Software at HPC Systems

National Supercomputer Centre (NSC), Linköping University

SNIC-PRACE training

Online @NSC 20th Apr 2021, 10:00 - ca. 15:00

LINKÖPINGS
UNIVERSITET

List of Animals (or Software)

- Those belonging to the emperor
- Embalmed ones
- Trained ones
- Suckling pigs
- Mermaids (or sirens)
- Fabled ones
- Stray dogs
- Those included in this classification
- …

Jorge Luis Borges, “The Analytical Language of John Wilkins”

https://en.wikipedia.org/wiki/Celestial_Emporium_of_Benevolent_Knowledge

Software at HPC Systems

- OS provided (Linux)
- Globally installed, provided as a **module**
- Locally, for compute project or user
- With different types of **licenses**
- Installed within a **container** (solution?)
- For *parallel* or *serial* calculations
- F77 program last updated in the ‘90s
- Nightly developer build pulled from git-repository
- …

Software at HPC Systems

- OS provided (Linux CentOS 7) - system default
- Globally installed, provided as a **module** - curated, different levels of support
- Locally, for compute project or user
- With different types of **licenses** - restrictions might apply
- Installed within a **container** (Singularity)
- For *parallel* or *serial* calculations
- F77 program last updated in the ‘90s
- Nightly developer build pulled from git-repository
- …

- difference in challenge to get it working

Where to Start?

- Suitable for my research, license?
- Is HPC needed?
- Is it already available at HPC center / expertise?
- Don’t forget utility software (plotting, analysis)!

How do I get Code X?

1. Check installed software webpage
2. Check module system (`module avail`)
3. Ask NSC support
4. Build and install yourself

Special wrappers/rec. e.g.: Gaussian

NSC software installation policy:

- Users encouraged to install in /home or /proj
- NSC can help to install on request

