1 2 3 4 5 6)
(2 4 6 8 10 12)
129
11.3578
0
0
$
```

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #2

- Without exiting your shell or logging out!
- Alter the platform parameters and rebuild the program
  - Change each parameter: DP, Opt, 32

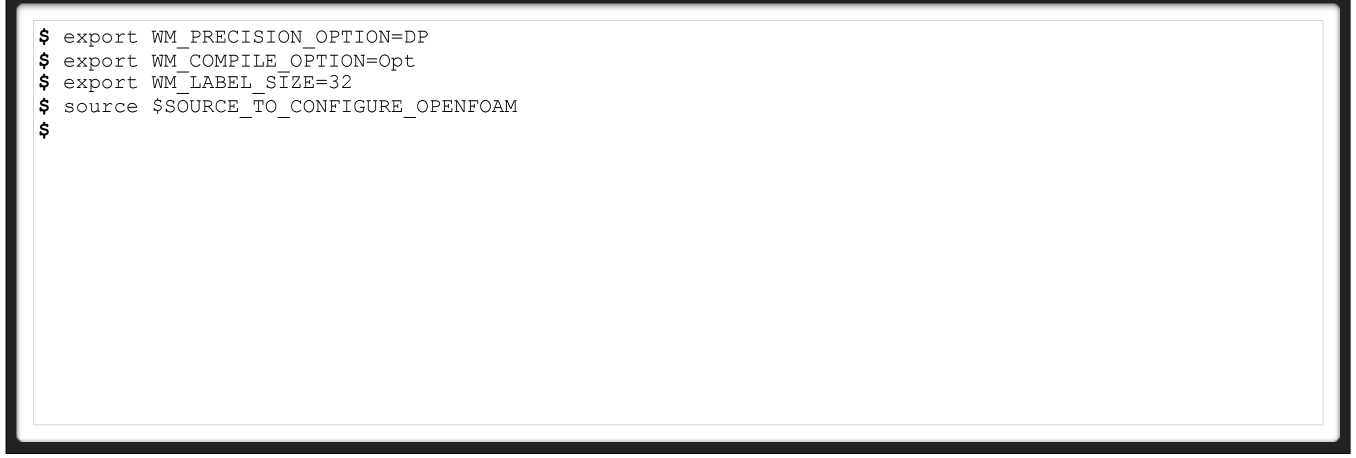


```
$ pwd
/home/1001/of-projects/tensor
$ echo $WM_COMPILE_OPTION
Debug
$ echo $WM_LABEL_SIZE
64
$ echo $WM_PRECISION_OPTION
SP
$
```

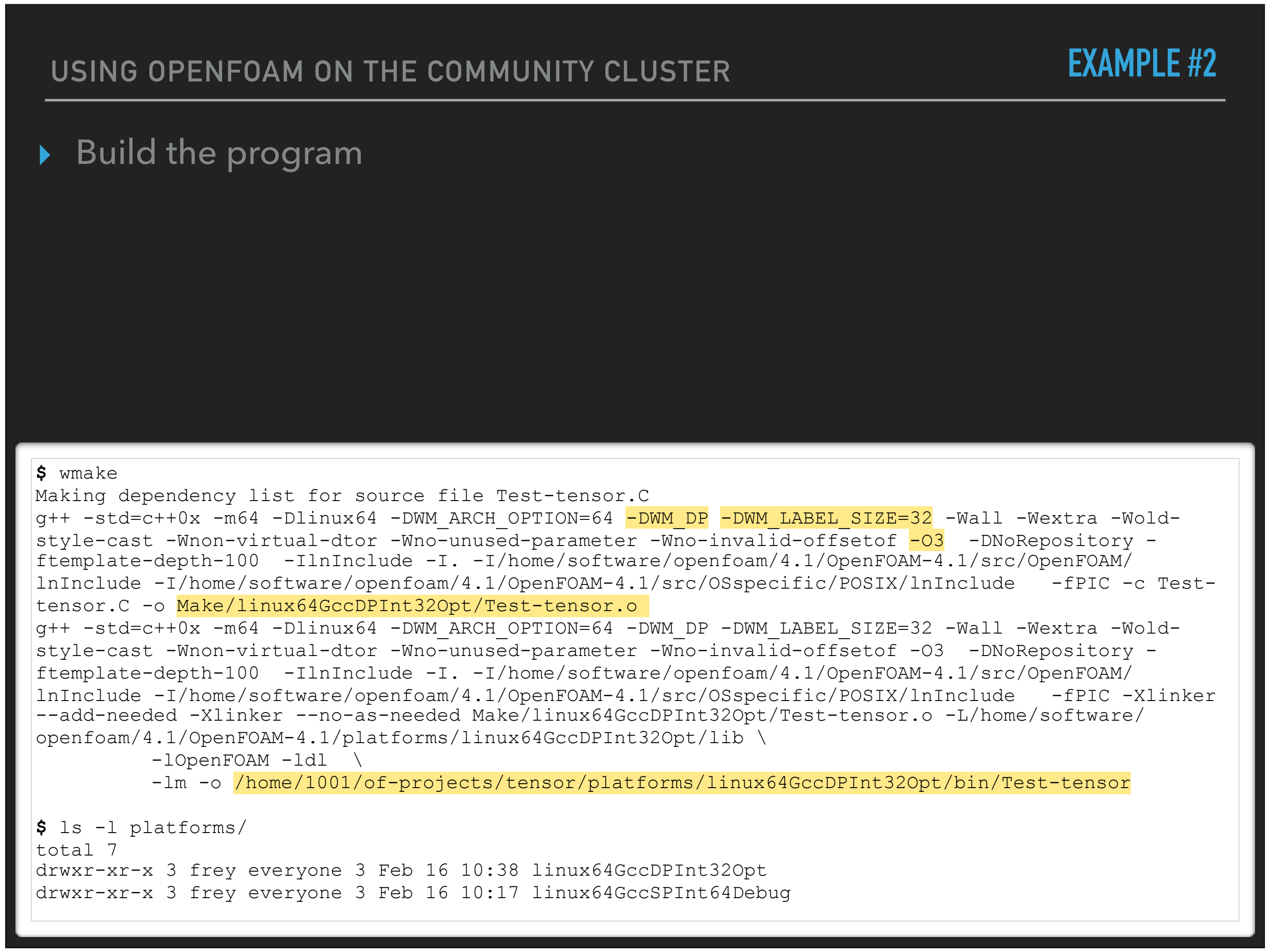
USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #2

- Without exiting your shell or logging out!
- Alter the platform parameters and rebuild the program
    - Change each parameter: DP, Opt, 32
    - Then, source the environment setup script again



```
$ export WM_PRECISION_OPTION=DP
$ export WM_COMPILE_OPTION=Opt
$ export WM_LABEL_SIZE=32
$ source $SOURCE_TO_CONFIGURE_OPENFOAM
$
```

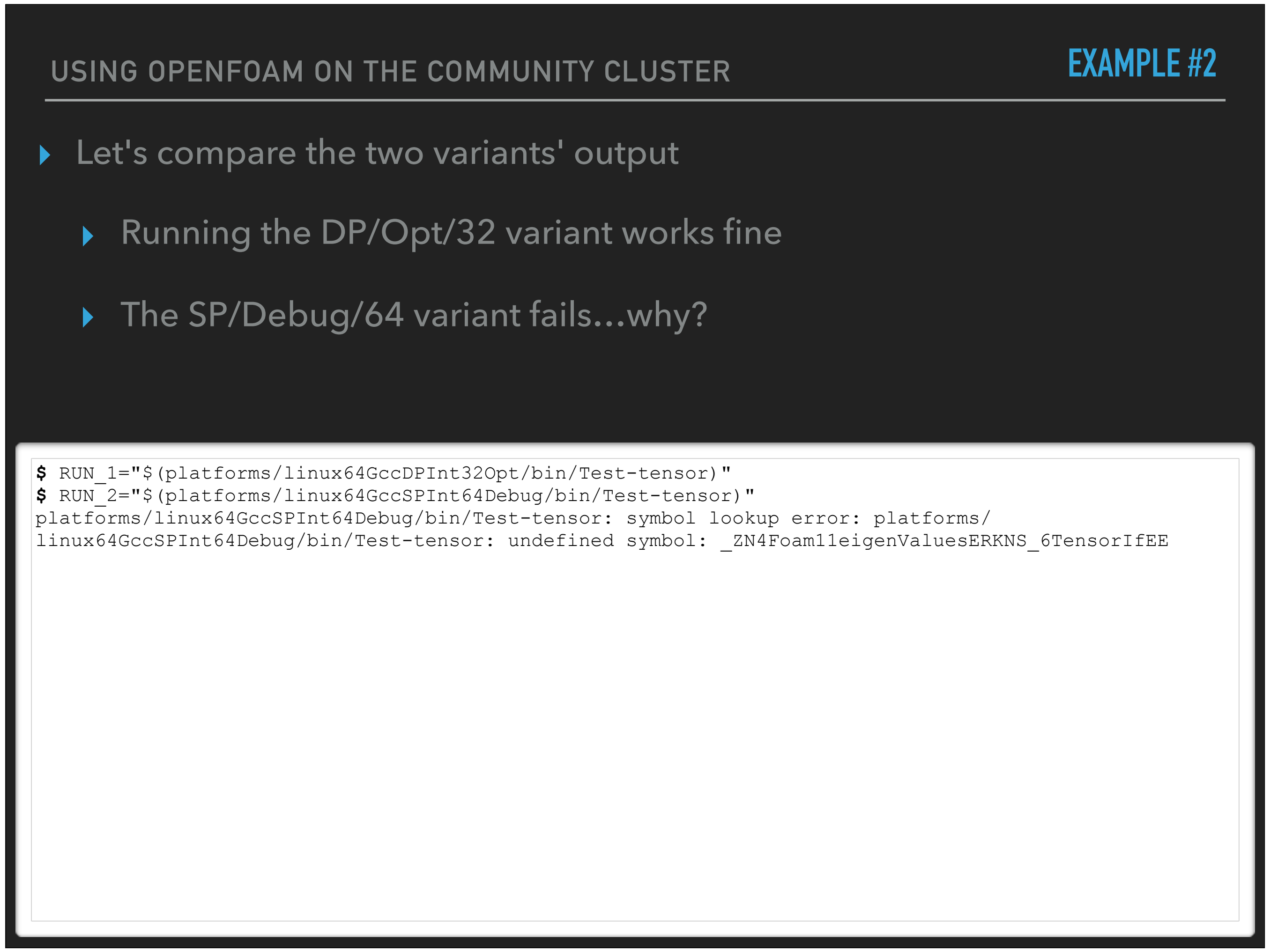


EXAMPLE #2

- Build the program

```
$ wmake
Making dependency list for source file Test-tensor.C
g++ -std=c++0x -m64 -Dlinux64 -DWM_ARCH_OPTION=64 -DWM_DP -DWM_LABEL_SIZE=32 -Wall -Wextra -Wold-
style-cast -Wnon-virtual-dtor -Wno-unused-parameter -Wno-invalid-offsetof -O3  -DNoRepository -
ftemplate-depth-100  -IlnInclude -I. -I/home/software/openfoam/4.1/OpenFOAM-4.1/src/OpenFOAM/
lnInclude -I/home/software/openfoam/4.1/OpenFOAM-4.1/src/OSspecific/POSIX/lnInclude   -fPIC -c Test-
tensor.C -o Make/linux64GccDPInt32Opt/Test-tensor.o
g++ -std=c++0x -m64 -Dlinux64 -DWM_ARCH_OPTION=64 -DWM_DP -DWM_LABEL_SIZE=32 -Wall -Wextra -Wold-
style-cast -Wnon-virtual-dtor -Wno-unused-parameter -Wno-invalid-offsetof -O3  -DNoRepository -
ftemplate-depth-100  -IlnInclude -I. -I/home/software/openfoam/4.1/OpenFOAM-4.1/src/OpenFOAM/
lnInclude -I/home/software/openfoam/4.1/OpenFOAM-4.1/src/OSspecific/POSIX/lnInclude   -fPIC -Xlinker
--add-needed -Xlinker --no-as-needed Make/linux64GccDPInt32Opt/Test-tensor.o -L/home/software/
openfoam/4.1/OpenFOAM-4.1/platforms/linux64GccDPInt32Opt/lib \
          -lOpenFOAM -ldl  \
          -lm -o /home/1001/of-projects/tensor/platforms/linux64GccDPInt32Opt/bin/Test-tensor

$ ls -l platforms/
total 7
drwxr-xr-x 3 frey everyone 3 Feb 16 10:38 linux64GccDPInt32Opt
drwxr-xr-x 3 frey everyone 3 Feb 16 10:17 linux64GccSPInt64Debug
```

- Note the new platform, no debug flags, level-3 optimization, precision and label size change



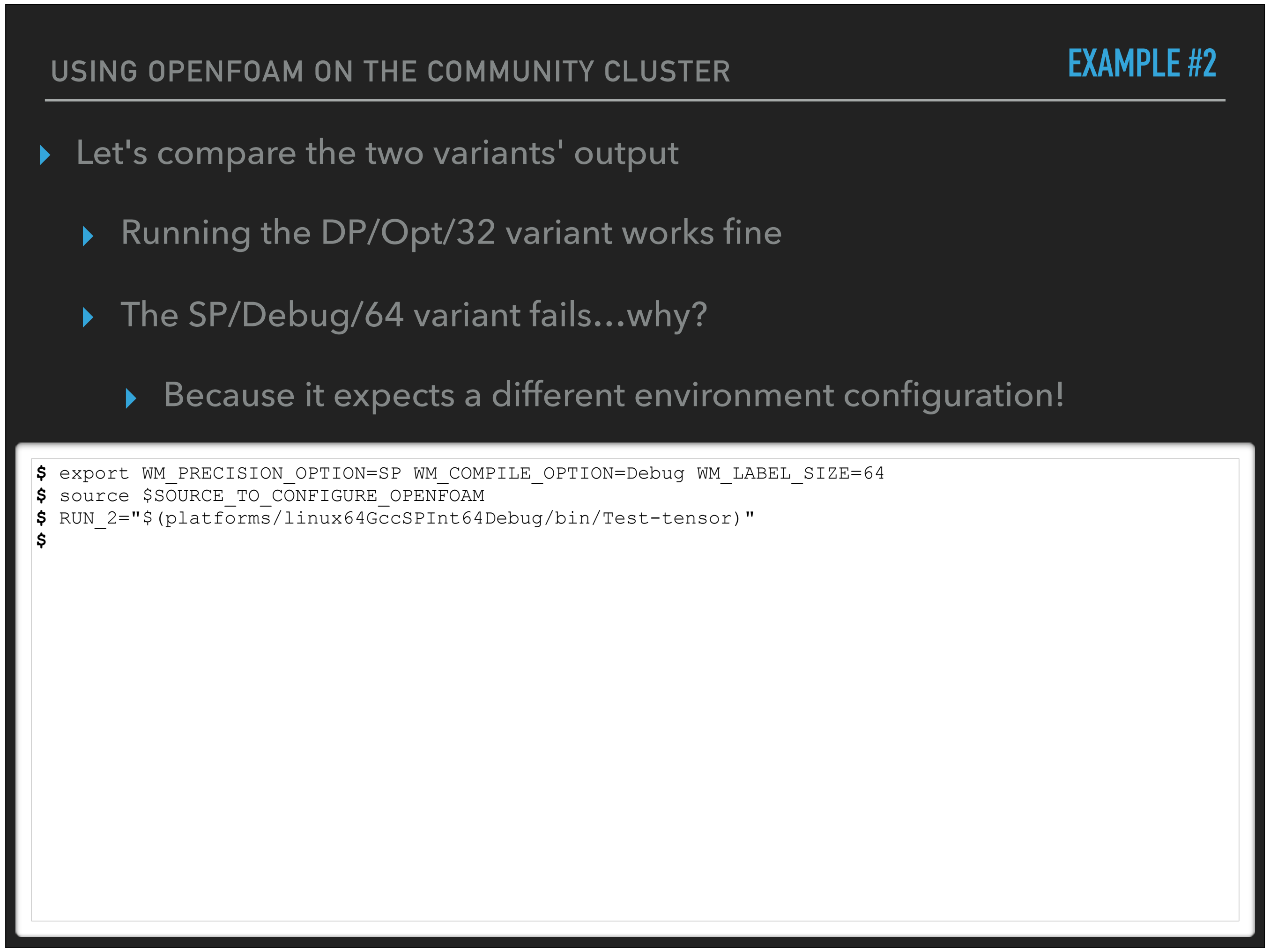
USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #2

- Let's compare the two variants' output
  - Running the DP/Opt/32 variant works fine
  - The SP/Debug/64 variant fails…why?

