Workshop "OMNI - New Features and Changes"

Jan Steiner

Zentrum für Informations- und Medientechnik

September 30th 2021

Motivation

- OMNI cluster in service for half a year
  - First hardware installed a year ago

- Review of what has been accomplished, what is yet to do

- Common problems we noticed with users

Agenda

- Filesystem changes
- Common pitfalls we have noticed
- Module system, software licenses, terms of use
- Using the GPUs
- Ongoing work (e.g. Jupyter)
- Questions and Discussion

Agenda

- **Filesystem changes**
- Common pitfalls we have noticed
- Module system, software licenses, terms of use
- Using the GPUs
- Ongoing work (e.g. Jupyter)
- Questions and Discussion

File systems

- **Home**: as before, but changed permissions
  - Group spaces under `/group`, explained later

- **Work**: as before (`ws_allocate`…)

- **Fast** (Burst Buffer, SSDs): new, use like workspaces

- **tmp**: used in background, modified by us
  - Make sure you do not use `/tmp` directly, better `$TMP`

- **Storage services** (NAS/XNAS): new, only on login nodes, can and should be used for transfer only
  - NAS for persons, XNAS for groups
  - Mounted under `/nas` and `/xnas`

https://cluster.uni-siegen.de/omni/usage/file-systems/?lang=en

Change in home permissions

- New permissions for home directories since August 2021

- Other users cannot see inside your home anymore: `rwx --- ---`
  - Changed from HoRUS to OMNI: previously `rwx r-x r-x`

- Advantage: personal data is now protected
  - Example: sensitive data used for data science

- Disadvantage: files in your home cannot be reached by colleagues
  - If you need to give access, you can change it:

    ```
    chmod 755 <your home directory>
    ```

Agenda

- Filesystem changes
- **Common pitfalls we have noticed**
- Module system, software licenses, terms of use
- Using the GPUs
- Ongoing work (e.g. Jupyter)
- Questions and Discussion

SLURM Pitfalls

Common problem we regularly see with new users:

- Job that does not use full node (e.g. one task, one core)
- Job then fails with “not enough memory” error

Reason: OMNI nodes configured for shared use

- If you use 1/64th of cores, you only get 1/64th of memory (4 GB)

https://cluster.uni-siegen.de/omni/usage/queuing-a-job/?lang=en

SLURM Pitfalls: memory

- Solution: request memory in job script

```
#SBATCH --mem=X
```

- Where X is value in Megabytes
- Maximum is 240000 (240 GB)
- Other units can be used: `mem=120G` will request 120 Gigabytes
- `mem=0` will request all memory on the node

https://cluster.uni-siegen.de/omni/usage/slurm/?lang=en

SLURM pitfalls: parallel execution

- Several users thought they run parallel computations, but didn’t

- Common example: multiple Python processes

- Reminder: `srun` inside of job script will launch parallel tasks
  - `$SLURM_NTASKS` processes by default
  - Use options to reduce number/modify using SLURM parameters

- **Bad idea**: *“It’s slow? I need to use more processes!”* blindly

Agenda

- Filesystem changes
- Common pitfalls we have noticed
- **Module system, software licenses, terms of use**
- Using the GPUs
- Ongoing work (e.g. Jupyter)
- Questions and Discussion

Module environment

- Our modules come from multiple different sources:
  - Bright cluster manager (GPU stack)
  - OpenHPC (C and Fortran compilers, MPI)
  - Software vendors (Intel, MATLAB)

