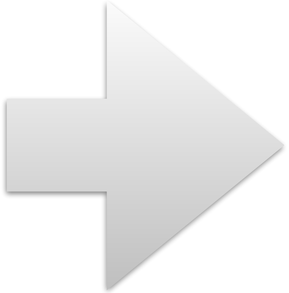
NSC introduction to Tetralith/Sigma

National Supercomputer Centre (NSC), Linköping University

SNIC training, online @NSC 26th Apr 2022, 10:00 - ca. 12:00

LINKÖPINGS UNIVERSITET

Information / Schedule



https://www.nsc.liu.se/support/Events/NSC_intro_Apr2022/

- this presentation as .pdf
- everything underlined is a link

10:00 Introduction to Tetralith/Sigma (Weine Olovsson)

~10:45 Using Python (Hamish Struthers)

~11:00 Using GPUs (Torben Rasmussen)

~11:15 Open session, questions?

National Supercomputer Centre (NSC)

NSC is part of:

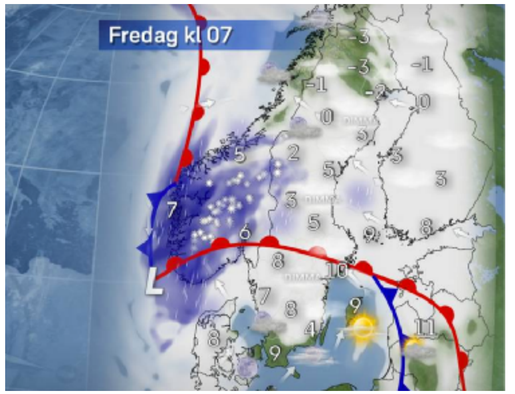
- SNIC Swedish National Infrastructure for Computing (10 Univ.)
- LINKÖPING UNIVERSITY liu.se

1983 - SAAB buys Cray1

1989 - NSC first supercomputer centre in Sweden / SAAB partner

1996 - SMHI partner

2016 - MET Norway partner



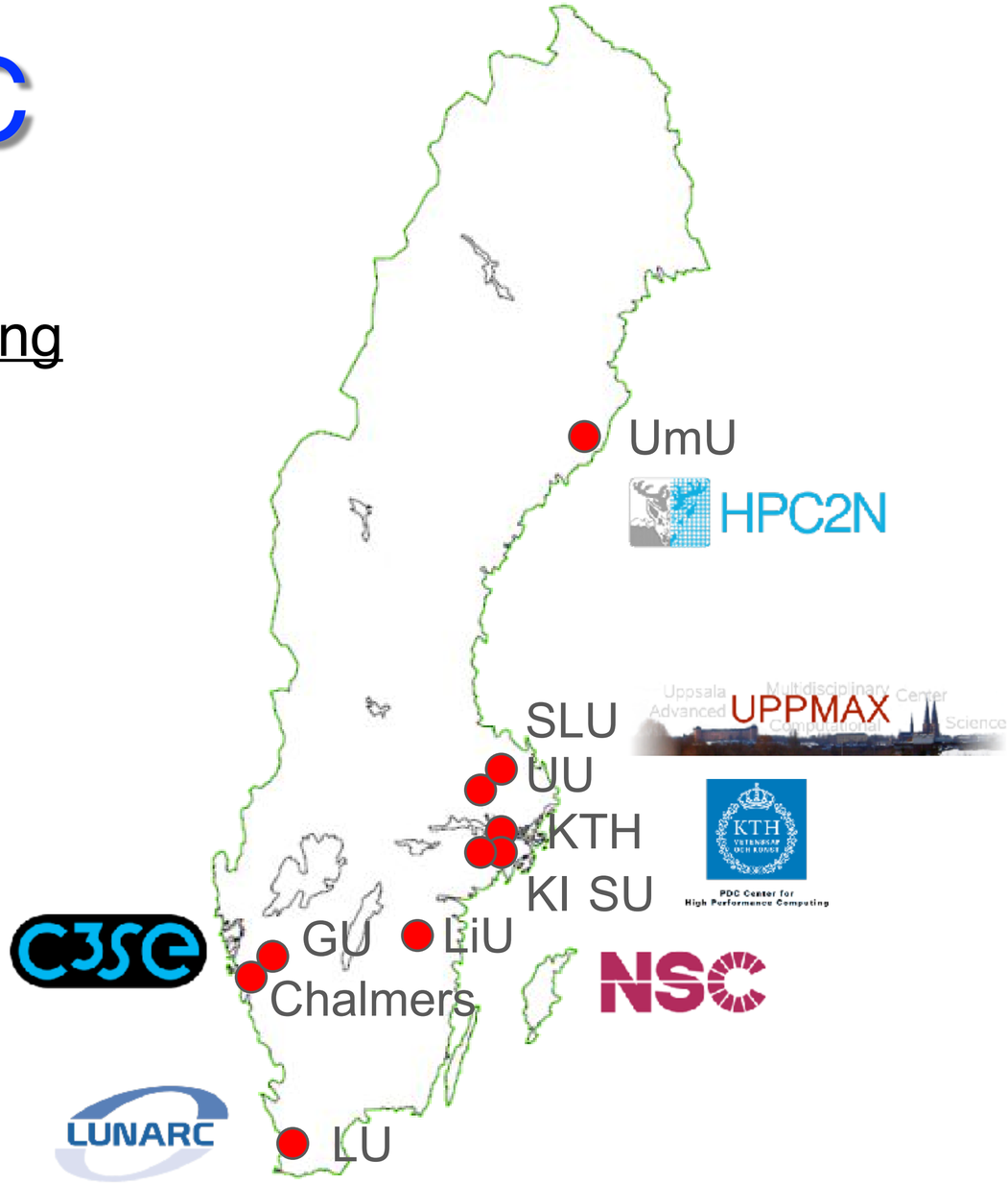
https://www.nsc.liu.se

SNIC

Swedish National Infrastructure for Computing

10 universities & 6 HPC centers:

Chalmers	C3SE
Göteborg	
Karolinska	
KTH	PDC
Linköping	NSC
Lund	LUNARC
SLU	
Stockholm	
Umeå	HPC2N
Uppsala	UPPMAX



https://snic.se/

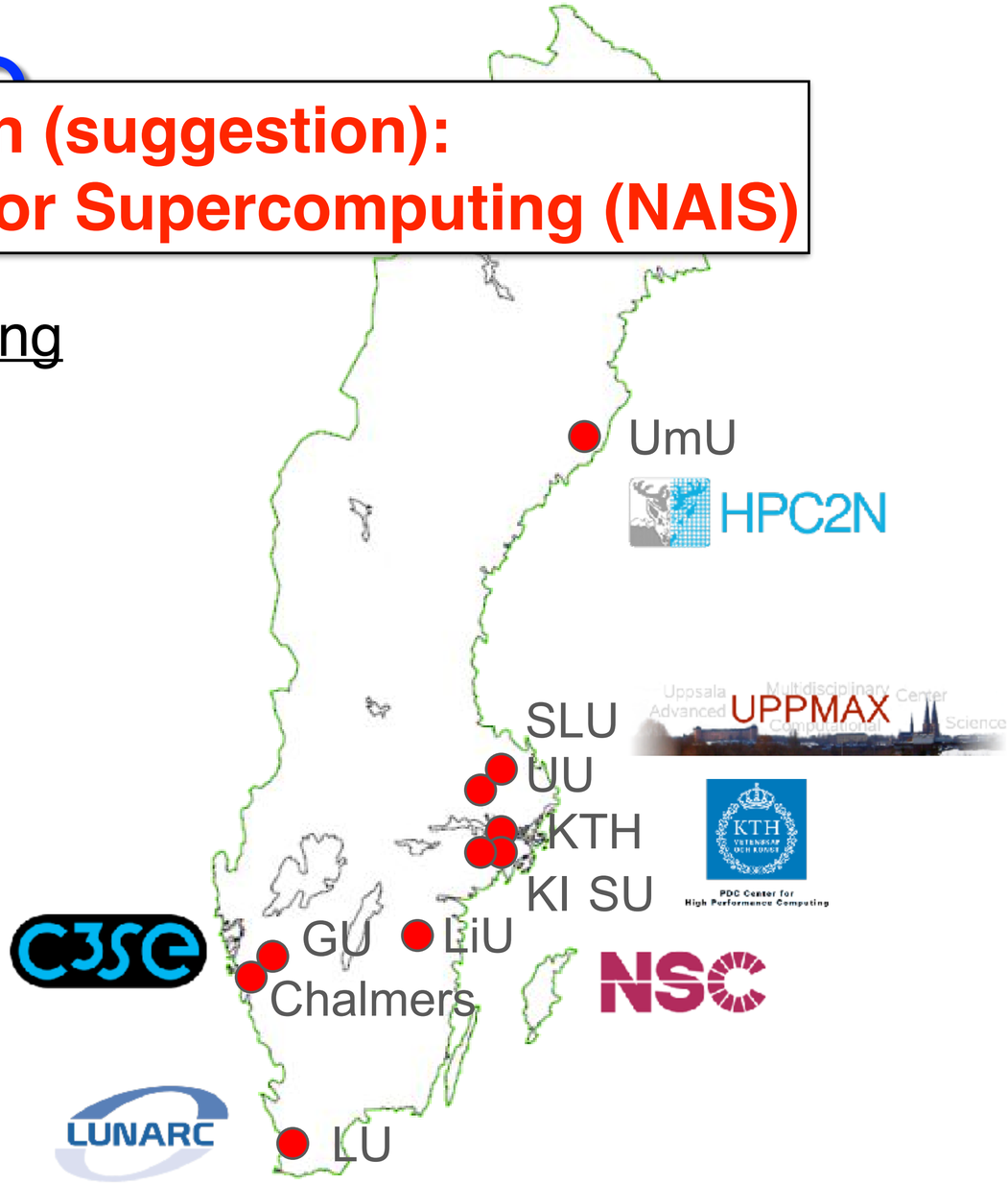
Funding: 10 univ. + Vetenskapsrådet (VR)

