Leibniz Supercomputing Centre
of the Bavarian Academy of Sciences and Humanities

# Leibniz Supercomputing Centre

Software Provisioning at LRZ | Nisarg Patel

# Motivation

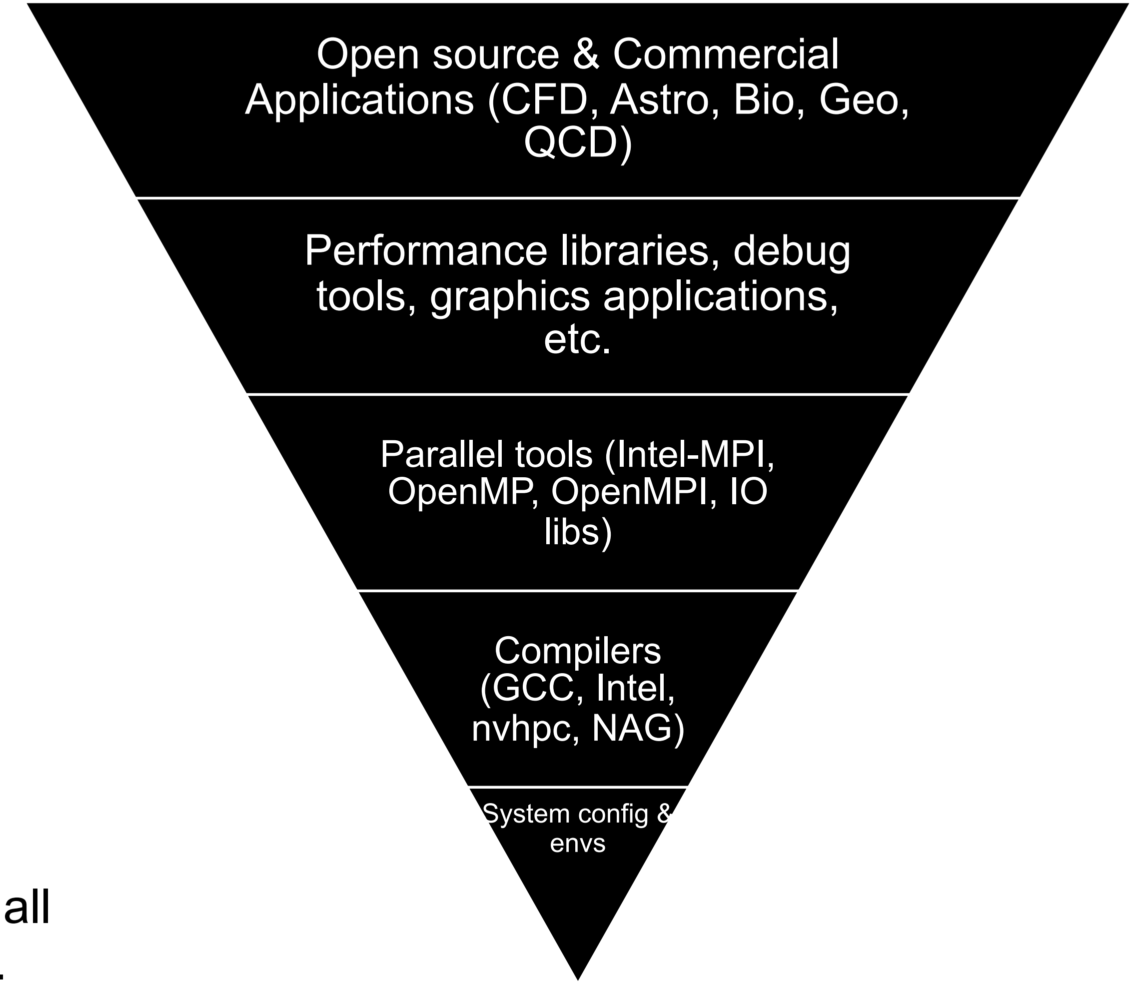
- Software stack on compute resources
- How do we manage HPC Software at LRZ
- General info about Software Stack
- Flow chart to get my application ready for HPC
- Present and Future developments w.r.t. Software Provisioning

# Software stack on compute resources

- In total about ~400 environment modules on each of these machines,
  - SuperMUC-NG
  - CoolMUC-2
  - CoolMUC-3
  - Housing Clusters

- Libraries and applications are build for,
  - Specific architectures
    - Skylake
    - Haswell
    - KNL
  - General build
    - x86_64

- Baring most commercial applications, **about >90%** of all software/libraries are provided with the help of **Spack**.



# How do we manage HPC Software at LRZ

- We manage software either,
  - Manually by each individual application maintainer
  - In semi-automated manner, using Spack