```
$ RUN_1="$(platforms/linux64GccDPInt32Opt/bin/Test-tensor)"
$ RUN_2="$(platforms/linux64GccSPInt64Debug/bin/Test-tensor)"
platforms/linux64GccSPInt64Debug/bin/Test-tensor: symbol lookup error: platforms/
linux64GccSPInt64Debug/bin/Test-tensor: undefined symbol: _ZN4Foam11eigenValuesERKNS_6TensorIfEE
```

- Note the new platform, no debug flags, level-3 optimization, precision and label size change



USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #2

- Let's compare the two variants' output
  - Running the DP/Opt/32 variant works fine
  - The SP/Debug/64 variant fails…why?
    - Because it expects a different environment configuration!

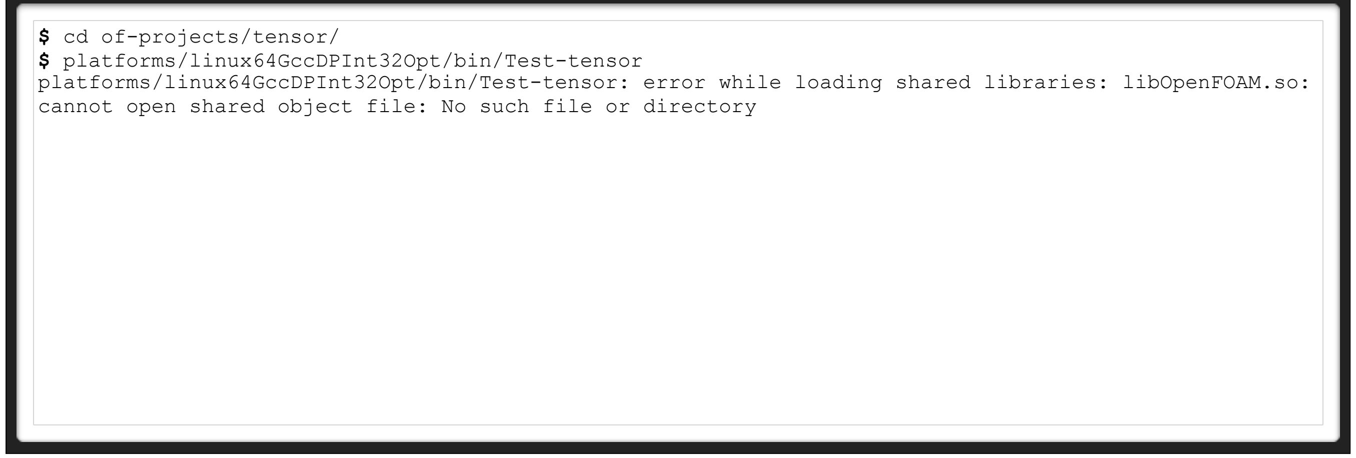
```
$ export WM_PRECISION_OPTION=SP WM_COMPILE_OPTION=Debug WM_LABEL_SIZE=64
$ source $SOURCE_TO_CONFIGURE_OPENFOAM
$ RUN_2="$(platforms/linux64GccSPInt64Debug/bin/Test-tensor)"
$
```

- Note the new platform, no debug flags, level-3 optimization, precision and label size change

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #3

- Let's logout and log back into Farber and try running the program again
  - What's wrong?

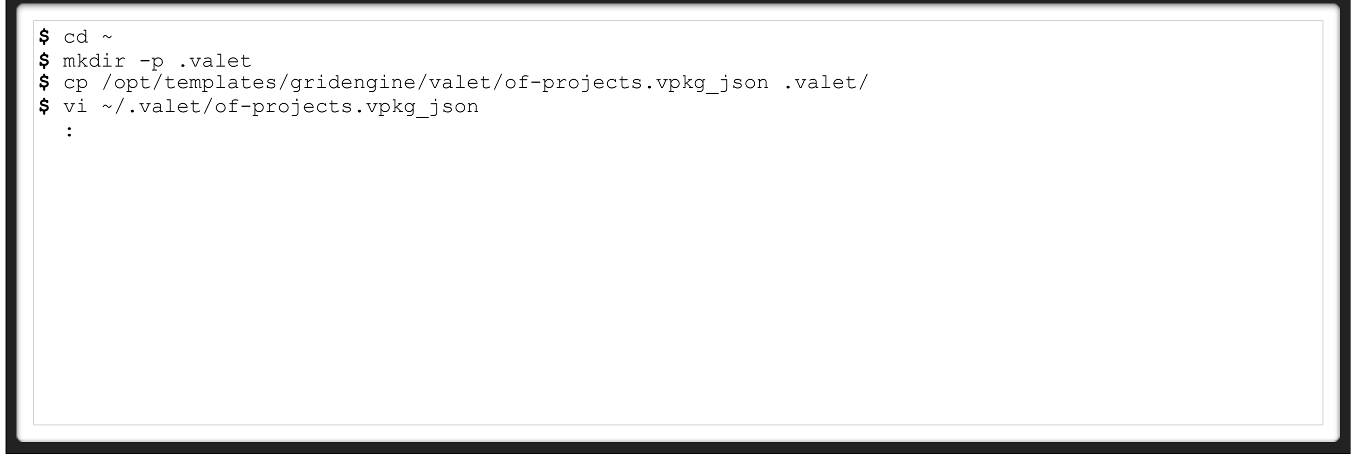


```
$ cd of-projects/tensor/
$ platforms/linux64GccDPInt32Opt/bin/Test-tensor
platforms/linux64GccDPInt32Opt/bin/Test-tensor: error while loading shared libraries: libOpenFOAM.so:
cannot open shared object file: No such file or directory
```

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #3

- Let's logout and log back into Farber and try running the program again
- Create a VALET package for the project
  - Major project with multiple versions? Create as its own VALET package
  - For simple projects, bundle them as versions of "of-projects"

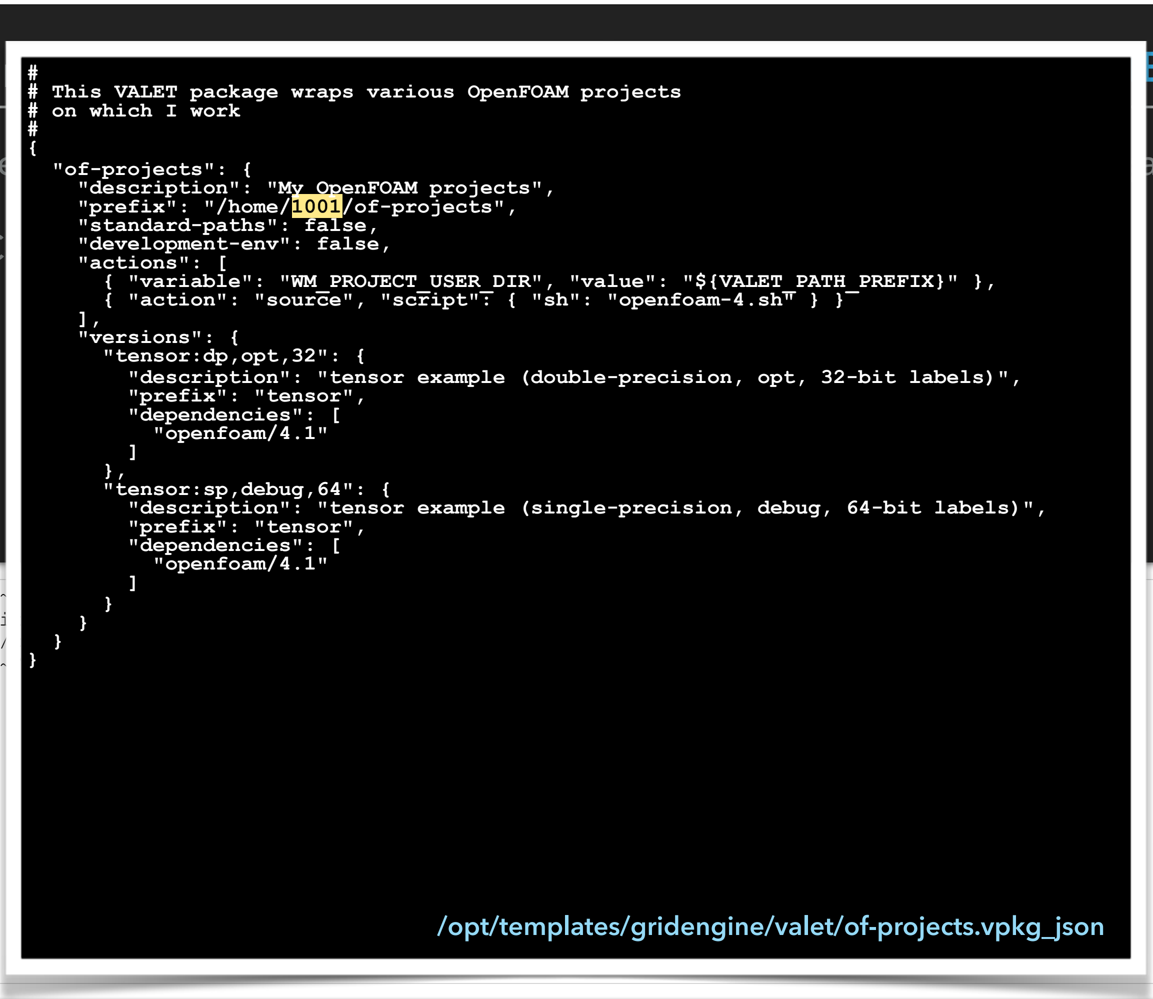


```
$ cd ~
$ mkdir -p .valet
$ cp /opt/templates/gridengine/valet/of-projects.vpkg_json .valet/
$ vi ~/.valet/of-projects.vpkg_json
   :
```

USI E #3

- Le ain
- C

```
$ cd ~
$ mkdi
$ cp /
$ vi ~
  :
```



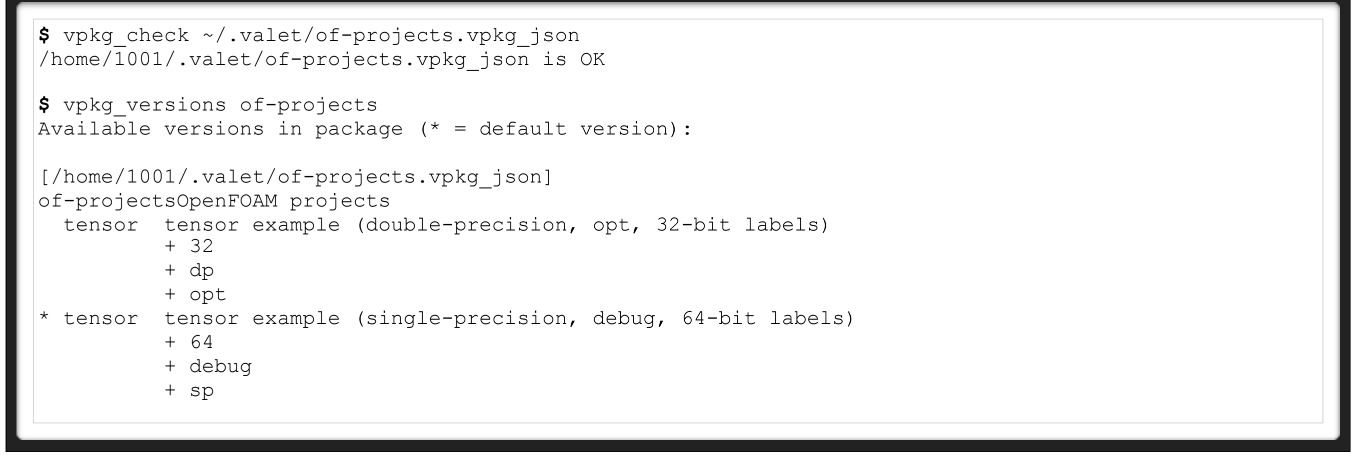
```
#
# This VALET package wraps various OpenFOAM projects
# on which I work
#
{
  "of-projects": {
    "description": "My OpenFOAM projects",
    "prefix": "/home/1001/of-projects",
    "standard-paths": false,
    "development-env": false,
    "actions": [
      { "variable": "WM_PROJECT_USER_DIR", "value": "${VALET_PATH_PREFIX}" },
      { "action": "source", "script": { "sh": "openfoam-4.sh" } }
    ],
    "versions": {
      "tensor:dp,opt,32": {
        "description": "tensor example (double-precision, opt, 32-bit labels)",
        "prefix": "tensor",
        "dependencies": [
          "openfoam/4.1"
        ]
      },
      "tensor:sp,debug,64": {
        "description": "tensor example (single-precision, debug, 64-bit labels)",
        "prefix": "tensor",
        "dependencies": [
          "openfoam/4.1"
        ]
      }
    }
  }
}
```

/opt/templates/gridengine/valet/of-projects.vpkg_json

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #3

- Let's logout and log back into Farber and try running the program again
- Create a VALET package for the project
  - Check the syntax of the package definition file
  - If OK, then list the versions (projects) defined therein

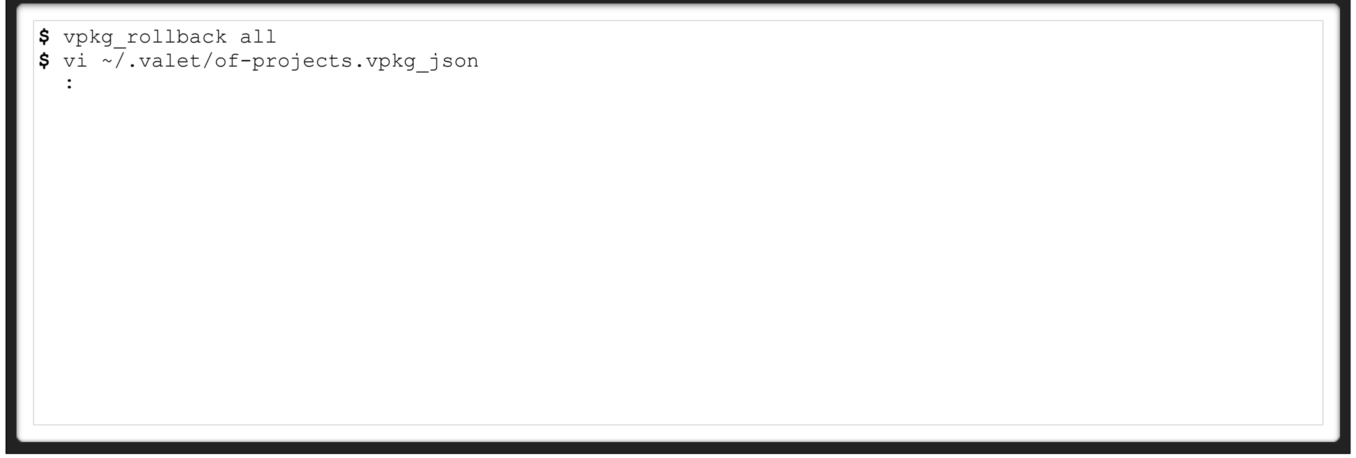


```
$ vpkg_check ~/.valet/of-projects.vpkg_json
/home/1001/.valet/of-projects.vpkg_json is OK

$ vpkg_versions of-projects
Available versions in package (* = default version):