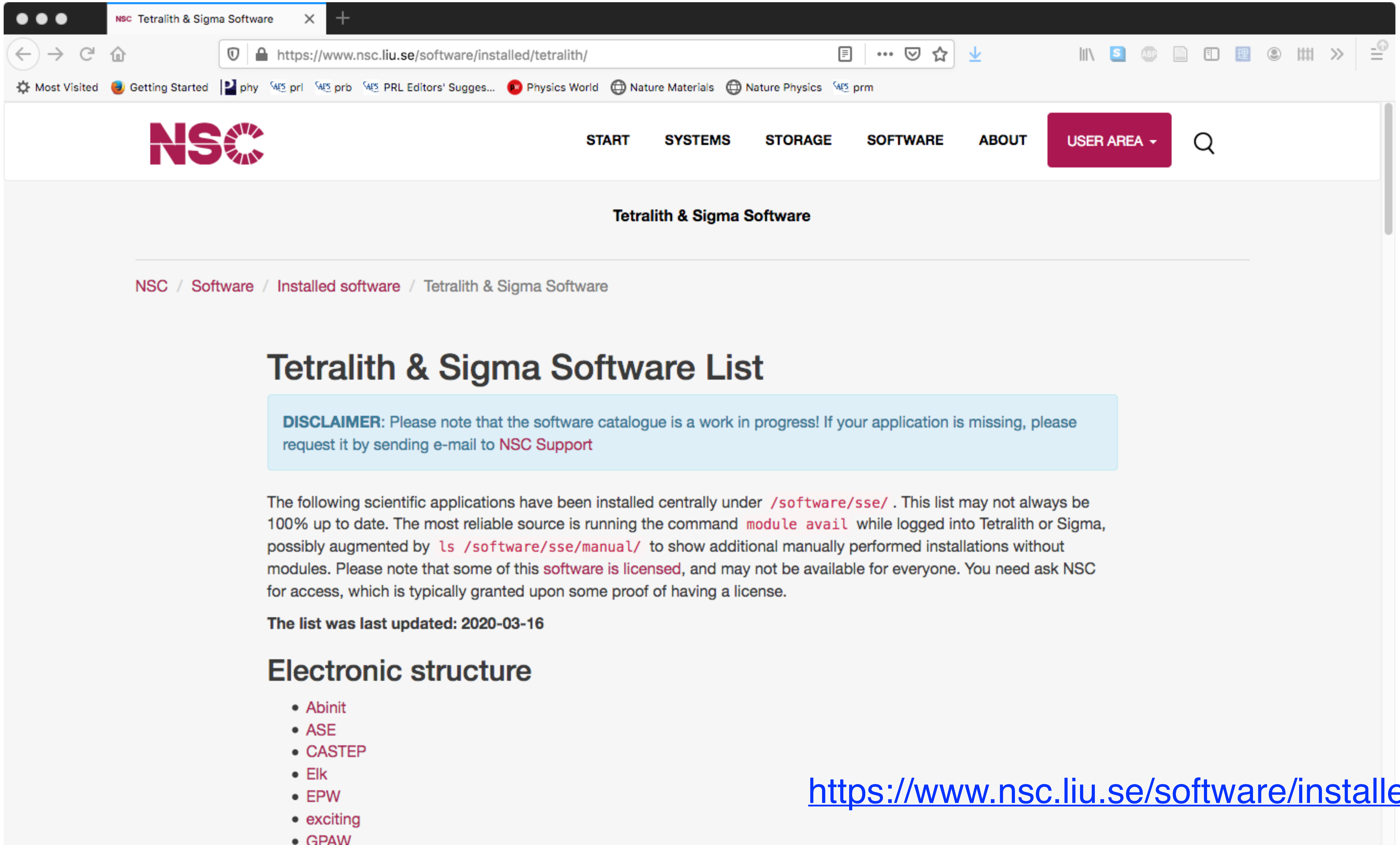
Testing, benchmarking, optimization

Global installation: wide or not usage, license?

https://www.nsc.liu.se/software/installed/tetralith/

https://www.nsc.liu.se/software/installation-policy/

Installation Webpage



Tetralith & Sigma Software

NSC / Software / Installed software / Tetralith & Sigma Software

Tetralith & Sigma Software List

DISCLAIMER: Please note that the software catalogue is a work in progress! If your application is missing, please request it by sending e-mail to NSC Support

The following scientific applications have been installed centrally under `/software/sse/`. This list may not always be 100% up to date. The most reliable source is running the command `module avail` while logged into Tetralith or Sigma, possibly augmented by `ls /software/sse/manual/` to show additional manually performed installations without modules. Please note that some of this software is licensed, and may not be available for everyone. You need ask NSC for access, which is typically granted upon some proof of having a license.

The list was last updated: 2020-03-16

Electronic structure

- Abinit
- ASE
- CASTEP
- Elk
- EPW
- exciting
- GPAW

https://www.nsc.liu.se/software/installed/tetralith/

Module System

```
[weiol@tetralith1 ~]$ module avail

-------------------------------------------------------------- /software/sse/modules --------------------------------------
   abaqus/2020-nsc1                                     libxc/recommendation                                          (D)
   ABINIT/recommendation                         (D)    libxc/4.2.1-nsc1-intel-2018a-eb
   ABINIT/8.8.2-nsc1-intel-2018a-eb                     LLVM/recommendation                                           (D)
   ABINIT/8.8.2-nsc2-intel-2018a                        LLVM/3.7.1-tc1-gcc-2018a-eb
   ABINIT/8.10.3-nsc1-intel-2018a                       LLVM/4.0.1-tc1-intel-2018a-eb
   allinea-DDT/recommendation                    (D)    LLVM/5.0.1-tc1-gcc-2018a-eb
   allinea-DDT/18.2.1                                   LSDalton/recommendation                                       (D)
   allinea-DDT/19.0.2                                   LSDalton/1.0-nsc1-intel-2018a-eb
   allinea-DDT/20.0.3                                   mathematica/recommendation                                    (D)
   allinea-DDT/20.1                                     mathematica/12.0.0-nsc1
   allinea-DDT/20.2                                     MATLAB/R2018a-nsc1
   allinea-DDT/20.2.1                                   MATLAB/R2018b-nsc1
   allinea-forge/recommendation                  (D)    MATLAB/R2019a-nsc1
   allinea-forge/18.2.1                                 MATLAB/R2019b_Update_3-nsc1
   allinea-forge/19.0.2                                 MATLAB/R2020a-nsc1
   allinea-forge/20.0.3                                 MATLAB/R2020b-nsc1
   allinea-forge/20.1                                   MATLAB/recommendation                                         (D)
   allinea-forge/20.2                                   mc/4.8.21
   allinea-forge/20.2.1                                 Molden/recommendation                                         (D)
   allinea-MAP/recommendation                    (D)    Molden/5.8-nsc1-intel-2018a-eb
   allinea-MAP/18.2.1                                   Molden/5.8-nsc1-system
   allinea-MAP/19.0.2                                   mpprun/3.0
   allinea-MAP/20.0.3                                   mpprun/4.0
   allinea-MAP/20.1                                     mpprun/4.1.2
```