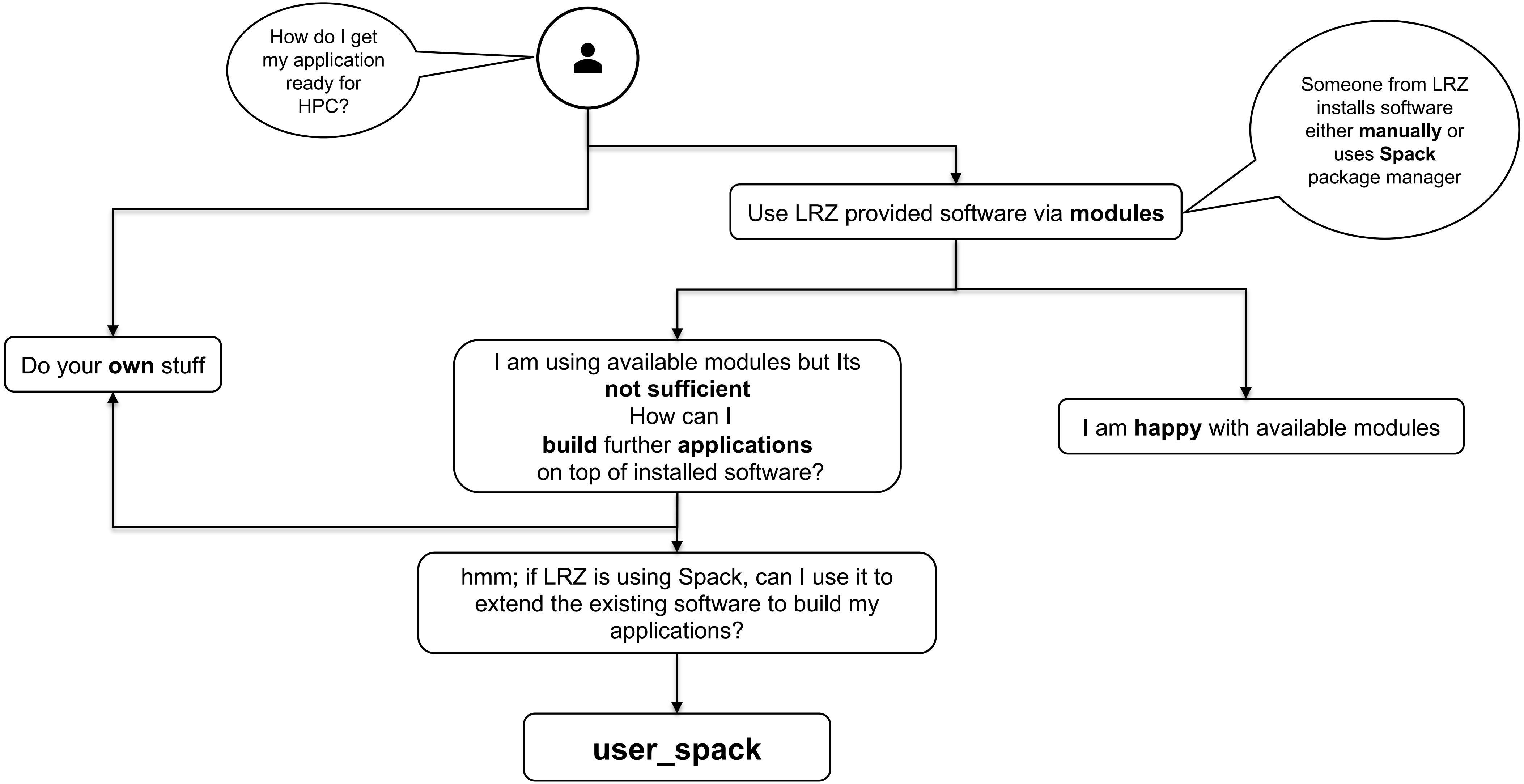
- Software managed using Spack
  - Core group manages Software Stack with the contribution from most Application maintainers.

- Each module that you see on LRZ systems, someone from CXS or HPC department is responsible
  - for managing the state of installations
  - providing modules
  - support users with individual software

# Inside of Software Stacks

- A few Compilers are supported on LRZ systems (GNU, Intel, NAG, NVHPC, LLVM)
- MPI (such as Intel-MPI and OpenMPI) are supported.
- Performance libraries and high level packages are built with architecture-specific optimization flags.
- “user_spack” available for each software stack to build on top of ~5000 pre-installed software
- Modules are automatically generated, with many LRZ system specific additions
- Modules are provided to users as a flat-view
- We provide a new software stack (usually ones in 12 months); depending on the need (This will change in future)
- We manage the software stack on SuperMUC-NG, CoolMUC-2 & -3, and a few housing clusters via a repository hosted on GitLab.

# Flow chart to get my application ready for HPC



# Screenshot of modules in the software stack

```
------------------------------------------------------------------ /lrz/sys/spack/release/23.1.0/modules/compilers ---------------------------------
gcc/8.5.0  gcc/10.4.0  gcc/12.2.0    intel/19.1.2  intel/2021.4.0  intel/2023.1.0  llvm/10.0.1  nag/7.1     nvhpc/22.11  nvhpc/23.3
gcc/9.5.0  gcc/11.3.0  intel/18.0.5  intel/20.0.4  intel/2022.2.0  llvm/9.0.1      llvm/16.0.2  nvhpc/22.9  nvhpc/23.1

--------------------------------------------------------------------- /lrz/sys/spack/release/23.1.0/modules/MPI ------------------------------------
intel-mpi/2018.4.274  intel-mpi/2019.12.320  intel-mpi/2021.6.0  intel-mpi/2021.7.0  intel-mpi/2021.9.0

----------------------------------------------------------------- /lrz/sys/spack/release/23.1.0/modules/MPI-haswell --------------------------------
openmpi/3.1.6-gcc12     openmpi/3.1.6-nag7.1  openmpi/4.0.7-intel23  openmpi/4.1.5-gcc12    openmpi/4.1.5-nag7.1
openmpi/3.1.6-intel23  openmpi/4.0.7-gcc12   openmpi/4.0.7-nag7.1   openmpi/4.1.5-intel23

------------------------------------------------------------------- /lrz/sys/spack/release/23.1.0/modules/x86_64 -----------------------------------
ace/7.1.0           cmake/3.26.3           git/2.40.0             libtool/2.4.6                 ncurses/6.4                    plplot/5.15.0
anaconda3/2020.02   coccinelle/gh-201904   gmake/4.4.1            libtool/2.4.7                 openjdk/1.8.0_202-b08          prng/3.0.2
anaconda3/2022.10   cube/4.8               gmp/6.2.1              ltrace/0.7.3                  openjdk/11.0.17_8              protobuf/3.22.2
autoconf/2.69       cuda/11.8.0            gnuplot/5.4.3          m4/1.4.18                     parallel/20220522              py-pymol/2.5.0
autoconf/2.71       cuda/12.0.0            gnuplot/5.4.3-X11      m4/1.4.19                     paraview-prebuild/5.6.0        py-testing-lrz/0.2.0
automake/1.16.1     doxygen/1.9.6          grace/5.1.25           matlab-mcr/R2022a_Update5     paraview-prebuild/5.6.0_mesa   python/3.7.16-base
automake/1.16.5     dyninst/12.3.0         imagemagick/7.0.8-7    matlab-mcr/R2022b_Update5     paraview-prebuild/5.8.0        python/3.7.16-extended
autotools/v1        emacs/27.2-console     intel-mkl/2020.4.304   matlab-mcr/R2023a_Update1     paraview-prebuild/5.8.0_mesa   python/3.8.16-base
binutils/2.31.1     emacs/27.2-gtk         intel-mkl/2021.4.0     matlab/R2022a_Update5-generic paraview-prebuild/5.10.0       python/3.8.16-extended
bison/3.8.2         emacs/28.2-console     intel-mkl/2023.1.0     matlab/R2022b_Update5-generic paraview-prebuild/5.10.0_mesa  python/3.10.10-base
cgdb/0.8.0          emacs/28.2-gtk         intel-toolkit/2021.4.0 matlab/R2023a_Update1-generic paraview-prebuild/5.11.0       python/3.10.10-extended
charliecloud/0.30   flex/2.6.3             intel-toolkit/2023.1.0 mercurial/5.8                 paraview-prebuild/5.11.0_mesa  qt/5.15.9-gl
charliecloud/0.32   gaussian/16-C.02       jmol/14.31.0           miniconda3/22.11.1            perl/5.36.0                    qt/5.15.9-gl-gtk
clingo/5.4.0        gdal/3.6.4             libszip/2.1.1          molden/6.7                    pkg-config/0.29.2              redis/7.0.5
cmake/3.14.5        gdb/13.1               libtirpc/1.2.6         nano/7.2                      pkgconf/1.8.0                  scons/3.1.2

------------------------------------------------------------------- /lrz/sys/spack/release/23.1.0/modules/haswell ----------------------------------
abinit/9.4.2-intel23-impi       gromacs/2022.5-intel       metis/5.1.0-intel23-i64-r64             parmetis/4.0.3-intel23-impi-
abinit/9.8.3-intel23-impi       gromacs/2022.5-intel-r64   mpfr/4.2.0-gcc12                        parmetis/4.0.3-intel23-ompi-
adios/1.13.1-gcc12-impi         gromacs/2022.5-plumed      mpi-bash/1.3-gcc12-impi                 parmetis/4.0.3-intel23-ompi-
adios2/2.9.0-intel23-impi       gromacs/2022.5-r64         mpifileutils/0.11.1-intel23-impi        petsc/3.19.1-gcc12-ompi-real
amber/18-gcc8-impi              gromacs/2023.1             mumps/5.5.1-gcc12-impi-openmp-shared    petsc/3.19.1-intel23-ompi-re
```

