

Bits & Bytes

No.219, August 2025

Computer Bulletin of the Max Planck Computing and Data Facility (MPCDF)*
<https://docs.mpcdf.mpg.de/bnb>

High-performance Computing

HPC system *Viper*

The new HPC system *Viper* was deployed in two phases, *Viper-CPU* and *Viper-GPU*. *Viper-CPU* has been in stable operation and fully utilized for over a year. The machine recently passed a number of final robustness and performance tests. These included a High-Performance Linpack (HPL) benchmark which delivered 4.3 PFlop/s, and a successful verification of the contracted application performance, based on the MPG benchmark suite consisting of major HPC application codes developed and used in the MPG. The machine is now formally accepted, resulting in the removal of the formal warning of “user-risk” mode operation.

The second part of the deployment, *Viper-GPU*, has been in early-operation mode since February, and was successfully benchmarked and ranked in the June 2025 issue of the Top500 list¹ with an HPL performance of 31.1 PFlop/s using all 300 nodes (600 AMD MI300A APUs). The Slurm-based scheduling and accounting was integrated with *Viper-CPU* during a maintenance at the end of July. This also marked the beginning of official accounting for *Viper-GPU*, with a weighting factor of two to be applied per node-hour on *Viper-GPU* relative to *Viper-CPU*. This part of the machine will continue to operate in “user-risk” mode until it passes final robustness and performance tests, which are planned for the next

few weeks.

MPCDF continuously updates the software stack on both *Viper* machines and provides application support for users, in collaboration with software engineers and application experts from AMD and Eviden.

Markus Rampp

Using Visual Studio Code (VSCode) with the Remote-SSH extension

Visual Studio Code (VSCode) is a popular code editor that can be enhanced with a wide variety of extensions. For working with MPCDF systems, the “Remote-SSH” extension is particularly useful. It allows users to connect to a remote machine over SSH, enabling them to edit files and run commands directly on the remote system as if it were local, combining the convenience of a local editor with the capabilities of the remote HPC environment.

Users can find up-to-date instructions on the necessary configuration for Linux and MacOS as part of the MPCDF FAQ². Please be aware that we cannot guarantee that the Remote-SSH extension will work in all cases due to the plethora of specifics in the configurations, operating systems, and versions across users.

Klaus Reuter

Software News

New ELPA version 2025.06.001

The latest ELPA eigensolver library³ (version 2025.06) delivers significant performance enhancements, particularly for GPU-based computations at MPCDF. Notably, the new release features improved support for AMD GPUs via ROCm 6.4.2 recently made available on *Viper*. Op-

timized rocSOLVER functions provided by this ROCm update achieve up to a 5x performance boost when using ELPA with RCCL (ROCm Collective Communication Library). Additionally, general GPU performance has increased by up to 15% for both standard and generalized eigenproblems.

*Editors: Dr. Renate Dohmen & Dr. Markus Rampp, MPCDF

¹<https://www.mpcdf.mpg.de/115266/news20250610-viperintop500>

²<https://docs.mpcdf.mpg.de/faq/connecting.html#how-can-i-connect-to-hpc-systems-and-clusters-using-visual-studio-code-vscode>

³<https://elpa.mpcdf.mpg.de>

This ELPA version also addresses a bug that previously prevented NCCL/RCCL builds from handling more than one MPI process per GPU in generalized eigenproblems and auxiliary ELPA routines.

Furthermore, ELPA is now available on *Raven* with a specialized toolchain combining the Intel compiler, CUDA, and excluding NCCL support, which is especially suitable for applications such as FHI-aims. To access this new toolchain on *Raven*, load the following modules: `intel/2025.2 impi/2021.16 mkl/2025.2 cuda/12.6` and use the module `elpa/mpi/standard/gpu/2025.06.001`. All other available module combinations can be queried with `find-module elpa.*2025.06.001`.