2023 - New organization (suggestion):
National Academic Infrastructure for Supercomputing (NAIS)

Swedish National Infrastructure for Computing

10 universities & 6 HPC centers:

Chalmers	C3SE
Göteborg	
Karolinska	
KTH	PDC
Linköping	NSC
Lund	LUNARC
SLU	
Stockholm	
Umeå	HPC2N
Uppsala	UPPMAX



https://snic.se/

Funding: 10 univ. + Vetenskapsrådet (VR)

NSC: Quick Overview

Current Director: Björn Alling, Nov 2021 -

~ **40** people (not all full-time)

Mostly **system experts** and **application experts**

- Provide computational resources
- Software installation (global / local)
- Troubleshooting / advice
- Training (SNIC, local and other)

NSC Academic Clusters

32 cores/node

Tetralith (2018 -) 1908 x 2 x 16 cores, Intel Xeon Gold 6130

SNIC (2020 -) 170 x T4 GPU-nodes Top500 no. 168 (74)

Sigma (2018 -) 110 x 2 x 16 cores, Intel Xeon Gold 6130 *"same" as Tetralith*

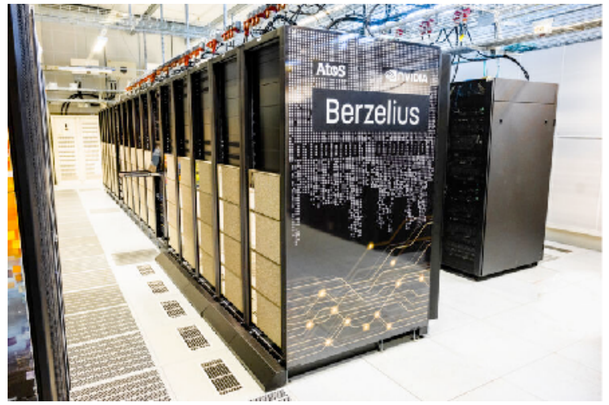
(2020 -) 2 x V100 GPU-nodes

LINKÖPING UNIVERSITY

BerzeLiUs (2021 -) Nvidia DGX SuperPOD, 60 x 8 A100 **GPUs**

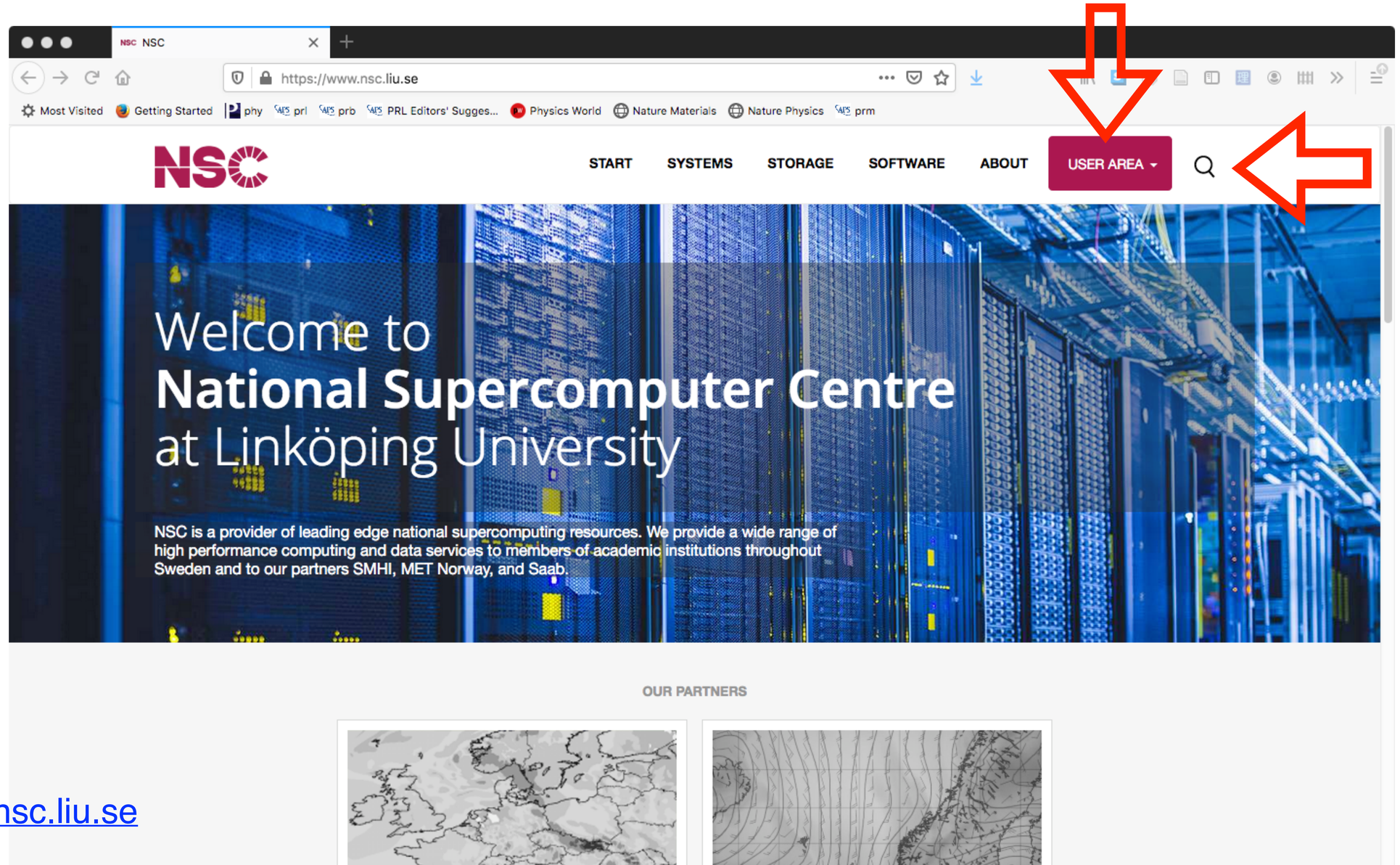
LINKÖPING UNIVERSITY

Knut and Alice Wallenberg Foundation



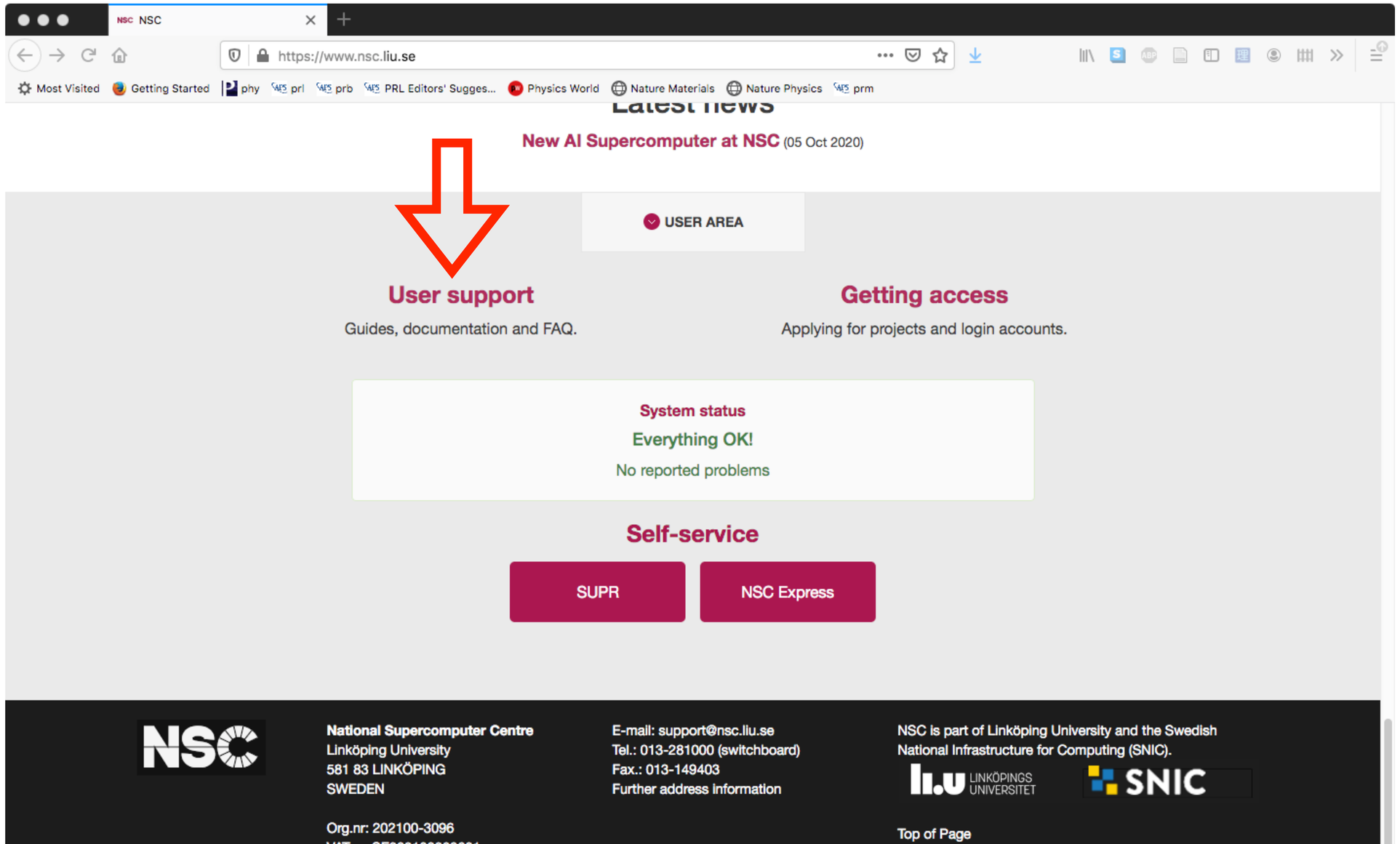
Top500 no. 94 (83)

