

Bits & Bytes

No.222, August 2026

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

High-performance Computing

Procurement of next-gen HPC and AI systems of the MPG

Earlier this year, MPCDF in collaboration with administrative headquarters of the Max-Planck Society (MPG) launched a Europe-wide call for tenders for procuring a successor for the HPC system *Raven* of the MPG. With additional budget allocated by the President the procurement included a second, GPU-only system dedicated to scientific AI applications. Bids from different vendors and with different technical characteristics were ranked primarily by the compute performance the offered system will deliver (based on a representative set of 10 HPC applications of the MPG for the HPC system, and the aggregated GPU-performance for the AI system, respectively). In summary, the main characteristics of the new systems are as follows:

HPC system:

- 320 CPU-only compute nodes based on AMD EPYC Turin 9655 (61440 CPU cores, 240 TiB RAM total),
- 59 GPU-accelerated compute nodes based on Nvidia GraceBlackwell GB200 NVL4 (236 GPUs, 43 TiB HBM total),
- Nvidia Infiniband NDR (nonblocking fat tree, 400 Gb/s per CPU node, 800 Gb/s per GPU node),
- IBM StorageScale with 24 PB HDD, 360 TB NVMe.

AI system:

- 106 4-way GPU-compute nodes based on Nvidia Grace-Blackwell GB200 NVL4 (424 GPUs, 77 TiB HBM total),
- Nvidia Infiniband XDR (nonblocking fat tree, 800 Gb/s per node)
- IBM StorageScale with 1 PB NVMe.

The new systems will both be delivered by the well known German company pro-com Datensysteme GmbH, in partnership with AMD, Nvidia, IBM and Lenovo.

Despite the currently very difficult global market situation, the new systems will provide MPG researchers with increased computational power compared to *Raven*, and in addition a new line of MPG resources for AI research. Installation of the systems will commence in autumn this year with the aim to have the systems in production in the course of 2027.

Erwin Laure, Markus Rampp

Viper-GPU: extension to final configuration

A number of 27 additional compute nodes (with 54 AMD MI300A APU's) are currently being added to *Viper-GPU* as a compensation for the delayed deployment of the entire *Viper* machine. With this extension *Viper-GPU* will reach its final configuration comprising 327 compute nodes (654 MI300A APU's) within the next few weeks.

Markus Rampp

Software News

Compilers

The latest GNU compiler 16.1.0 (module `gcc/16`) and the Intel oneAPI version 2026.0 (modules `intel/2026.0`, `impi/2021.18`, and `mk1/2026.0`) have been installed together with the corresponding software stacks on the HPC ma-

chines and on many institute clusters. The Intel compiler provides new features, including more fine-grained loop optimization control and the ability to filter optimization reports.

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

The new version 23.2.0 of the LLVM-based AMD compiler (module `amd-llvm/23.2`) has been made available on *Viper-GPU*.

Tobias Melson

CUDA and Nvidia HPC SDK

CUDA 13.2 (module `cuda/13.2`) has been installed on *Raven*, together with the matching NCCL module `ncccl/2.30.4`.

The Nvidia HPC SDK version provided by the module `nvhpcsdk/26` has been upgraded to version 26.5 and will further be updated to the most recent version once available. The current version ships with CUDA 13.2 supplied by the module `cuda/13.2-nvhpcsdk_26`. See our previous announcement¹ for an explanation about CUDA modules. The update has also been applied to the CI module images in our GitLab instance.

Tobias Melson, Tilman Dannert

Amber molecular dynamics package

The new version 26 of the popular classical molecular dynamics (MD) package Amber² is available on the MPCDF HPC systems. Notably, with HIP support added, the new version is provided on *Viper-GPU* for the first time. With the new release we also update the naming of the corresponding environment modules, as the licensing distinction that motivated the earlier split no longer applies.

Starting with Amber26, environment modules previously

named `amberMD` (providing the license-restricted `pmemd`) and `amber` (providing the free AmberTools) are now provided on our HPC systems as a single unified module named `amber` containing both `pmemd` and tools. Earlier module versions stay untouched, but note that there is no `amberMD` module for Amber26 and job scripts loading `amberMD` need to be adapted accordingly. Since `pmemd` releases happen only every other year, an AmberTools-only package (i.e. without `pmemd`) will continue to be provided as module `ambertools`.

The new version can be loaded using `module load amber/26`. Using tools that depend on Python requires additionally loading *Water Boa Python 2024.06* (environment module `python-waterboa/2024.06`). Note that `pmemd` is free of charge for non-commercial use; users affiliated with for-profit organizations require a separate license agreement with UCSF (see the Amber26 license terms³).

Amber's protein force fields are widely used beyond Amber itself: MPCDF, in close collaboration with the Max Planck Institute for Multidisciplinary Sciences and Johnson & Johnson and together with co-authors representing Amber, has ported the protein force fields `ff14SB` and `ff19SB` for use with the GROMACS⁴ MD package, and cross-validated their consistency in both packages. The results are detailed in the recent preprint *Force Fields and Simulation Engines: Lessons Learned from Porting Amber ff14SB and ff19SB to GROMACS*⁵.