Petr Karpov, Tobias Melson, Andreas Marek

Compilers

New compilers have been made available on *Raven*, *Viper-CPU*, *Viper-GPU*, and other clusters: the GNU compiler collection 15.1 and Intel oneAPI 2025.2. The former can be used by loading the module `gcc/15`. The latter provides the compiler module `intel/2025.2`, the corresponding MPI module `impi/2021.16`, the MKL module `mkl/2025.2`, and modules for the Intel profiling tools. As usual, the full scientific software stack is compiled with these toolchains.

Tobias Melson

Water Boa Python 2025.06

To address a significant change in the licensing of the popular Anaconda Python Distribution, MPCDF has been providing a free drop-in replacement for a scientific Python software stack since June 2024, labeled “Water Boa Python”. This ensures that all users have access to a comprehensive scientific Python environment without licensing concerns.

Recently, version 2025.06 was rolled out on the HPC systems and clusters. It is built entirely from the open-source `conda-forge` channel and offers a robust and up-to-date collection of packages for scientific computing, data analysis, and visualization. It is based on cPython 3.13. The package list and versions are very similar to those of the commercial counterpart, making it easy to transition existing workflows from Anaconda. On the MPCDF systems, it is available via the environment module `python-waterboa/2025.06`. After loading the environment module, running `conda list` will show the full list of the available Python packages.

We encourage users to adopt Water Boa Python for their scientific workflows. The project’s scripts and package lists are available internally on MPCDF GitLab⁴.

Klaus Reuter

⁴<https://gitlab.mpcdf.mpg.de/mpcdf/water-boa-python>

⁵<https://github.com/sokrypton/ColabFold>

⁶<https://github.com/soedinglab/MMseqs2>

⁷<https://github.com/YoshitakaMo/localcolabfold>

⁸<https://developer.nvidia.com/blog/boost-alphafold2-protein-structure-prediction-with-gpu-accelerated-mmseqs2/>

Accelerated prediction of protein structures and complexes with ColabFold

ColabFold⁵ is a user-friendly tool that streamlines protein structure prediction by combining the capabilities of AlphaFold2 with the rapid Multiple Sequence Alignment (MSA) computation of MMseqs2.

MPCDF’s local installations of ColabFold have recently been upgraded to include GPU-accelerated MSA based on MMseqs2-GPU⁶ for Nvidia hardware. The databases are stored on special NVMe-backed file systems separate from the ‘/ptmp’ and ‘/u’ file systems, to optimize for the intense disk-IO during the MSA phase and to avoid a slowdown of the general-purpose file systems. ColabFold runs locally⁷ on MPCDF resources, without putting load on ColabFold’s public MSA servers.

Users of the plain AlphaFold2 installations might be interested in considering a transition to ColabFold. Information on how to get started is available via the command `module help colabfold/202507` on *Raven* as well as on some institute clusters.

Technical information on how the accelerated MSA works is given in a post on the Nvidia Developer Blog⁸. Please note that the speedups reported in the blog were achieved under ideal conditions on optimized hardware and will be more modest in multi-user HPC environments.

Klaus Reuter

AlphaFold2 available on Viper-GPU

Similarly to the frequently used installations for Nvidia GPUs on *Raven* and on selected clusters, AlphaFold2 is now available on *Viper-GPU* as well. To port the original codebase from Nvidia GPUs to AMD, several core dependencies such as JAX and OpenMM have been replaced with specific builds for the AMD MI300A APUs. Information on how to use AlphaFold2 on *Viper-GPU* can be obtained from running the command `module help alphafold/2.3.2-2025`.

Klaus Reuter

DataShare command-line client (ds/pocli) modernized

In preparation for the migration of the MPCDF DataShare service to the Nextcloud platform (see announcement below), the lightweight command-line client `ds` (also known as `pocli`) has been updated to ensure compatibility with the new backend.

It is crucial that all users of the `ds` client upgrade, as older versions will no longer function correctly after the migration.