…very long list which continues…

```
$ module avail | less
```

https://www.nsc.liu.se/software/modules/

Module System: Commands

`module help …`	Show information for module …
`module avail`	List available modules
`module avail …`	Search after module containing … in its name
`module add …`	Add a module (same as module load …)
`module list`	List your loaded modules
`module rm …`	Remove the … module
`module purge`	Remove all loaded modules (useful to start “clean”)

https://www.nsc.liu.se/software/modules/

Module System @NSC

NSC module usage:

- **Only** load specific software module (not dependencies)

  at many other centers, **must** load all dependencies

- **Only** load build environment when building

  gives access to specific build time modules

“clean up” with `$ module purge`

https://www.nsc.liu.se/software/modules/

Note: difference with most HPC centers

Module System: Search

- An example, searching for “vasp”:

```
[weiol@tetralith1 ~]$ module avail vasp

------------------------------------------------------- /software/sse/modules -------------------------------------------------------
   p4vasp/recommendation                                             (D)    VASP/5.4.4.16052018-nsc1-intel-2018b-eb
   p4vasp/tmp1                                                              VASP/5.4.4.16052018-nsc2-intel-2018a-eb
   p4vasp/0.3.30-nsc1                                                       VASP/5.4.4.16052018-vanilla-nsc1-intel-2018a-eb
   VASP-OMC/5.4.4.16052018-nsc1-intel-2018a-eb                              VASP/5.4.4.16052018-wannier90-nsc1-intel-2018a-eb
   VASP-VTST/3.2-sol-5.4.4.16052018-nsc2-intel-2018a-eb                     VASP/6.1.0.28012020-nsc1-intel-2018a-eb
   VASP-VTST/3.2-sol-5.4.4.16052018-vanilla-nsc1-intel-2018a-eb (D)    VASP/6.1.2.25082020-nsc1-intel-2018a-eb
   VASP/recommendation                                               (D)    VASP/6.1.2.25082020-omp-nsc1-intel-2018a-eb
   VASP/5.4.4.16052018-nsc1-intel-2018a-eb                                  vasptools/0.3

  Where:
   D:  Default Module

Use "module spider" to find all possible modules.
Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".


[weiol@tetralith1 ~]$
```

https://www.nsc.liu.se/software/modules/

Compilers and Libraries

NSC recommendation: to compile your own software, load a build environment

- Compilers
  - Intel: icc, ifort
  - Gcc: gcc, gfortran
- MPI libraries
  - Intel (impi), OpenMPI
- Math libraries
  - e.g. MKL
- Build environments
  - e.g. `buildenv-intel/2018a-eb`

https://www.nsc.liu.se/software/index.html

Build Environment

What buildenvs are available @NSC?

```
[weiol@tetralith1 ~]$ module avail buildenv

----------------------------------------------- /software/sse/modules -----------------------------------------------
   buildenv-gcc/recommendation      (D)    buildenv-intel/recommendation                (D)
   buildenv-gcc/7.3.0-bare                 buildenv-intel/2015.1.133-impi-2018.1.163-eb
   buildenv-gcc/2016b-eb                   buildenv-intel/2016b-eb
   buildenv-gcc/2018a-eb                   buildenv-intel/2017.u7-bare
   buildenv-gcccuda/recommendation  (D)    buildenv-intel/2018a-eb
   buildenv-gcccuda/10.2-7.3.0-bare        buildenv-intel/2018b-eb
   buildenv-impi-gcc/recommendation (D)    buildenv-intel/2018.u1-bare
   buildenv-impi-gcc/2018a-eb

  Where:
   D:  Default Module

Use "module spider" to find all possible modules.
Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".


[weiol@tetralith1 ~]$
```

https://www.nsc.liu.se/software/modules/

Build Environment

(1) add a buildenv

```
[weiol@tetralith1 ~]$ module add buildenv-intel/2018a-eb
***************************************************
You have loaded an intel buildenv module
***************************************************
The buldenv-intel module makes available:
 - Compilers: icc, ifort, etc.
 - Mpi library with mpi-wrapped compilers: intel mpi with mpiicc, mpiifort, etc.
 - Numerical libraries: intel MKL

It also makes a set of dependency library modules available via
the regular module command. Just do:
  module avail
to see what is available.

NOTE: You shoud never load build environments inside submitted jobs.
(with the single exception of when using supercomputer time to compile code.)
```