- Complex structure
  - Interdependencies
  - Mutual incompatibilities

- Brief overview here to reduce confusion

https://cluster.uni-siegen.de/omni/usage/modules/?lang=en

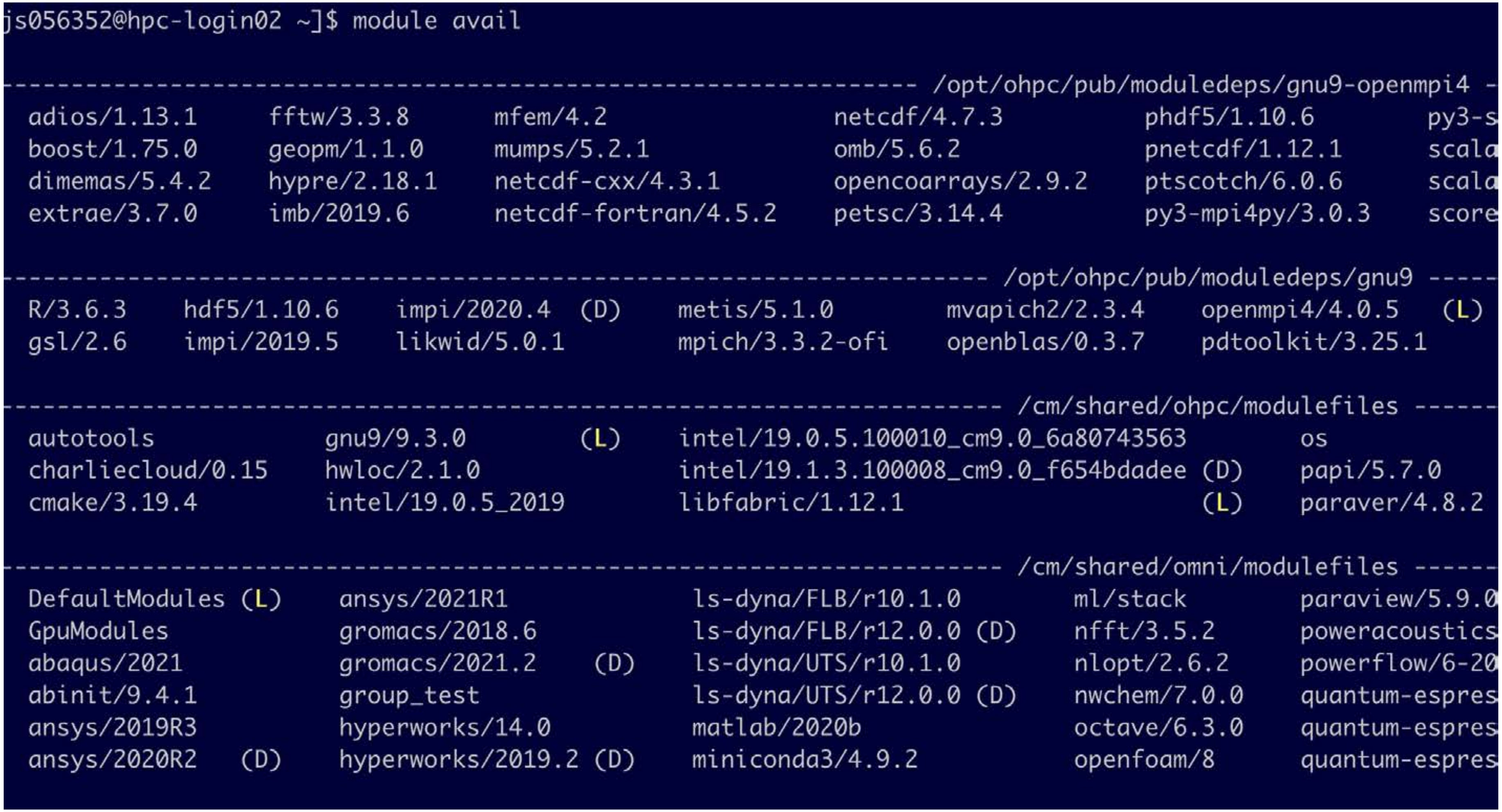
Module environment

- Incompatible modules cannot be loaded together
  - **Hidden from view** in `module avail`
  - Can still be searched with `module spider`