- On MPCDF systems, the updated client is provided via the `datashare` environment module. Please ensure you are loading the latest available version of this module.
- On local machines (e.g., laptops), you must upgrade your installation by running the command: `pip install --upgrade pocli`. This will fetch the latest

version from the Python Package Index (PyPI⁹).

Further information including the source code for `pocli` is available on MPCDF GitLab¹⁰.

Florian Kaiser, Klaus Reuter

Using Access Tokens in GitLab

GitLab access tokens (GAT) are an easy and secure way to access your GitLab repositories. They are meant to establish a machine-to-machine communication between GitLab on the one side and scripts or workflow engines on the other one. Think of a GAT as a substitute for your user account, but you can exactly specify in which context the token is allowed to do what.

Beside some special kinds of access tokens, GitLab supports the following main and most important types of access tokens (left side of the graphic):

- **Personal Access Token:** these are the most powerful tokens, as they can act directly “in the name of the user” with all his permissions
- **Group Access Token:** tokens assigned to a GitLab group, allowing for permissions across multiple projects within that group
- **Project Access Token:** created at the project level, allowing restricted access to that specific project

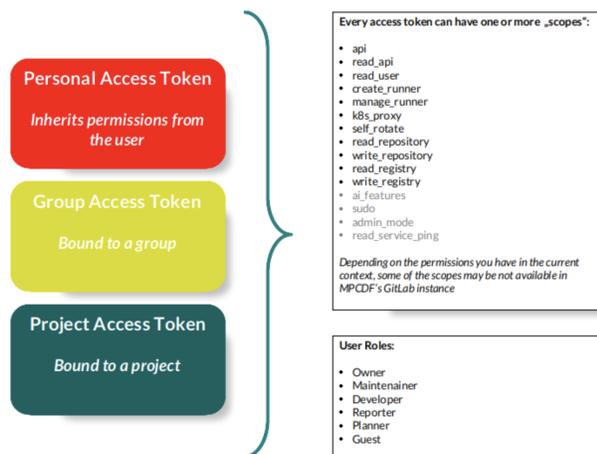


Figure 1: Overview of the GitLab Access Tokens

When creating a token, the user can specify in a fine-grained manner which permissions - so called scopes - the token should have. For example, the scopes `read_repository` and `write_repository` can be used to give the token read/write access to the Git repositories it

belongs to. Depending on the permissions the user has in the current context, also setting the *User Role* of an access token needs to be done accordingly.

When creating an access token in GitLab, all of these three dimensions (type, scope and role) have to be taken into account. Choosing the proper configuration of an access token can be somewhat tricky depending on the use case.

Using access tokens on the command line

One main use case for access tokens is authentication on the command line. This can be either a human interaction (avoiding SSH keys) or a script (maybe as part of a continuous-integration pipeline).

To allow an access token to *clone/pull* and *push* a repository via the HTTPS protocol, the following setup is necessary:

- **Type:** Could be a personal, group or project access token. Preferable is always a token with lowest capabilities.
- **Scope:** `read_registry` (clone, pull) and its analogue, `write_registry` (push) are necessary for accessing a repository.
- **User Role:** As the user roles *guest* and *planner* have no write access to a repository, at least the user role *reporter* needs to be assigned for write permissions.

In addition to these settings, the user can set an expiration date for any token. Once the token is saved in GitLab, the user has only on the resulting web page the possibility to copy the token to the clipboard. Once the web page is closed, GitLab will never show the token again. During its life time, the token can be used on the command line, in continuous-integration pipelines or scripts to act as a proxy for the user himself. When asked for username and password, the user's account name and instead of his real password the token can be used.

More information on GitLab's access token system can be found in the official documentation¹¹.

