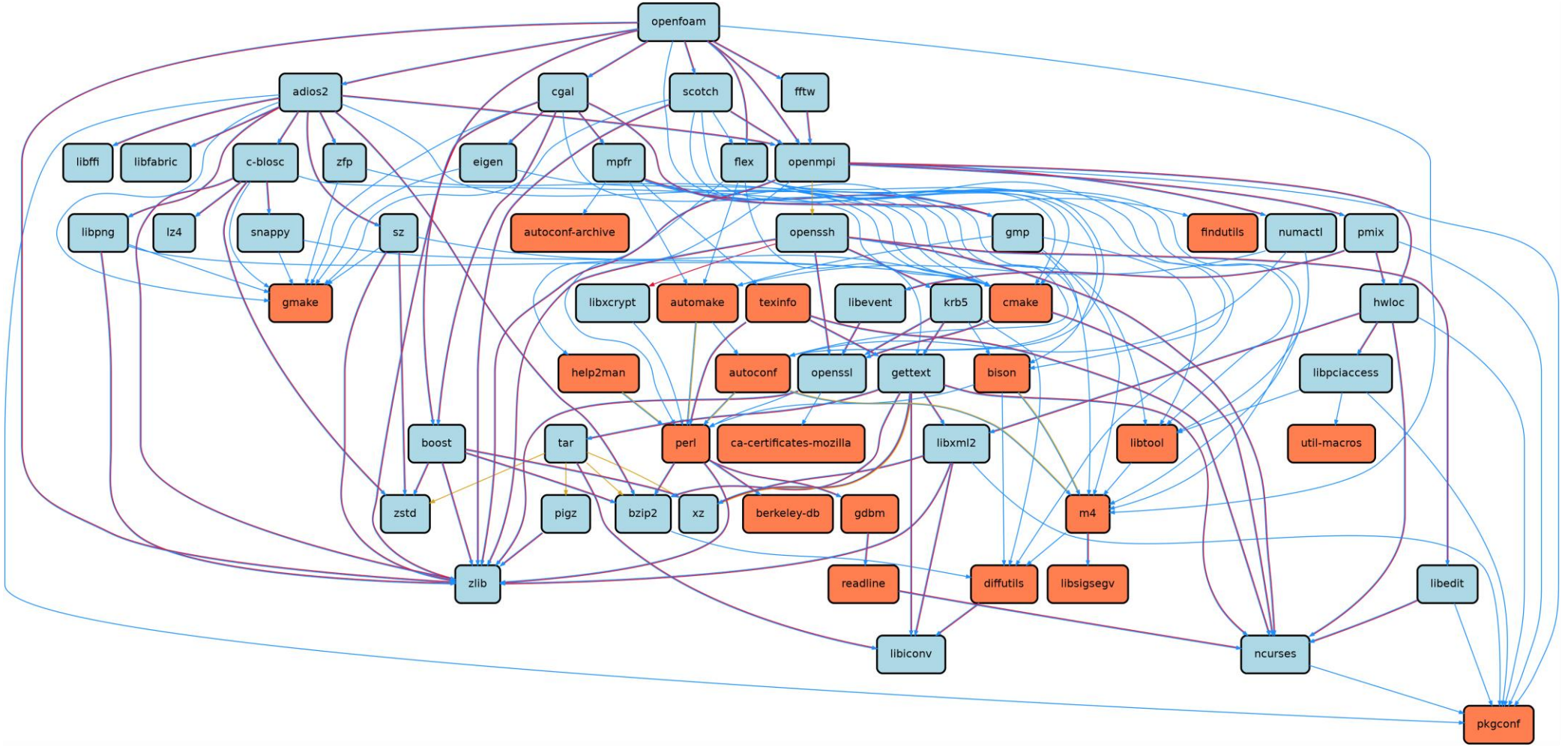
Using OpenFoam with Open Source Package Manager Spack

2023 10th OpenFOAM Korea Users' Community Conference
2023. 10.19~20

Samsung SDS Jong Rok Kim
삼성 SDS 김종록

Entry barrier of OpenFoam

- Many packages are tangled up in a complex relationship, making installation not easy.
- Containers may need to be rebuilt to support many different hosts, anyway.