- GPU stack is incompatible with almost everything else

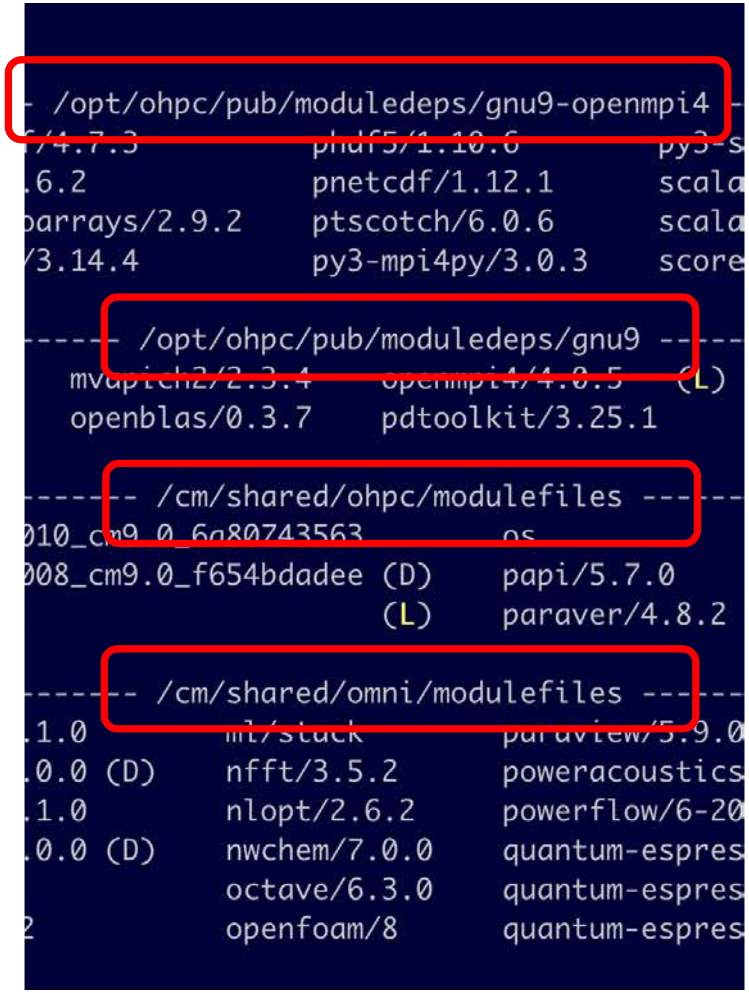
- Regular stack: four main “branches”
  - Almost everything depends on either GCC or Intel compiler
  - A lot of things depend on either OpenMPI or Intel MPI (and those depend on GCC or Intel compiler)
  - Default: GCC (`gnu9` module) and OpenMPI (`openmpi4` module)

Regular stack



```
js056352@hpc-login02 ~]$ module avail

------------------------------------------------------------------ /opt/ohpc/pub/moduledeps/gnu9-openmpi4 -
  adios/1.13.1      fftw/3.3.8      mfem/4.2                 netcdf/4.7.3          phdf5/1.10.6        py3-s
  boost/1.75.0      geopm/1.1.0     mumps/5.2.1              omb/5.6.2             pnetcdf/1.12.1      scala
  dimemas/5.4.2     hypre/2.18.1    netcdf-cxx/4.3.1         opencoarrays/2.9.2    ptscotch/6.0.6      scala
  extrae/3.7.0      imb/2019.6      netcdf-fortran/4.5.2     petsc/3.14.4          py3-mpi4py/3.0.3    score

----------------------------------------------------------------------- /opt/ohpc/pub/moduledeps/gnu9 -----
  R/3.6.3     hdf5/1.10.6    impi/2020.4   (D)    metis/5.1.0         mvapich2/2.3.4    openmpi4/4.0.5    (L)
  gsl/2.6     impi/2019.5    likwid/5.0.1         mpich/3.3.2-ofi     openblas/0.3.7    pdtoolkit/3.25.1

------------------------------------------------------------------------ /cm/shared/ohpc/modulefiles ------
  autotools             gnu9/9.3.0          (L)    intel/19.0.5.100010_cm9.0_6a80743563         os
  charliecloud/0.15     hwloc/2.1.0                intel/19.1.3.100008_cm9.0_f654bdadee (D)     papi/5.7.0
  cmake/3.19.4          intel/19.0.5_2019          libfabric/1.12.1                     (L)     paraver/4.8.2

------------------------------------------------------------------------ /cm/shared/omni/modulefiles ------
  DefaultModules (L)     ansys/2021R1              ls-dyna/FLB/r10.1.0           ml/stack          paraview/5.9.0
  GpuModules             gromacs/2018.6            ls-dyna/FLB/r12.0.0 (D)       nfft/3.5.2        poweracoustics
  abaqus/2021            gromacs/2021.2     (D)    ls-dyna/UTS/r10.1.0           nlopt/2.6.2       powerflow/6-20
  abinit/9.4.1           group_test                ls-dyna/UTS/r12.0.0 (D)       nwchem/7.0.0      quantum-espres
  ansys/2019R3           hyperworks/14.0           matlab/2020b                  octave/6.3.0      quantum-espres
  ansys/2020R2    (D)    hyperworks/2019.2 (D)     miniconda3/4.9.2              openfoam/8        quantum-espres
```