[/home/1001/.valet/of-projects.vpkg_json]
of-projectsOpenFOAM projects
  tensor  tensor example (double-precision, opt, 32-bit labels)
          + 32
          + dp
          + opt
* tensor  tensor example (single-precision, debug, 64-bit labels)
          + 64
          + debug
          + sp
```

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #3

- What versions are available?
- We can use VALET to help us build other variants, too!
  - Add a SP/Opt/64 variant



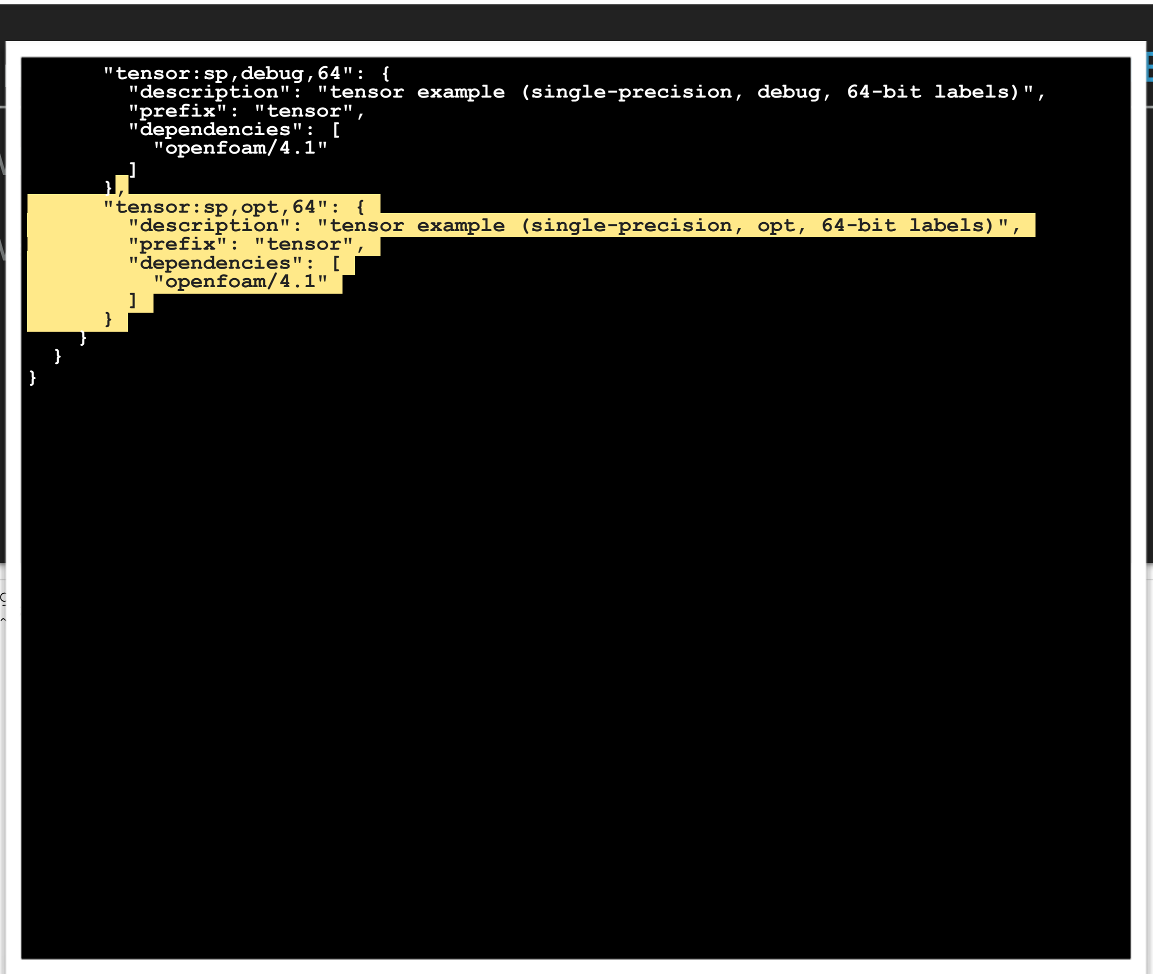
```
$ vpkg_rollback all
$ vi ~/.valet/of-projects.vpkg_json
  :
```

USI

E #3

- W
- W
  -

```
$ vpkg
$ vi ~
  :
```

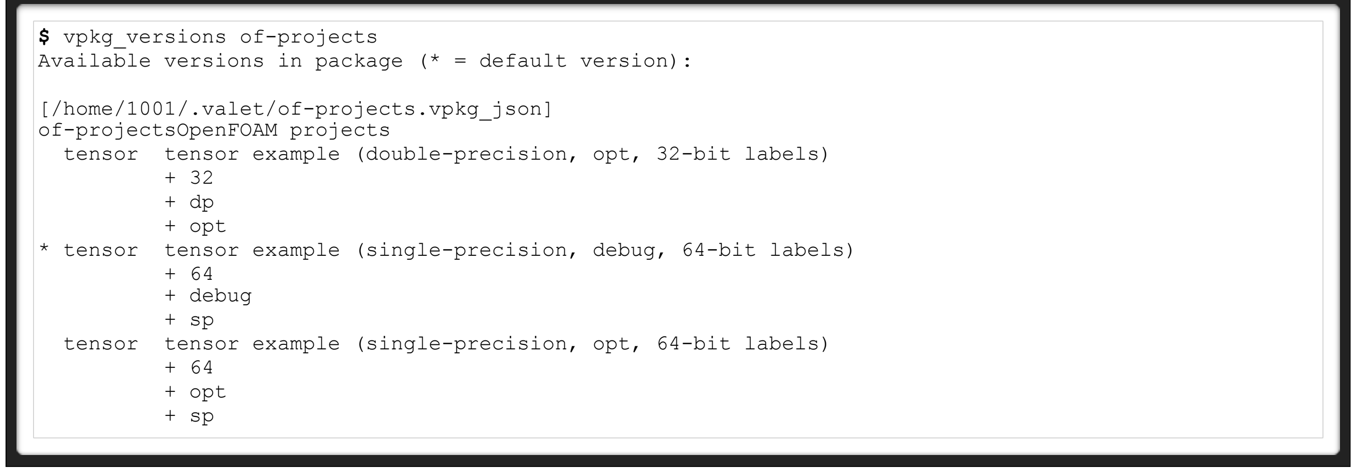


```
      "tensor:sp,debug,64": {
        "description": "tensor example (single-precision, debug, 64-bit labels)",
        "prefix": "tensor",
        "dependencies": [
          "openfoam/4.1"
        ]
      },
      "tensor:sp,opt,64": {
        "description": "tensor example (single-precision, opt, 64-bit labels)",
        "prefix": "tensor",
        "dependencies": [
          "openfoam/4.1"
        ]
      }
    }
  }
}
```

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #3

- What versions are available?
- We can use VALET to help us build other variants, too!
  - Assuming syntax is OK, list versions:

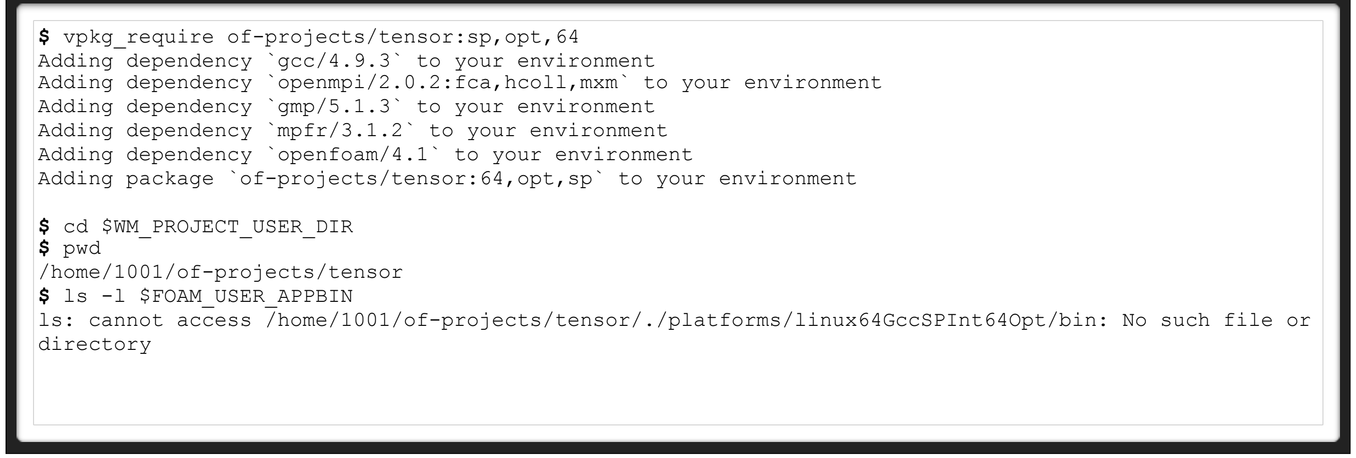


```
$ vpkg_versions of-projects
Available versions in package (* = default version):

[/home/1001/.valet/of-projects.vpkg_json]
of-projectsOpenFOAM projects
  tensor  tensor example (double-precision, opt, 32-bit labels)
          + 32
          + dp
          + opt
* tensor  tensor example (single-precision, debug, 64-bit labels)
          + 64
          + debug
          + sp
  tensor  tensor example (single-precision, opt, 64-bit labels)
          + 64
          + opt
          + sp
```

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #3

- What versions are available?
- We can use VALET to help us build other variants, too!
  - Add a SP/Opt/64 variant
  - Doing the vpkg_require sets up the environment – regardless of whether or not you've built the project already

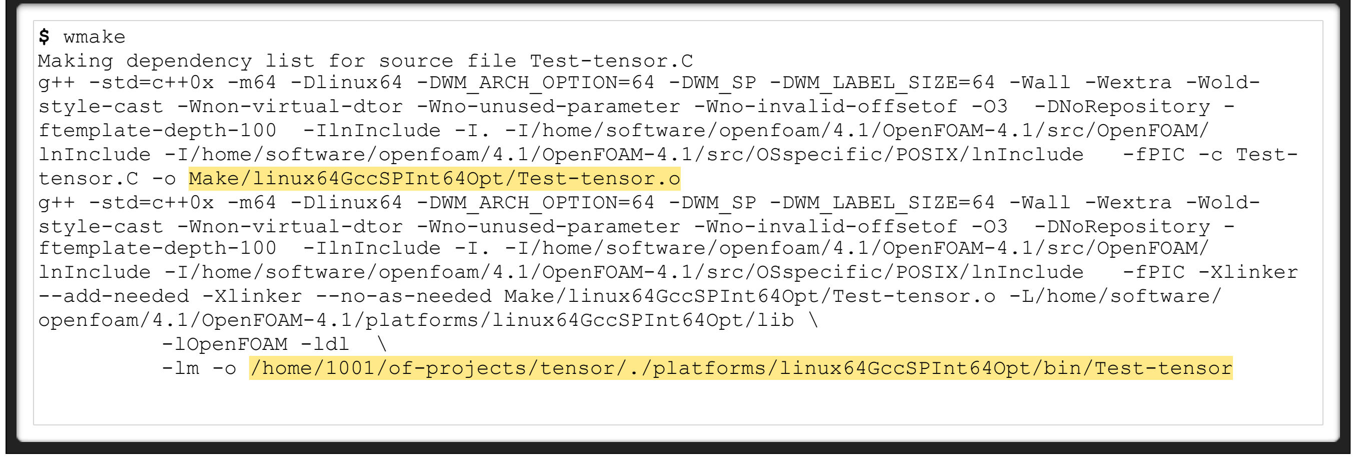


```
$ vpkg_require of-projects/tensor:sp,opt,64
Adding dependency `gcc/4.9.3` to your environment
Adding dependency `openmpi/2.0.2:fca,hcoll,mxm` to your environment
Adding dependency `gmp/5.1.3` to your environment
Adding dependency `mpfr/3.1.2` to your environment
Adding dependency `openfoam/4.1` to your environment
Adding package `of-projects/tensor:64,opt,sp` to your environment

$ cd $WM_PROJECT_USER_DIR
$ pwd
/home/1001/of-projects/tensor
$ ls -l $FOAM_USER_APPBIN
ls: cannot access /home/1001/of-projects/tensor/./platforms/linux64GccSPInt64Opt/bin: No such file or
directory
```

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #3

- What versions are available?
- We can use VALET to help us build other variants, too!
  - Add a SP/Opt/64 variant
  - Doing the vpkg_require sets up the environment – regardless of whether or not you've built the project or not

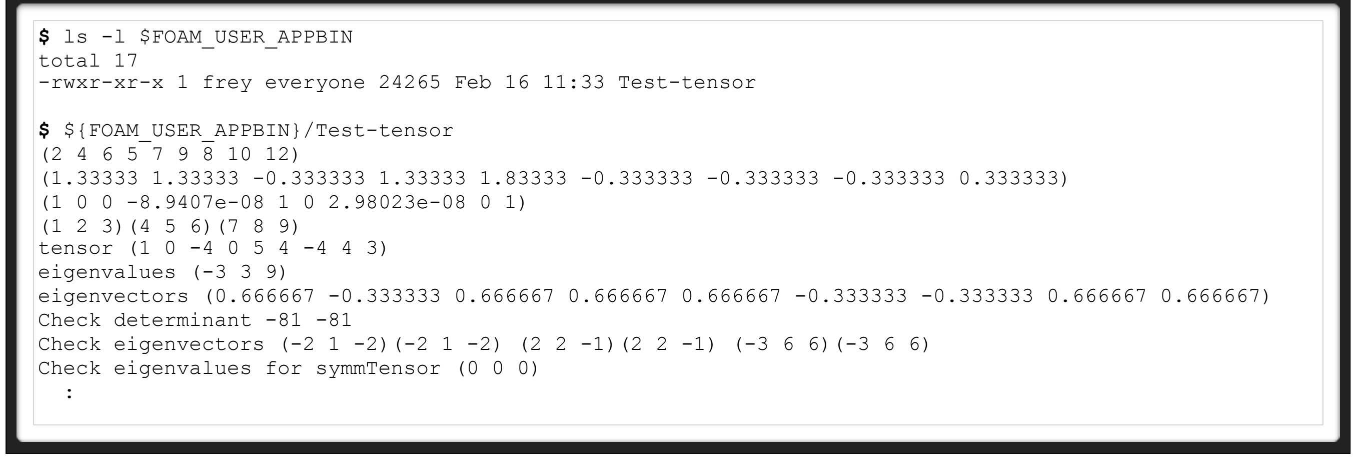


```
$ wmake
Making dependency list for source file Test-tensor.C
g++ -std=c++0x -m64 -Dlinux64 -DWM_ARCH_OPTION=64 -DWM_SP -DWM_LABEL_SIZE=64 -Wall -Wextra -Wold-
style-cast -Wnon-virtual-dtor -Wno-unused-parameter -Wno-invalid-offsetof -O3  -DNoRepository -
ftemplate-depth-100  -IlnInclude -I. -I/home/software/openfoam/4.1/OpenFOAM-4.1/src/OpenFOAM/
lnInclude -I/home/software/openfoam/4.1/OpenFOAM-4.1/src/OSspecific/POSIX/lnInclude   -fPIC -c Test-
tensor.C -o Make/linux64GccSPInt64Opt/Test-tensor.o
g++ -std=c++0x -m64 -Dlinux64 -DWM_ARCH_OPTION=64 -DWM_SP -DWM_LABEL_SIZE=64 -Wall -Wextra -Wold-
style-cast -Wnon-virtual-dtor -Wno-unused-parameter -Wno-invalid-offsetof -O3  -DNoRepository -
ftemplate-depth-100  -IlnInclude -I. -I/home/software/openfoam/4.1/OpenFOAM-4.1/src/OpenFOAM/
lnInclude -I/home/software/openfoam/4.1/OpenFOAM-4.1/src/OSspecific/POSIX/lnInclude   -fPIC -Xlinker
--add-needed -Xlinker --no-as-needed Make/linux64GccSPInt64Opt/Test-tensor.o -L/home/software/
openfoam/4.1/OpenFOAM-4.1/platforms/linux64GccSPInt64Opt/lib \
          -lOpenFOAM -ldl  \
          -lm -o /home/1001/of-projects/tensor/./platforms/linux64GccSPInt64Opt/bin/Test-tensor
```

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #3

- What versions are available?
- We can use VALET to help us build other variants, too!
- Now we can run that variant, too

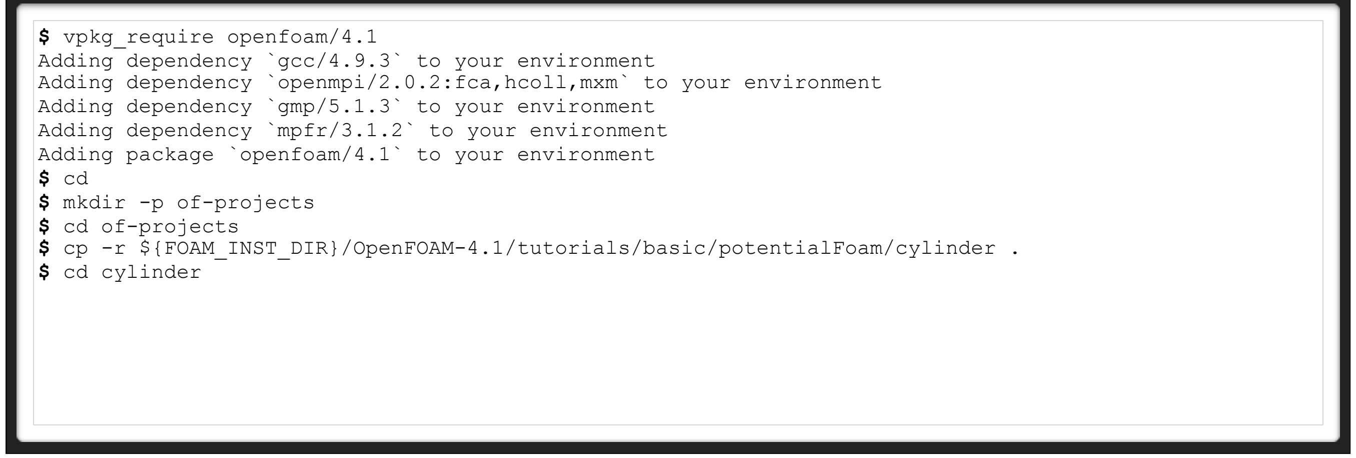


