

Bits & Bytes

No.218, April 2025

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

High-performance Computing

Viper-GPU operational

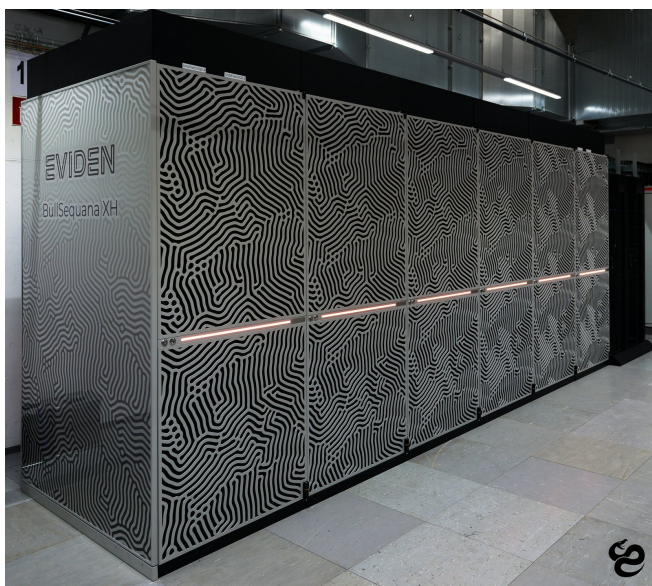


Figure 1: Deployment of *Viper-GPU*. Image credit: K. Zilker, MPCDF

The second, GPU-powered phase of the new supercomputer *Viper* of the Max Planck Society was opened for early user operation (still in “user-risk” mode, and with some hardware and software tuning ongoing) in February 2025. The machine currently comprises 228 compute nodes with two AMD Instinct MI300A APUs (accelerated processing unit), each with 128 GB of high-bandwidth memory (HBM3) which is shared coherently between the 228 CDNA3 GPU compute units and 24 Zen4 CPU cores. Another 72 nodes are added in the course of April. The nodes are interconnected with an Nvidia InfiniBand (NDR 400 Gb/s) network with a non-blocking fat-tree topology, and are connected to online storage based on

IBM Storage Scale (aka GPFS) parallel file systems (`/u` and `/ptmp`) with a total capacity of ca. 11 PB, and a 480 TB NVMe-based filesystem (`/ri`) for projects with read-intensive applications.

A high-level overview of *Viper-GPU* was given in a MeetMPCDF seminar¹ on February 6th. Users can find all relevant technical details on the hardware and software and its usage, including example batch submission scripts, in a comprehensive user guide². Users are invited to register for our AMD-GPU workshop and *Viper-GPU* hackathon in May (see “Events” below).

Among the most notable differences to the *Raven* supercomputer, users should be aware that

- *Viper-CPU* and *Viper-GPU* each have their individual sets of login nodes, file systems, and software (module) stacks
- transitioning from an Nvidia-GPU programming environment requires adapting build systems and compiler tool chains. Technical details and practical tips can be found in a detailed migration guide³.

For code developers a new gitlab shared runner with an AMD GPU is available in the MPCDF GitLab-CI environment (see below).

Markus Rampp

HPC performance monitoring on *Viper-GPU*

The MPCDF is running a comprehensive performance monitoring system⁴ on the HPC systems that allows support staff as well as users to check on a plethora of performance metrics of compute jobs. Recently, the system was deployed to the *Viper-GPU* supercomputer. Unlike *Raven* which uses Nvidia GPUs, *Viper-GPU* features AMD MI300A APUs that combine a GPU, CPU cores, and high-bandwidth memory on the same socket.

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

¹<https://www.mpcdf.mpg.de/training/meetmpcdf>

²<https://docs.mpcdf.mpg.de/doc/computing/viper-gpu-user-guide.html>

³<https://docs.mpcdf.mpg.de/doc/computing/viper-gpu-user-guide.html#migration-guide-for-users-coming-from-intel-and-nvidia-based-hpc-systems>

⁴<https://docs.mpcdf.mpg.de/doc/computing/performance-monitoring.html>

These APUs and the available software tools currently do not support high-level metrics such as GFLOPs and memory bandwidths. Users can access their performance data under these limitations, with the metric “GPU uti-

lization” currently being the most important indicator if a job actually makes use of the GPUs.

Klaus Reuter, Sebastian Kehl

Software News

Activation of python-waterboa as the new Python default

As announced in Bits and Bytes No.216, Aug 2024, new versions of the Anaconda Python distribution cannot be provided anymore due to licensing restrictions. Therefore, MPCDF has come up with its own Python basis “Water Boa Python” which is not dissimilar to the original but entirely based on the “conda-forge” project from which packages are freely available. Comprehensive information on the topic was presented in MeetMPCDF of Oct 2024⁵.