Regular stack



Depends on compiler and MPI

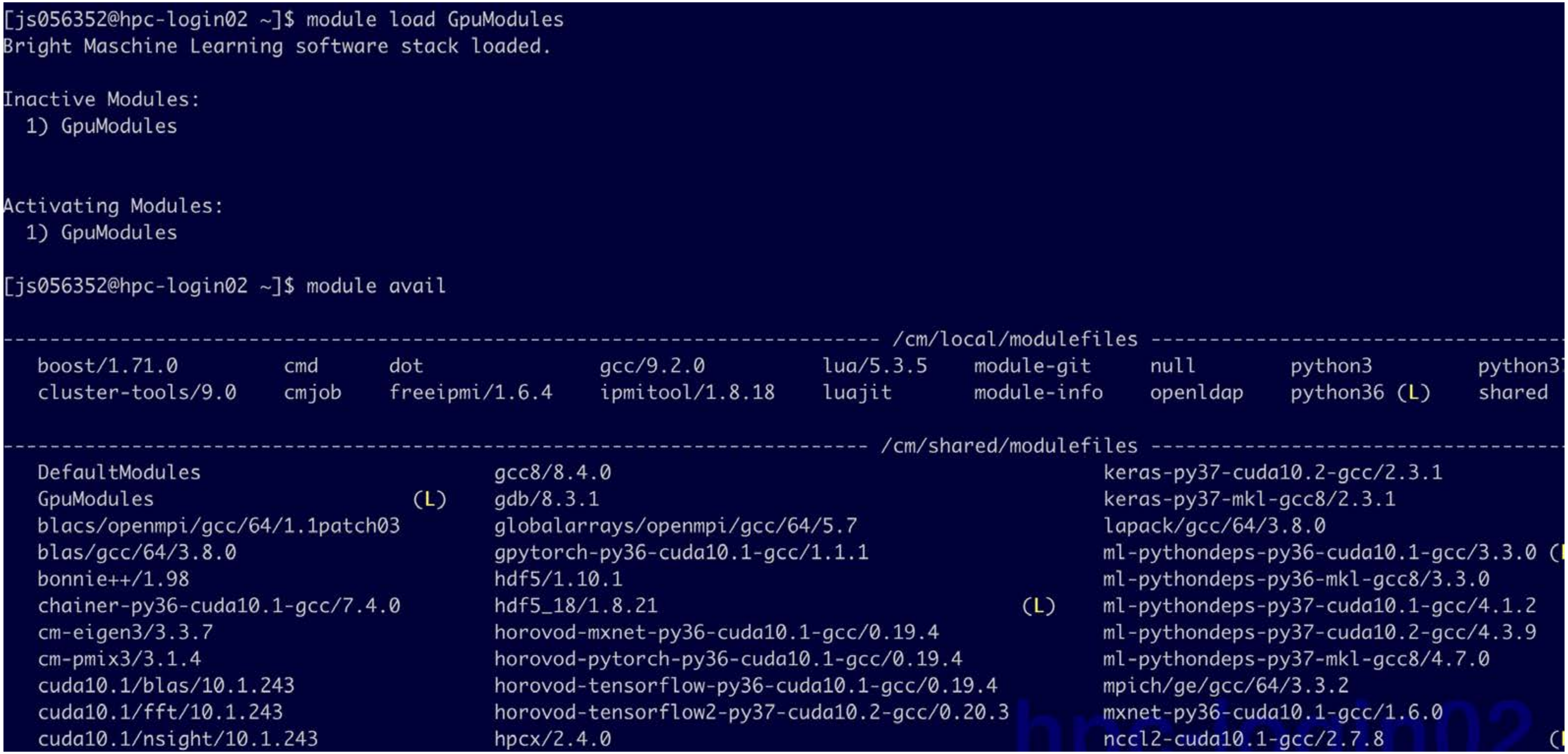
Depends on compiler

OpenHPC, no dependencies

Unrelated to OpenHPC

```
- /opt/ohpc/pub/moduledeps/gnu9-openmpi4 -
f/4.7.3             phdf5/1.10.6          py3-s
.6.2                pnetcdf/1.12.1        scala
parrays/2.9.2       ptscotch/6.0.6        scala
/3.14.4             py3-mpi4py/3.0.3      score

------ /opt/ohpc/pub/moduledeps/gnu9 -----
   mvapich2/2.3.4     openmpi4/4.0.5   (L)
   openblas/0.3.7     pdtoolkit/3.25.1

------- /cm/shared/ohpc/modulefiles -------
010_cm9.0_6a80743563         os
008_cm9.0_f654bdadee (D)     papi/5.7.0
                     (L)     paraver/4.8.2

------- /cm/shared/omni/modulefiles -------
.1.0         ml/stack         paraview/5.9.0
.0.0 (D)     nfft/3.5.2       poweracoustics
.1.0         nlopt/2.6.2      powerflow/6-20
.0.0 (D)     nwchem/7.0.0     quantum-espres
             octave/6.3.0     quantum-espres
2            openfoam/8       quantum-espres
```

After loading GpuModules



```
[js056352@hpc-login02 ~]$ module load GpuModules
Bright Maschine Learning software stack loaded.

Inactive Modules:
  1) GpuModules


Activating Modules:
  1) GpuModules

[js056352@hpc-login02 ~]$ module avail

--------------------------------------------------------------------------- /cm/local/modulefiles -----------------------------------
   boost/1.71.0         cmd      dot               gcc/9.2.0          lua/5.3.5    module-git     null        python3         python3
   cluster-tools/9.0    cmjob    freeipmi/1.6.4    ipmitool/1.8.18    luajit       module-info    openldap    python36 (L)    shared

-------------------------------------------------------------------------- /cm/shared/modulefiles -----------------------------------
   DefaultModules                         gcc8/8.4.0                                          keras-py37-cuda10.2-gcc/2.3.1
   GpuModules                      (L)    gdb/8.3.1                                           keras-py37-mkl-gcc8/2.3.1
   blacs/openmpi/gcc/64/1.1patch03        globalarrays/openmpi/gcc/64/5.7                     lapack/gcc/64/3.8.0
