Introduction to LRZ HPC Systems

2020-06-08 | Gerald Mathias

Leibniz-Rechenzentrum (LRZ)/ Leibniz Supercomputing Centre

of the Bayerischen Akademie der Wissenschaften/ Bavarian Academy of Sciences and Humanities

250
employees
approx.

56
years of
IT support

Computer Centre
for all Munich Universities

Regional Computer Centre
for all Bavarian Universities

National Supercomputing Centre
(GCS)

European Supercomputing Centre
(PRACE)

For European Research

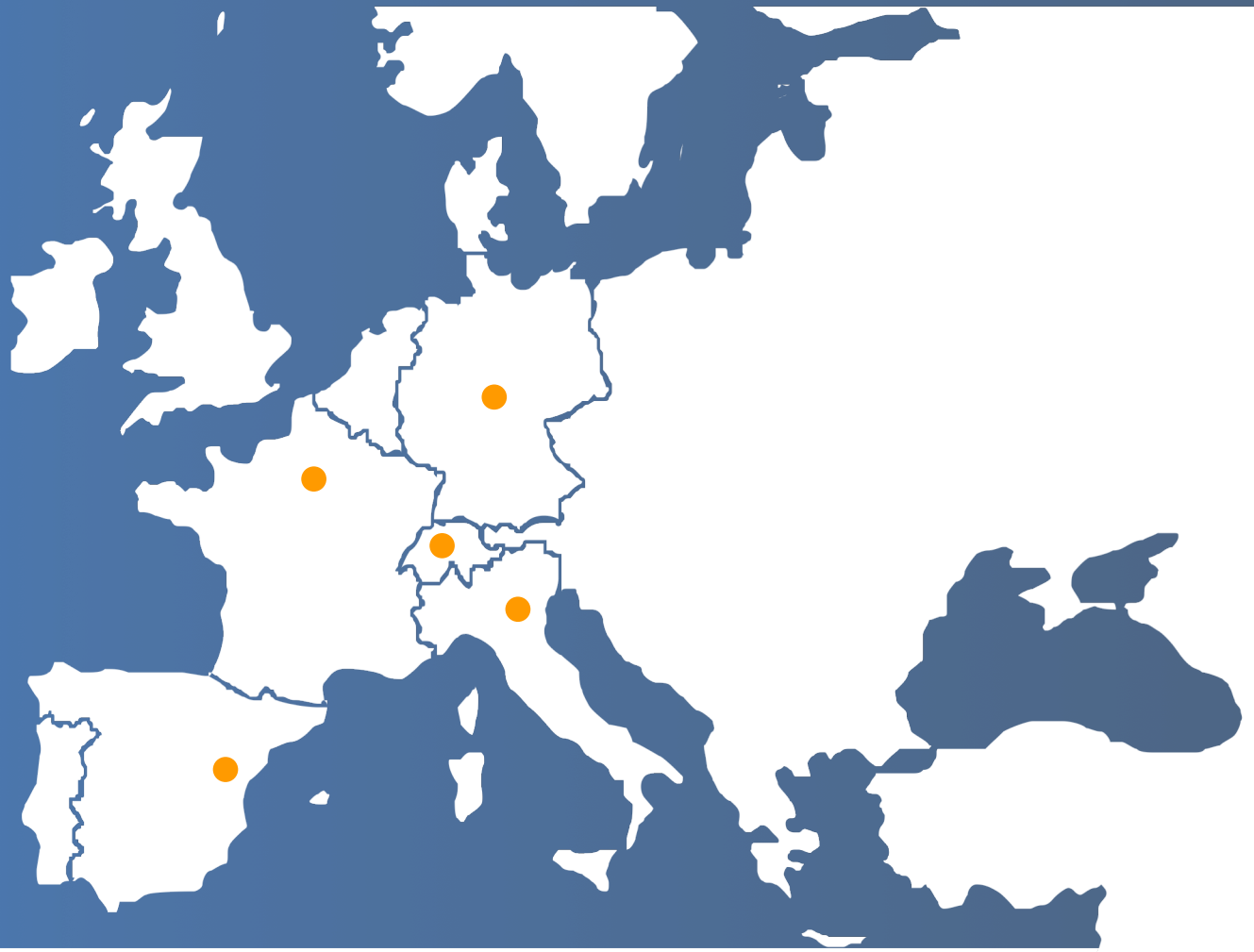
Partnership for Advanced Computing in Europe (PRACE) | 25 Countries