Thomas Zastrow

⁹<https://pypi.org/project/pocli/>

¹⁰<https://gitlab.mpcdf.mpg.de/mpcdf/pocli>

¹¹<https://docs.gitlab.com/security/tokens/>

News/Announcements

DataShare: Migration to Nextcloud

The MPCDF DataShare sync&share service based on the *ownCloud* product has been in operation for more than 10 years, being used daily by hundreds of users to store, edit and share their documents and data. Already in 2016, ownCloud was forked into a new product called *Nextcloud*. Both, ownCloud and Nextcloud co-existed in parallel for most of the time, focussing on different niches in the market. Recently however, momentum around ownCloud has slowed significantly, with the vendor Kiteworks finally announcing its end of life for the end of 2026. For this reason and after careful deliberation and testing, the MPCDF DataShare service will be migrated to Nextcloud on September 13th and 14th, 2025.

On this weekend, the service will be unavailable!

All data, shares, calendars etc. will be preserved by the migration, and the general functionality and look&feel should remain very similar. Nevertheless, there will be some important changes after the migration:

New desktop and mobile clients: If you used the *ownCloud* or branded *DataShare* client on your device before, you will need to **install the Nextcloud client**¹² instead in order to continue being able to synchronize your data on that device.

You may re-use the ownCloud data directory on your machine in order to avoid downloading all of the files again, provided you **ensure that it is no longer accessed by the ownCloud client** by either uninstalling it, or removing the DataShare account configuration from it.

Shares by expired users no longer accessible: Currently, data belonging to an expired user can still be accessed via public links (e.g. <https://datashare.mpcdf.mpg.de/s/<token>>) or by other DataShare users it was shared with. **After the migration to Nextcloud, this will no longer be the case!** If you or your collaborators are still using data belonging to an expired user, **please transfer it to an active user** if possible and re-share it from there. In special cases where this is not feasible, for example if the shared link must remain the same, contact support@mpcdf.mpg.de for assistance.

Expired accounts will be deleted after 6 months: Expired accounts including all data, shares, calendars etc. associated with them will be deleted after 6 months. Please make sure to transfer data that should be preserved to another user before your account expires. This will be possible via the Nextcloud web interface after the migration.

Pending or rejected shares will not be migrated: Pending shares that you rejected or never accepted will not be migrated, e.g. can no longer be accepted afterwards.

Old style “v1” chunking API no longer supported: The old-style “v1” chunking API¹³ for uploading large files will no longer be supported. This should only impact a small number of users using old versions of the *pocli*, *pyoclient* or similar.

Custom groups will be converted to Nextcloud Teams: Any ownCloud “custom groups” that you may have created will be converted to *Nextcloud Teams*, also called *Circles*. They generally work very similar and can be managed through the new *Contacts* app.

Multi-factor authentication mandatory: Multi-factor authentication via MPCDF Login will become mandatory for enhanced security. This is the same Single sign-on (SSO) login that was already introduced for Gitlab earlier this year, i.e. you will only have to sign in once per day for any of DataShare, Gitlab, or other MPCDF services that will use SSO in the future. In case you have not set up 2FA in our SelfService for other services such as Gitlab already, we recommend doing so any time in advance in order to ensure you will be able to login to DataShare after the maintenance. If you use third-party clients that do not support MFA, you may create device specific passwords¹⁴ for them.

Florian Kaiser, Michele Compostella

A farewell to the AFS cell “ipp-garching.mpg.de”

The Andrew Filesystem¹⁵, commonly known as AFS, is reaching its end of life at MPCDF. Introduced to the Max Planck Institute for Plasma Physics (IPP) by Hartmut Reuter and publicly announced in the Bits&Bytes of June 1993 by the then-director of RZG, Stefan Heinzl, it quickly gained importance at RZG, IPP and the neighbouring institutes at the Garching Campus. AFS had several features of a cloud file system: global access, redundant metadata, and fine-grained access control lists. This allowed users to log in to any machine and find the same home directory. Scientists could write their programs on their office computers and compile them on the systems they were meant to run on without needing to copy them around.