Where to find Information?

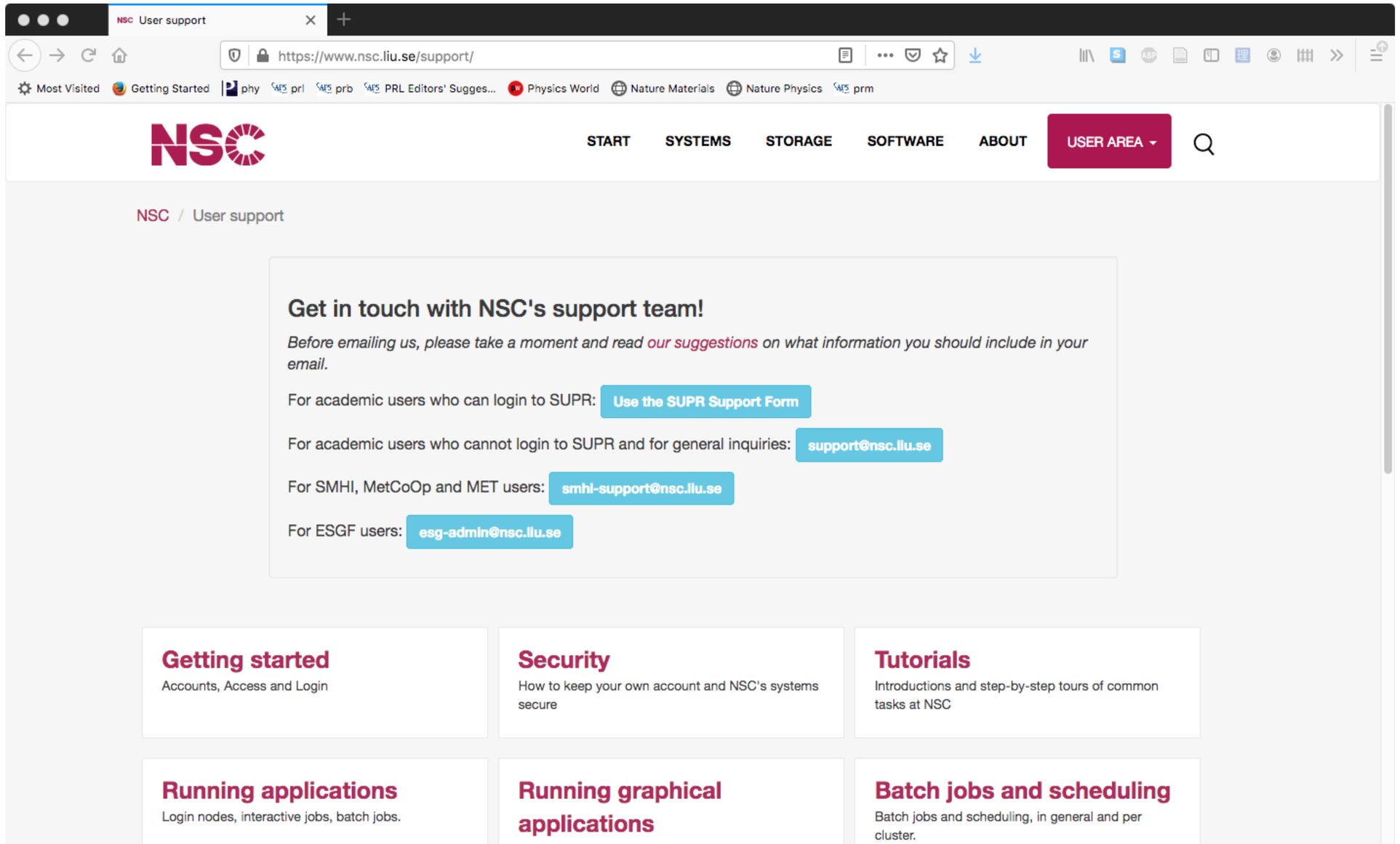


https://www.nsc.liu.se

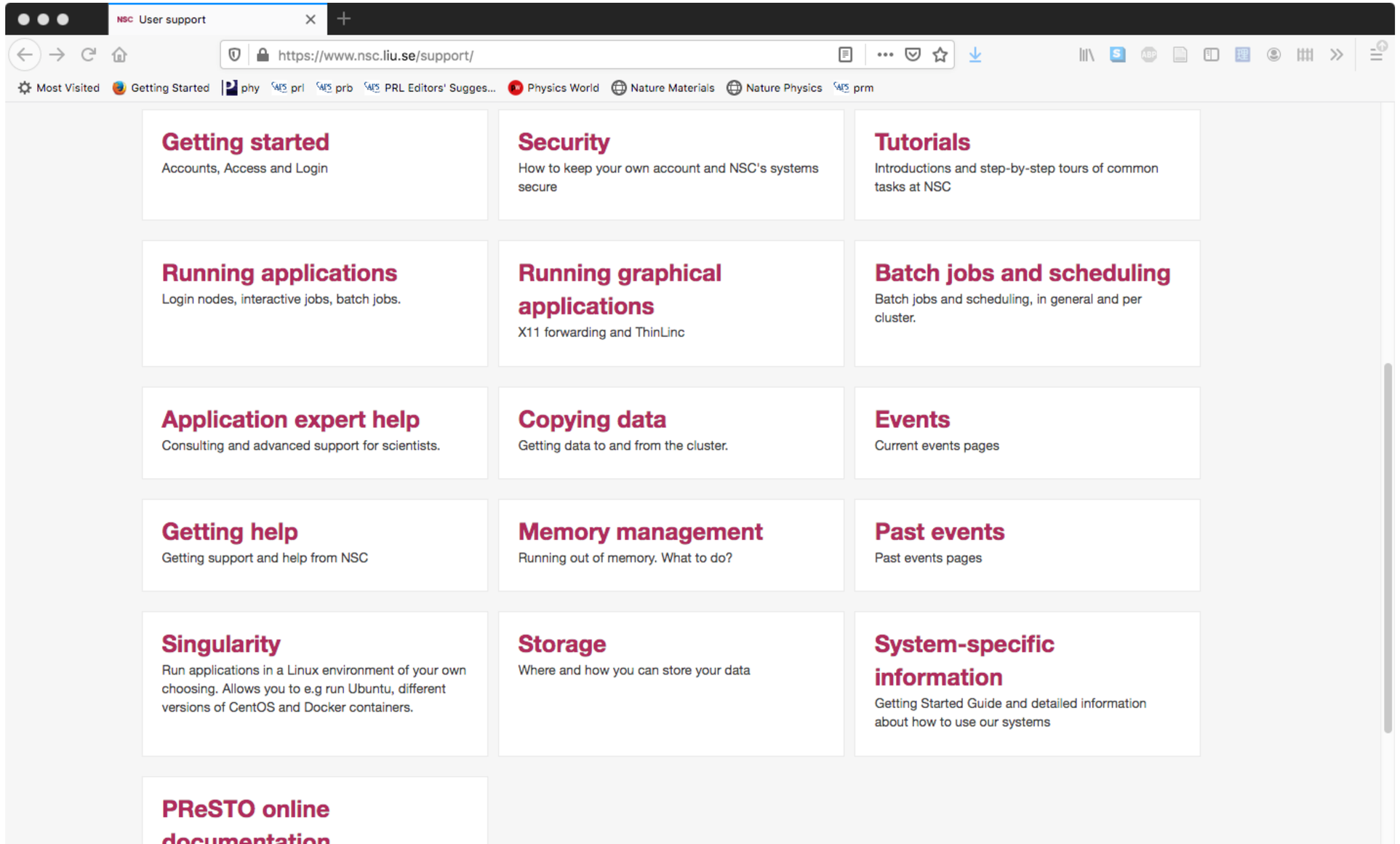
Where to find Information?