# Introducing the New Software Stack: Spack/23.1.0

- Today we roll out the a software stack, Spack/23.1.0.

- Highlights
  - As with previous software stack updates, this release brings new versions of various packages, ensuring you have access to the latest features and improvements.
  - LLVM based drivers (e.g., icx, ifx, etc.) are provided by default with Intel modules.
- Improved Module Interactions
  - We have made a few changes to enhance the maintainability and long-term support of the software stack, particularly in terms of module interactions.
    - Such as removing redundant module suffixes, adding required prerequisites, etc..
- Bundle Modules
  - We are supporting a few bundle modules that combine commonly used software into a single module.
  - Such as
    - python-base & python-extended
    - intel-toolkit
    - autotools
- Enhanced Documentation
  - To assist users in utilizing the software effectively, documentation and usage instructions will be directly accessible through the command: $> module help <package_name>.

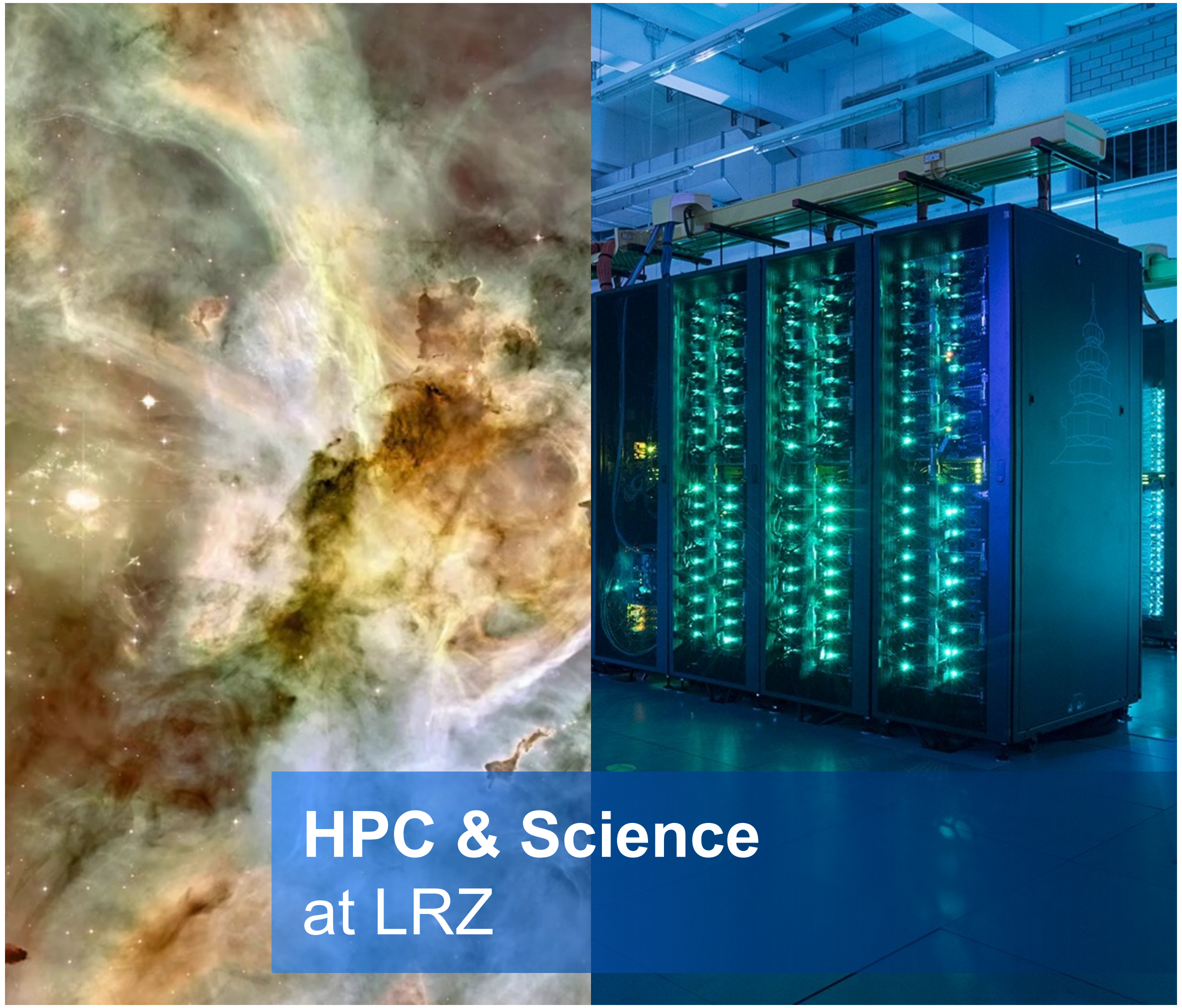
# Present and Future developments w.r.t. Software Provisioning