   blas/gcc/64/3.8.0                      gpytorch-py36-cuda10.1-gcc/1.1.1                    ml-pythondeps-py36-cuda10.1-gcc/3.3.0 (
   bonnie++/1.98                          hdf5/1.10.1                                         ml-pythondeps-py36-mkl-gcc8/3.3.0
   chainer-py36-cuda10.1-gcc/7.4.0        hdf5_18/1.8.21                               (L)    ml-pythondeps-py37-cuda10.1-gcc/4.1.2
   cm-eigen3/3.3.7                        horovod-mxnet-py36-cuda10.1-gcc/0.19.4              ml-pythondeps-py37-cuda10.2-gcc/4.3.9
   cm-pmix3/3.1.4                         horovod-pytorch-py36-cuda10.1-gcc/0.19.4            ml-pythondeps-py37-mkl-gcc8/4.7.0
   cuda10.1/blas/10.1.243                 horovod-tensorflow-py36-cuda10.1-gcc/0.19.4         mpich/ge/gcc/64/3.3.2
   cuda10.1/fft/10.1.243                  horovod-tensorflow2-py37-cuda10.2-gcc/0.20.3        mxnet-py36-cuda10.1-gcc/1.6.0
   cuda10.1/nsight/10.1.243               hpcx/2.4.0                                          nccl2-cuda10.1-gcc/2.7.8               (
```

Software Licences

Software situation more complex on OMNI than HoRUS

- Cluster is three times the size
- More diverse userbase (data science)
- More software that not everyone is allowed to use

→Considerably more maintenance overhead
→We cannot install and maintain everything centrally for everyone

Software Policy

Three-tiered software installation policy:

1. More ways of installing software yourself
   - Singularity (container system like Docker)
   - Package manager `conda` (not just Python)

2. Group spaces (work in progress)
   - Directory where group can install software
   - No guarantees, but we advise and guide

3. Central installation as before
   - If multiple groups use software
   - If we believe maintenance overhead is manageable

Software Policy

- Centrally installed software will need a responsible person among the userbase
- “PAPA” – Primary Approach Partner for the Application
- One for each software application
- Should be experienced user who knows userbase
  - Common: one person responsible for license
- We will start approaching people soon

PAPA responsibilities

Not much extra work (!)