The environment module “python-waterboa/2024.06” has been available since summer 2024 on the HPC systems and clusters. Please be aware that it meanwhile also serves as the basis for the central software builds provided by MPCDF, replacing Anaconda Python 2023.03. Integrated into the hierarchical environment modules this affects new versions of ‘mpi4py’, ‘h5py-mpi’, and other domain-specific software. As usual, existing branches within the modules hierarchy and hence software already installed on the clusters stay unchanged.

Klaus Reuter

New ELPA version 2025.01.001

The new 2025.01 release of the ELPA eigensolver library features significant optimizations for GPUs supporting NCCL/RCCL (Nvidia/ROCm Collective Communication Libraries), which enables efficient direct memory transfers between GPU devices. NCCL/RCCL requires the number of MPI processes to match the total number of GPUs; otherwise, ELPA automatically falls back to the previous GPU+MPI (no NCCL/RCCL) implementation.

The new ELPA is available as a module, for example, `elpa/mpi/standard/gpu/2025.01.001` after loading `gcc/13 openmpi/5.0 cuda/12.6` on *Raven* or `gcc/14 openmpi/5.0 rocm/6.3` on *Viper-GPU*. All available module combinations can be queried with `find-module elpa.*2025.01.001`.

Petr Karpov, Andreas Marek, Tobias Melson

GitLab shared CI runner with AMD GPU available

The MPCDF offers a new shared GitLab runner in its GitLab CI environment. It is based on a virtual machine with 8 CPU cores, 32 GB of RAM, 1 TB of disk space and an AMD Instinct MI210 GPU. The runner is named “MPCDF-GPU-05” and will accept only CI-jobs tagged with `amd-mi200`.

Accordingly, the MPCDF CI module images were recently extended with AMD software. Currently, the `gcc_14` flavour of these images contains the environment modules `rocm/6.3` and `amd-llvm/6.0`. As both packages are undergoing rapid development, updates will be regularly applied to the CI module images. Unlike our usual policy of retaining all versions, older versions of `rocm` and `amd-llvm` will need to be removed from the image after some time. In order to always load the latest versions, simply run `module load rocm` or `module load amd-llvm` without specifying a version number. An up-to-date list of

all available images and their contents can be inspected and searched in our documentation⁶. A snippet from a file `.gitlab-ci.yaml` testing a simple HIP code on the CI runner could look as follows:

```
hip_hello_world:
  tags:
    - amd-mi200
  image: gitlab-registry.mpcdf.mpg.de/mpcdf/ci-module-image/gcc_14:latest
  script:
    - module load gcc/14
    - module load rocm
    - rocm-smi --showhw
    - hipcc --offload-arch=native -o hello.x
    - hip_hello_world.cpp
    - ./hello.x
```

This example uses the flag `--offload-arch=native` which automatically infers the correct architecture for the locally

⁵<https://datashare.mpcdf.mpg.de/s/CZR64ip6fvi6TqA>

⁶<https://mpcdf.pages.mpcdf.de/ci-module-image/>

available GPU (which is `gfx90a` for the MI200 generation of GPUs available with the new CI runner). Please be aware that the flag `--offload-arch=gfx942` which targets the MI300A GPUs on *Viper-GPU* must not be used on

these CI runners. While code compiled using that flag might even run it will likely produce bogus results.

Francesco Turcinovich, Nicolas Fabas, Tobias Melson, Klaus Reuter, Thomas Zastrow

HowTo: External Usage of GitLab Wikis

If you need to write project documentation or any other technical document, the “GitLab Wiki” is the right place to do it. Integrated into a GitLab repository, it offers the common Wiki functionalities (please see GitLab’s documentation for a complete overview):

- Support for several common markup languages (Markdown, RDoc, AsciiDoc, Org)
- Nesting of pages
- Integration of visualization libraries (LaTeX math formulas, graph structures, ...)
- Searching for a page title

Nevertheless, working with a Wiki in GitLab’s web interface is not very comfortable: some functionality like a full-text search is still missing. Therefore, it makes sense to clone your Wiki and work on your local machine on it. Cloning a GitLab Wiki? Yes, because behind the scenes a GitLab Wiki is nothing else than another Git repository. Independent of your project’s repository, it can be cloned with the command:

```
git clone git@gitlab.mpcdf.mpg.de:NAMESPACE/PROJECT-NAME.wiki.git
```

where `NAMESPACE` is the path to your GitLab repository (normally your user name or the group(s) of the repository) and `PROJECT-NAME` is the name of your GitLab project. You can also find the link to the Wiki repository in the web interface:

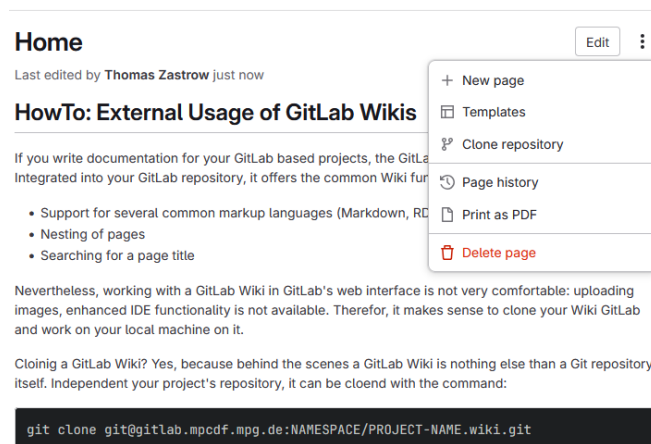


Figure 2: Cloning a GitLab Wiki as repository to your local system

Once you have cloned the Wiki repository to your local file system, you can work with the pages and folders in your preferred text editor or IDE.

Note-taking Apps

As GitLab Wikis consist of markup formatted files organized in a folder structure, it is also possible to integrate them into common note-taking apps like Obsidian or Joplin. For example in Obsidian, you can open the local folder which contains the cloned Wiki as a “Vault” (File menu / Open Vault). The full functionality of Obsidian like full-text search, linking between pages etc. is now available for working with your GitLab Wiki. Don’t forget to add, commit and push your changes once you are done. The screenshot in Fig. 3 shows the current article, being part of a GitLab Wiki and edited locally in Obsidian.

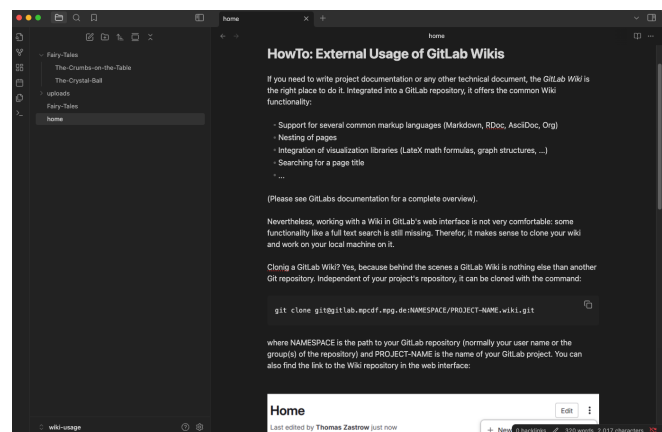


Figure 3: Working with the cloned Wiki in the Obsidian note-taking app

Thomas Zastrow

Enhanced SSH Configuration

OpenSSH is the most commonly used SSH client on the command line. It is available for Linux, MacOS and Windows (via Powershell). A useful feature is the configuration file⁷ located in `$HOME/.ssh/config`, which is an alternative to using flags with the `ssh` command. It allows to define an alias for each host, along with specific connection parameters. Such aliases are then available to all programs based on SSH, like `sftp`, `scp`, Ansible or VS Code.

Since OpenSSH 7.3p1, separate configuration files can be included in the main config file via the `include` directive.

```
include config.d/*.conf
include config.d/cloud/*.conf
```

Each remote host can be defined with the `Host` directive. It introduces a block containing several connection options, e.g.:

```
Host webserver
  Hostname webserver.mpcdf.mpg.de
  User MPCDF_USER_NAME
  IdentityFile /home/user/.ssh/my-key.pem
```

This is equivalent to:

```
ssh -i /home/user/.ssh/my-key.pem \
  MPCDF_USER_NAME@webserver.mpcdf.mpg.de
```

Now, with the configuration block defined, we can simply connect to webserver with:

```
ssh webserver
```

If the same option is reused often, it can be defined separately in a wildcard block, e.g.:

```
Host *.mpcdf.mpg.de
  User MPCDF_USER_NAME
```

This means that all connections will implicitly be as the user given in the wildcard block. Wildcards do not work in Windows.

You can refer to the MPCDF documentation⁸ about login to machines at MPCDF with SSH.

Nicolas Fabas

News

MPCDF Status Page

On a new service-status webpage⁹, the MPCDF provides real-time information about:

- the status of the most common web services of the MPCDF: the homepage, SelfService, Helpdesk, GitLab, DataShare, the MPCDF documentation and the HPC Cloud Admin Interface
- the availability of the HPC systems *Raven* and *Viper*.