Federated, pan-European Tier-0 supercomputing infrastructure

Hosting Members:

- GCS (Germany: LRZ, HLRS, JSC)
- BSC (Spain)
- CSCS (Switzerland)
- CINECA (Italy)
- GENCI (France)

PRACE 2: 2017 – 2020




lrz
SuperMUC-NG

SuperMUC-NG: Intel / Lenovo

	thin	fat
Processor type (205/240 W TDP) Xeon Platinum 8174, 24 cores	Intel Skylake	Intel Skylake
Number of cores per node	48 (2x24)	48 (2x24)
Memory per node	96 GB	768 GB
Nominal frequency	2.7 GHz	2.7 GHz
AVX-512 frequency (all cores active), current default frequency	2.3 GHz	2.3 GHz
Floating point operations per clock (Fused MulAdd = 2)	32	32
Total number of nodes of this type	6336	144
Number of islands with this node type (792 nodes / 144 Nodes)	8	1
Fat Tree with island, pruning 1:3.8 between islands		
Total Cores	311,040	
Total Nodes	6480	
Total Memory	719 TByte	
„Peak Performance“	26.9 PF	
Linpack (Nov 18: rank 8, **Jun 19 and Nov 19**: rank 9)	19.5 PF	
*Cf. Vector Triad RINF vd=va*vb+vc (Average/Large+OpenMP)*	*1.0 / 0.2 PF*	