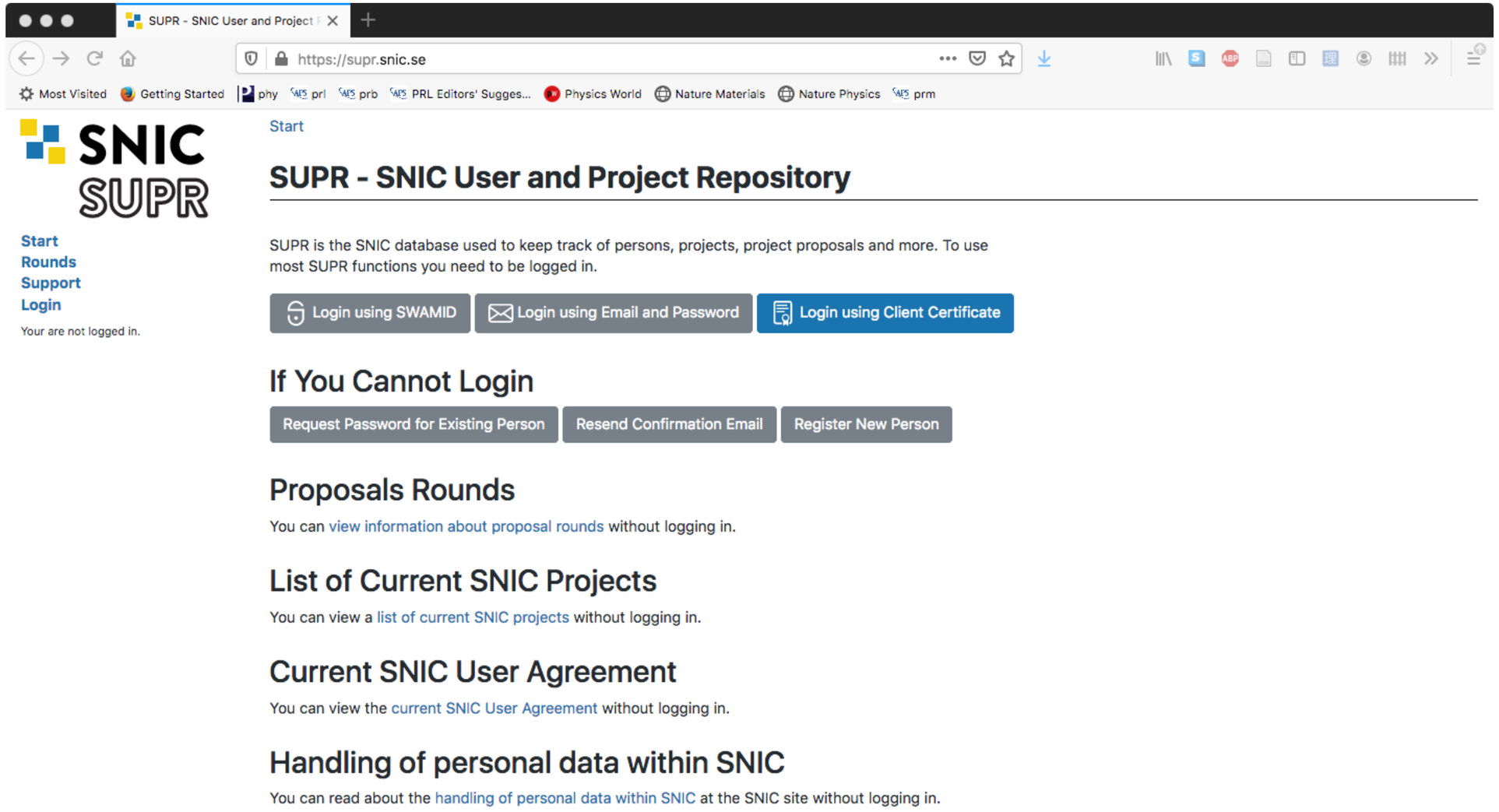
Where to find Information?



Where to find Information?