In 1994, Hartmut Reuter introduced Multi-Resident AFS (MR-AFS), which managed the storage of large data not only on disk, but also on tape. At that time (and actu-

¹²<https://nextcloud.com/install/#install-clients>

¹³<https://github.com/owncloud/core/wiki/spec:-big-file-chunking>

¹⁴https://docs.nextcloud.com/server/30/user_manual/en/session_management.html#managing-devices

¹⁵https://en.wikipedia.org/wiki/Andrew_File_System

ally also today), the available disk space was insufficient to store all the experimental data. Therefore, it could move data to a tape backend while still being visible in the filesystem. When a file was accessed, it was automatically copied back to disk. This HSM (Hierarchical Storage Management) system had been in operation until 2008, when it then was replaced by AFS-RXOSD, an RZG/MPCDF in-house development. Later on, IBM's GHI/HPSS took over the role of automatic file transfer to tape, and MPCDF switched back to plain OpenAFS, also since Hartmut had already retired, and further development of AFS-RXOSD had come to an end.

Throughout its lifetime at MPCDF, AFS provided many services: users' home directories, storage for experimental data, and software for clusters and office machines alike. However, all kind of technologies eventually come to an end. The AFS network protocol, amongst other things, no longer meets today's standards.

The AFS cell "ipp-garching.mpg.de" will therefore be set to **read-only** at the end of **November 2025** and will be **finally shut down in November 2026**.

MPCDF would not be where it is today without AFS. Hence, we would like to again thank Hartmut Reuter and all involved colleagues, particularly also from IPP, for their vision and dedication over more than three decades.

Christof Hanke

Multifactor authentication: deactivation of E-mail tokens

All MPCDF services are meanwhile protected by two-factor (2FA) authentication. Until now the following token types can be used: app, external hardware tokens, SMS, E-mail, and TAN list.

To improve overall security, MPCDF will deactivate the usage of E-mail tokens. Since May, the creation of new E-mail tokens has been disabled, but existing E-mail tokens can still be used until end of September 2025. While you still have access to your E-mail token, please make sure that you also have a valid app or hardware token, ideally in combination with an SMS token as backup. See also our documentation on 2FA¹⁶.

In case you loose access to all registered tokens, a token reset process can be triggered through the MPCDF SelfService interface. For security reasons, in order to make sure that your E-mail account has not been hacked, the contact person registered for your account needs to authenticate your identity and confirm your request.

Kathrin Beck, Andreas Schott

Account re-applications

The workflow for MPCDF-account applications and for the reopening of expired accounts has been updated mainly to enable a more comfortable way to reactivate existing accounts.

If an MPCDF user account has been deactivated less than six months ago and the user metadata are still valid, it can easily be reactivated by the user through opening a ticket in the helpdesk. If an account is closed for longer than six months, it should be reopened by filling a new account application instead of sending a ticket. This will update the user's metadata automatically instead of manual adjustment. Thus, users can now submit multiple applications, regardless of whether they have a closed account or a previously rejected account.

Kathrin Beck, Andreas Schott

Export control and updated Terms of Use

Export control regulations impose restrictions on the access to and usage of high-performance computing (HPC) systems whose components (compute nodes) exceed certain performance thresholds. The new AI machine for several Max Planck Institutes¹⁷, which is about to become operational, is the first system of such kind at MPCDF, and we anticipate that more HPC systems at the MPCDF will fall under this category in the future.

Users with access to such systems must now accept an updated version of our Terms of Use. The main relevant change is the new § 7 Sec. 2p), which covers new obligations. In particular, users are required *to comply with all currently applicable export control regulations under EU and national law, including sanctions and embargoes, and, in the event of intended access from outside the European Union, have a corresponding export control review carried out by the export control authorities in good time prior to access and, if approval by the competent authorities is required, only access the data after such approvals have been obtained*. Details can be found in the OHB of MPG¹⁸.