```
$ ls -l $FOAM_USER_APPBIN
total 17
-rwxr-xr-x 1 frey everyone 24265 Feb 16 11:33 Test-tensor

$ ${FOAM_USER_APPBIN}/Test-tensor
(2 4 6 5 7 9 8 10 12)
(1.33333 1.33333 -0.333333 1.33333 1.83333 -0.333333 -0.333333 -0.333333 0.333333)
(1 0 0 -8.9407e-08 1 0 2.98023e-08 0 1)
(1 2 3)(4 5 6)(7 8 9)
tensor (1 0 -4 0 5 4 -4 4 3)
eigenvalues (-3 3 9)
eigenvectors (0.666667 -0.333333 0.666667 0.666667 0.666667 -0.333333 -0.333333 0.666667 0.666667)
Check determinant -81 -81
Check eigenvectors (-2 1 -2)(-2 1 -2) (2 2 -1)(2 2 -1) (-3 6 6)(-3 6 6)
Check eigenvalues for symmTensor (0 0 0)
  :
```

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #4

- An example computation task making use of standard OpenFOAM tools
  - No software development involved

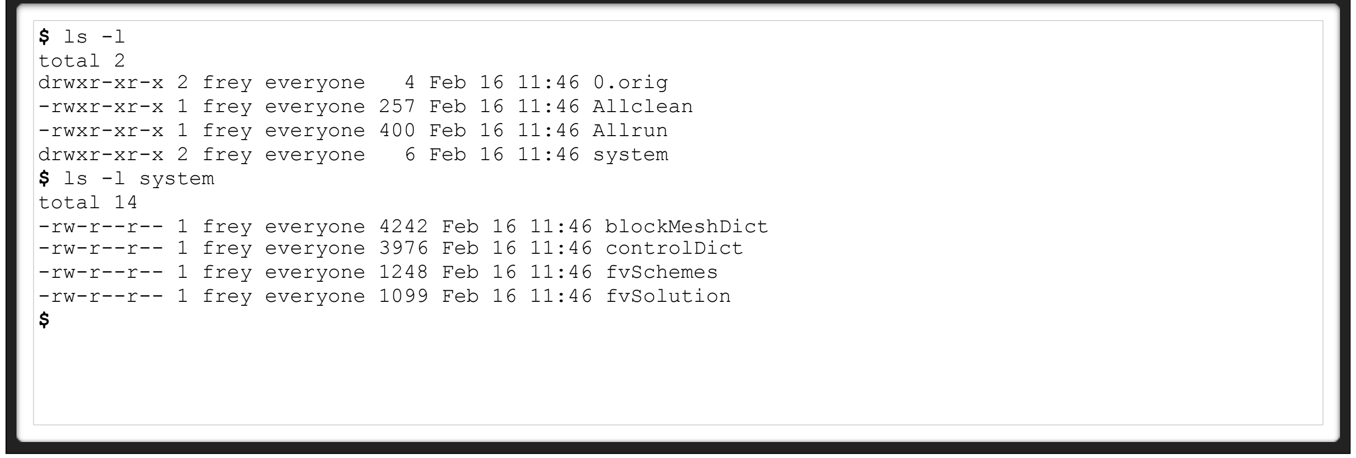


```
$ vpkg_require openfoam/4.1
Adding dependency `gcc/4.9.3` to your environment
Adding dependency `openmpi/2.0.2:fca,hcoll,mxm` to your environment
Adding dependency `gmp/5.1.3` to your environment
Adding dependency `mpfr/3.1.2` to your environment
Adding package `openfoam/4.1` to your environment
$ cd
$ mkdir -p of-projects
$ cd of-projects
$ cp -r ${FOAM_INST_DIR}/OpenFOAM-4.1/tutorials/basic/potentialFoam/cylinder .
$ cd cylinder
```

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #4

- An example computation task making use of standard OpenFOAM tools
  - The "0.orig" directory has the input files
  - The "system" directory contains parameters for the computation
  - The "Allrun" script calls the tools in order

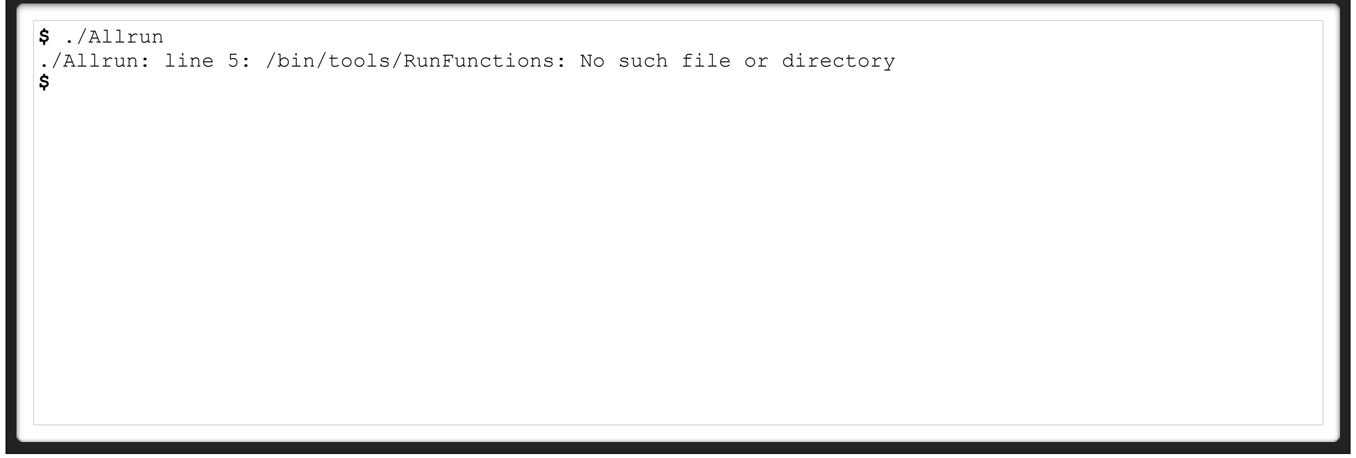


```
$ ls -l
total 2
drwxr-xr-x 2 frey everyone   4 Feb 16 11:46 0.orig
-rwxr-xr-x 1 frey everyone 257 Feb 16 11:46 Allclean
-rwxr-xr-x 1 frey everyone 400 Feb 16 11:46 Allrun
drwxr-xr-x 2 frey everyone   6 Feb 16 11:46 system
$ ls -l system
total 14
-rw-r--r-- 1 frey everyone 4242 Feb 16 11:46 blockMeshDict
-rw-r--r-- 1 frey everyone 3976 Feb 16 11:46 controlDict
-rw-r--r-- 1 frey everyone 1248 Feb 16 11:46 fvSchemes
-rw-r--r-- 1 frey everyone 1099 Feb 16 11:46 fvSolution
$
```

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #4

- An example computation task making use of standard OpenFOAM tools
- Let's try running it
  - Why doesn't it work?

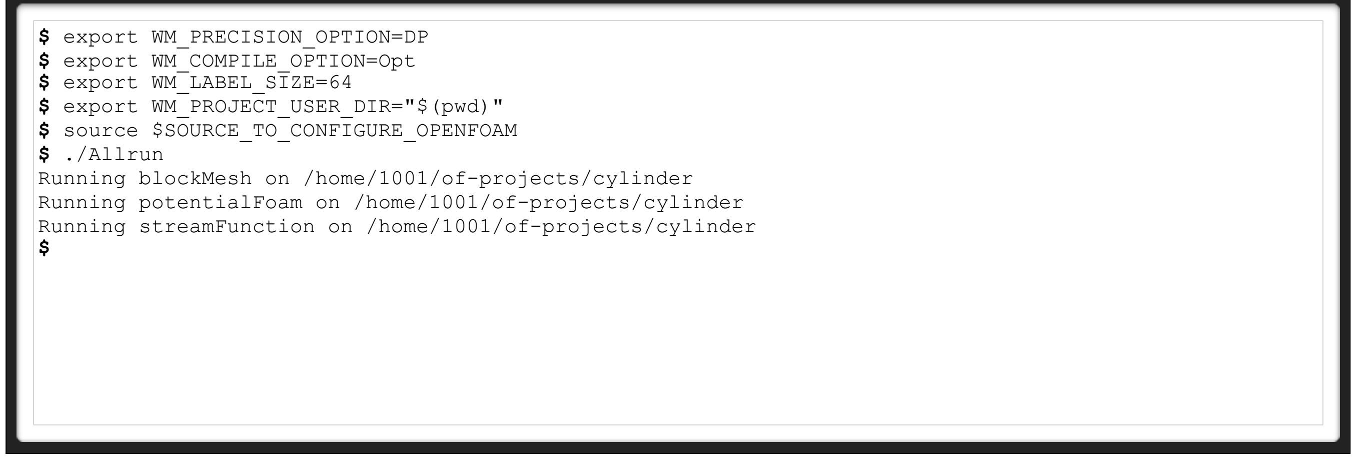


```
$ ./Allrun
./Allrun: line 5: /bin/tools/RunFunctions: No such file or directory
$
```

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #4

- An example computation task making use of standard OpenFOAM tools
- Let's try running it
  - Always have to finish the environment setup
  - Input files copied to "0" directory (output objects will be written there)

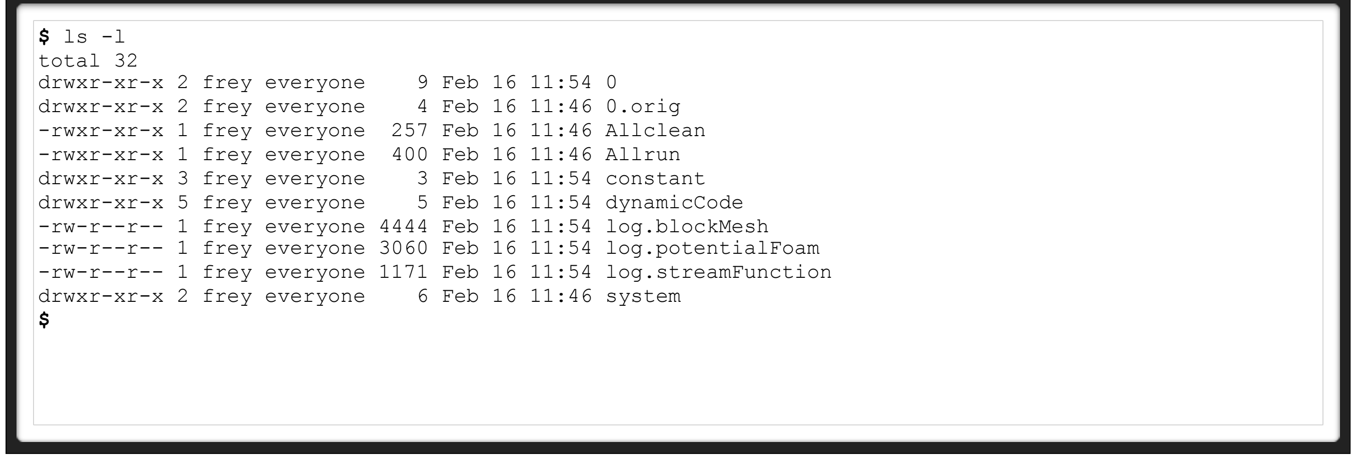


```
$ export WM_PRECISION_OPTION=DP
$ export WM_COMPILE_OPTION=Opt
$ export WM_LABEL_SIZE=64
$ export WM_PROJECT_USER_DIR="$(pwd)"
$ source $SOURCE_TO_CONFIGURE_OPENFOAM
$ ./Allrun
Running blockMesh on /home/1001/of-projects/cylinder
Running potentialFoam on /home/1001/of-projects/cylinder
Running streamFunction on /home/1001/of-projects/cylinder
$
```

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #4

- An example computation task making use of standard OpenFOAM tools
- Let's try running it
  - Log files for each of the three steps are now present
  - The grid for the computation is in the "constant" directory

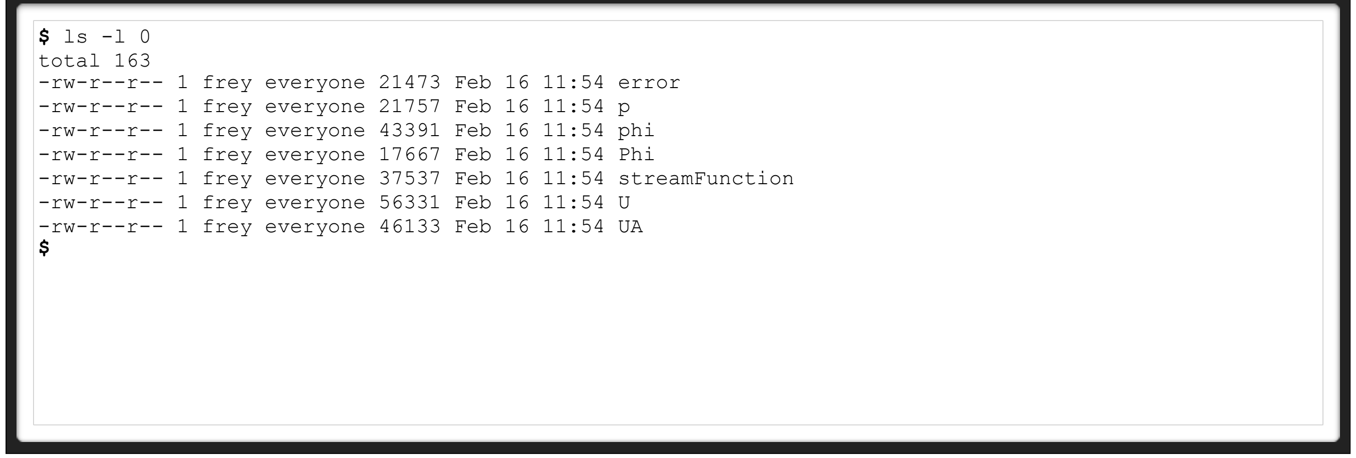


```
$ ls -l
total 32
drwxr-xr-x 2 frey everyone    9 Feb 16 11:54 0
drwxr-xr-x 2 frey everyone    4 Feb 16 11:46 0.orig
-rwxr-xr-x 1 frey everyone  257 Feb 16 11:46 Allclean
-rwxr-xr-x 1 frey everyone  400 Feb 16 11:46 Allrun
drwxr-xr-x 3 frey everyone    3 Feb 16 11:54 constant
drwxr-xr-x 5 frey everyone    5 Feb 16 11:54 dynamicCode
-rw-r--r-- 1 frey everyone 4444 Feb 16 11:54 log.blockMesh
-rw-r--r-- 1 frey everyone 3060 Feb 16 11:54 log.potentialFoam
-rw-r--r-- 1 frey everyone 1171 Feb 16 11:54 log.streamFunction
drwxr-xr-x 2 frey everyone    6 Feb 16 11:46 system
$
```

USING OPENFOAM ON THE COMMUNITY CLUSTER

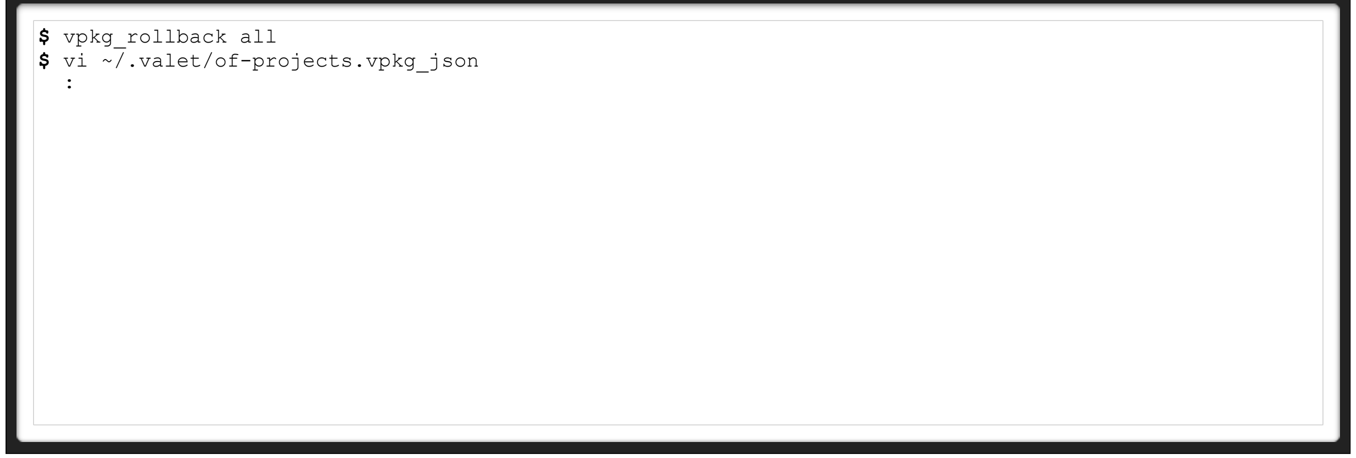
EXAMPLE #4

- An example computation task making use of standard OpenFOAM tools
- Let's try running it
  - Input objects ("p" and "U") have been updated
  - Output objects in OpenFOAM format in the "0" directory