(2) what's included?

```
[weiol@tetralith1 ~]$ module list

Currently Loaded Modules:
  1) mpprun/4.1.3                        5) buildtool-easybuild/4.3.0-nscde3532a       9) ifort/.2018.1.163-GCC-6.4.0-2.28 (H)   13)
buildenv-intel/2018a-eb
  2) nsc/.1.1                   (H,S)    6) GCCcore/6.4.0                             10) impi/.2018.1.163                 (H)
  3) EasyBuild/4.3.0-nscde3532a          7) binutils/.2.28                       (H)  11) imkl/.2018.1.163                 (H)
  4) nsc-eb-scripts/1.2                  8) icc/.2018.1.163-GCC-6.4.0-2.28       (H)  12) intel/2018a

  Where:
   S:  Module is Sticky, requires --force to unload or purge
   H:             Hidden Module


[weiol@tetralith1 ~]$
```

Build Environment: Dependencies

Buildenv → Many dependencies/libraries becomes available `$ module avail`

(1) tcl available?

```
[weiol@tetralith1 ~]$ module avail tcl
No modules found!
Use "module spider" to find all possible modules.
Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".
```

(2) load buildenv

```
[weiol@tetralith1 ~]$ module add buildenv-intel/2018a-eb

. . .
```

(3) check for tcl modules

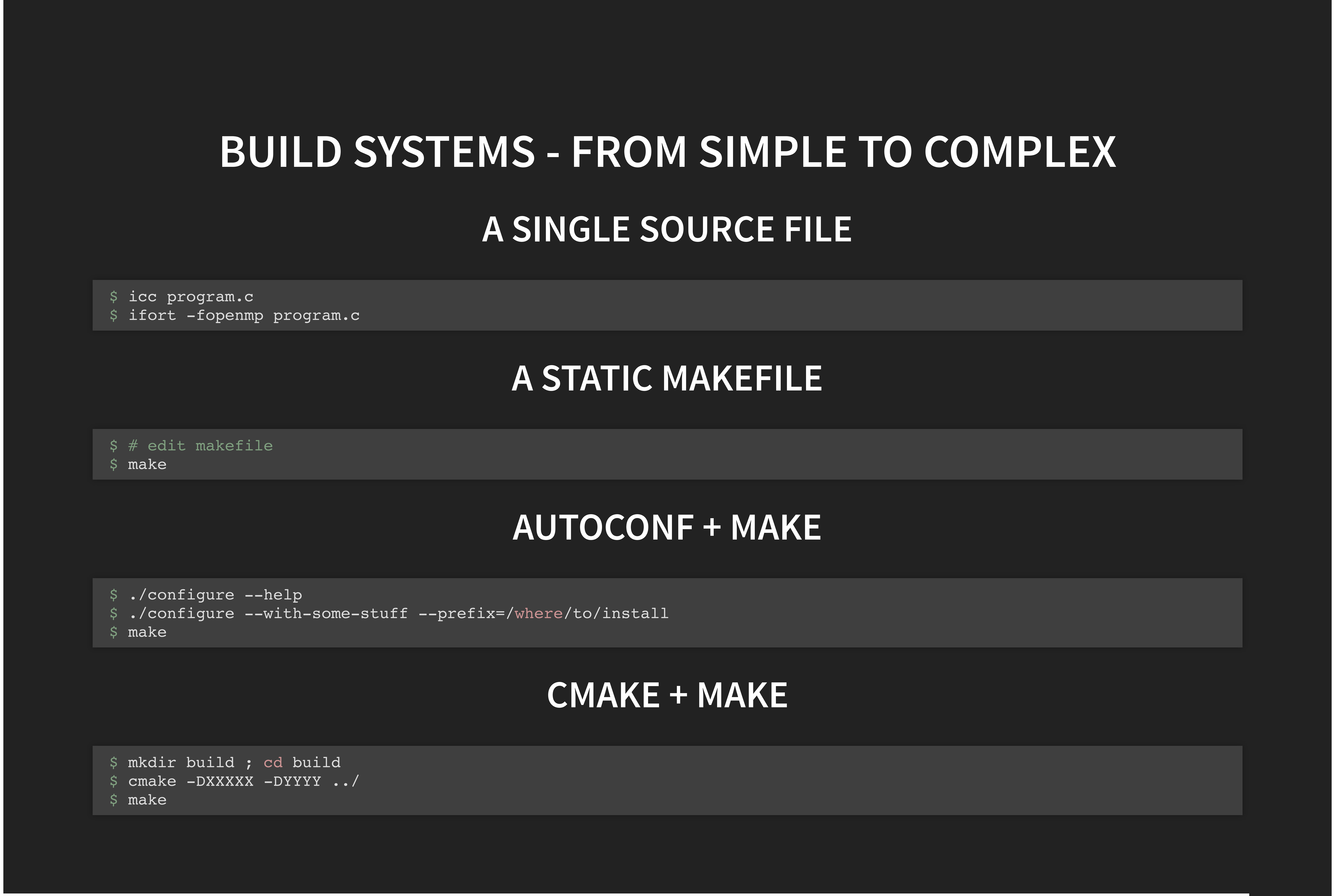
```
[weiol@tetralith1 ~]$ module avail tcl

-------- /software/sse/easybuild/prefix/modules/all/MPI/intel/2018.1.163-GCC-6.4.0-2.28/impi/2018.1.163 ----------------
   Tcl/8.6.5-nsc1

-------- /software/sse/easybuild/prefix/modules/all/Compiler/GCCcore/6.4.0 ---------------------------------------------
   Tcl/8.6.8-nsc1    Tcl/8.6.10-nsc1

Use "module spider" to find all possible modules.
Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".


[weiol@tetralith1 ~]$
```