SuperMUC-NG

IBM Spectrum Scale (GPFS) Parallel File System

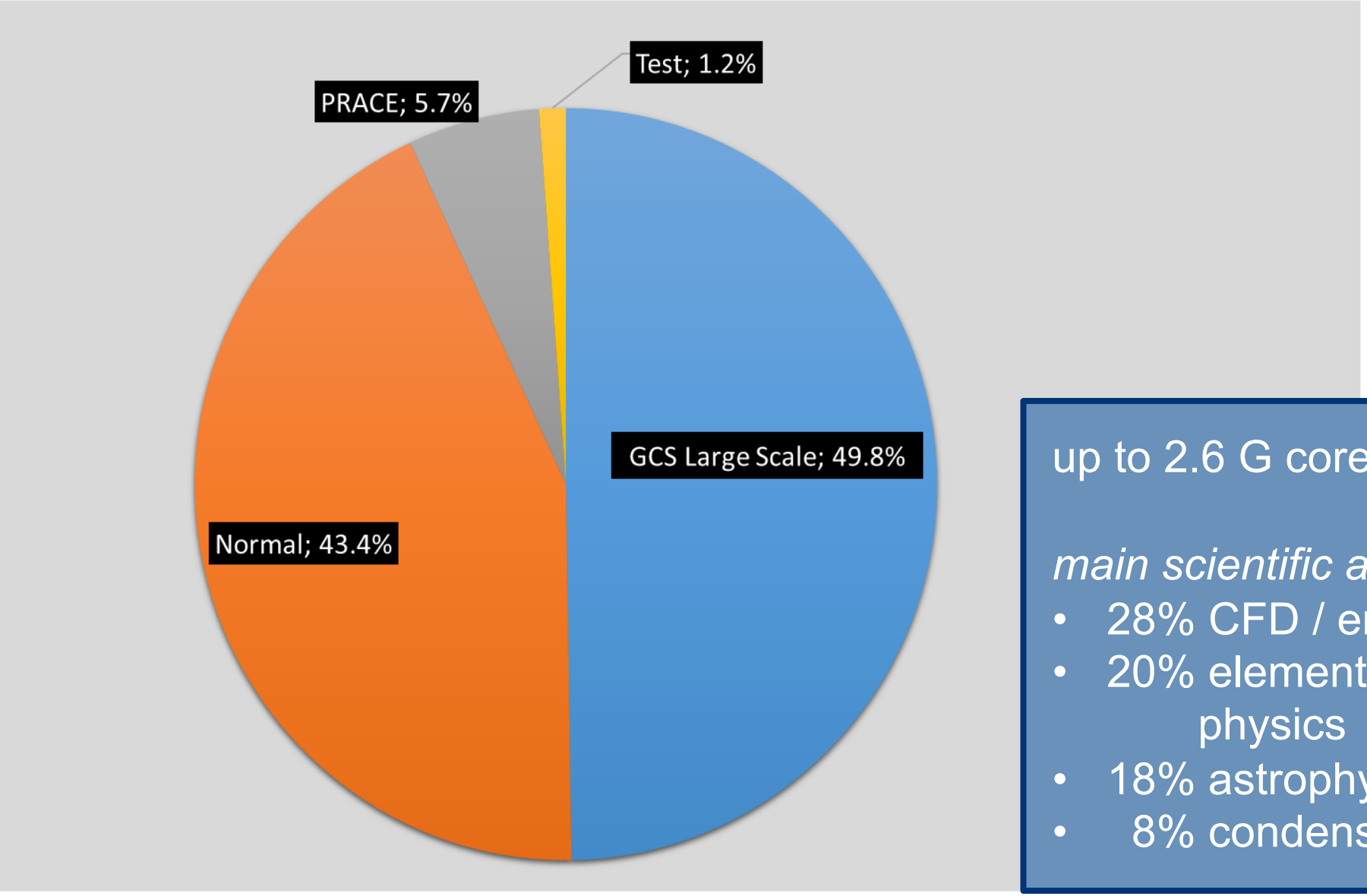
- SCRATCH/WORK
  - 50 PByte capacity
  - 500 GByte/s I/O bandwidth
- LRZ DSS: Data Science Storage for Long Term Data Storage
  - 20 PByte capacity and
  - 70 GByte/s I/O bandwidth
- HOME
  - 256 TB + 256 TB Replika
  - 28 Gbytes/s SSD Tier, 7 Gbyte/s HDD Tier, 40000 IOPS

OpenStack Compute Cloud (100 GigE)

- 32 nodes with 2x Intel Xeon 6148 processors, 192 Gbyte
- 32 nodes with 2x Intel Xeon 6148 processors,
  2x Nvidia Volta 100 GPUs, 768 GByte memory
- 1 huge memory node with 8x Intel Xeon 8160 processors,
  6144 GByte memory

Cloud Computing

Usage 2019/2020



up to 2.6 G core-h / year

main scientific areas

- 28% CFD / engeneering
- 20% elementary particle physics
- 18% astrophysics
- 8% condensed matter

System Access
Test Accounts and Compute Projects

Scientists in Europe are eligible, proposals for computing time are reviewed.