```
$ ls -l 0
total 163
-rw-r--r-- 1 frey everyone 21473 Feb 16 11:54 error
-rw-r--r-- 1 frey everyone 21757 Feb 16 11:54 p
-rw-r--r-- 1 frey everyone 43391 Feb 16 11:54 phi
-rw-r--r-- 1 frey everyone 17667 Feb 16 11:54 Phi
-rw-r--r-- 1 frey everyone 37537 Feb 16 11:54 streamFunction
-rw-r--r-- 1 frey everyone 56331 Feb 16 11:54 U
-rw-r--r-- 1 frey everyone 46133 Feb 16 11:54 UA
$
```

- An example computation task making use of standard OpenFOAM tools
- Let's try running it
- Computational projects can be added to the "of-projects" package, too

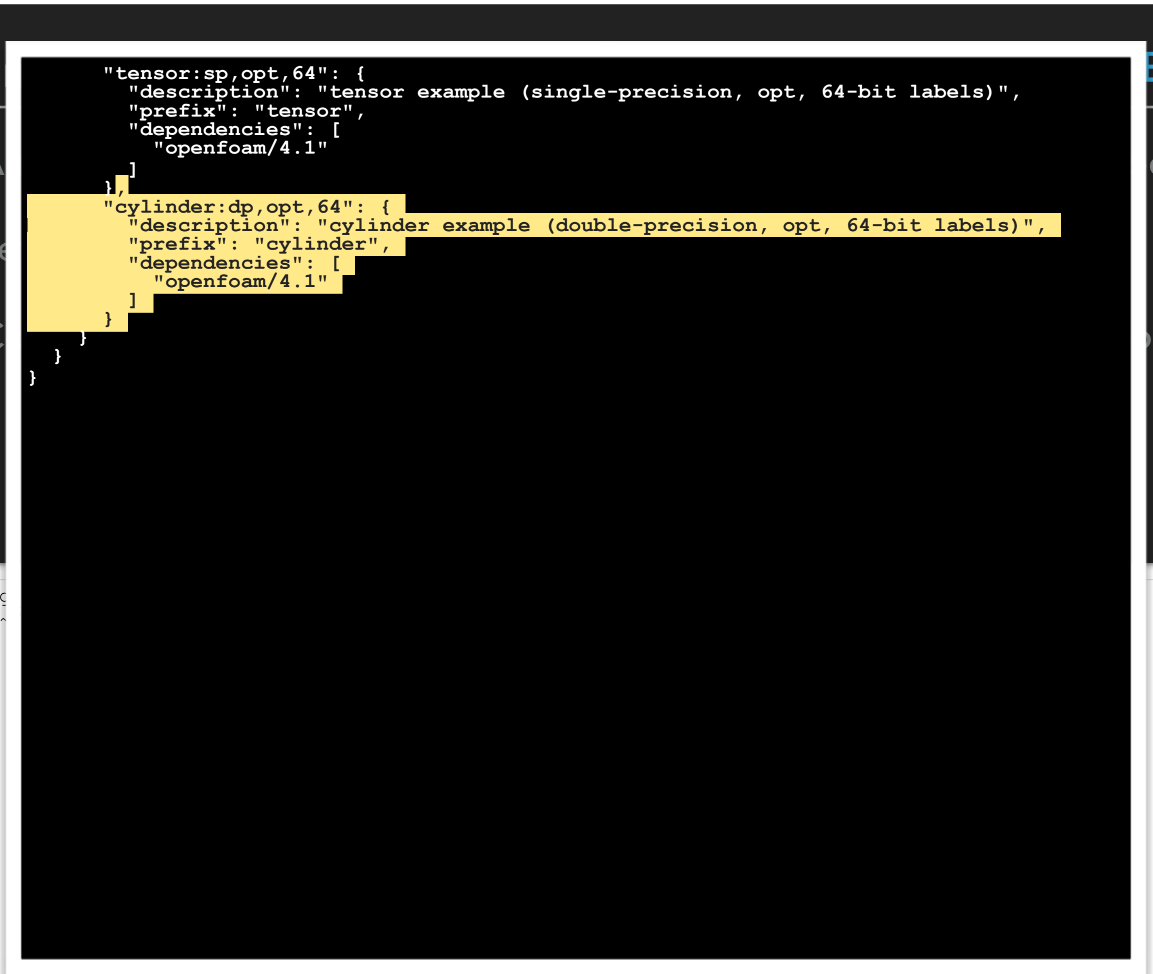


```
$ vpkg_rollback all
$ vi ~/.valet/of-projects.vpkg_json
  :
```

USI … E #4

- A … ols
- Le
- C … o

```
$ vpkg
$ vi ~
  :
```

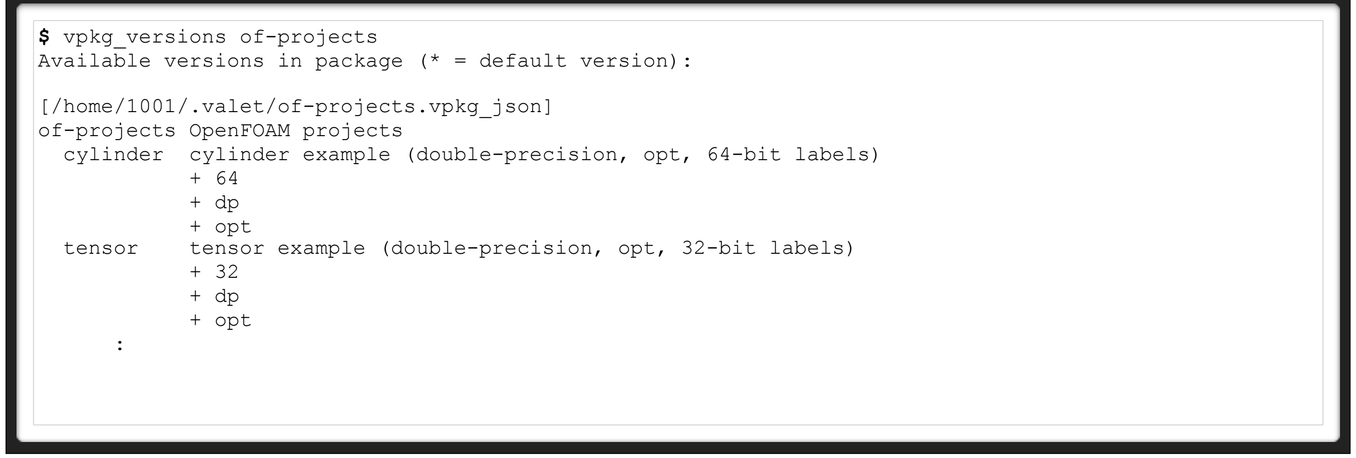


```
      "tensor:sp,opt,64": {
        "description": "tensor example (single-precision, opt, 64-bit labels)",
        "prefix": "tensor",
        "dependencies": [
          "openfoam/4.1"
        ]
      },
      "cylinder:dp,opt,64": {
        "description": "cylinder example (double-precision, opt, 64-bit labels)",
        "prefix": "cylinder",
        "dependencies": [
          "openfoam/4.1"
        ]
      }
    }
  }
}
```

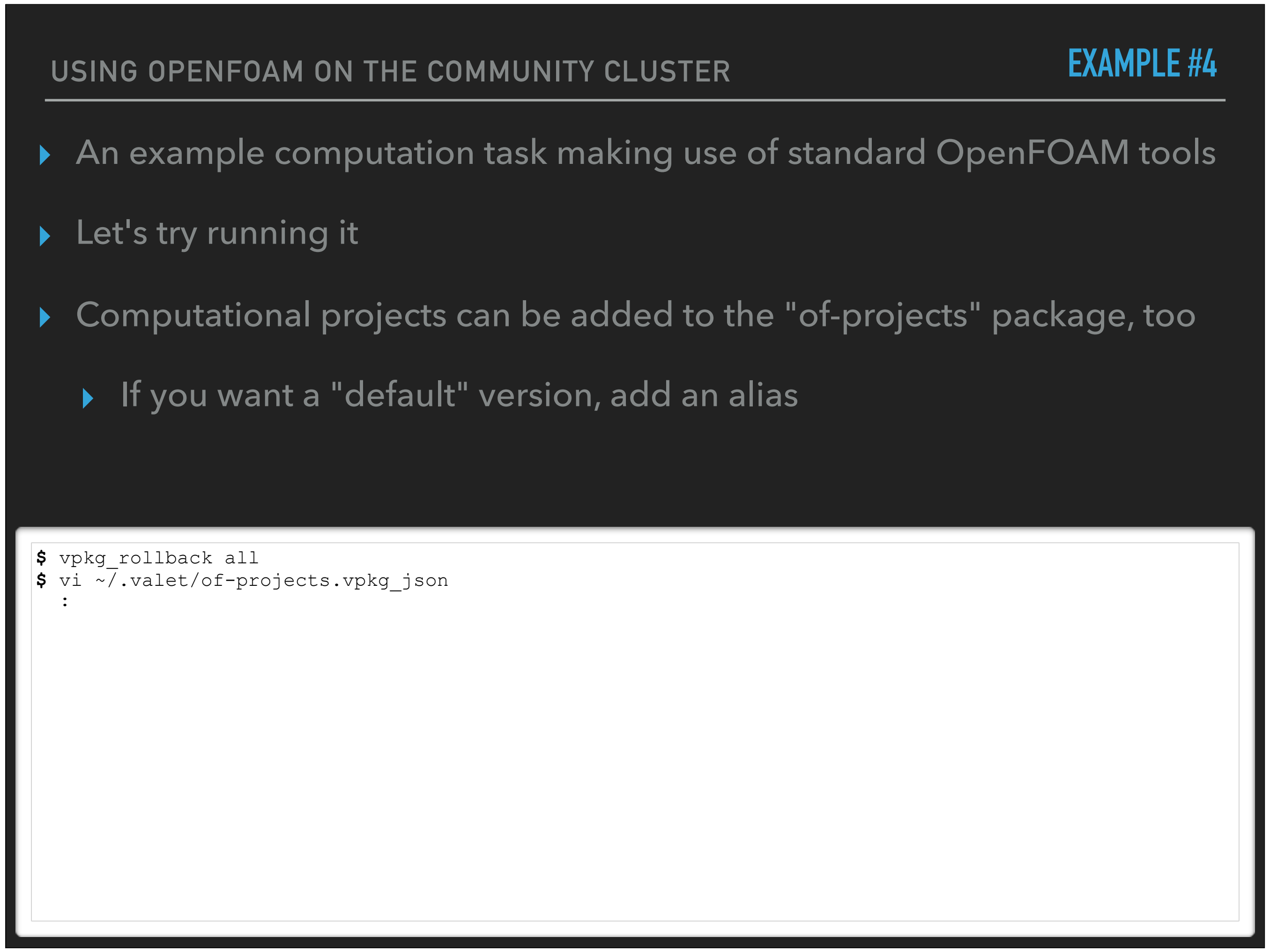
USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #4

- An example computation task making use of standard OpenFOAM tools
- Let's try running it
- Computational projects can be added to the "of-projects" package, too
  - After a successful syntax check, the "cylinder" project is present



```
$ vpkg_versions of-projects
Available versions in package (* = default version):

[/home/1001/.valet/of-projects.vpkg_json]
of-projects OpenFOAM projects
  cylinder  cylinder example (double-precision, opt, 64-bit labels)
            + 64
            + dp
            + opt
  tensor    tensor example (double-precision, opt, 32-bit labels)
            + 32
            + dp
            + opt
      :
```



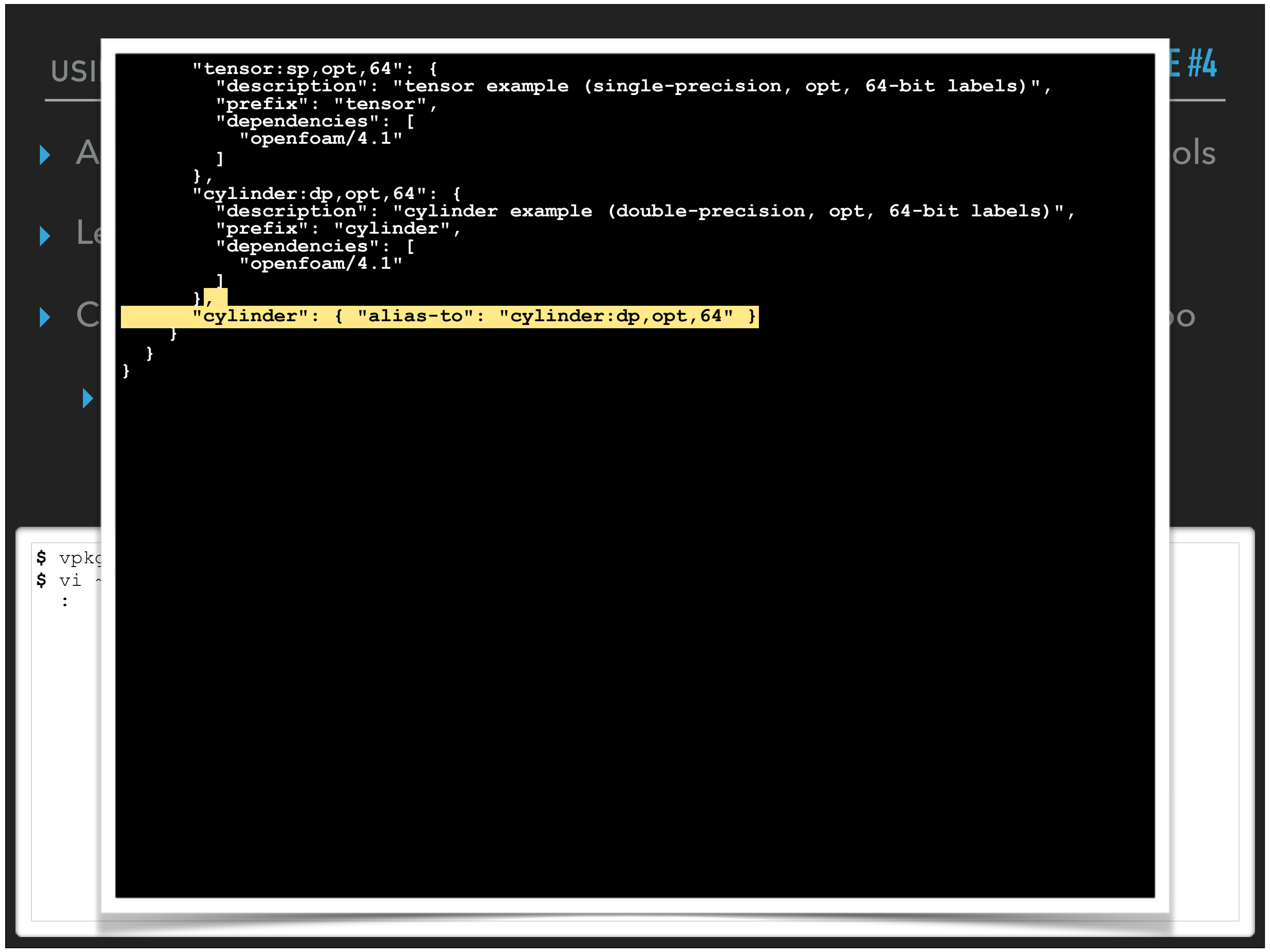
USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #4

- An example computation task making use of standard OpenFOAM tools
- Let's try running it
- Computational projects can be added to the "of-projects" package, too
  - If you want a "default" version, add an alias