At the bottom of the page, upcoming maintenances and emergency incidents are announced. The new status page is also linked at the MPCDF homepage under “operational information”

Thomas Zastrow

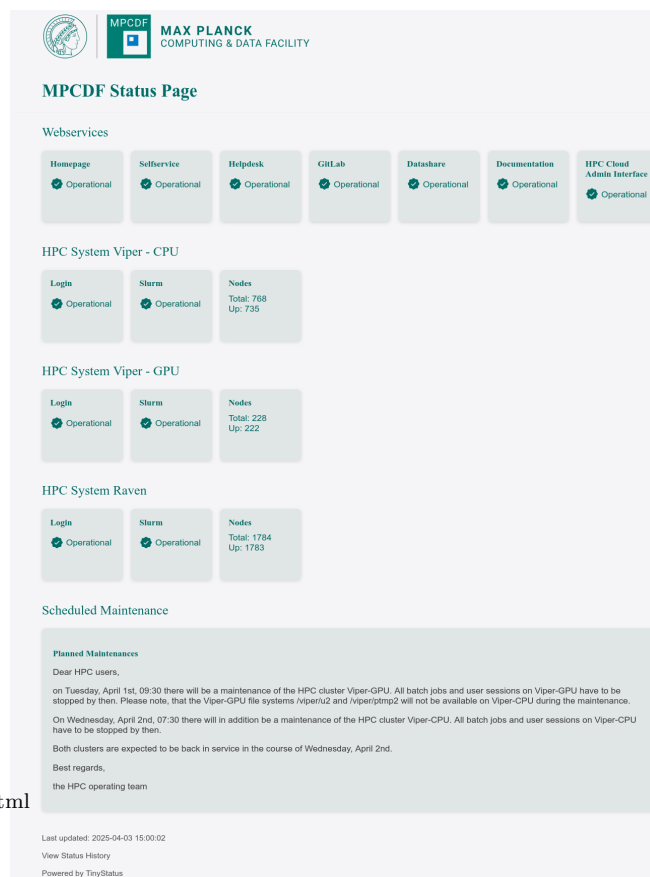


Figure 4: The new MPCDF status page

⁷https://www.man7.org/linux/man-pages/man5/ssh_config.5.html

⁸<https://docs.mpcdf.mpg.de/faq/connecting.html>

⁹<https://status.mpcdf.mpg.de>

Mandatory 2FA for GitLab

Starting June 2nd, it will no longer be possible to login to the MPCDF GitLab webinterface¹⁰ without a second authentication factor. The current login via username and password will be deactivated and only the “2FA Login” (username and password in combination with a second factor, see our FAQ¹¹ for further information) will be possible. Please note that this login functionality is handled by the MPCDF user management and is independent of GitLab’s internal 2FA functionality (which you can additionally activate in your profile). Usage of access tokens and the SSH-based Git functionality on the command line

are not affected. If you want to try out the 2FA Login, you can choose it already now on GitLab’s login page:

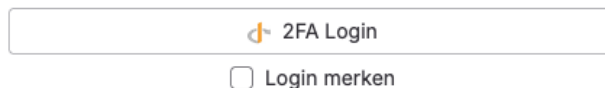


Figure 5: 2FA Login button on MPCDF GitLab

Thomas Zastrow

Events

AMD workshop

Very similar to our last AMD-GPU workshop in November 2024, we will host a second workshop and hackathon with a focus on the new *Viper-GPU* system. It will be held online on three afternoons from May 13th to May 15th, 2025. On the first day, AMD will give presentations on how to use the system with different programming approaches and introduce their tools. The second and third day are dedicated to hands-on work on the participants’ applications, using the *Viper-GPU* system. Experts from AMD and MPCDF will assist in this hackathon. More information and the registration can be found here¹².

Tilman Dannert

Introduction to MPCDF services

The next issue of our semi-annual seminar series on the introduction to MPCDF services will be given on May 8th, 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 given on our webpage.

Tilman Dannert

MeetMPCDF

Save-the-date: The next editions of our monthly online-seminar series “MeetMPCDF” are scheduled for May 8th, June 5th, July 3rd, 15:30 (CEST). Topics will be announced in due time via the all-users mailing list. All

announcements and presentations of previous seminars can be found on our training webpage¹³.

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.de¹⁴.

Tilman Dannert

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 will follow as they become available.

Raphael Ritz

Research Data Alliance - Deutschland-Tagung 2025

As in previous years, MPCDF contributed to the organization of the RDA Deutschland-Tagung 2025 which took place at the Geoforschungszentrum in Potsdam on February 18th and 19th this year. Various topics in the area of research data management were presented and discussed. Focus areas included the newly established “Datenkompetenzzentren” as well as the role of research data management in artificial intelligence. The program and presentation materials are available from the conference website¹⁵.

Raphael Ritz

¹⁰<https://gitlab.mpcdf.mpg.de>

¹¹<https://docs.mpcdf.mpg.de/faq/2fa.html>

¹²<https://plan.events.mpg.de/e/amd-mpcdf-workshop-2025>

¹³<https://www.mpcdf.mpg.de/services/training>

¹⁴<mailto:training@mpcdf.mpg.de>

¹⁵<https://indico.desy.de/event/47204/timetable/#all.detailed>