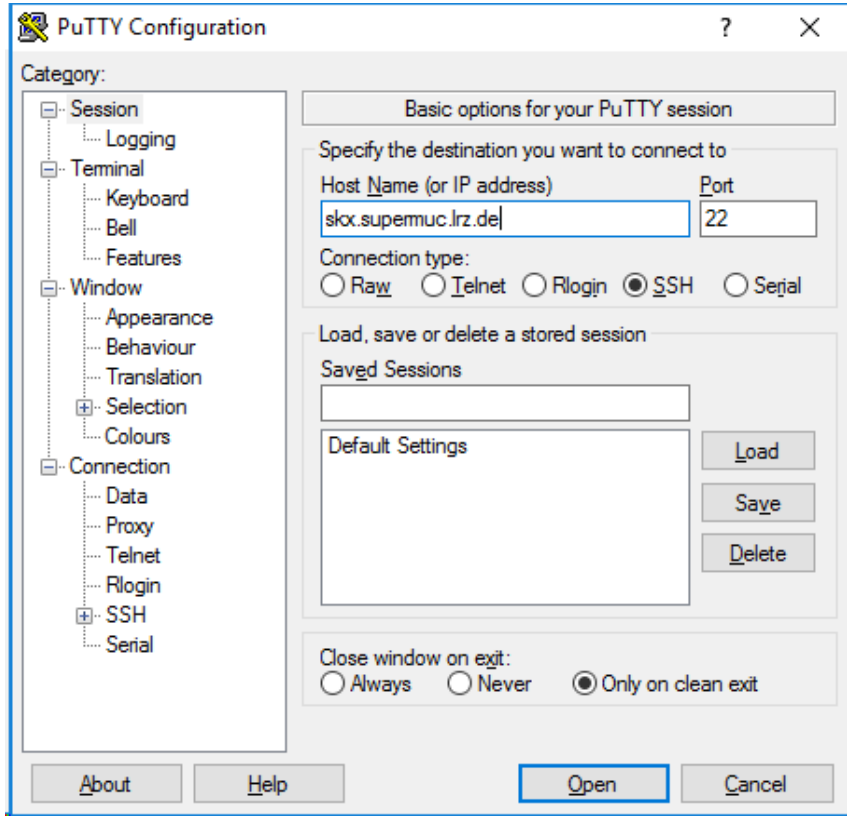
- Scientists with affiliation in Germany
  https://doku.lrz.de/display/PUBLIC/Access+and+Login+to+SuperMUC-NG
- European scientists apply via
  https://prace-ri.eu/hpc-access/preparatory-access/
  https://prace-ri.eu/hpc-access/project-access/

Login onto the system via secure shell from
a terminal window (Linux/macOS)

```
:~> ssh -Y skx.supermuc.lrz.de -l xxyyyzz
```

On Windows systems, install putty https://www.putty.org/

System used today: Meggie@RRZE (instructions in the next talk)



Environment Modules

- Used by most centres
- Variant: lmod
- provide standard software environment by the computing centre
- modular installation of software
- provide multiple versions of a package/compiler/library
- central software installation

Which modules are loaded?

```
:~> module list
Currently Loaded Modulefiles:
 1) admin/1.0                  5) intel/19.0.5
 2) tempdir/1.0                6) intel-mkl/2019.5.281
 3) lrz/1.0                    7) intel-mpi/2019.7.217
 4) spack/staging/20.1.1
```

Good practise

Use `module list` in batch scripts and compile scripts to document run/compile conditions.

➡️ helps debugging problems

Environment Modules

Which modules are provided?

```
# complete list
:~> module available
/lrz/sys/spack/.../linux-sles15-skylake -
abinit/8.10.3-intel19-impi
adios/1.13.1-gcc8-impi
adios2/2.5.0-intel19-impi
bigmpi/0.1-intel19-impi
blitz/1.0.1-gcc8
...

# selected package
:~> module av likwid
-/lrz/sys/spack/.../linux-sles15-skylake -
likwid/4.3.3-gcc8-msr     likwid/5.0.1-gcc8-msr
likwid/4.3.3-intel19-msr  likwid/5.0.1-intel19-msr

------ /lrz/sys/.../files_sles15/tools -------
likwid/4.2  likwid/4.3  likwid/4.3-perf
```