- We will consult you whether we should upgrade version
- Should have an overview over how licensing works
- Helps if you know users personally
- If we notice software is not used for extended period, we might ask whether we can uninstall

PAPA responsibilities

Things being a PAPA does NOT mean:

- You do not need to be an expert user
- You do not need to be a sysadmin for the software
- You do not need to keep an overview over who uses software
- You are not liable for mistakes

Terms of Use

- ZIMT leadership demands stricter enforcement of rules than before

- Each user now has to agree to “Terms of Use” when applying for cluster use in Nutzerkontenverwaltung (unisim.zimt.uni-siegen.de)

- Key terms/restrictions
  - Enforcement of trade embargoes (e.g. Iran citizens need special permissions)
  - Cite OMNI use in publications (just mention in Acknowledgements)
  - Regular ZIMT Terms of Use apply too

- Terms (in German): http://cluster.uni-siegen.de/Nutzungsbedingungen/

MATLAB licenses

- We were informed of misunderstanding/changed conditions

- Fak. 4 (Bernd Klose) manages MATLAB licenses
  - Collects money for MATLAB use annually
  - Contacts professors
  - ZIMT not involved in process

- Rule: money collected per parallel usable instance of MATLAB
  - Your PC and the cluster count as two instances
  - Pool workers do not count extra

Unfortunately, that means MATLAB usage on cluster is not free

https://cluster.uni-siegen.de/omni/application-software/matlab/?lang=en

Agenda

- Filesystem changes
- Common pitfalls we have noticed
- Module system, software licenses, terms of use
- **Using the GPUs**
- Ongoing work (e.g. Jupyter)
- Questions and Discussion

GPU tips and tricks

- 10 GPU-nodes with 24 GPUs in total
  `#SBATCH --partition=gpu` or
  `#SBATCH --gres=gpu:2`

Node	# GPUs
gpu-node[001-004]	4
gpu-node[005-008]	1
gpu-node[009-010]	2

- You can exclude nodes:
  `#SBATCH -x gpu-node010`
  - Some software cannot handle running on GPUs other than the first

- Remember different module stack: `module load GpuModules`

https://cluster.uni-siegen.de/omni/usage/gpus/?lang=en

GPUs: use efficiently

- GPUs are heavily used → long wait times

- Occasionally: GPU not even used (misconfiguration)

- What can you do?
  - Make sure GPUs are actually faster **including wait time**
  - Split work: only run in GPU queue what really needs GPU
  - Make sure software actually uses GPU

- Many applications do not strictly need GPUs (e.g. Tensorflow)

Agenda

- Filesystem changes
- Common pitfalls we have noticed
- Module system, software licenses, terms of use
- Using the GPUs
- **Ongoing work (e.g. Jupyter)**
- Questions and Discussion

Ongoing Work

- Two more cluster features being worked on:
  - Jupyter: web platform for interactive programming, visualization
  - Group Spaces: install software for your group
- Both in “beta state”
- We will officially announce them when ready

Jupyter status

- Jupyter working on OMNI
- Some kernels run as SLURM jobs
  - Same problems as regular jobs: **do not run on front end**
- Not yet stable (regular technical problems)
- Documentation not yet ready (easy to break e.g. Python installs)

Group spaces status

- 3 groups as test cases in temporary setup
- Coordination with other ZIMT departments complete (storage, group management etc.)
- Processes basically fully defined
- Not yet migrated
- Documentation not complete

Group spaces status

Preview of eventual procedure (not final):

- Professor will create ticket asking for group space (not yet possible)
  - Professor needs HPC access
- Group will either be created or existing group used
- Professor will be able to add/remove users via https://selfservice.zimt.uni-siegen.de
  - Users still need to apply for HPC access separately
- Each group member can install software in group space

Thank you for your attention!

Questions?

Part 2: Open Discussion

Appendix

How not to run in parallel

```
# Wrong, don’t do this. Will not run in parallel
#SBATCH --ntasks 64

python myscript.py
```

```
# Still not ideal (circumvents SLURM)
#SBATCH --ntasks 64

for i in {1..64}
do
    python myscript.py &
done
wait
```

How to run in parallel

```
# Better
#SBATCH --ntasks 64

srun wrapper.sh
```

```
# Corresponding wrapper.sh

# Run a different case in each task
case_id=$SLURM_PROCID

# Now srun task 0 will run case 0 etc.
python myscript.py $case_id
```