```
$ vpkg_rollback all
$ vi ~/.valet/of-projects.vpkg_json
  :
```

- A VALET *version alias* is (usually) a simpler version identifier referencing a longer, more complex identifier



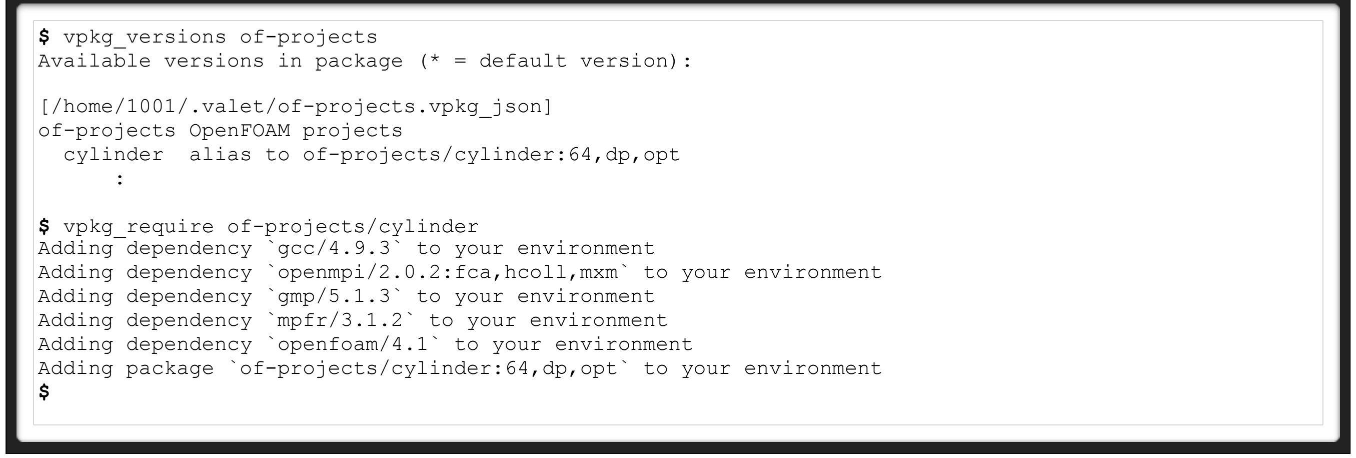
```
$ vpkg
$ vi ~
  :
```

```
      "tensor:sp,opt,64": {
        "description": "tensor example (single-precision, opt, 64-bit labels)",
        "prefix": "tensor",
        "dependencies": [
          "openfoam/4.1"
        ]
      },
      "cylinder:dp,opt,64": {
        "description": "cylinder example (double-precision, opt, 64-bit labels)",
        "prefix": "cylinder",
        "dependencies": [
          "openfoam/4.1"
        ]
      },
      "cylinder": { "alias-to": "cylinder:dp,opt,64" }
    }
  }
}
```

- A VALET *version alias* is (usually) a simpler version identifier referencing a longer, more complex identifier

EXAMPLE #4

- An example computation task making use of standard OpenFOAM tools
- Let's try running it
- Computational projects can be added to the "of-projects" package, too
  - After a successful syntax check, the "cylinder" alias is present, too

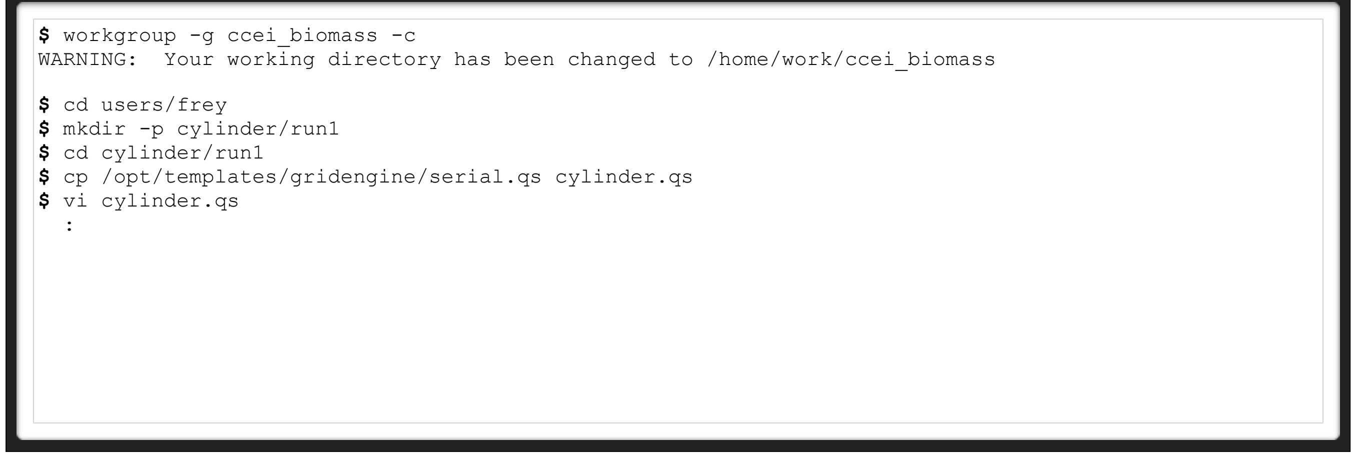


```
$ vpkg_versions of-projects
Available versions in package (* = default version):

[/home/1001/.valet/of-projects.vpkg_json]
of-projects OpenFOAM projects
  cylinder  alias to of-projects/cylinder:64,dp,opt
      :

$ vpkg_require of-projects/cylinder
Adding dependency `gcc/4.9.3` to your environment
Adding dependency `openmpi/2.0.2:fca,hcoll,mxm` to your environment
Adding dependency `gmp/5.1.3` to your environment
Adding dependency `mpfr/3.1.2` to your environment
Adding dependency `openfoam/4.1` to your environment
Adding package `of-projects/cylinder:64,dp,opt` to your environment
$
```

USING OPENFOAM ON THE COMMUNITY CLUSTER

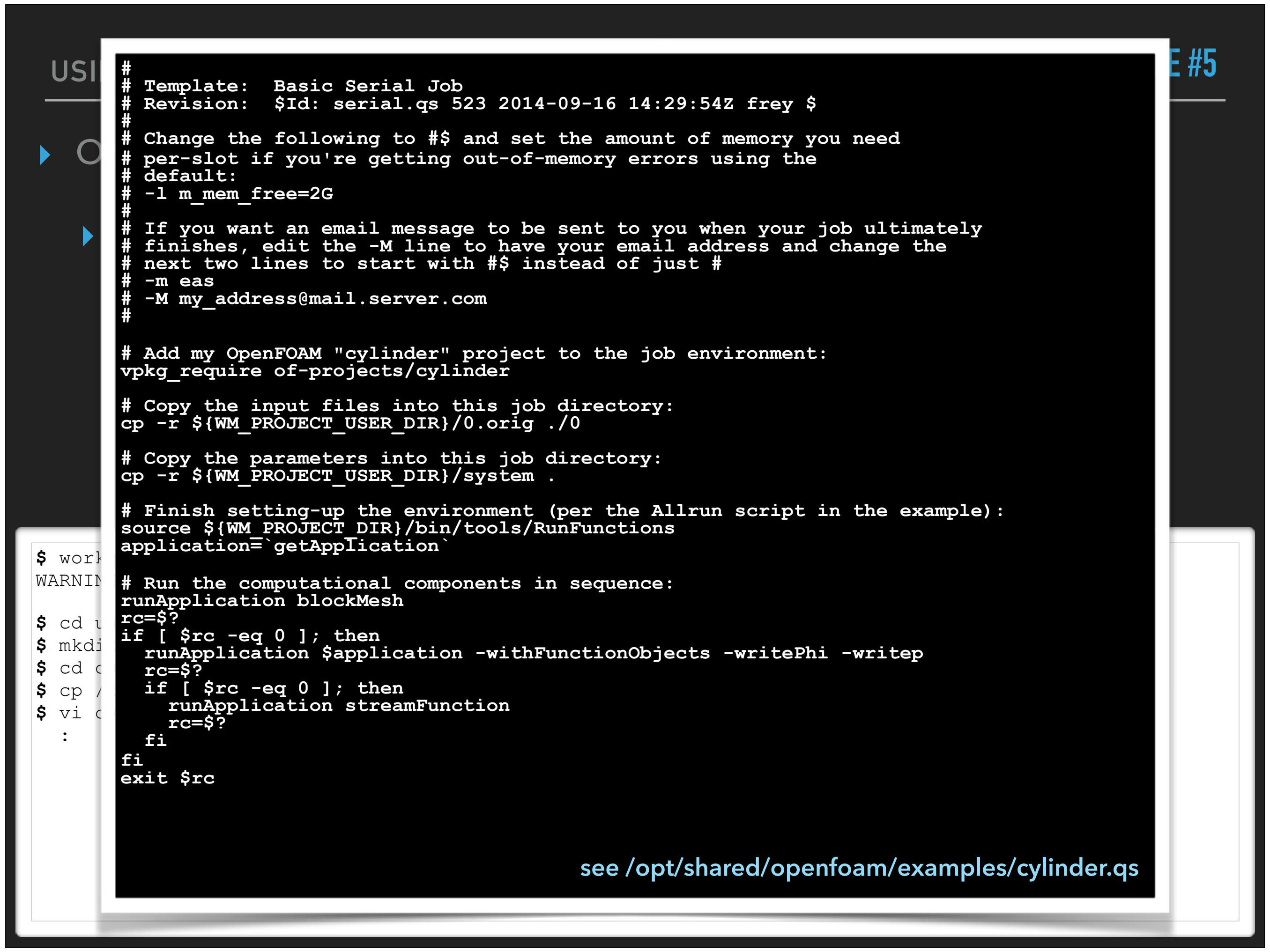
EXAMPLE #5

- OpenFOAM jobs should be run in the cluster's job scheduler
  - The "cylinder" example is run serially (a single process)



```
$ workgroup -g ccei_biomass -c
WARNING:  Your working directory has been changed to /home/work/ccei_biomass

$ cd users/frey
$ mkdir -p cylinder/run1
$ cd cylinder/run1
$ cp /opt/templates/gridengine/serial.qs cylinder.qs
$ vi cylinder.qs
  :
```

- This assumes your workgroup has a "users" directory for each group member's personal storage, with each member's directory therein named accordingly



USI E #5

O

```
$ work
WARNIN

$ cd u
$ mkdi
$ cd c
$ cp /
$ vi c
  :
```

```
#
# Template:  Basic Serial Job
# Revision:  $Id: serial.qs 523 2014-09-16 14:29:54Z frey $
#
# Change the following to #$ and set the amount of memory you need
# per-slot if you're getting out-of-memory errors using the
# default:
# -l m_mem_free=2G
#
# If you want an email message to be sent to you when your job ultimately
# finishes, edit the -M line to have your email address and change the
# next two lines to start with #$ instead of just #
# -m eas
# -M my_address@mail.server.com
#

# Add my OpenFOAM "cylinder" project to the job environment:
vpkg_require of-projects/cylinder

# Copy the input files into this job directory:
cp -r ${WM_PROJECT_USER_DIR}/0.orig ./0

# Copy the parameters into this job directory:
cp -r ${WM_PROJECT_USER_DIR}/system .

# Finish setting-up the environment (per the Allrun script in the example):
source ${WM_PROJECT_DIR}/bin/tools/RunFunctions
application=`getApplication`

# Run the computational components in sequence:
runApplication blockMesh
rc=$?
if [ $rc -eq 0 ]; then
  runApplication $application -withFunctionObjects -writePhi -writep
  rc=$?
  if [ $rc -eq 0 ]; then
    runApplication streamFunction
    rc=$?
  fi
fi
exit $rc
```

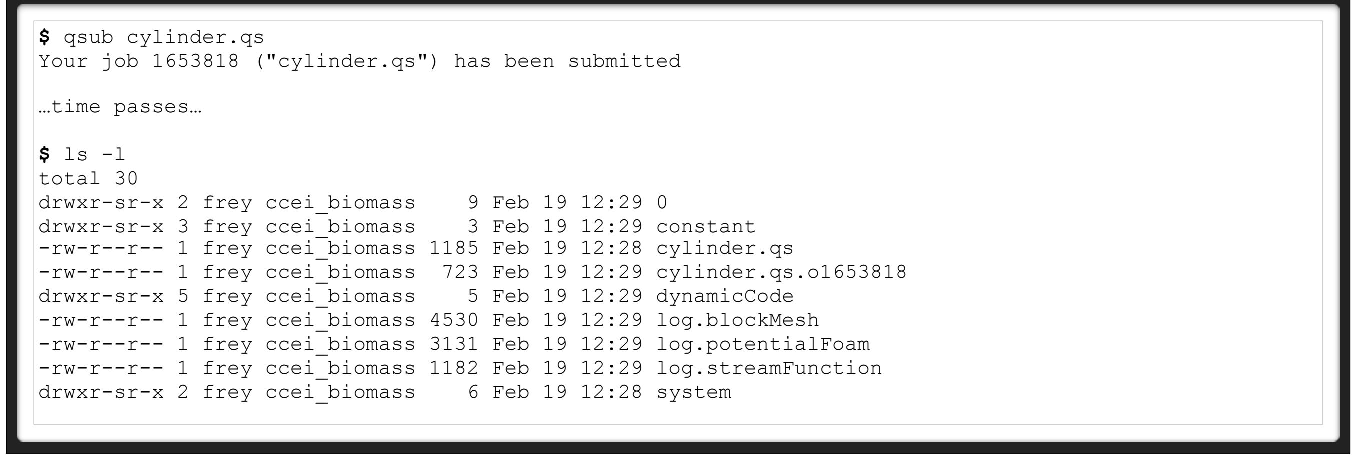
see /opt/shared/openfoam/examples/cylinder.qs

- This assumes your workgroup has a "users" directory for each group member's personal storage, with each member's directory therein named accordingly

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #5

- OpenFOAM jobs should be run in the cluster's job scheduler
  - The "cylinder" example is run serially (a single process)
    - Submit the job…and wait for it to execute…



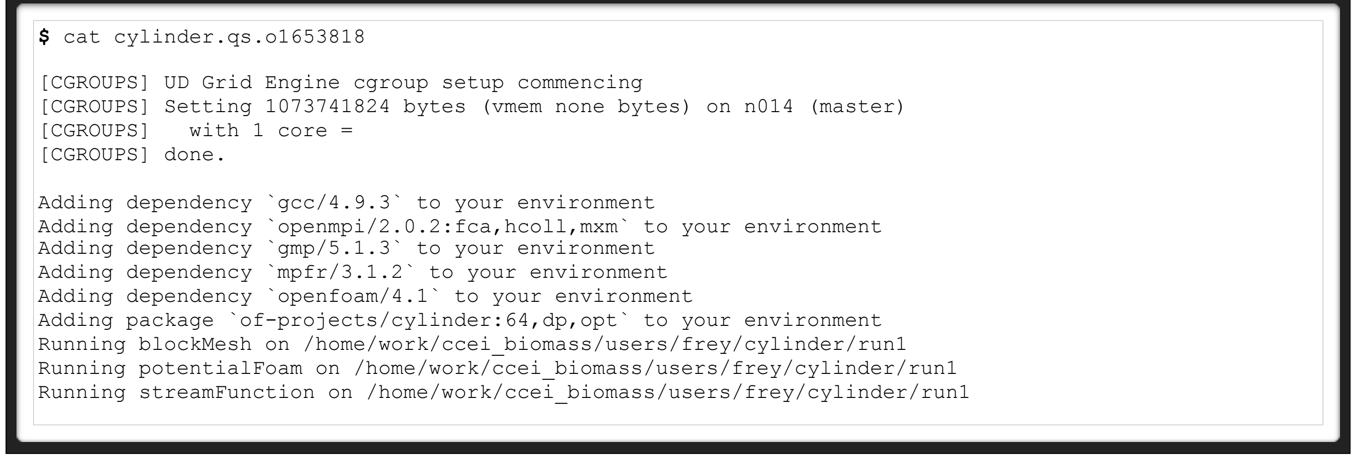
```
$ qsub cylinder.qs
Your job 1653818 ("cylinder.qs") has been submitted