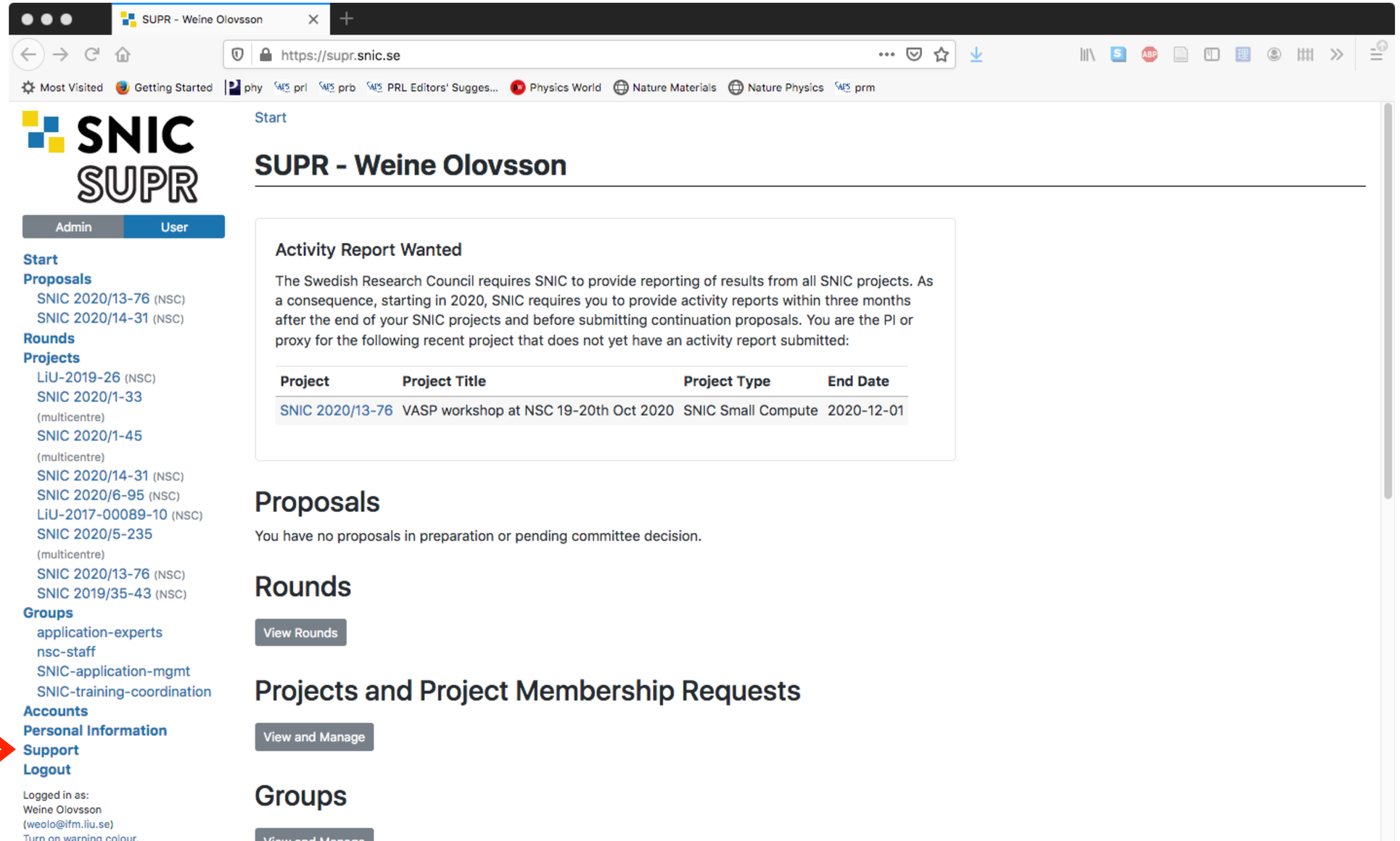


Getting Acess to HPC - SUPR

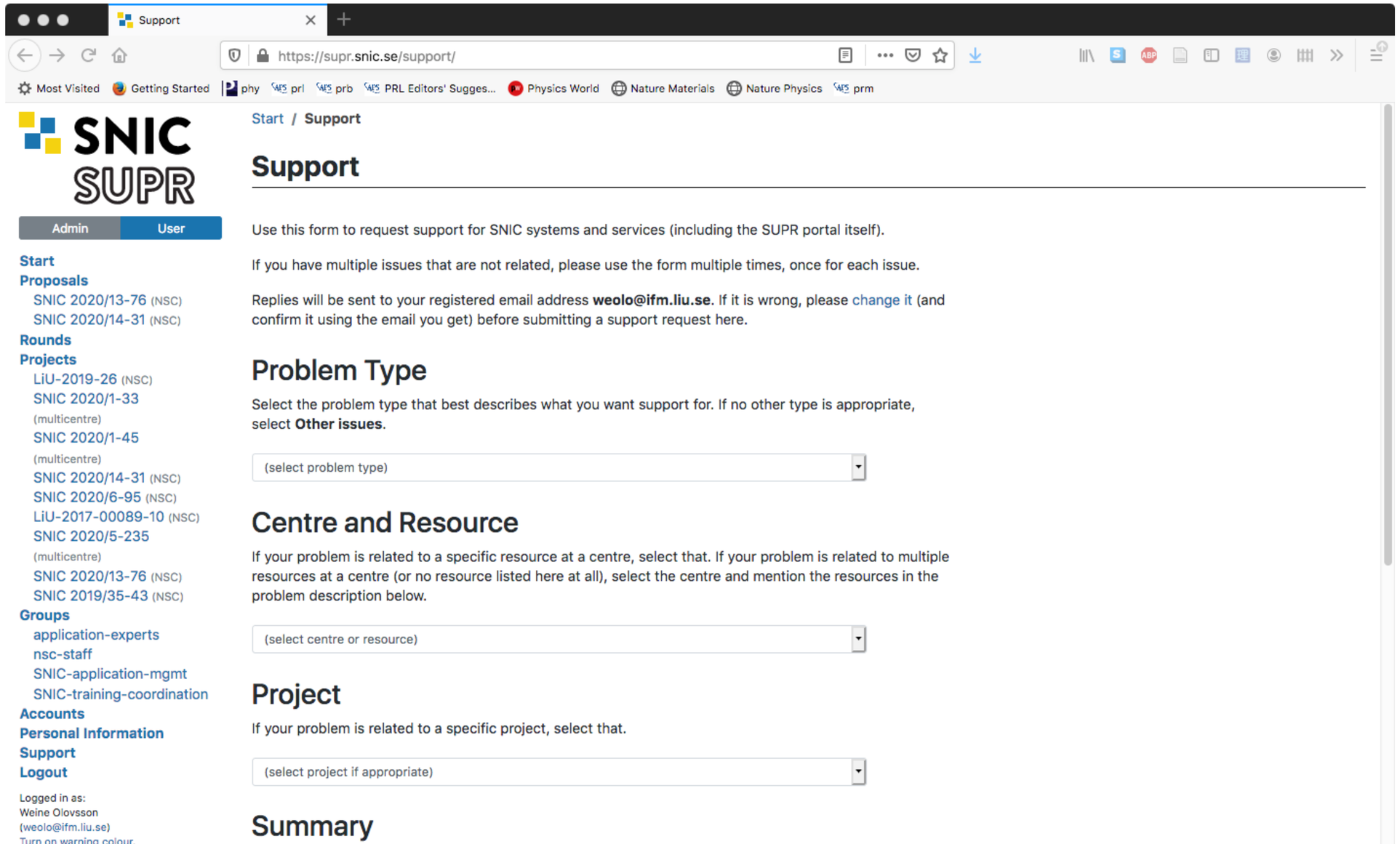


https://supr.snic.se

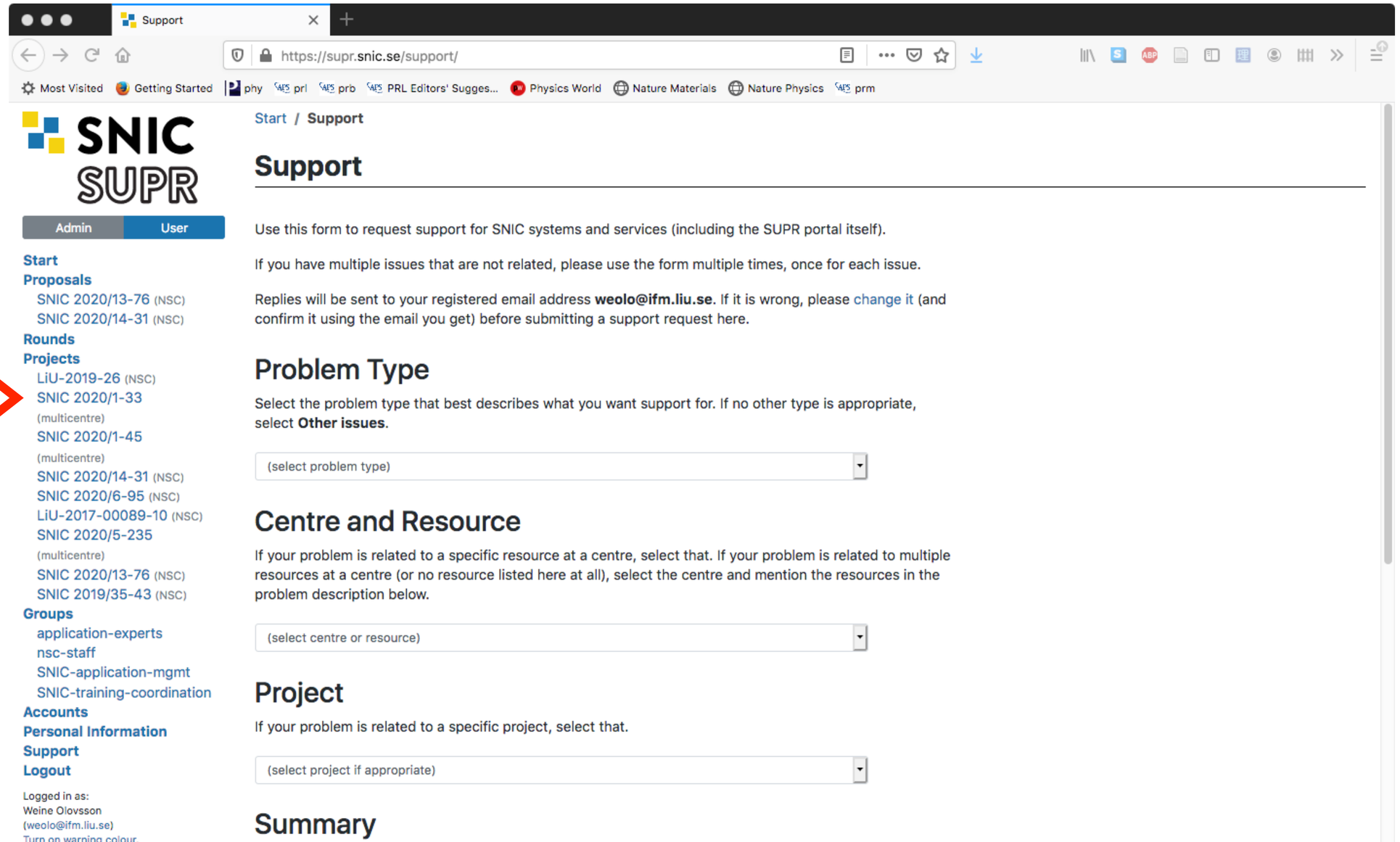
Support via SUPR



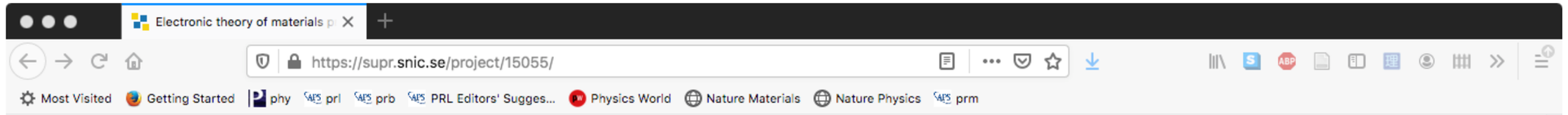
Support via SUPR



Projects in SUPR



Projects in SUPR



Storage projects linked to this compute project

Members of this compute project become extended members of the linked storage project and can access its storage.

Storage Project	Title	PI
SNIC 2020/6-95	Storage for theoretical physics environm...	Rickard Armiento

Resources

Allocation shows the current allocation.

Compute

Total Allocation during the whole project is shown with a **Percentage** field to the right, that compares **Total Usage** with the total allocation. The **Allocation until Today** field shows the allocation until today, also with a **Percentage** comparison.

Resource	Allocation	Unit	Total Usage	Allocation until Today	Percentage	Total Allocation	Percentage
Beskow @ PDC	1 400	x 1000 core-h/month	6 797.0	7 000.0	97.1 %	16 800.0	40.5 %
Tetralith @ NSC	900	x 1000 core-h/month	5 163.2	4 500.0	114.7 %	10 800.0	47.8 %
Kebnekaise @ HPC2N	150	x 1000 core-h/month	554.7	750.0	74.0 %	1 800.0	30.8 %
Tegner @ PDC	23	x 1000 core-h/month	0.0	115.0		276.0	

Storage

Percentage field to the right, compares **Usage** with the allocation. **Last Updated** shows the time at which the usage was last updated.

Resource	Allocation	Usage	Unit	Percentage	Allocation	Usage	Unit	Percentage	Last Updated
Centre Storage @ NSC	0		GiB		0		files		

Projects in SUPR

When & Why to use HPC?