- We care about user experience that is why we work hard towards improving the software side of things
  - Developing a comprehensive test & validation framework to cross check the installation and performance
  - We will start collecting usage of modules, so we can focus better on most used software
  - Unified documentation, Providing updated documentation on a webpage and also in the terminal.
  - Automated software roll outs using CI/CD
- For all the points mentioned above, and many more, we need you!
  - We need your feedback. Please write us with your questions, suggestions or even criticism.
  - Help us improve your LRZ HPC usage as smooth as possible.

# Leibniz Supercomputing Centre

Scivis Software and High-level Support | Salvatore Cielo

# The LRZ **CXS** group and **Application Labs**




Supporting basic **research**:

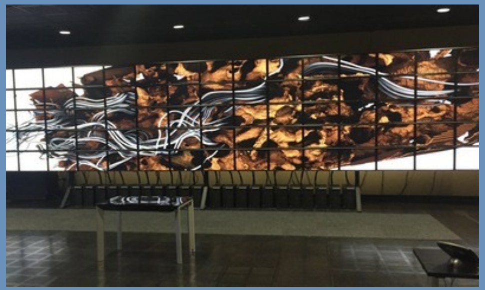
- GCS/PRACE mentoring
- Maintaining **software stack** at LRZ
- HPC **courses** (scivis, parallel coding, ... )
- Collaborations for **code modernization**

# Parallel and Scalable

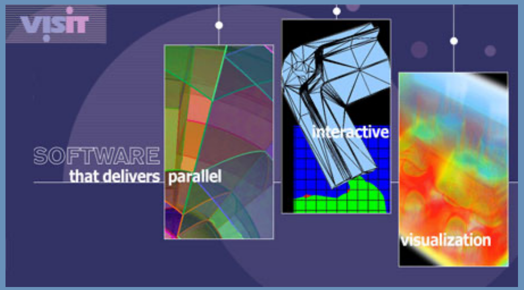
https://doku.lrz.de/display/PUBLIC/VisIt

## VisIt + Intel OSPRay …

- Now also public, since VisIt 3.0 (now 3.3.1)
- Scalable on non-accelerated HPC systems
- TBB workers + multi-node (via MPI)
- Interactive GUI or batch, cross-code



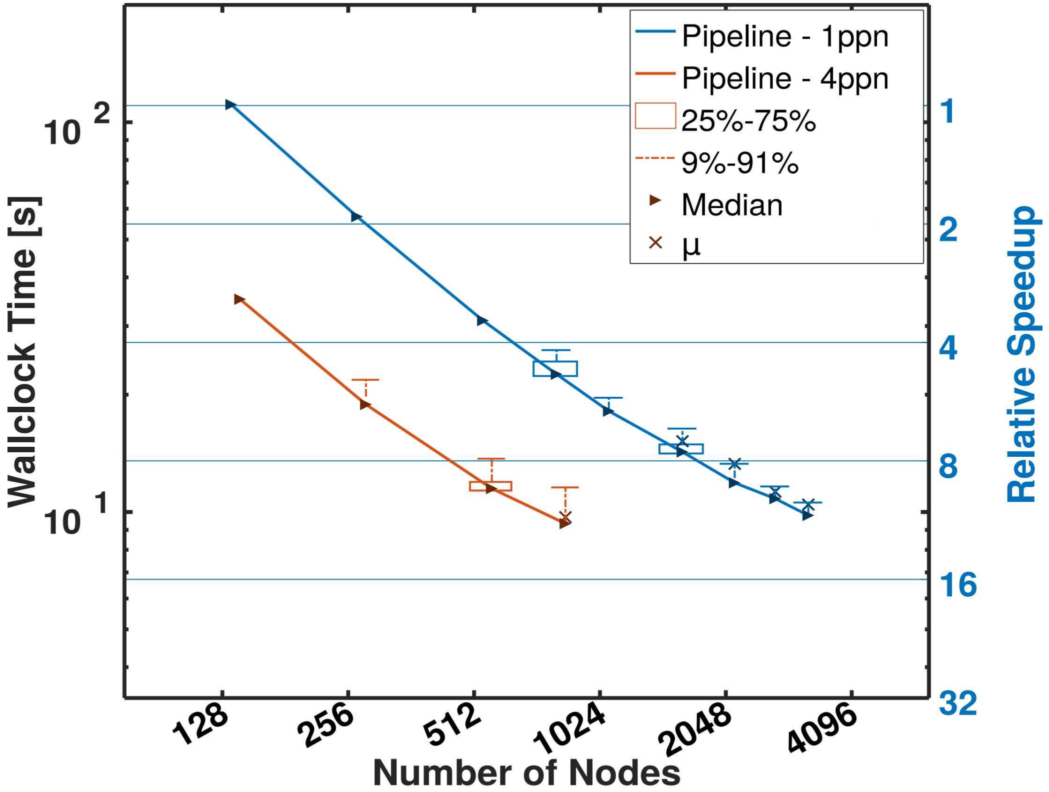
OSPRay: **O**pen, **S**calable, **P**ortable, **Ray** Tracing Engine for High-Fidelity Visualization.