…time passes…

$ ls -l
total 30
drwxr-sr-x 2 frey ccei_biomass    9 Feb 19 12:29 0
drwxr-sr-x 3 frey ccei_biomass    3 Feb 19 12:29 constant
-rw-r--r-- 1 frey ccei_biomass 1185 Feb 19 12:28 cylinder.qs
-rw-r--r-- 1 frey ccei_biomass  723 Feb 19 12:29 cylinder.qs.o1653818
drwxr-sr-x 5 frey ccei_biomass    5 Feb 19 12:29 dynamicCode
-rw-r--r-- 1 frey ccei_biomass 4530 Feb 19 12:29 log.blockMesh
-rw-r--r-- 1 frey ccei_biomass 3131 Feb 19 12:29 log.potentialFoam
-rw-r--r-- 1 frey ccei_biomass 1182 Feb 19 12:29 log.streamFunction
drwxr-sr-x 2 frey ccei_biomass    6 Feb 19 12:28 system
```

- The stdout for the job is present as the script filename with a .o#### extension
- Same log files and new directories (constant, dynamicCode) as before are present — hooray!

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #5

- OpenFOAM jobs should be run in the cluster's job scheduler
  - The "cylinder" example is run serially (a single process)
    - The stdout for the job should show:
      - Same VALET environment setup seen when run on the head node
      - Same three "Running…" lines as when run on the head node

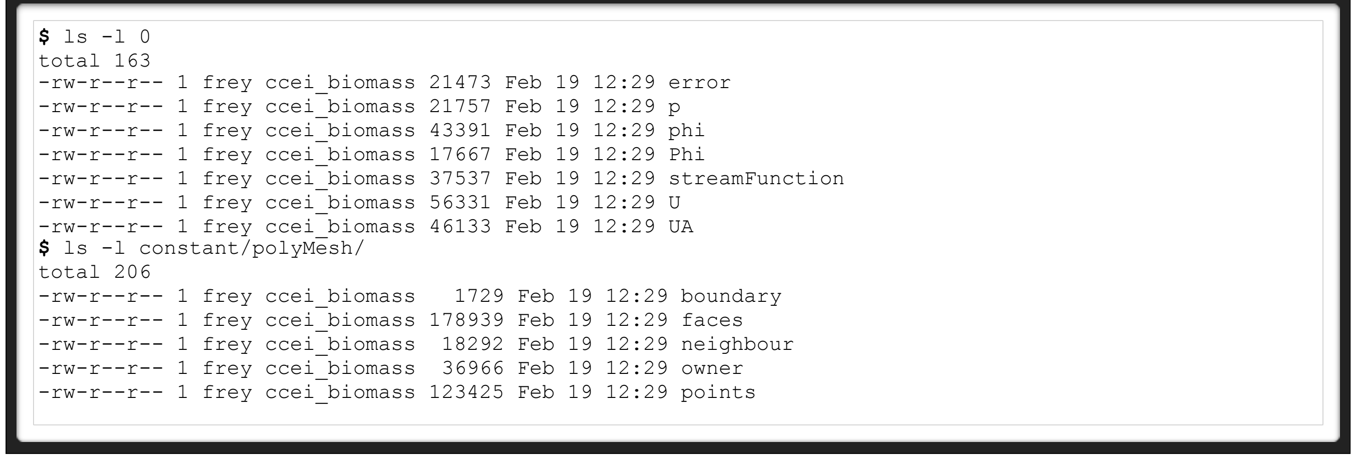


```
$ cat cylinder.qs.o1653818

[CGROUPS] UD Grid Engine cgroup setup commencing
[CGROUPS] Setting 1073741824 bytes (vmem none bytes) on n014 (master)
[CGROUPS]   with 1 core =
[CGROUPS] done.

Adding dependency `gcc/4.9.3` to your environment
Adding dependency `openmpi/2.0.2:fca,hcoll,mxm` to your environment
Adding dependency `gmp/5.1.3` to your environment
Adding dependency `mpfr/3.1.2` to your environment
Adding dependency `openfoam/4.1` to your environment
Adding package `of-projects/cylinder:64,dp,opt` to your environment
Running blockMesh on /home/work/ccei_biomass/users/frey/cylinder/run1
Running potentialFoam on /home/work/ccei_biomass/users/frey/cylinder/run1
Running streamFunction on /home/work/ccei_biomass/users/frey/cylinder/run1
```

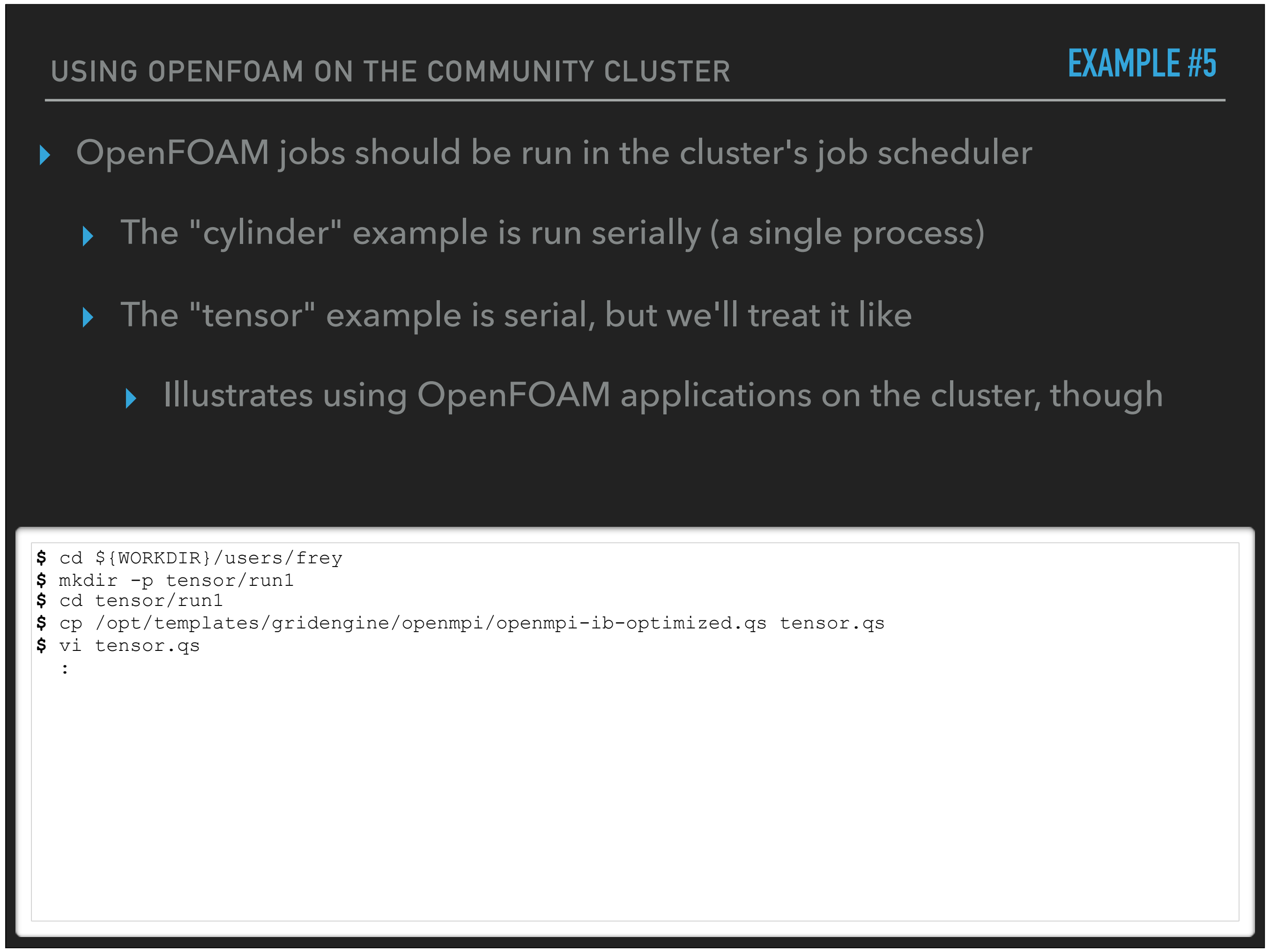
USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #5

- OpenFOAM jobs should be run in the cluster's job scheduler
  - The "cylinder" example is run serially (a single process)
    - The stdout for the job should show:
    - New and updated objects in the "0" directory (same as before)
    - Mesh description in "constant/polyMesh" directory (same as before)



```
$ ls -l 0
total 163
-rw-r--r-- 1 frey ccei_biomass 21473 Feb 19 12:29 error
-rw-r--r-- 1 frey ccei_biomass 21757 Feb 19 12:29 p
-rw-r--r-- 1 frey ccei_biomass 43391 Feb 19 12:29 phi
-rw-r--r-- 1 frey ccei_biomass 17667 Feb 19 12:29 Phi
-rw-r--r-- 1 frey ccei_biomass 37537 Feb 19 12:29 streamFunction
-rw-r--r-- 1 frey ccei_biomass 56331 Feb 19 12:29 U
-rw-r--r-- 1 frey ccei_biomass 46133 Feb 19 12:29 UA
$ ls -l constant/polyMesh/
total 206
-rw-r--r-- 1 frey ccei_biomass   1729 Feb 19 12:29 boundary
-rw-r--r-- 1 frey ccei_biomass 178939 Feb 19 12:29 faces
-rw-r--r-- 1 frey ccei_biomass  18292 Feb 19 12:29 neighbour
-rw-r--r-- 1 frey ccei_biomass  36966 Feb 19 12:29 owner
-rw-r--r-- 1 frey ccei_biomass 123425 Feb 19 12:29 points
```



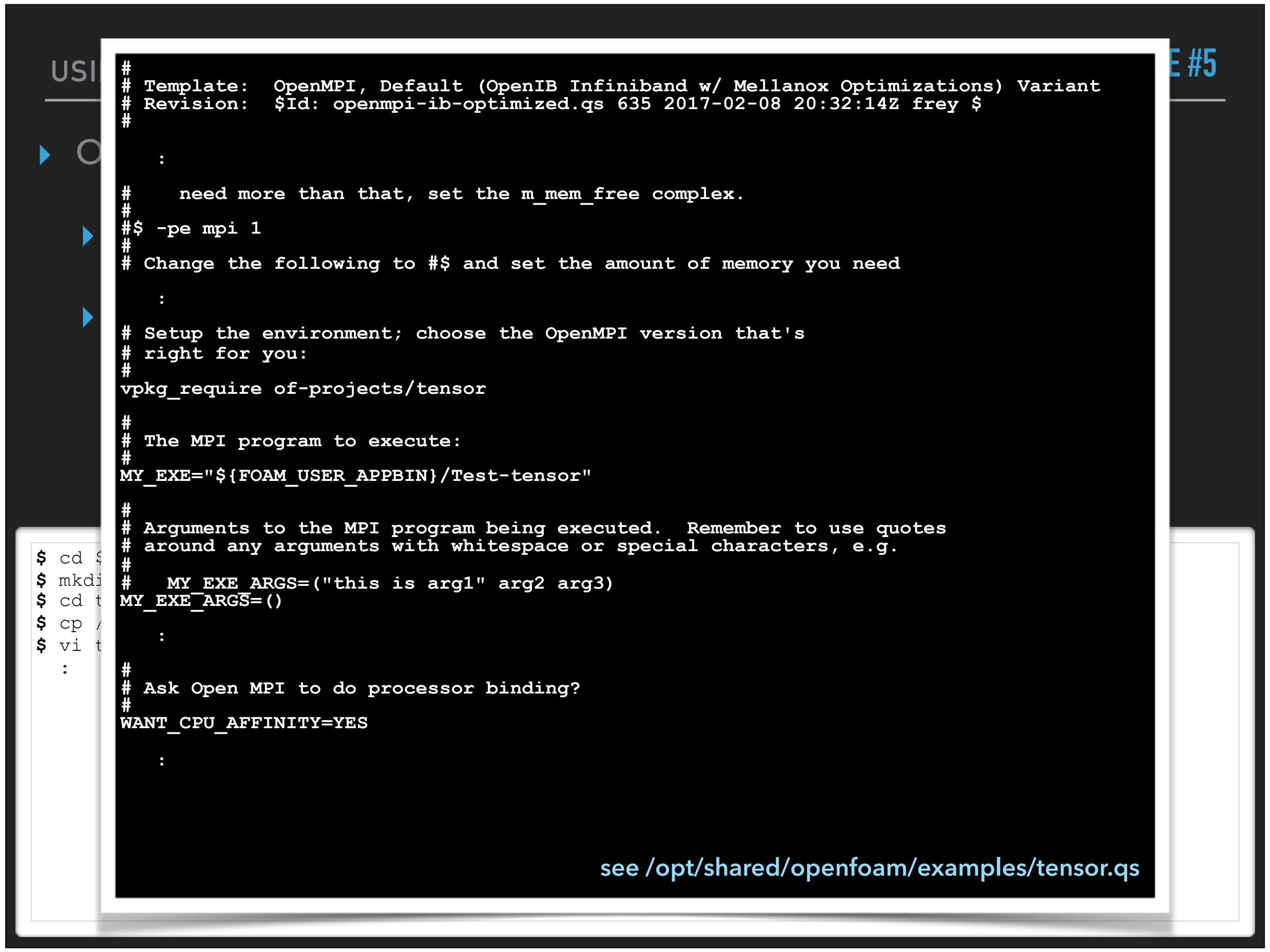
USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #5

- OpenFOAM jobs should be run in the cluster's job scheduler
  - The "cylinder" example is run serially (a single process)
  - The "tensor" example is serial, but we'll treat it like
    - Illustrates using OpenFOAM applications on the cluster, though

```
$ cd ${WORKDIR}/users/frey
$ mkdir -p tensor/run1
$ cd tensor/run1
$ cp /opt/templates/gridengine/openmpi/openmpi-ib-optimized.qs tensor.qs
$ vi tensor.qs
  :
```

- On Farber, OpenFOAM 4.1 is linked against an Open MPI library that includes Mellanox IB optimizations
  - Thus, we use the "ib-optimized" template; previous OpenFOAM versions should use the "openmpi-ib.qs" template
- Only the **changes** to the template script are shown in the black-and-white listing

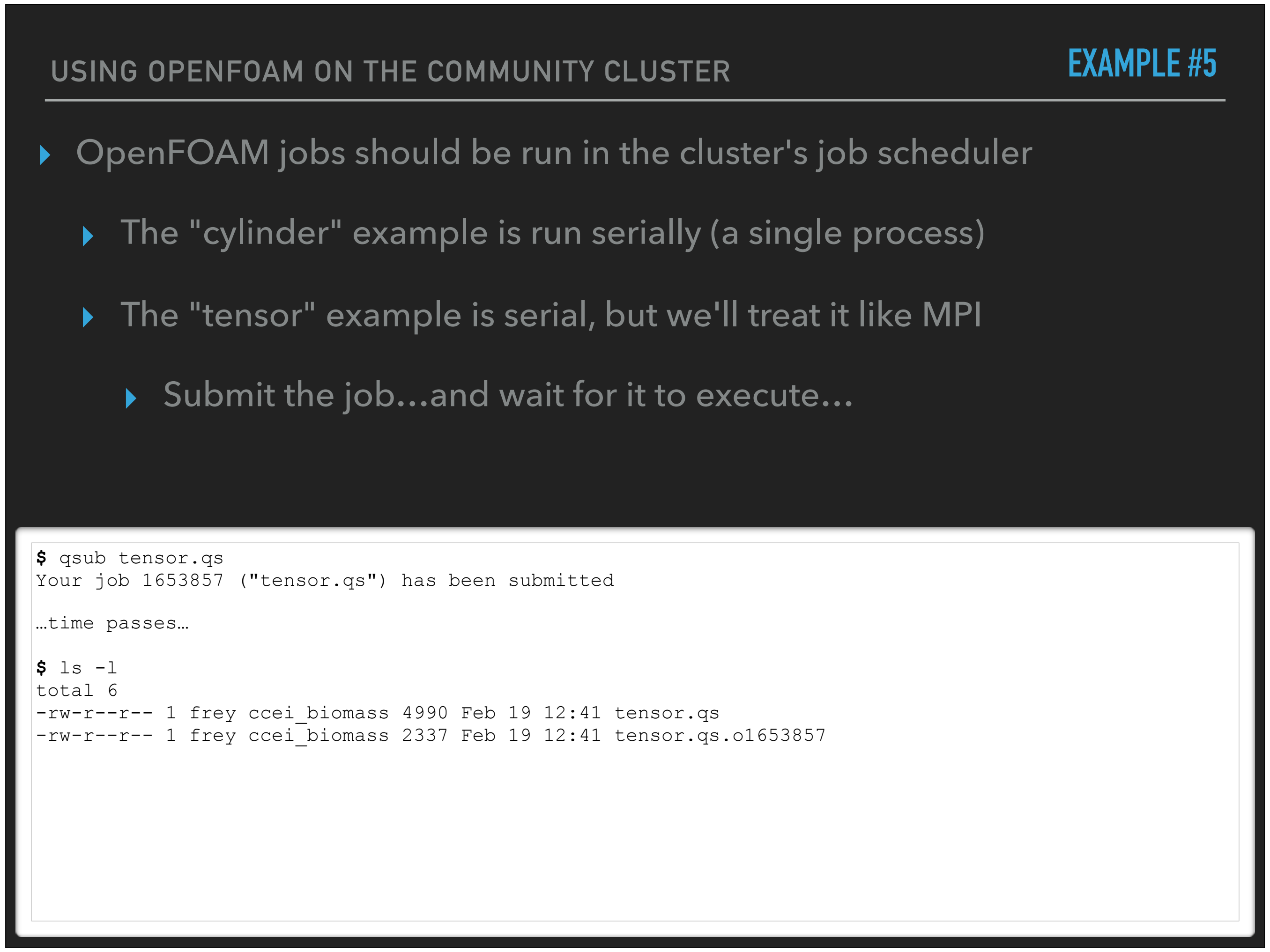


