

New Software Environment

What and why?

In order to make use of the performance advantages of modern CPUs, applications and all their dependencies must be compiled for the CPU in question.

Doing this in a consistent manner across multiple CPU types is a non-trivial problem and, as such, the current stack is built for a lowest common denominator architecture which will work everywhere. This has the disadvantage of not taking advantage of the previous 10 years of technological improvements: the code will be as fast on a modern Xeon CPU as it will on a CPU from 2008.

The DCSR are currently putting in place a new software stack to resolve these issues which uses the Spack (**S**cientific **pack**age manager) framework: <https://spack.readthedocs.io/en/latest/> and <https://github.com/spack/spack>.

The new stack will be available on an optional basis during 2020 and will become the default stack in 2021.

If you ask for a new package to be installed then it may well be provided in the new environment.

The new environment also makes use of the Lmod (**L**ua **mod**ule) system <https://lmod.readthedocs.io/en/latest/> which offers a number of advantages over the classical module tool.

There will be a new release of our stack each year with it being put in place during the annual January downtime.

Each release fixes certain components including:

- The base compiler versions - e.g. GCC 8.3.0 and Intel 19.1.106
- The MPI flavours and versions - e.g. MPICH 3.3.2 and Intel MPI 19.1.106
- The base linear algebra (BLAS) flavours and versions

Installing new packages or versions during the year is always possible as long as they are compatible with the fixed components.

Using the environment

Please be aware that the new environment is under active development and that the instructions might change without warning - if something doesn't work please re-read the documentation before asking for help.

Finding and loading modules

In order to use the new environment you need to source a script -

`/dcsrsoft/spack/bin/setup_dcsrsoft` - which will load LMod and set up the correct paths:

```
$ source /dcsrsoft/spack/bin/setup_dcsrsoft
Activating LMod with: /dcsrsoft/spack/meleze/v1/share/spack/lmod/S6g1-IB/linux-centos7-
x86_64/Core

$ module avail

----- /dcsrsoft/spack/meleze/v1/share/spack/lmod/S6g1-IB/linux-centos7-x86_64/Core -----
gcc/8.3.0      intel/19.1.0.166
```

In order to see more modules one needs to load their dependencies. If we load a compiler we see:

```
$ module load gcc
$ module avail

----- /dcsrsoft/spack/meleze/v1/share/spack/lmod/S6g1-IB/linux-centos7-x86_64/gcc/8.3.0
-----
fftw/3.3.8      mpich/3.3.2      openblas/0.3.8      python/2.7.16      python/3.7.6 (D)      r/3.6.2

----- /dcsrsoft/spack/meleze/v1/share/spack/lmod/S6g1-IB/linux-centos7-x86_64/Core
-----
gcc/8.3.0 (L)      intel/19.1.0.166

Where:
L:  Module is loaded
D:  Default Module
```

If you don't know where something is or what dependencies are needed then try `module spider`

```
$ module spider gromacs
```

```
-----  
--  
gromacs: gromacs/2020-mpi-openmp  
-----  
--
```

You will need to load all module(s) on any one of the lines below before the "gromacs/2020-mpi-openmp" module is available to load.

```
gcc/8.3.0 mpich/3.3.2
```

Use in batch scripts

There are a few subtle changes needed to make use of the new environment in batch scripts

```
#!/bin/bash -l  
  
#SBATCH -N 1  
#SBATCH -n 1  
#SBATCH -c 4  
  
source /dcsrcsoft/spack/bin/setup_dcsrcsoft  
  
module load gcc  
module load python  
  
python mycode.py
```

As seen before we need to activate the new environment but the big difference is to tell the script to behave as if it was in an interactive session with `#!/bin/bash -l`

Revision #9

Created 5 March 2020 09:09:06 by Ewan Roche

Updated 3 February 2021 11:48:23 by Etienne Orliac