**VisIt** is an **open source**, interactive, scalable, multiplatform visualization, animation and analysis tool.

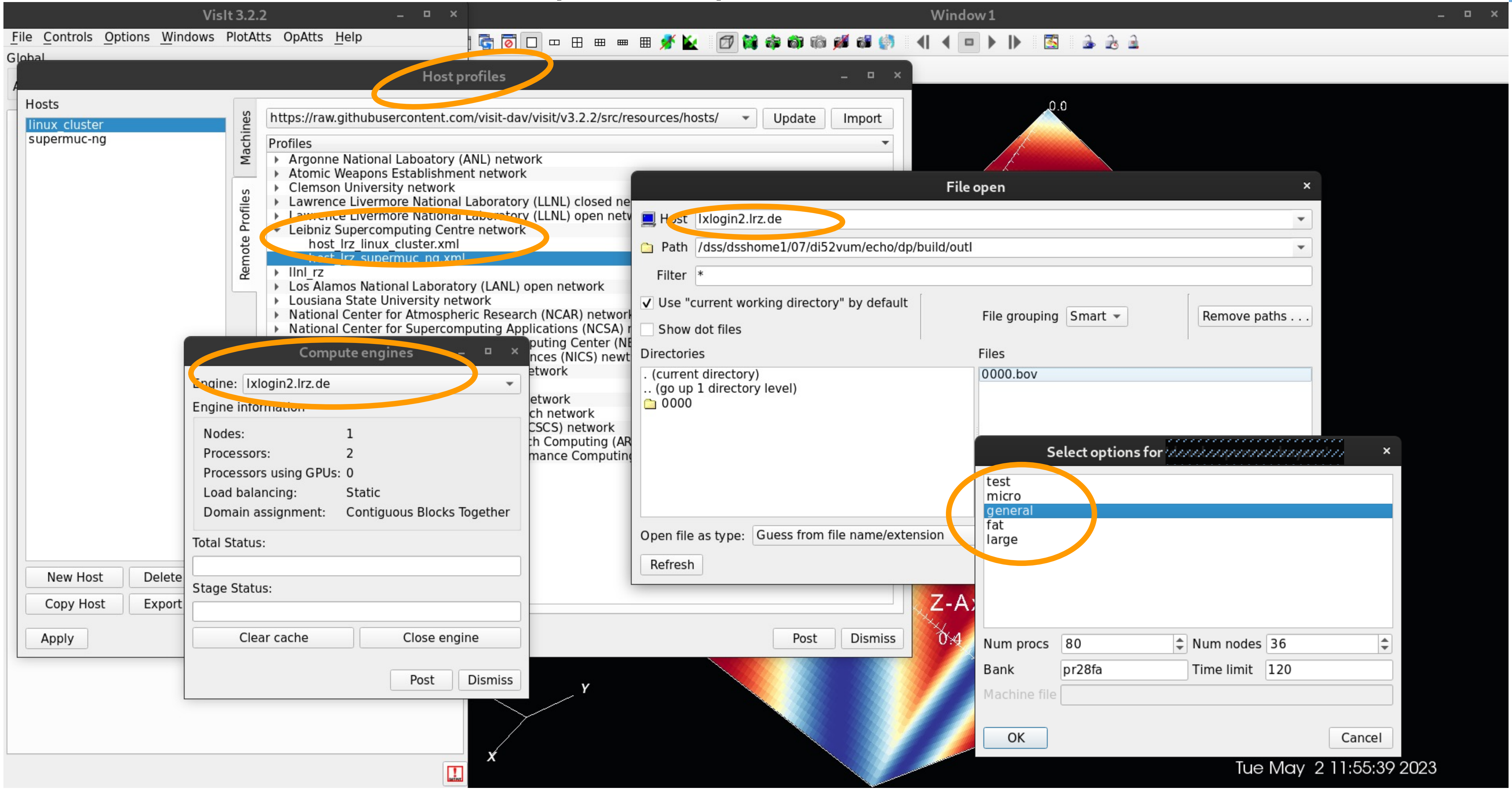
## … on SuperMUC-NG

- In the software stack, for all users
- Strong-scaling to half machine (VisIt)