The Impact of Compilation Options on Performance

- Installing software with different compilation options can lead to varying performance outcomes.
- "Intel Cascade Lake microarchitecture gives an additional 18% performance improvement relative to using AVX2 instructions, with a speedup of about 70% compared to a generic GROMACS installation with only SSE2."[1]

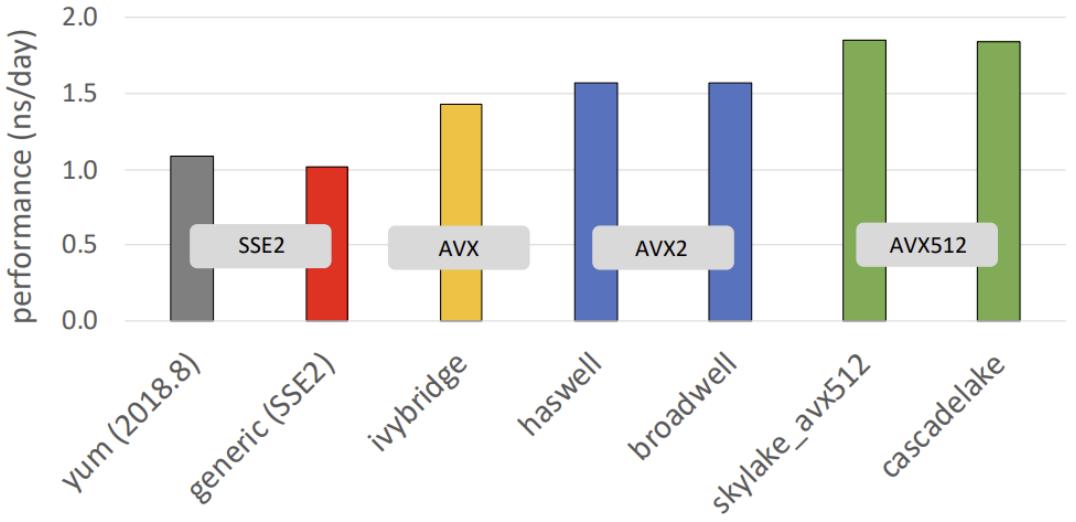
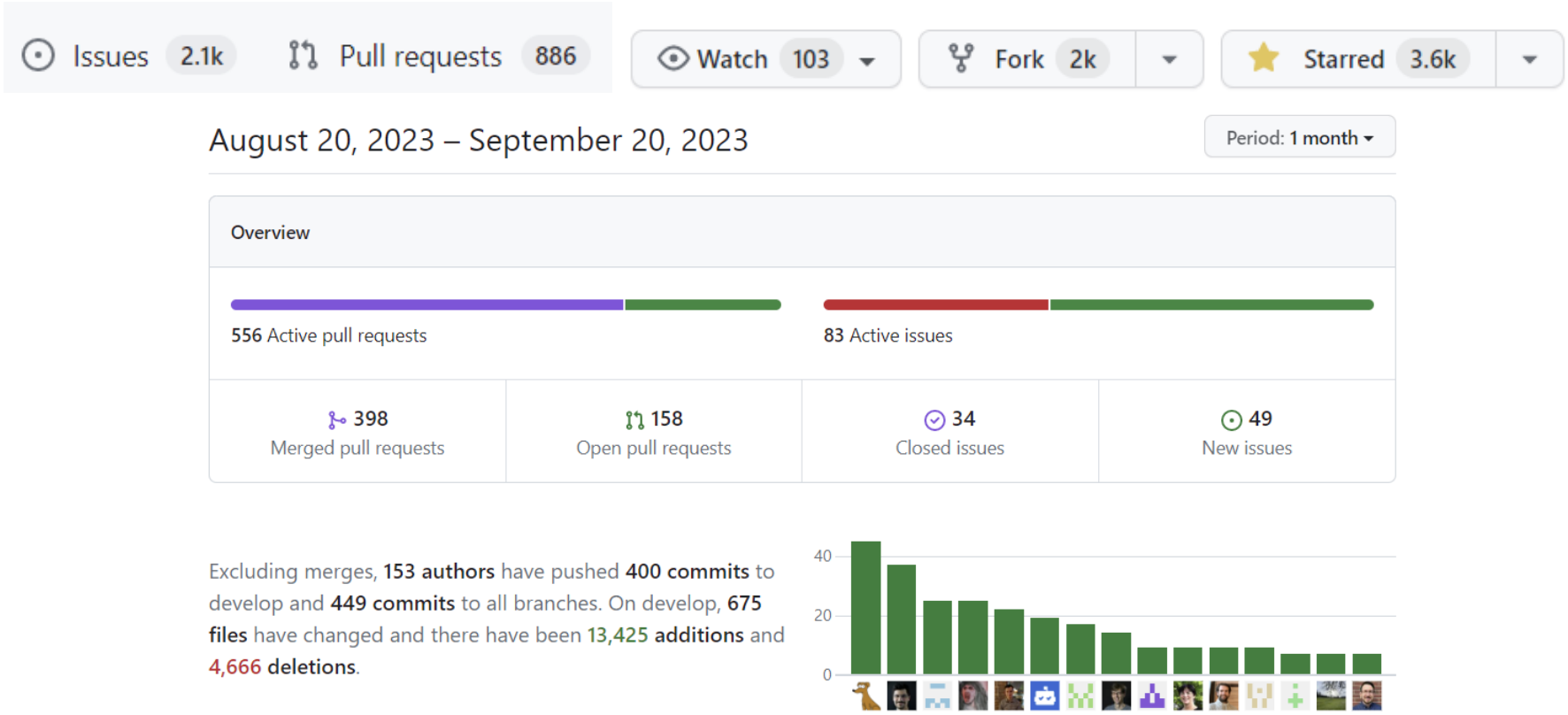