In connection with the new Terms of Use, we have introduced a new workflow for accepting the Terms of Use. Users must accept the new Terms of Use in the MPCDF SelfService within one month of their release. If this does not happen, the password's lifetime will be shortened to one month, during which time the user can accept the Terms of Use and extend the password for another year. Furthermore, any extension or change of the password will require re-acceptance of the current Terms of Use.

Kathrin Beck, Andreas Schott

¹⁶<https://docs.mpcdf.mpg.de/faq/2fa.html>

¹⁷<https://docs.mpcdf.mpg.de/bnb/217.html#a-shared-ai-system-for-eleven-mpis>

¹⁸https://ohb.mpg.de/Policies/MPG%20VII.201%20en%204.0/VII.2.01_Establishment%20of%20an%20MPG-wide%20Export%20Control%20System_en.pdf

Events

WAMTA 2026

MPCDF is co-organizing and hosting the 2026 Workshop on Asynchronous Many-Task Systems and Applications (WAMTA) which will take place from Feb 16-18, 2026 at MPCDF in Garching. The objectives of this workshop are to bring together experts in asynchronous many-task frameworks, developers of science codes, performance experts, and hardware vendors to discuss the state-of-the-art techniques needed to program, analyze, benchmark, and profile these codes to achieve maximum performance possible from modern machines. For the first time in this well-established series, the WAMTA 2026 workshop will be held in Europe. Invited speakers include Rosa M. Badia (BSC-CNS), Michael Klemm (AMD, OpenMP ARB), and Nick Brown (EPCC). Further details, including registration information and the call for papers can be found on the workshop webpage¹⁹.

Erwin Laure, Markus Rampp

MPG-DV-Treffen

Save-the-date: The 42nd edition of the MPG-DV-Treffen is scheduled for September 23-25, 2025, as part of the *IT4Science Days* in Bremen. Details are available from the event page.²⁰

Raphael Ritz

MPG-NFDI and Research Data Management Workshops

Registration is open for the third MPG-NFDI Workshop (October 27-28) providing a forum for MPG researchers engaged in NFDI to exchange experiences and successes. It is followed by the 7th MPG-Workshop on Research Data Management (October 28-30). Both events take place in Leipzig at the MPI for Evolutionary Anthropology. Further information is available from the event page.²¹

Raphael Ritz

Introduction to MPCDF services

The next issue of our semi-annual seminar series “Introduction to MPCDF services” will be given on October 16th, 14:00-16:30 online. Topics comprise login, file systems, HPC systems, the Slurm batch system, and the MPCDF services remote visualization, Jupyter notebooks and DataShare, together with a concluding question & answer session. No registration is required, just connect at the time of the workshop via the zoom link provided on our webpage.

Tilman Dannert

Meet MPCDF

In our “Meet MPCDF” series, two seminars are planned in autumn:

- October 2nd, 15:30-16:30, Diagonalization of Sparse Matrices
- November 6th, 15:30-16:30, Possibilities for Enhanced Security in Gitlab

We encourage our users to propose further topics of their interest, e.g. in the domains of high-performance computing, data management, artificial intelligence or high-performance data analytics. Please send an E-mail to training@mpcdf.mpg.de²².

Tilman Dannert

Python for HPC

On popular demand, the next MPCDF course on “Python for HPC” will be given in November or early December 2025. Presented online via Zoom, the course will cover essential tools and techniques for using the Python ecosystem efficiently on HPC clusters. The exact date will be announced in September on the MPCDF website, together with a detailed list of topics, a schedule, and a registration link.

Tilman Dannert

¹⁹<https://wamta26.stellar-group.org/>

²⁰<https://plan.events.mpg.de/event/474/>

²¹<https://rdm.mpdl.mpg.de/mpdl-services/workshops/7-fdm-workshop-2025-forschungsdatenmanagement-in-der-max-planck-gesellschaft/>

²²<mailto:training@mpcdf.mpg.de>