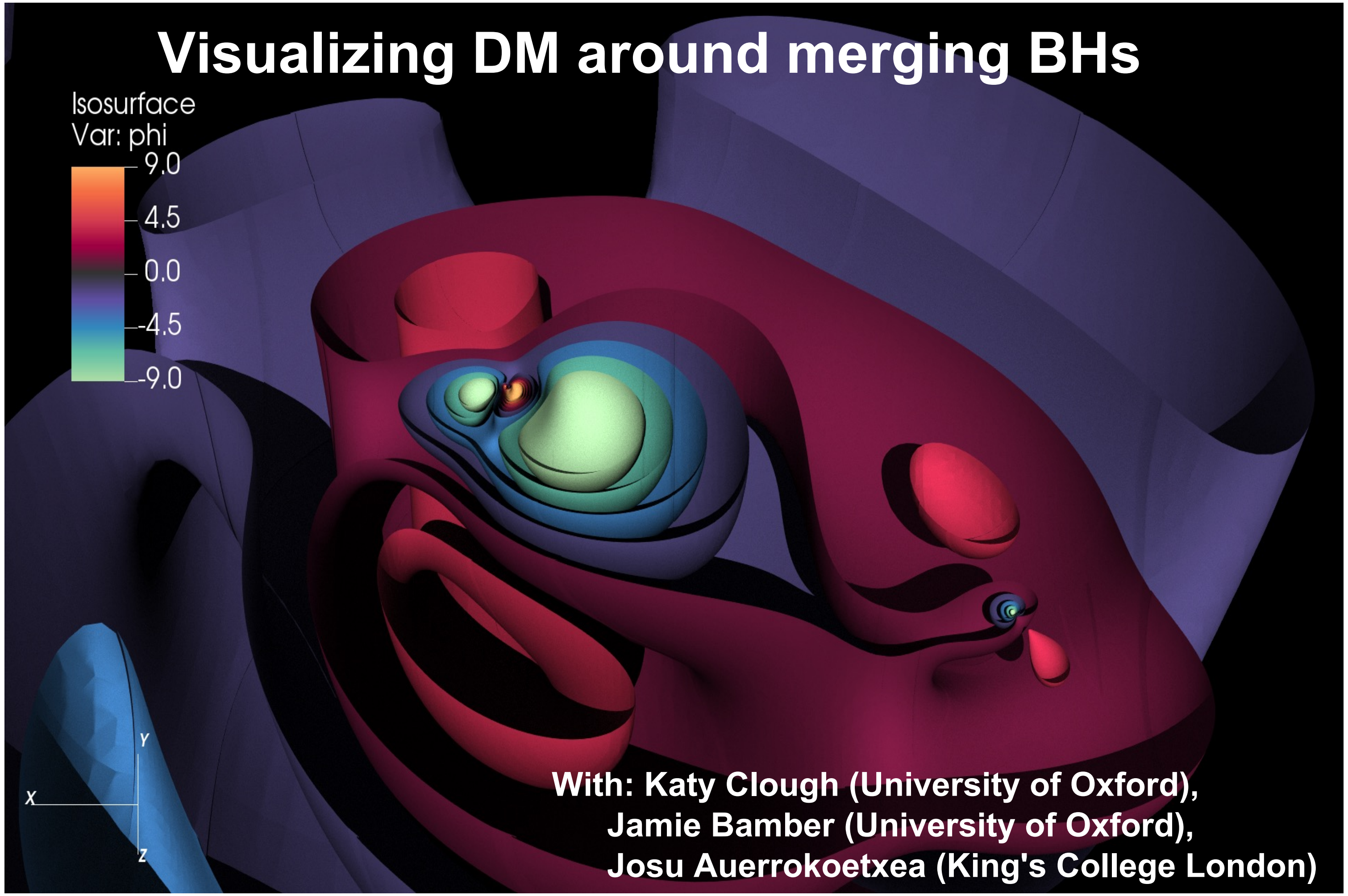


## Scivis Software

# VisIt remote GUI on LRZ Supercomputers

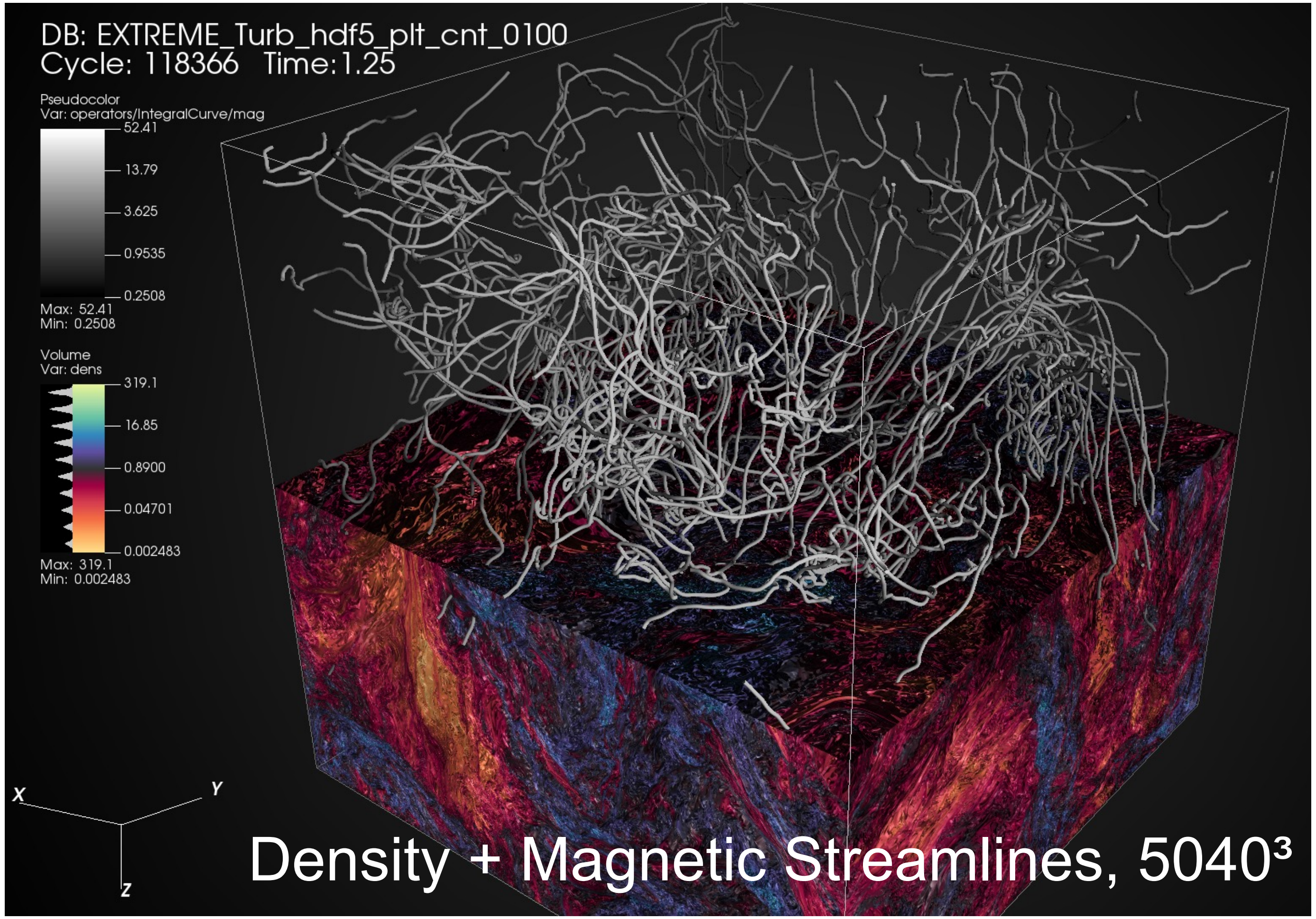


# Scientific Visualization Highlights with LRZ V2C

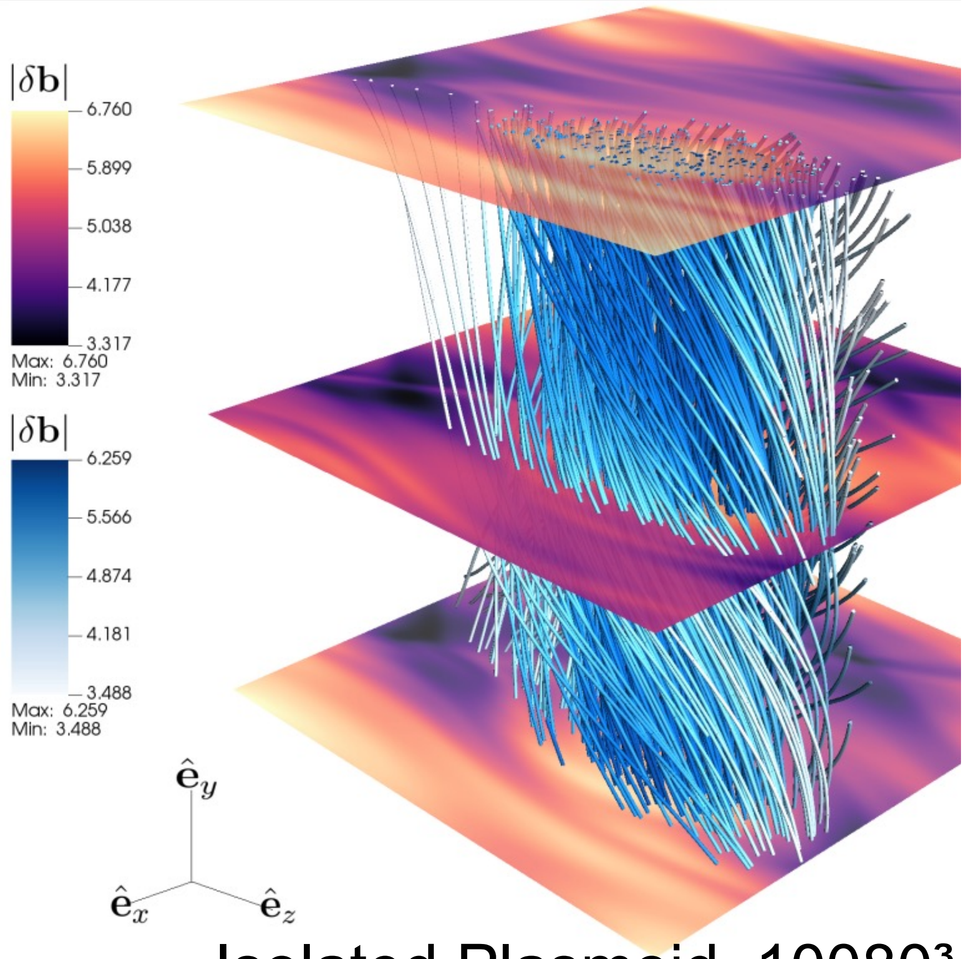



# Visualizing Magnetic field structures

**(Beattie et al., in prep.)**



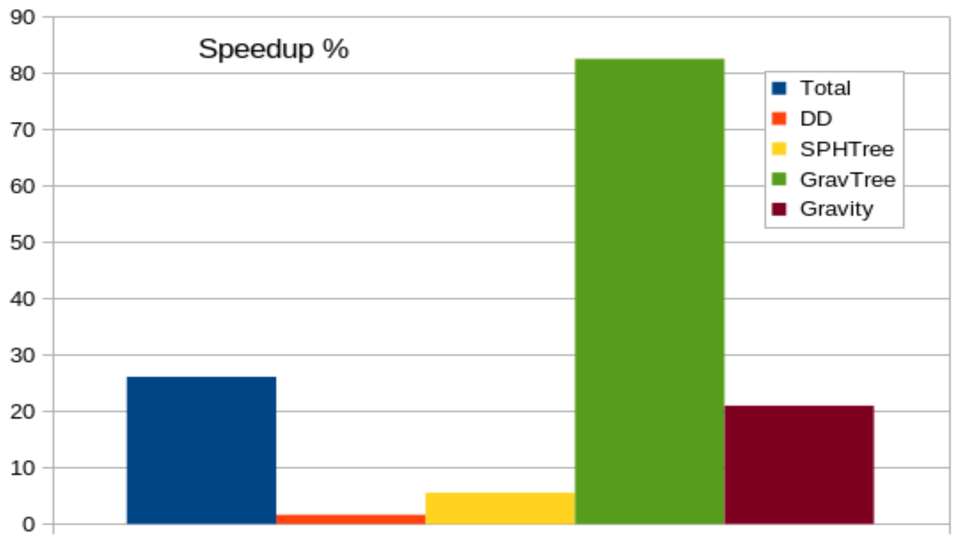

Density + Magnetic Streamlines, $5040^3$

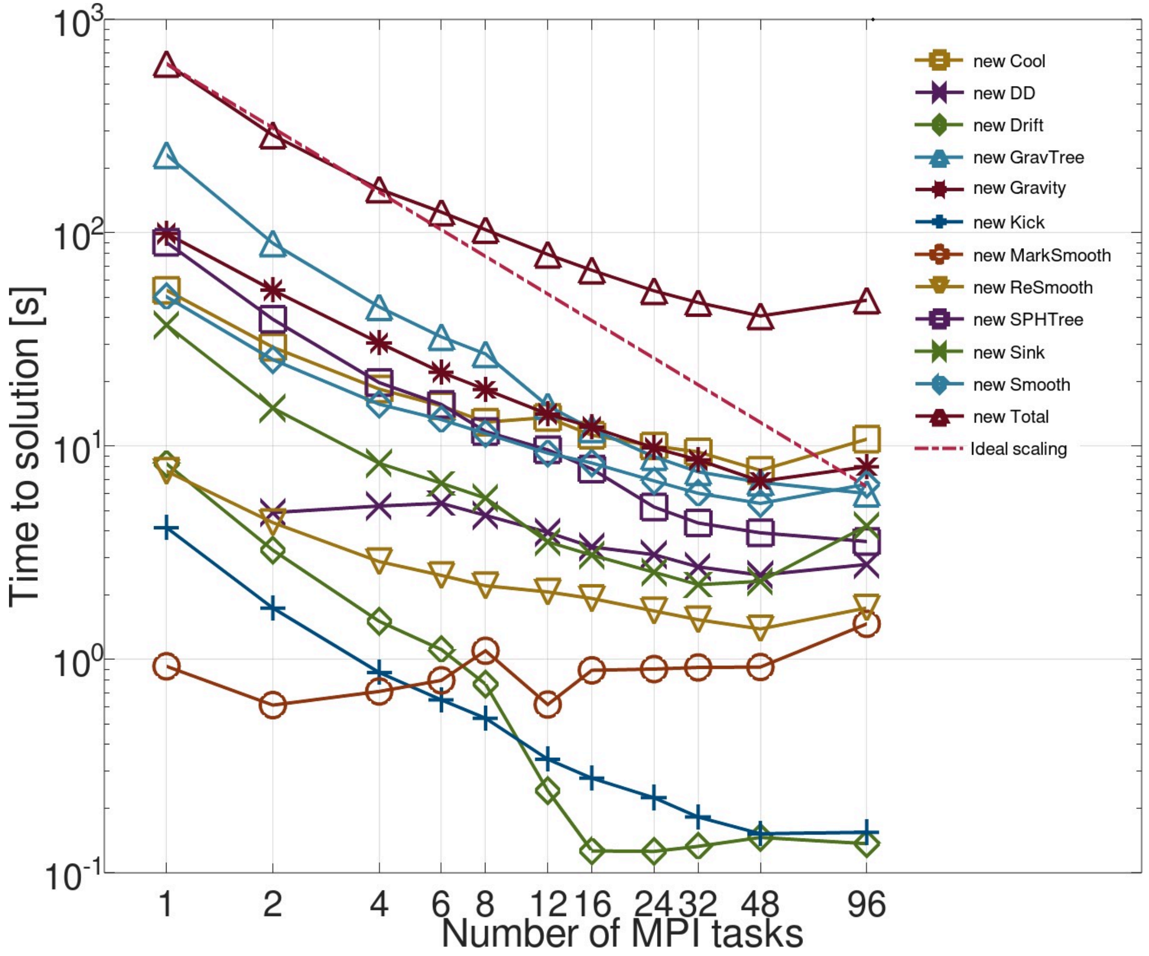