Figure 1. Performance of GROMACS 2020.1 built for different generations of CPUs.[2]Vertical axis shows performance expressed in ns/day, a GROMACS-specific measure of simulation speed (higher is better).

* Compiled with different compilation options on the same hardware and identical software version.

Spack enables Software distribution for HPC

- Spack is an open source package manager, https://github.com/spack/spack
- Feature: Build-from-source, Focused on HPC applications
- Spack will be used to build software for the three upcoming U.S. exascale systems, including the 2023 frontrunner, Frontier.
- 7,179 total packages('23.05)



How to use Spack to install OpenFOAM

$ spack install openfoam	Install OpenFOAM with a single CLI command.
$ spack install openfoam@2206	Use "@" to specify the version.
$ spack install openfoam@2206 %gcc@9.4.0	Use "%" to specify the compiler.
$ spack install openfoam@2206 +paraview	Use "+" to specify variants(build option).
$ spack install openfoam@2206 ^openmpi@4.1.0	Use "^" to specify dependencies.
$ spack install openfoam@2206 cppflags="-O3 –g3"	Specify compiler options.
$ spack install openfoam@2206 target=cascadelake	Target a specific microarchitecture.

Retrieving Information about OpenFOAM(1/2)

$ spack info openfoam

```
Package:   openfoam

Description:
  OpenFOAM is a GPL-opensource C++ CFD-toolbox. This offering is supported
  by OpenCFD Ltd, producer and distributor of the OpenFOAM software via
  www.openfoam.com, and owner of the OPENFOAM trademark. OpenCFD Ltd has
  been developing and releasing OpenFOAM since its debut in 2004.

Homepage: https://www.openfoam.com/
```

Overview of Package: OpenFoam

```
Preferred version:
  2206           https://sourceforge.net/projects/openfoam/files/v2206/OpenFOAM-v2206.tgz

Safe versions:
  develop        [git] https://develop.openfoam.com/Development/openfoam.git on branch develop
  master         [git] https://develop.openfoam.com/Development/openfoam.git on branch master
  2206           https://sourceforge.net/projects/openfoam/files/v2206/OpenFOAM-v2206.tgz
  2112           https://sourceforge.net/projects/openfoam/files/v2112/OpenFOAM-v2112.tgz
  2106           https://sourceforge.net/projects/openfoam/files/v2106/OpenFOAM-v2106.tgz
  …
  1612           https://sourceforge.net/projects/openfoam/files/v1612+/OpenFOAM-v1612+.tgz

Deprecated versions:
  None
```