BUILD SYSTEMS - FROM SIMPLE TO COMPLEX

A SINGLE SOURCE FILE

```
$ icc program.c
$ ifort -fopenmp program.c
```

A STATIC MAKEFILE

```
$ # edit makefile
$ make
```

AUTOCONF + MAKE

```
$ ./configure --help
$ ./configure --with-some-stuff --prefix=/where/to/install
$ make
```

CMAKE + MAKE

```
$ mkdir build ; cd build
$ cmake -DXXXXX -DYYYY ../
$ make
```

https://www.nsc.liu.se/support/past-events/Tetralith_training2018/tetralith_training_2018-11-28.pdf

Common Ecosystems & Frameworks

- Python - prog. language
- Conda - packet manager (first for python, Anaconda)
- R - statistical computing and graphics
- Matlab - commercial prog. language (often University license available, alternative: Octave)

Python

- Python 2.7 (end of support), Python 3.9 (latest)
- Very widely used in Academia/Industry
- Lots of libraries and packages
- Many & increasing number of research software
- Recommendations @NSC:
    - Load module & use virtual environments
    - Avoid default install in `~/.local`

https://www.nsc.liu.se/software/python/

Python

- be aware of system python!

```
[weiol@tetralith1 ~]$ which python
/usr/bin/python

[weiol@tetralith1 ~]$ python --version
Python 2.7.5

[weiol@tetralith1 ~]$ module avail python

----------------------------------------------- /software/sse/modules ------------------------------------------------
   GaussSum/3.0.2-Python-3.6-nsc1                           Python/2.7.15-anaconda-5.3.0-extras-nsc1
   GDAL/2.3.2-Python-2.7.14-nsc1-intel-2018a-eb             Python/2.7.15-env-nsc1-gcc-2018a-eb
   OpenBabel/2.4.1-Python-2.7.14-nsc1-intel-2018a-eb        Python/3.6.3-anaconda-5.0.1-nsc1
   phonopy/1.13.0.64-Python-2.7-nsc2                        Python/3.6.4-nsc1-intel-2018a-eb
   pymatgen/2019.5.28-Python-3.6.3-nsc1                     Python/3.6.4-nsc2-intel-2018a-eb
   Python/recommendation                             (D)    Python/3.6.7-env-nsc1-gcc-2018a-eb
   Python/2.7.14-anaconda-5.0.1-nsc1                        Python/3.7.0-anaconda-5.3.0-extras-nsc1
   Python/2.7.14-nsc1-gcc-2018a-eb                          Python/3.8.3-anaconda-2020.07-extras-nsc1
   Python/2.7.14-nsc1-intel-2018a-eb                        Vim/8.1-Python-2.7.12-nsc1-gcc-2018a-eb

  Where:
   D:  Default Module

Use "module spider" to find all possible modules.
Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".


[weiol@tetralith1 ~]$
```

https://www.nsc.liu.se/software/python/

Containers

- Docker, Singularity, Shifter
- Run program inside its own environment (OS image)
- Complex dependencies (OS packages, Python, …)
- Sometimes *“easiest way”* to run
- Serial calculations fine / parallel, with some work
- **@HPC centers**: restrictions due to security

➡ **@NSC**: using **Singularity**

- docker images ok

https://www.nsc.liu.se/support/singularity/

Caution

- Even center installed software can have bugs
- Bugs or special quirks can be expected in software
- Some testing always necessary