Isolated Plasmoid, $10080^3$

# Modernizing the GASOLINE SPH code

- Correctness, debugging
- Removal of nested pointers
- Domain decomposition over work size
- Improved multipole calculation, order and scheme
- Optimized radiation treatment within tree

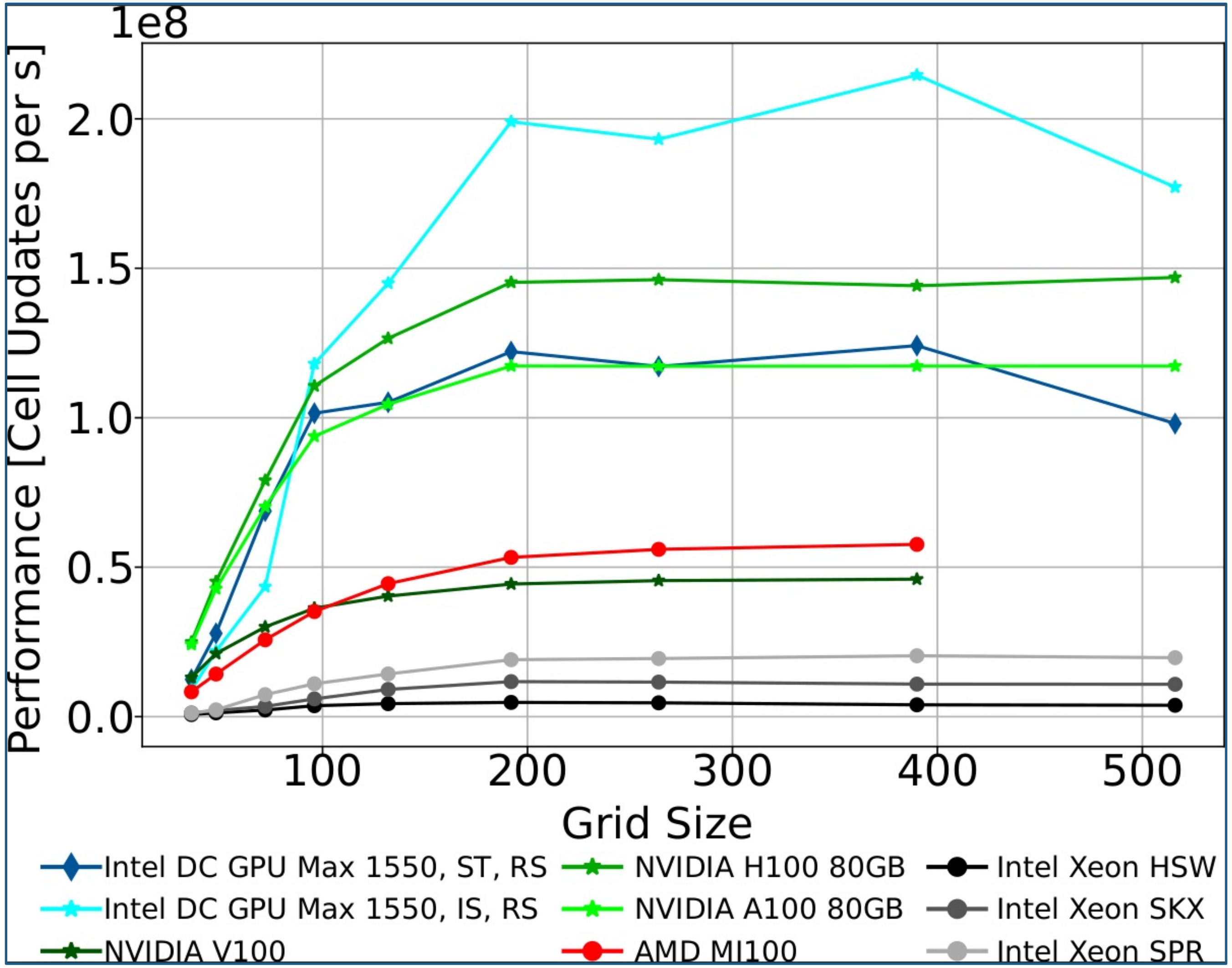



**By:** Jonathan Coles, Aura Obreja, Tobias Buck, Salvatore Cielo, Christoph Pospiech

# DPEcho: SYCL porting of GR-MHD simulation code

- Classic and relativistic MHD supported, both in Minkowski or any general relativistic metric
- Instabilities, turbulence, stellar winds and magnetospheres, disk accretion onto BHs and jets, MHD waves
- SYCL / Intel DPC++ with MPI, CMake
- Improved performance on CPU + GPU. Targeting next-gen Intel GPU (PVC +)




**Developers:** S. Cielo, A. Pöppl, M. Egelhofer, L. Del Zanna (University of Florence), M. Bugli (CEA-Saclay)

# Resources

lrz.de

# Questions?