Versions of OpenFoam

Retrieving Information about OpenFOAM(2/2)

```
Variants:
  Name [Default]            When    Allowed values    Description
  ======================    ====    ==============    ===================================================

  build_system [generic]    --      generic           Build systems supported by the package
  float32 [off]             --      on, off           Use single-precision
  int64 [off]               --      on, off           With 64-bit labels
  kahip [off]               --      on, off           With kahip decomposition
  knl [off]                 --      on, off           Use KNL compiler settings
  metis [off]               --      on, off           With metis decomposition
  mgridgen [off]            --      on, off           With mgridgen support
  paraview [off]            --      on, off           Build paraview plugins and runtime post-processing
  scotch [on]               --      on, off           With scotch/ptscotch decomposition
  source [on]               --      on, off           Install library/application sources and tutorials
  spdp [off]                --      on, off           Use single/double mixed precision
  vtk [off]                 --      on, off           With VTK runTimePostProcessing
  zoltan [off]              --      on, off           With zoltan renumbering
```

Variants of OpenFoam

```
Build Dependencies:
    adios2  boost  cgal  cmake  fftw-
api  flex  kahip  m4  metis  mpi  paraview  parmgridgen  scotch  vtk  zlib  zoltan

Link Dependencies:
    adios2  boost  cgal  fftw-api  flex  kahip  metis  mpi  paraview  scotch  vtk  zlib  zoltan

Run Dependencies:
    None
```

Dependencies of OpenFoam

Writing a Package recipe using Python DSL

- Each package includes a recipe file written in Python DSL(Domain Specific Language).

```
class Openjpeg(CMakePackage):
    """OpenJPEG is an open-source JPEG 2000 codec written in C language"""

    homepage = "https://github.com/uclouvain/openjpeg"
    url = "https://github.com/uclouvain/openjpeg/archive/v2.3.1.tar.gz"

    version("2.4.0", sha256="8702ba68b442657f11aaeb2b338443ca8d5fb95b0d845757968a7be31ef7f16d")

    variant("codec", default=False, description="Build the CODEC executables")
    depends_on("libpng", when="+codec")

    def url_for_version(self, version):
        if version >= Version("2.1.1"):
            return super().url_for_version(version)
        url_fmt = "https://github.com/uclouvain/openjpeg/archive/version.{0}.tar.gz"
        return url_fmt.format(version)

    def cmake_args(self):
        args = [
            self.define_from_variant("BUILD_CODEC", "codec"),
            self.define("BUILD_MJ2", False),
            self.define("BUILD_THIRDPARTY", False),
        ]
        return args
```

Base Package(CMake Build)

Metadata

Versions

Variants
dependencies

Install logic

OpenFOAM Distribution and Package Management

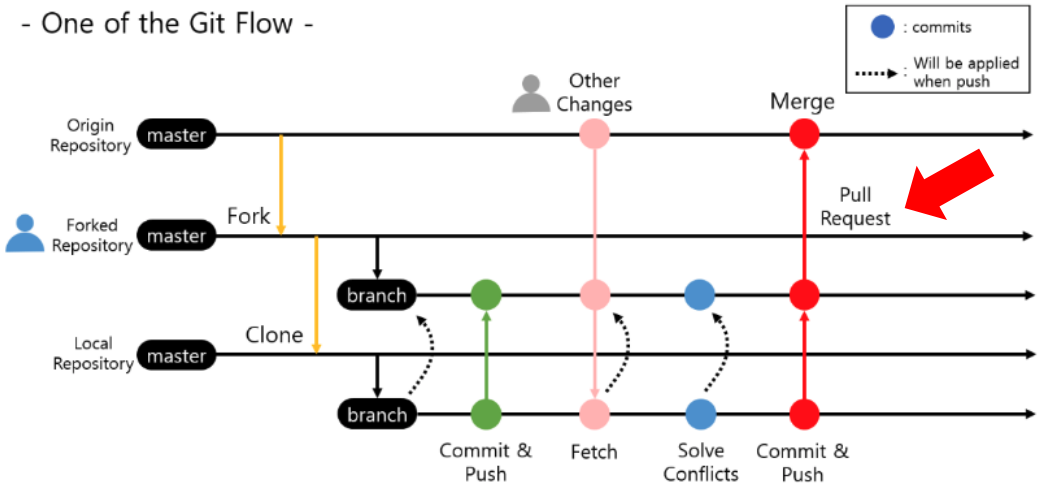
- In May 2017, based on OpenCFD's distribution(Openfoam.com), various other distributions were created.
- While the OpenCFD distribution is actively maintained with maintainers, other distributions have faced challenges in terms of maintenance.
- I have decided to focus on improving packages within the OpenFOAM.org distribution.