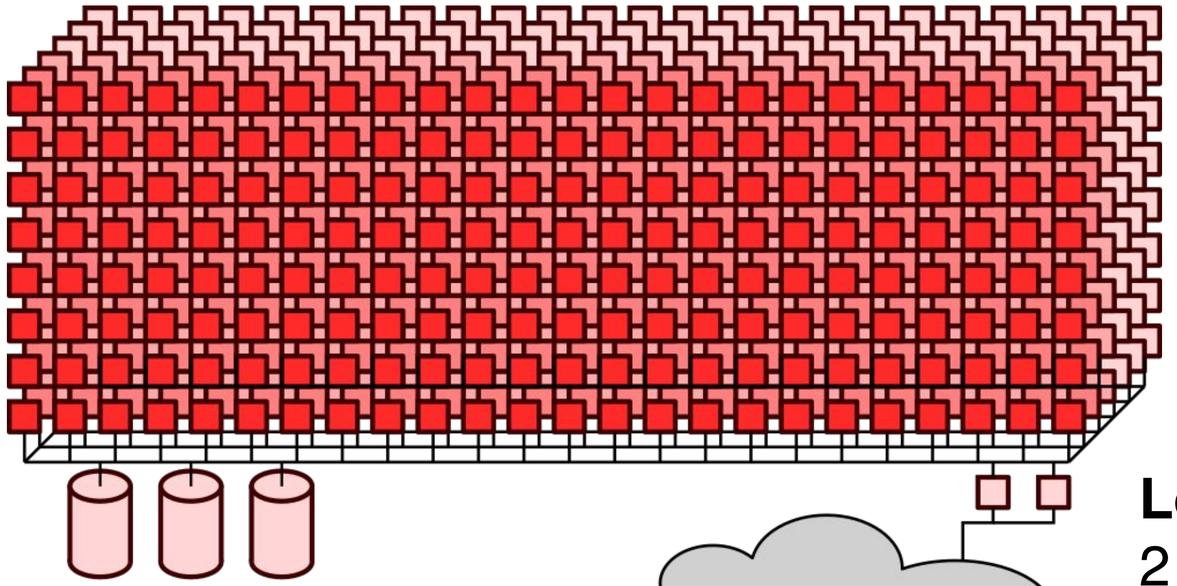
HPC = High Performance Computing

- High number of simulation or data analysis jobs
- The jobs are too large for a desktop/laptop
- Used in most research fields today
  - Numerical weather prediction
  - Climate simulations
  - Flow simulations
  - Materials science
  - Many disciplines within Chemistry, Physics, Biology
  - …

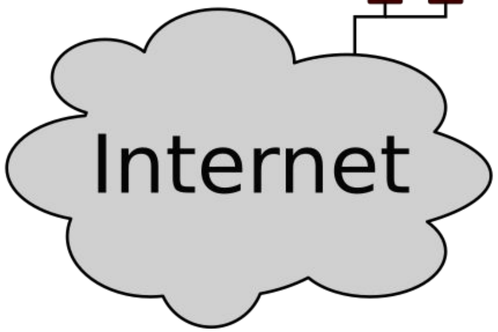
Desktop PC vs HPC

Desktop/laptop:
8 cores
16 GB RAM
Windows, MacOS (Unix), Linux
1 user

Tetralith: 1908 nodes, **Sigma:** 110 nodes



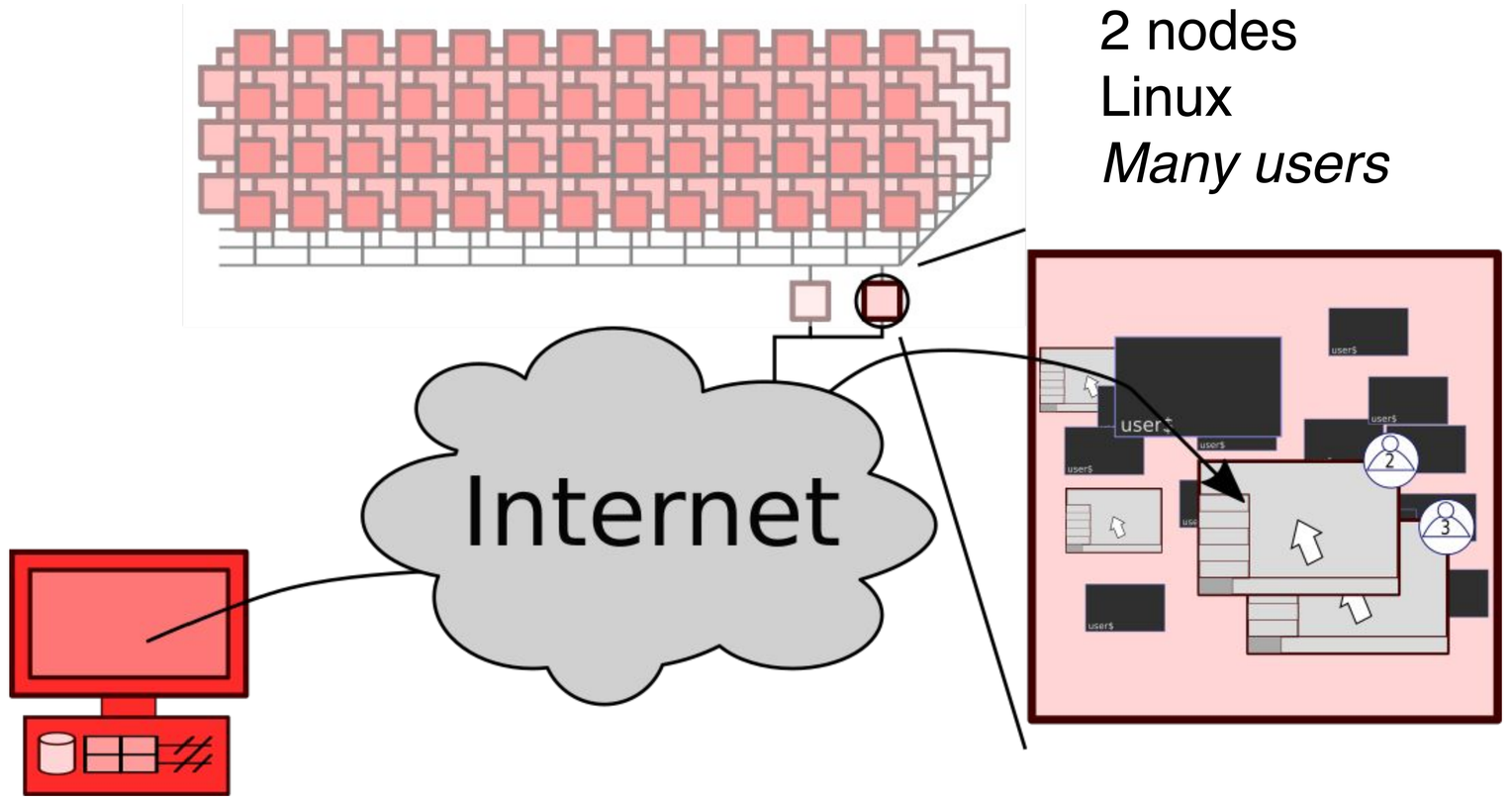
Work node:
32 cores
96 (384) GB RAM
Linux
Omni-Path network
1 - few users at a time



Login nodes:
2 nodes
Linux
Many users

Access to Tetralith

- Typical access: using ssh
- For graphics, use ThinLinc
- Many users share login node
- Be mindful of login node usage
- Work node access via queue system (Slurm)



Login nodes:
2 nodes
Linux
Many users

https://www.nsc.liu.se/support/getting-started/

Access to Tetralith: ssh

ssh: the common, classical way, to login

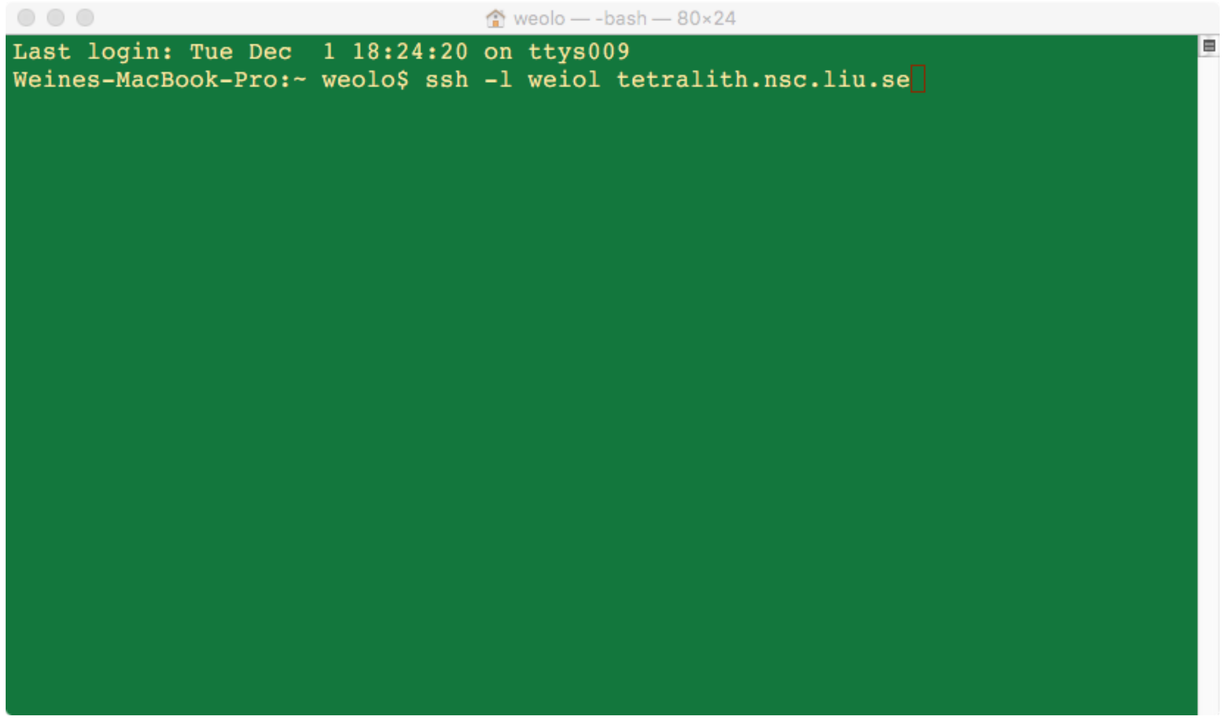
Typical login via terminal from Linux / Mac:

```
ssh username@tetralith.nsc.liu.se
```