```
$ cd $
$ mkdi
$ cd t
$ cp /
$ vi t
  :
```

```
#
# Template:  OpenMPI, Default (OpenIB Infiniband w/ Mellanox Optimizations) Variant
# Revision:  $Id: openmpi-ib-optimized.qs 635 2017-02-08 20:32:14Z frey $
#

   :

#    need more than that, set the m_mem_free complex.
#
#$ -pe mpi 1
#
# Change the following to #$ and set the amount of memory you need

   :

# Setup the environment; choose the OpenMPI version that's
# right for you:
#
vpkg_require of-projects/tensor

#
# The MPI program to execute:
#
MY_EXE="${FOAM_USER_APPBIN}/Test-tensor"

#
# Arguments to the MPI program being executed.  Remember to use quotes
# around any arguments with whitespace or special characters, e.g.
#
#   MY_EXE_ARGS=("this is arg1" arg2 arg3)
MY_EXE_ARGS=()

   :

#
# Ask Open MPI to do processor binding?
#
WANT_CPU_AFFINITY=YES

   :
```

see /opt/shared/openfoam/examples/tensor.qs

- On Farber, OpenFOAM 4.1 is linked against an Open MPI library that includes Mellanox IB optimizations
    - Thus, we use the "ib-optimized" template; previous OpenFOAM versions should use the "openmpi-ib.qs" template
- Only the **changes** to the template script are shown in the black-and-white listing



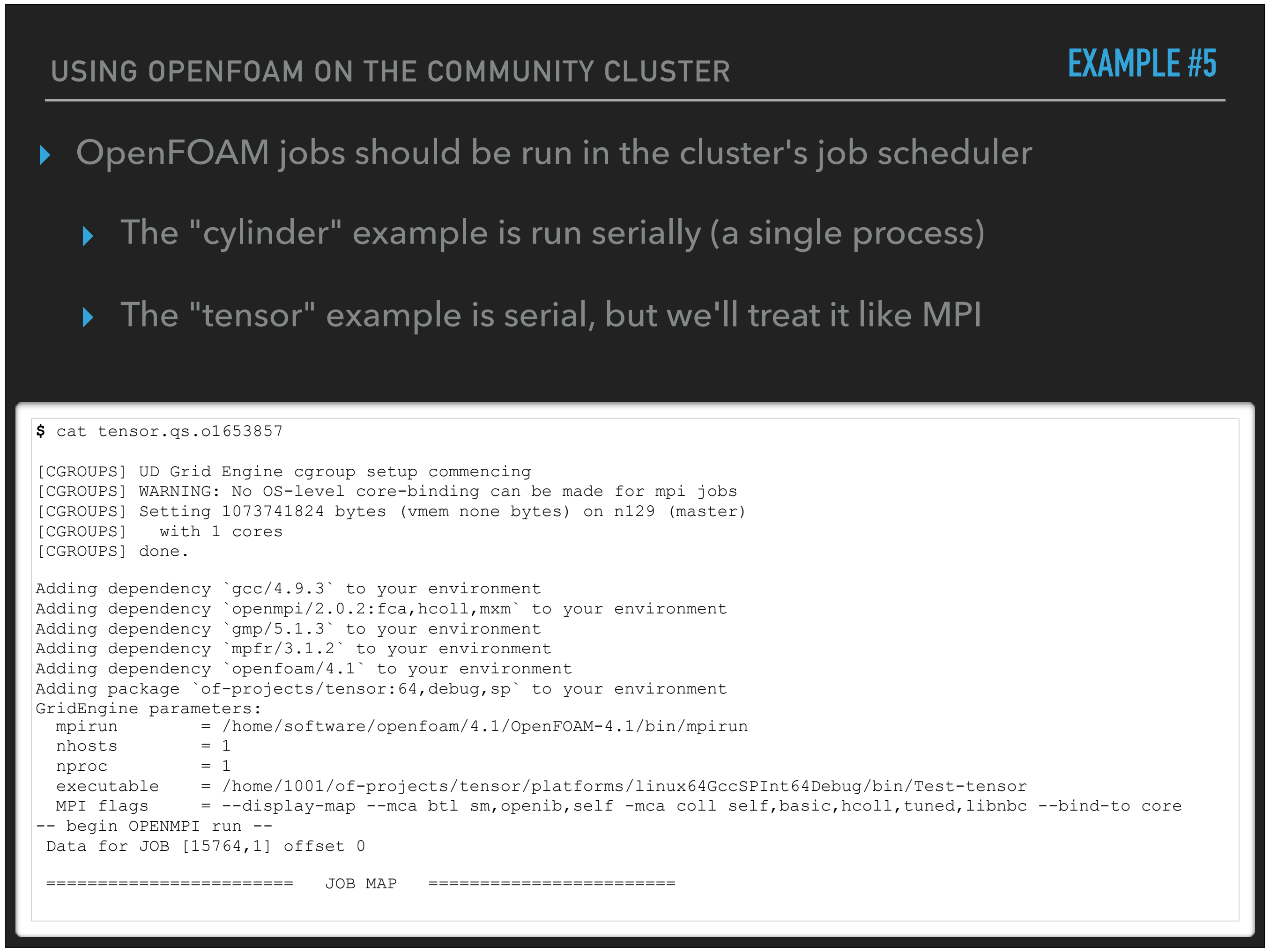
USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #5

- OpenFOAM jobs should be run in the cluster's job scheduler
  - The "cylinder" example is run serially (a single process)
  - The "tensor" example is serial, but we'll treat it like MPI
    - Submit the job…and wait for it to execute…

```
$ qsub tensor.qs
Your job 1653857 ("tensor.qs") has been submitted

…time passes…

$ ls -l
total 6
-rw-r--r-- 1 frey ccei_biomass 4990 Feb 19 12:41 tensor.qs
-rw-r--r-- 1 frey ccei_biomass 2337 Feb 19 12:41 tensor.qs.o1653857
```

- The stdout for the job is present as the script filename with a .o#### extension



USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #5

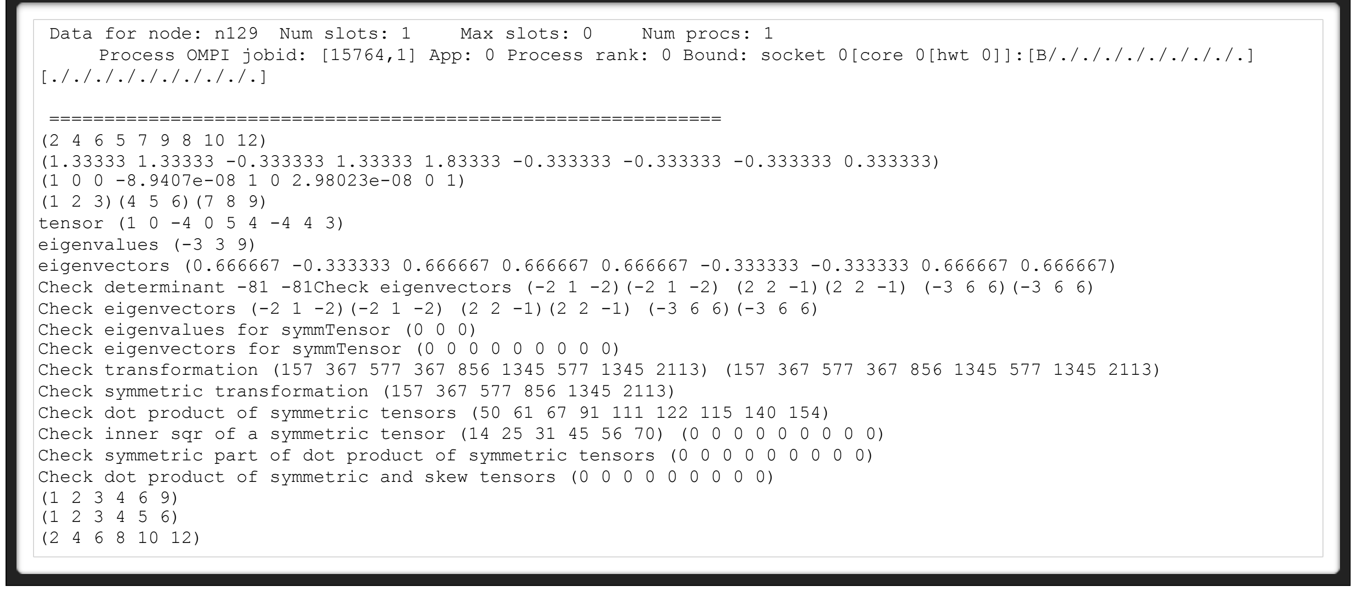
- OpenFOAM jobs should be run in the cluster's job scheduler
  - The "cylinder" example is run serially (a single process)
  - The "tensor" example is serial, but we'll treat it like MPI

```
$ cat tensor.qs.o1653857

[CGROUPS] UD Grid Engine cgroup setup commencing
[CGROUPS] WARNING: No OS-level core-binding can be made for mpi jobs
[CGROUPS] Setting 1073741824 bytes (vmem none bytes) on n129 (master)
[CGROUPS]   with 1 cores
[CGROUPS] done.

Adding dependency `gcc/4.9.3` to your environment
Adding dependency `openmpi/2.0.2:fca,hcoll,mxm` to your environment
Adding dependency `gmp/5.1.3` to your environment
Adding dependency `mpfr/3.1.2` to your environment
Adding dependency `openfoam/4.1` to your environment
Adding package `of-projects/tensor:64,debug,sp` to your environment
GridEngine parameters:
  mpirun        = /home/software/openfoam/4.1/OpenFOAM-4.1/bin/mpirun
  nhosts        = 1
  nproc         = 1
  executable    = /home/1001/of-projects/tensor/platforms/linux64GccSPInt64Debug/bin/Test-tensor
  MPI flags     = --display-map --mca btl sm,openib,self -mca coll self,basic,hcoll,tuned,libnbc --bind-to core
-- begin OPENMPI run --
 Data for JOB [15764,1] offset 0

 ========================   JOB MAP   ========================
```

- The stdout for the job is present as the script filename with a .o#### extension

EXAMPLE #5

- OpenFOAM jobs should be run in the cluster's job scheduler
  - The "cylinder" example is run serially (a single process)
  - The "tensor" example is serial, but we'll treat it like MPI



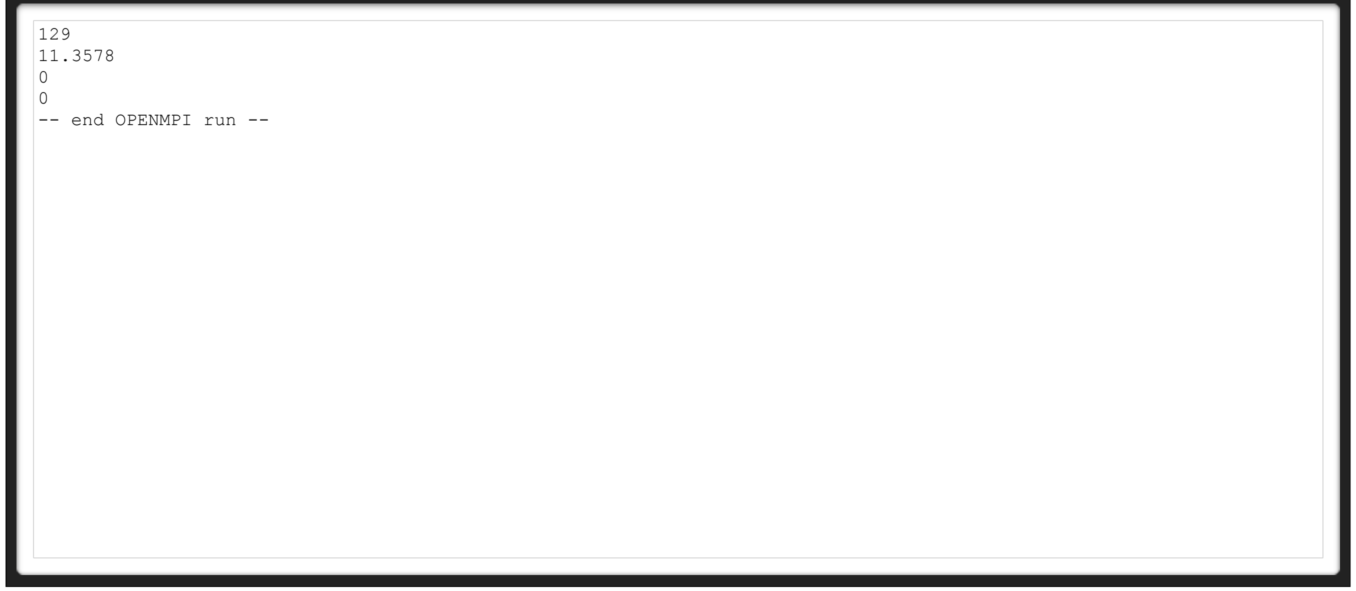
```
 Data for node: n129  Num slots: 1    Max slots: 0    Num procs: 1
      Process OMPI jobid: [15764,1] App: 0 Process rank: 0 Bound: socket 0[core 0[hwt 0]]:[B/./././././././.]
[./././././././././.]

 =============================================================
(2 4 6 5 7 9 8 10 12)
(1.33333 1.33333 -0.333333 1.33333 1.83333 -0.333333 -0.333333 -0.333333 0.333333)
(1 0 0 -8.9407e-08 1 0 2.98023e-08 0 1)
(1 2 3)(4 5 6)(7 8 9)
tensor (1 0 -4 0 5 4 -4 4 3)
eigenvalues (-3 3 9)
eigenvectors (0.666667 -0.333333 0.666667 0.666667 0.666667 -0.333333 -0.333333 0.666667 0.666667)
Check determinant -81 -81Check eigenvectors (-2 1 -2)(-2 1 -2) (2 2 -1)(2 2 -1) (-3 6 6)(-3 6 6)
Check eigenvectors (-2 1 -2)(-2 1 -2) (2 2 -1)(2 2 -1) (-3 6 6)(-3 6 6)
Check eigenvalues for symmTensor (0 0 0)
Check eigenvectors for symmTensor (0 0 0 0 0 0 0 0 0)
Check transformation (157 367 577 367 856 1345 577 1345 2113) (157 367 577 367 856 1345 577 1345 2113)
Check symmetric transformation (157 367 577 856 1345 2113)
Check dot product of symmetric tensors (50 61 67 91 111 122 115 140 154)
Check inner sqr of a symmetric tensor (14 25 31 45 56 70) (0 0 0 0 0 0 0 0 0)
Check symmetric part of dot product of symmetric tensors (0 0 0 0 0 0 0 0 0)
Check dot product of symmetric and skew tensors (0 0 0 0 0 0 0 0 0)
(1 2 3 4 6 9)
(1 2 3 4 5 6)
(2 4 6 8 10 12)
```

- Continued

USING OPENFOAM ON THE COMMUNITY CLUSTER

EXAMPLE #5

- OpenFOAM jobs should be run in the cluster's job scheduler
  - The "cylinder" example is run serially (a single process)
  - The "tensor" example is serial, but we'll treat it like MPI



```
129
11.3578
0
0
-- end OPENMPI run --
```

- Continued

SUMMARY

- OpenFOAM represents an effort to deliver optimized numerical algorithms in a single package
  - No more reinvent-the-wheel for each project you create
  - Parallelism wherever it's appropriate
  - Speed: compiled code, not interpreted (a'la Matlab or Python)

SUMMARY

- OpenFOAM represents an effort to deliver optimized numerical algorithms in a single package
- University community clusters provide a number of OpenFOAM releases, with varying platform parameterizations

SUMMARY

- OpenFOAM represents an effort to deliver optimized numerical algorithms in a single package
- University community clusters provide a number of OpenFOAM releases, with varying platform parameterizations
- VALET can be used to make OpenFOAM environment setup a cinch

QUESTIONS?