Distribution	Maintainers	Version	Dependencies	Variants	Required by
Openfoam.com (OpenCFD)	olsenm(Mark Olesn)	1612~2306	mpi, zlib-api, fftw-api, boost, cgal, flex, cmake, m4, scotch, kahip, metis, parmgridgen, zoltan, vtk, adios2, paraview	build_system, int64, knl, kahip, metis, scotch, zoltan, mgridgen, paraview, vtk, float32, spdp	of-precice, of-catalyst
Openfoam.org (Foundation)	-	2.3.1~10	mpi, zlib-api, flex, cmake, scotch, metis	build_system, int64, float32, source, metis	-
Foam-extend	-	3.0~4.1	Mpi, python, zlib-api, flex, cmake, scotch, metis, parmetis, parmgrdgen, paraview	Build_system, float32, paraview, scotch, ptscotch, metis, parmetis, parmgridgen, source	-

Spack Package PR(Pull Request)

Five examples where I made contributions by submitting Pull Requests directly to the Spack repository.

1. Version URL Function
2. Precision Options
3. Decomposition Method(Zoltan)
4. New Solver
5. Etc.



Pull Request
https://velog.io/@leejpsd/Pull-Request

PR 1 - URL Function 1/2

- Issue: Specifying URLs for each version separately, with special rules for versions 5.0 and below
- Solution: "url_for_version" method to provide standardized URLs for all versions.

```
url = "https://github.com/OpenFOAM/OpenFOAM-4.x/archive/version-4.1.tar.gz"
version(
    "10",
    sha256="59d712ba798ca44b989b6ac50bcb7c534eeccb82bcf961e10ec19fc8d84000cf",
    url=baseurl + "/OpenFOAM-10/archive/version-10.tar.gz",
)
version(
    "9",
    sha256="0c48fb56e2fbb4dd534112811364d3b2dc12106e670a6486b361e4f864b435ee",
    url=baseurl + "/OpenFOAM-9/archive/version-9.tar.gz",
)
...
version(
    "5.0",
    sha256="9057d6a8bb9fa18802881feba215215699065e0b3c5cdd0c0e84cb29c9916c89",
    url=baseurl + "/OpenFOAM-5.x/archive/version-5.0.tar.gz",
)
...
version(
    "2.3.1",
    sha256="2bbcf4d5932397c2087a9b6d7eeee6d2b1350c8ea4f455415f05e7cd94d9e5ba",
    url="http://downloads.sourceforge.net/foam/OpenFOAM-2.3.1.tgz",
)
...
```

Original code

```
def url_for_version(self, version):
    """If the version number is 5.0 or lower, the returned URL includes
    the ".x" suffix in the OpenFOAM directory name to reflect
    the old directory naming convention for these versions.
    """
    if version <= Version("5.0"):
       url= "https://github.com/OpenFOAM/OpenFOAM-{0}.x/archive/version-
{1}.tar.gz"
       return url.format(version.up_to(-1), version)
```