Find module by keyword

```
:~>  module search compiler
- /lrz/sys/spack/.../linux-sles15-x86_64 -
          gcc/8.4.0: Compilers:GNU compiler collection:GCC
          gcc/9.3.0: Compilers:GNU compiler collection:GCC
       gcc/9.3.0-nv: Compilers:GNU compiler collection:GCC
       intel/19.0.5: Compilers:HPC languages:Intel Fortran/C/C++
       intel/19.1.1: Compilers:HPC languages:Intel Fortran/C/C++
         llvm/8.0.0: compiler:clang:llvm
         llvm/9.0.0: compiler:clang:llvm
matlab-mcr/R2019a-generic: applications:scientific programming
           framework:MATLAB compiler runtime
matlab-mcr/R2019b-generic: applications:scientific programming
           framework:MATLAB compiler runtime>
        perl/5.30.0: Compilers:tools:perl interpreter:scripting
```

Environment Modules

What does a module do?

```
:~> module show elpa
-------------------------------------------------------------------
/lrz/sys/spack/.../elpa/2019.11.001-intel19-impi-openmp:

conflict        elpa
prepend-path    PATH /lrz/sys/spack/.../elpa/2019.11.001-intel-qztc5oz/bin
prepend-path    MANPATH /lrz/sys/spack/.../elpa/2019.11.001-intel-qztc5oz/share/man
prepend-path    LD_LIBRARY_PATH /lrz/sys/spack/.../elpa/2019.11.001-intel-qztc5oz/lib
setenv          ELPA_BASE /lrz/sys/spack/.../elpa/2019.11.001-intel-qztc5oz
setenv          ELPA_SHLIB {-L/lrz/sys/spack/.../elpa/2019.11.001-intel-qztc5oz/lib -lelpa_openmp}
setenv          ELPA_INC -I/lrz/sys/spack/.../elpa/2019.11.001-intel-qztc5oz/include/elpa_openmp-
2019.11.001/
setenv          ELPA_WWW http://elpa.mpcdf.mpg.de/html/Documentation/ELPA-2017.11.001/html/
module-whatis   {Libraries:elpa:eigenvalue:Eigenvalue SoLvers for Petaflop-Applications}
```

`prepend-path:` extend path variable

`setenv:` provide environment variable

Hint: Use module-provided variables in makefiles, the command line and in shell scripts!

Environment Modules

Loading and unloading

```
:~> module av gcc
--- /apps/modules/data/development -----
gcc/4.9.4 gcc/5.4.0 gcc/6.2.0 gcc/7.3.0 gcc/8.1.0

# load new module
:~> module load gcc
:~> module list
Currently Loaded Modulefiles:
 1) gcc/8.1.0
# change current module (unload & load)
:~> module switch gcc gcc/7.3.0
:~> module list
Currently Loaded Modulefiles:
 1) gcc/7.3.0
# unload the module
:~> module unload gcc
:~> module list
No Modulefiles Currently Loaded.
```

Aliases and defaults

```
:~> module alias
--- Aliases ------------------------
intel-mkl/2019 -> intel-mkl/2019.5.281
intel-mkl/2019-gcc8 -> intel-mkl/2019.5.281
intel-mkl/2019-seq -> intel-mkl/2019.5.281
intel-mpi/2019-gcc -> intel-mpi/2019.7.217
intel-mpi/2019-intel -> intel-mpi/2019.7.217
intel/19.1 -> intel/19.1.1

---------- Versions ----------------
admin/default -> admin/1.0
allinea-reports/default -> allinea-reports/18.3
amber/default -> amber/18
amira/default -> amira/2019.3
```

depends on the local module-file setup

Tasks for the hands-on

1. Log on to the `meggie` system @RRZE.
2. Which modules are automatically loaded?
3. Identify available compilers.
4. Why are module files for intel compilers in different directories?
5. Find the module of the latest Intel MKL version.
6. How can you link MKL as a shared library with thread support?

Advanced:

6. Modify your ~/.profile to automatically load the intel compiler on login.
7. Why is more than one module loaded after login?
8. Use `module initadd` and `module initrm` to modify the list of automatically loaded modules.