- Windows: can use PuTTY

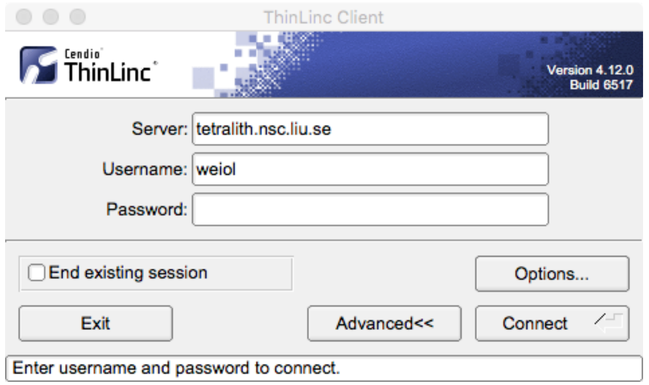
Note: to end up on a specific login node use:

```
tetralith1.nsc.liu.se
tetralith2.nsc.liu.se
```

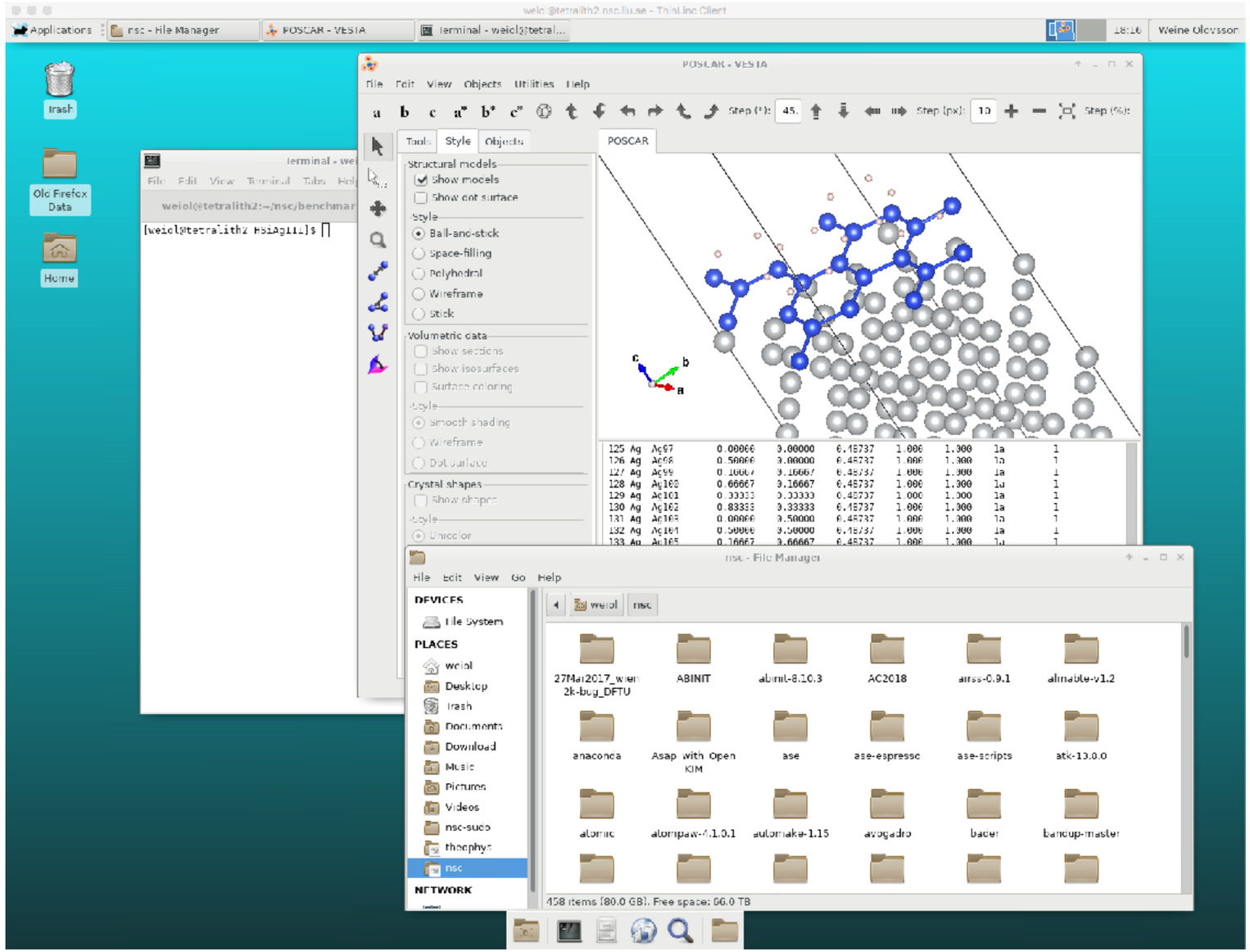


https://www.nsc.liu.se/support/getting-started/

Access to Tetralith: ThinLinc



- ThinLinc - *virtual desktop*
- Persistent sessions (compare screen, tmux)
- Recommended for graphics
- Hardware acc. graphics (vglrun) in some cases



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

Some Basics

- **Linux**, see e.g. guide and forum
  - Basic commands: `cd, pwd, ls, mkdir, mv, grep, less, cat, …`
- Common tools
  - Text editors: `vi, gedit, emacs, nano, …`
  - Plotting graphs: `gnuplot, grace, …`
  - Analysis (basic/complex): `python, R, Matlab, …`
- Useful things
  - Persistent terminal session: `screen, tmux`
  - Check compute usage: `projinfo`
  - Check disk usage: `snicquota`

Files & Storage

Three types of storage areas available:	Backup?	Recover deleted files? Snapshot?
1. Personal home directory, e.g. `/home/x_user`	yes!	yes!
2. Project storage, owned by PI, e.g. `/proj/ourstuff`	no!	yes!
3. Work node local disk (during runs)	no!	no!

Some notes:

- Use `snicquota` to check available disk space
- Project storage is linked to specific project allocation and life time
- Good idea to have your own backup
- Data is never 100% safe, there's always some risk

https://www.nsc.liu.se/support/storage/snic-centrestorage/recover-deleted-files/

https://www.nsc.liu.se/support/storage/index.html

Basic Security

- Unique password (non-trivial but not overly complicated)
- Suspicion that your account is compromised -> contact NSC

  - Don’t hesitate to contact us!
- Sharing accounts is not allowed (accounts are personal)

  Share files e.g. by managing project memberships and use /proj

https://www.nsc.liu.se/support/security/  Also, on how to use key-pair authentication