First proposal

PR 1 - URL Function 2/2

- During the code review process, I received negative feedback on the initial implementation.
- Feedback: While the initial implementation worked, there were concerns regarding the completeness of the method. It was suggested that the method should provide a single location to determine "standard" URLs for the package.
- Lesson: This case highlights how even seemingly minor contributions require attention to detail and adherence to project conventions, and that addressing code review feedback is crucial for successful integration into open-source projects.

```
    def url_for_version(self, version):
        """If the version number is 5.0 or lower, the returned URL includes
        the ".x" suffix in the OpenFOAM directory name to reflect
        the old directory naming convention for these versions.
        """
        if version == Version("2.3.1"):
            return "http://downloads.sourceforge.net/foam/OpenFOAM-2.3.1.tgz"
        elif version <= Version("5.0"):
            version_prefix = str(version.up_to(-1)) + ".x"
        else:
            version_prefix = version

        url = "https://github.com/OpenFOAM/OpenFOAM-{}/archive/version-{}.tar.gz".format(
            version_prefix, version
        )
        return url
```

Final proposal

PR 2 – Precision Options 1/2

- Issue
  - In OpenFOAM.com distribution, the 'precision' option is defined as having three possible values, but it is implemented using two boolean variants.
  - The 'LP' option is missing in the OpenFOAM.org distribution.
- Solution
  - New multi-valued variants for the precision option.
  - Add "LP" value of precision variant in the Openfoam.org distribution

	etc/bashrc				**Spack Variants**			
Distribution	SP	DP	Mixed	LDP(LP)	SP	DP	Mixed	LDP(LP)
OpenFoam.com	O	O	O(1906)	X	float32	(default)	spdp	-
OpenFoam.org	O	O	X	O(6.0 `18)	float32	(default)	-	X
Foam-extend	O	O	X	O(3.2 `15)	float32	(default)	-	X

Current Status of OpenFOAM Precision Option

SP: Single Precision
DP: Double Precision
SPDP: Mixed(Single+Double) Precision
LDP, LP: Long Double Precision

PR 2 – Precision Options 2/2

- Lesson: It's essential to maintain a clear code version history that allows us to track when specific options or features were introduced.
- Using the GitHub Blame feature, I have identified the commits that have modified the code.

6 years ago OpenFOAM: Added support for e... 75 # WM_PRECISION_OPTION = SP | DP | LP

OpenFOAM: Added support for extended precision scalar

OpenFOAM can now be compiled with single, double or long double scalars by setting the WM_PRECISION_OPTION environment variable to either SP, DP or LP respectively.

On most 64bit systems long double is stored as 128bit but computed in the floating point hardware to 80bit. Due to the increased storage compared to double precision cache and memory access is significantly more time consuming causing a slow-down of floating point intensive operations by a factor of 2 to 3.

master

version-11 version-10 version-9 version-8 version-7 version-6 20230829 20230707 20230702 2023

```
variant(
    "precision",
    default="dp",
    description="Precision option",
    values=("sp", "dp", conditional("lp", when="@6:")),
    multi=False,
)
```

```
    class OpenfoamOrgArch(OpenfoamArch):
    """An openfoam-org variant of OpenfoamArch"""

    def __init__(self, spec, **kwargs):
        super().__init__(spec, **kwargs)
        if "precision=lp" in spec:
            self.precision_option = "LP"
        elif "precision=sp" in spec:
            self.precision_option = "SP"
        self.update_options()
```

Final proposal

PR 3 – Decomposition Methods 1/2

- Issue: Zoltan package serves two main functions: "renumber" and "decomposition." The "renumber" functionality has been available since the initial version, starting from version 2.3.1. However, it wasn't initially included in Spack. On the other hand, the "decomposition" functionality was introduced in version 10 and later.
- Solution: To register the Zoltan package as a dependency and add it to the Variants.

Distribution	**Third Party**				**Spack Variants**			
	Metis	Scotch	Kahip	Zoltan	Metis	Scotch	Kahip	Zoltan
OpenFoam.com	O	O	O(1712)	X*	O	O	O	O*
OpenFoam.org	O	O	X	O(10 '22)	O	X**	-	X
Foam-extend	O	O	X	X***	O	O	-	X

Current Status of OpenFOAM Third Party Decomposition Methods

* Variants are currently applied only for renumbering purposes.
** There is no option to select it as a variant; it is a mandatory dependency.
*** It requires installation but I'm unsure of its purpose.