Software: 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

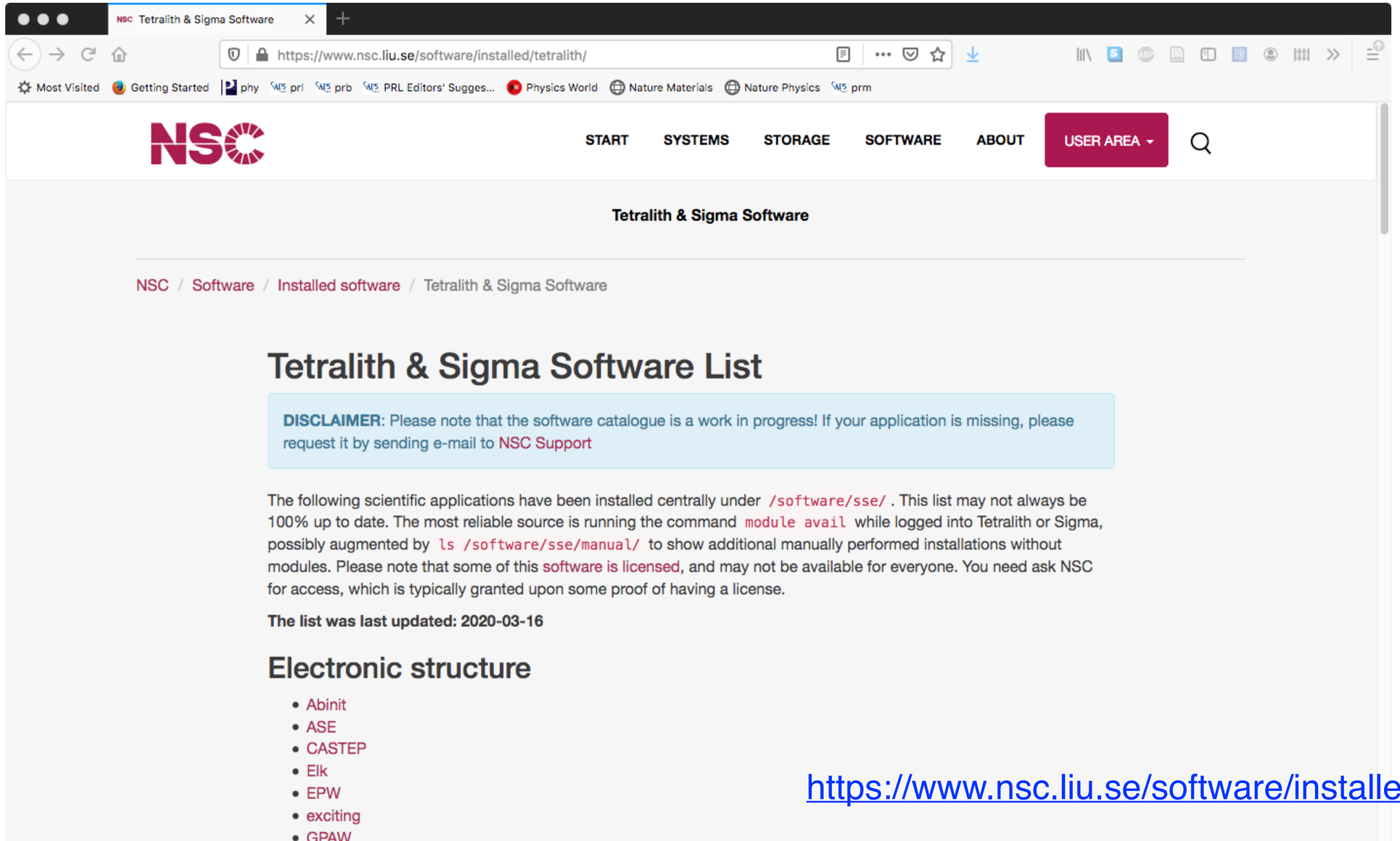
Testing, benchmarking, optimization

Global installation: wide or not usage, license?

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

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

Software: Installation Webpage



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

Software: Module System

`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/

Software: Module System

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

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

Software: Module System

```
[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/

Software: 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

Software: Build Environment

```
[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 ~]$
```

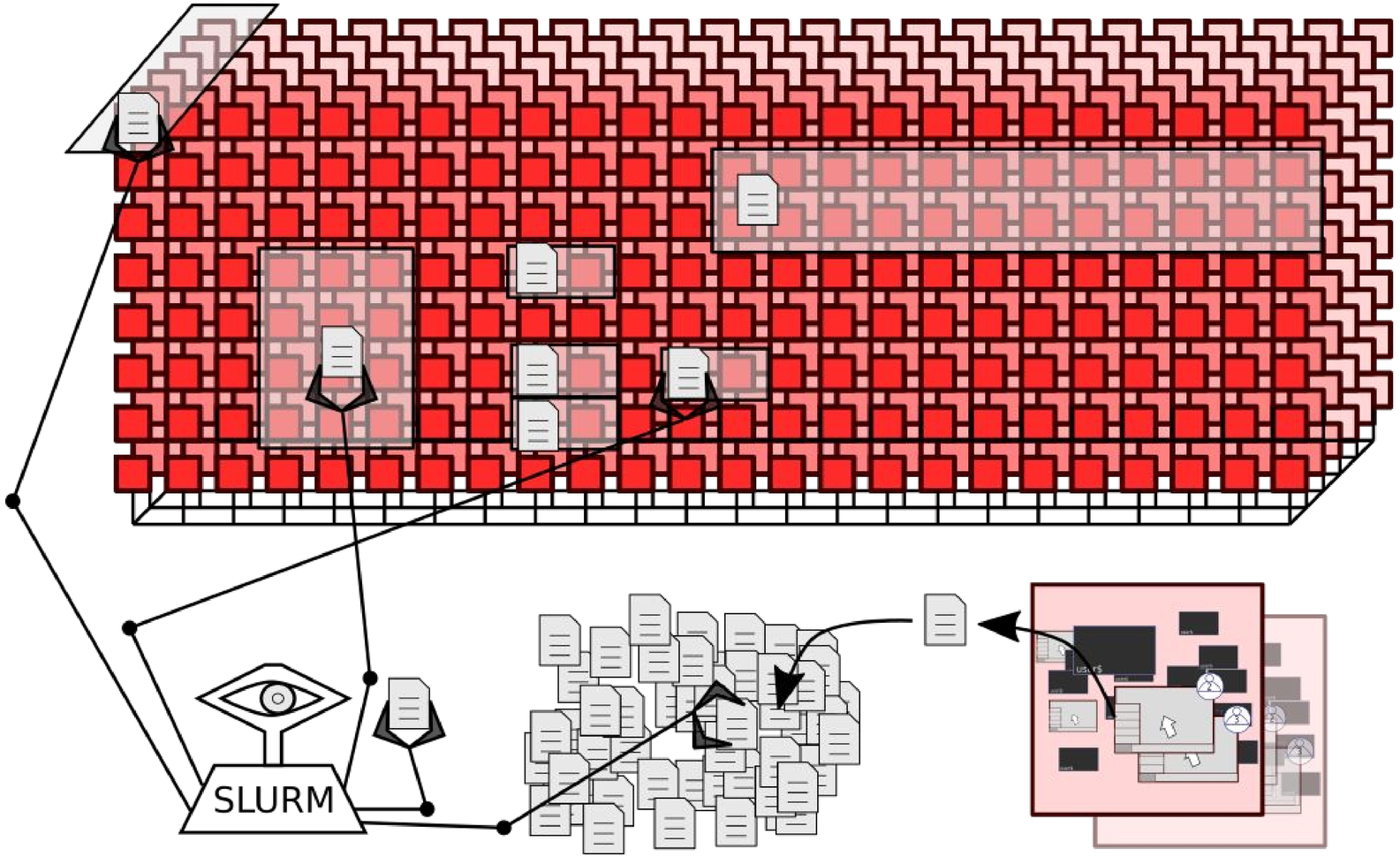
https://www.nsc.liu.se/software/modules/

Software: Build Environment

```
[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.)

[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 ~]$
```

Queue System: Slurm

- Many jobs & users
- Resource access via Slurm
- Several methods:
  - `sbatch`
  - `interactive`
- Run as much possible, based on prior usage
- Fairshare scheduling with backfill
- 168 hours (7d) walltime limit
- Avoid short time wide jobs, “flat jobs”
- Priority boosting available



https://www.nsc.liu.se/support/batch-jobs/tetralith/

Slurm: Running Batch Job

- Regular production runs
- Output to files

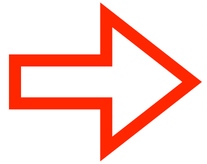
  `slurm-JOBID.out`

Example: a job script called “run.sh”

```
#!/bin/bash
#SBATCH -A snic2020-13-76
#SBATCH -t 1:00:00
#SBATCH -n 32
#SBATCH -J vasptst

module load VASP/6.2.1.29042021-omp-nsc1-intel-2018a-eb
mpprun vasp_std
```

project
time
MPI ranks
job name

NSC MPI job launching tool



Submit job:
`sbatch run.sh`

Check queue:
`squeue -u USERNAME`

Checking jobs:
`jobload JOBID`
`jobsh NODE` login to node, run “top”
`seff JOBID`
`lastjobs`

https://www.nsc.liu.se/support/batch-jobs/introduction/

Slurm: Interactive Job

- Testing, debugging
- Hands-on, direct node access

  Example: similar settings as for the job script

```
[weiol@tetralith1 ~]$interactive -A snic2020-13-76 -n 32 -t 1:00:00
salloc: Pending job allocation 11193334
salloc: job 11193334 queued and waiting for resources
salloc: job 11193334 has been allocated resources
salloc: Granted job allocation 11193334
srun: Step created for job 11193334
[weiol@n405 ~]$
```

- Special queue for brief testing, max 1h, max 1 node (also with job script)

```
--reservation=devel
```

https://www.nsc.liu.se/support/batch-jobs/tetralith/

Best Practices & Suggestions

In general:

- Be careful how you use Tetralith/Sigma login nodes
- Use SUPR to follow project usage
- Use the NSC documentation

contact us if problems!
we try to describe everything…

- Be careful about what you put in .bashrc (keep as simple as possible)
- Don’t hesitate to contact support@nsc.liu.se for help/questions

Best Practices & Suggestions

Common problems:

- My job **failed/crashed**. What now?
  - First, try to understand the cause
  - Contact support@nsc.liu.se / fill in form https://supr.snic.se

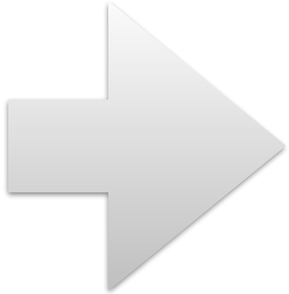
    

    provide details! username, system, jobid, job path, …
- Odd problems (lots of things set in .bashrc?)
- Don’t run heavy stuff / production work on the login node
  - For brief testing e.g. run interactively `--reservation=devel`

https://www.nsc.liu.se/support/getting-help/

Further Resources

- Working effectively on Tetralith / Sigma 2018
- Working with Python on Tetralith 2019
- NSC introduction day 2017 More details, e.g. running calcs.



Check links for presentations (.pdf)

- Presentations available at webpage!

- Working effectively with HPC systems