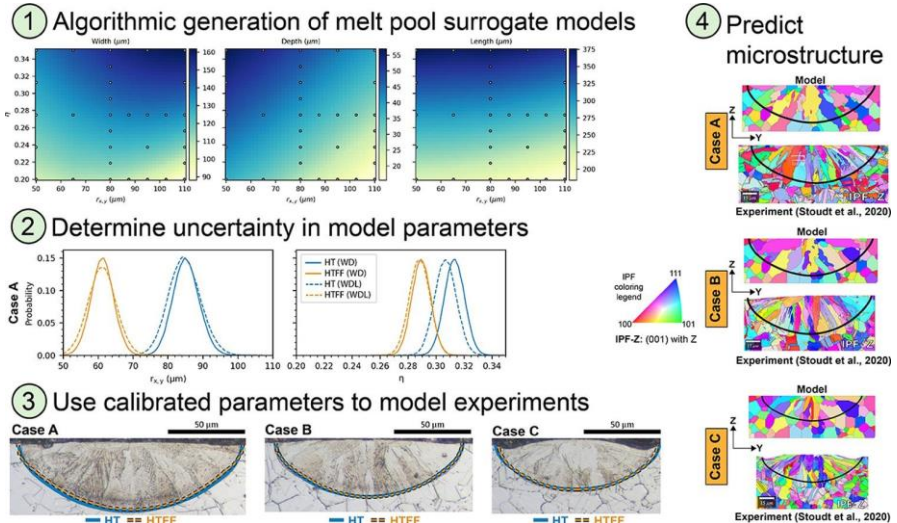
PR 3 – Decomposition Methods 2/2

- Lesson : I have looked into how to install packages that are specified as dependencies in the Spack and made the necessary variants and patches.

```
variant("zoltan", default=False, description="Enable Zoltan renumbering and decomposition")
depends_on("zoltan+shared", when="+zoltan")

...
def patch
...
    if self.spec.satisfies("@10:") and "+zoltan" in self.spec:
        filter_file("libzoltan.a", "libzoltan.so", join_path("src", "renumber", "Allwmake"))
        filter_file(
            "libzoltan.a",
            "libzoltan.so",
            join_path("src", "parallel", "decompose", "Allwmake"),
        )

def configure
...
    if "+zoltan" in spec:
        if spec.satisfies("@:9"):
            self.etc_prefs["ZOLTAN_ARCH_PATH"] = spec["zoltan"].prefix
            self.etc_prefs["ZOLTAN_VERSION"] = "Zoltan-{0}".format(spec["zoltan"].version)
        else:
            self.etc_config["zoltan"] = {
                "ZOLTAN_ARCH_PATH": spec["zoltan"].prefix,
                "ZOLTAN_VERSION": "Zoltan-{0}".format(spec["zoltan"].version),
            }
```

PR 4 – New Solver 2/2

- Issue:
  - Adding New Solver Package "AdditiveFoam" with OpenFOAM-Org Dependency
  - The package was implemented using Python's os.system library instead of the Spack Package DSL.
- Package Overview: AdditiveFOAM is a heat and mass transfer software for Additive Manufacturing (AM) released by Oak Ridge National Laboratory. https://github.com/ORNL/AdditiveFOAM
- Solution: The package has been refactored to maximize the use of the Spack Package DSL approach.



[3]

PR 4 – New Solver 2/2

```
class Additivefoam(Package):
    …

    depends_on("openfoam-org@10")

    def install(self, spec, prefix):
             # Clone repo to prefix
        mkdirp(prefix)
        with working_dir(prefix):
            os.system(f"git clone {self.git}")

        # Build movingHeatSource
        with working_dir(
            join_path(
                prefix,
                "AdditiveFOAM",
                "applications",
                "solvers",
                "additiveFoam",
                "movingHeatSource",
            )
        ):
            os.system("wmake libso")

        # Build additiveFoam
        with working_dir(
            join_path(prefix, "AdditiveFOAM", "applications", "solvers", "add
itiveFoam")
        ):
            os.system("wmake")
```

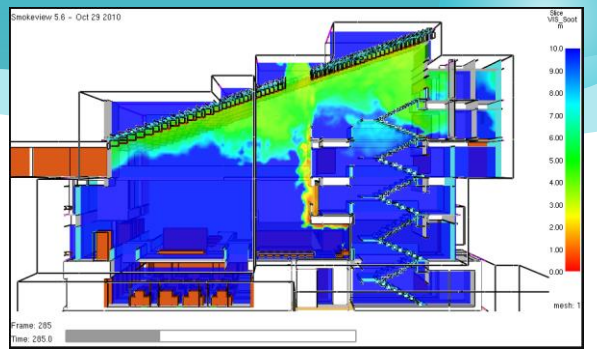
First proposal

```
class Additivefoam(Package):
    …

    depends_on("openfoam-org@10")

    common = ["spack-derived-Allwmake"]
    assets = ["applications/Allwmake", "Allwmake"]

    build_script = "./spack-derived-Allwmake"

    phases = ["configure", "build", "install"]

    def patch(self):
        add_extra_files(self, self.common, self.assets)
    …
    def build(self, spec, prefix):
        args = []
        if self.parallel:  # Parallel build? - pass via environment
            os.environ["WM_NCOMPPROCS"] = str(make_jobs)
        builder = Executable(self.build_script)
        builder(*args)

    def install(self, spec, prefix):
        for f in ["README.md", "LICENSE"]:
            if os.path.isfile(f):
                install(f, join_path(self.prefix, f))

        dirs = ["tutorials", "applications"]
        for d in dirs:
            if os.path.isdir(d):
                install_tree(d, join_path(self.prefix, d), symlinks=True)
```

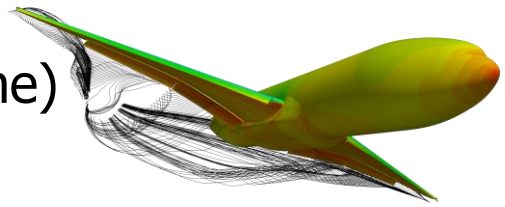
Final proposal

PR 5. Other Contributions

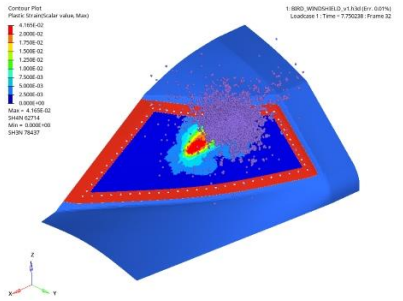
- FDS
  - CFD(LES) code for low-speed flows, Fortran
  - Lesson: The first experience of creating package
  - Maintainer
- SU2
  - CFD, Aerodynamic Optimization, C++, Python
  - Lesson: Meson Build System(Less configuration time, Less build time)
  - Maintainer
- OpenRadioss(Starter, Engine)
  - Explicit structure dynamics code, Fortran
  - Lesson: One Repository, but two spack packages
  - Maintainer



https://pages.nist.gov/fds-smv/
https://fdstutorial.com/what-is-fds/



https://su2code.github.io



https://www.openradioss.org

Finally

- I volunteered to become the maintainer for OpenFOAM-org. This decision was a pivotal moment in my involvement with the project, as it allowed me to take on a central role in overseeing and ensuring the quality of the OpenFOAM-org package within the Spack ecosystem.
- More detailed information on today's presentation can also be found on my blog(https://kjrstory.netlify.app/).

Reference

- [1] archspec: A library for detecting, labeling, and reasoning about microarchitectures.(2020).
- [2] Managing HPC Software Complexity with Spack. RADIUSS Tutorial Series 2023. Virtual event. August 8-9, 2023.
- [3] Knapp et al. "Calibrating uncertain parameters in melt pool simulations of additive manufacturing", Comp. Mat. Sci. (2023) 111904