Vedran Miletic, Markus Rampp

Using AI Coding Agents on MPCDF Systems

Command-line AI coding agents such as Claude Code, OpenAI Codex, Google Antigravity, OpenCode, or Cline have quickly become popular tools for software development. They read and modify source files, run commands, execute tests, and iterate largely on their own. Increasingly, MPCDF users are using them to develop their scientific software, refactor simulation codes, write analysis scripts, or explore an unfamiliar codebase. To help users do this more safely, MPCDF now provides a containerized environment that runs these agents with reduced access to the surrounding file system — a meaningful improvement over running them directly on the machine, though it offers no guarantee of security.

Why a sandbox is needed

An AI coding agent does more than suggest text: it runs with your user account and executes real actions. It opens

files, runs shell commands, and edits code, often with only minimal confirmation. By default it inherits all your permissions, so it can read any file you can read and write any file you can write. Started in your home directory, it could therefore access SSH keys, API tokens, and unrelated project data and send them to an external service. A single misunderstood instruction can also overwrite or delete files. Running an agent with unrestricted access to your account is therefore strongly discouraged.

A practical mitigation is to confine the agent as far as possible to just the project it is supposed to work on. This is what the MPCDF AI agents container is designed to do: it reduces what an agent can reach, without claiming to make its use completely safe.

¹<https://docs.mpcdf.mpg.de/bnb/215.html#cuda-modules-on-raven>

²<https://ambermd.org/>

³<https://ambermd.org/GetAmber.php>

⁴<https://www.gromacs.org/>

⁵<https://doi.org/10.26434/chemrxiv.15006112/v1>

The MPCDF AI agents container

The environment is built on Apptainer⁶, the user-space container runtime already used across MPCDF HPC systems. When you launch it, only two things are visible to the agent inside:

- the **current working directory** (your project), and
- a dedicated **fake home directory**, which holds the agents' own configuration, credentials, and caches, kept separate from your real \$HOME.

Everything else — the rest of your home directory, other users' data, and other file systems — remains invisible. The container image itself is read-only, so the agents cannot modify their own installation. Launching directly from \$HOME is also blocked, to avoid accidentally exposing all of your files.

It is worth being clear about what this does and does not achieve. A container adds a layer of isolation on top of the ordinary Linux user account, but it is not an impenetrable security boundary. Container runtimes are themselves software, and known or as-yet-undiscovered vulnerabilities may in principle allow an agent to break out of the sandbox. Running an agent inside the container is therefore considerably safer than running it directly on the machine, but it does not make it entirely safe, and it does not remove your responsibility for what the agent does.

The container ships with a set of pre-installed command-line agents, so you can pick the one you prefer without installing anything yourself:

Command	Provider
<code>agy</code>	Google Antigravity
<code>claude</code>	Anthropic Claude Code
<code>cline</code>	Cline
<code>codex</code>	OpenAI Codex
<code>copilot</code>	GitHub Copilot CLI
<code>gemini</code>	Google Gemini CLI
<code>opencode</code>	OpenCode
<code>vibe</code>	Mistral Vibe

The container comes in three flavours — based on the Linux distributions SLES, RHEL, and Ubuntu — and the matching one is detected automatically from the host system. On the SLES- and RHEL-based HPC systems (e.g. *Raven* and *Viper*) the MPCDF software tree is bind-mounted read-only into the container, so the familiar `module load` command works exactly as on the login node, and agents can build and test software against the usual HPC software stack.

Getting started

Clone the project, build the container image, and install the `agents` launcher once (make sure `~/bin` is in your `PATH`):

```
git clone https://gitlab.mpcdf.mpg.de/mpcdf/ai-cli-agents-container.git
```

```
cd ai-cli-agents-container
# builds the container image (auto-detects
# the OS flavour):
./build-containers.sh
# creates the ~/bin/agents launcher:
./install-agents-launcher.sh
```

After that, the typical workflow is simply:

```
cd /path/to/your/project
agents
```

This drops you into a shell inside the container, indicated by an `[Agents]` prompt. From there you start the agent of your choice, for example `claude` or `codex`, and work as usual. When you are done, exit the shell to leave the sandbox.

For all details (prerequisites, configuration options, and the two-stage build) please refer to the `README.md` in the AI agents container project repository.

Best practices

Because the sandbox reduces but does not eliminate the risks, a few habits are essential to using AI coding agents on MPCDF systems safely and effectively:

- **Start in the project directory, never in \$HOME.** The isolation is only as good as the directory you open. Change into the specific project you want to work on before launching the container.
- **Mind confidentiality and data protection.** Prompts and file contents are sent to external cloud services operated by the respective providers. Do not expose personal data, sensitive research data, credentials, or otherwise confidential material to an agent. Check that using a given service is compatible with the terms under which your data was obtained.
- **Keep credentials in the fake home.** API keys and login tokens accumulate in the fake home directory. Treat it like any other secret store, and wipe it if you want to remove all agent state.
- **Review everything the agent produces.** AI agents make mistakes, introduce subtle bugs, and can “hallucinate” plausible-looking but incorrect code or results. Read the diffs, run your tests, and keep your work under version control so that unwanted changes can be reverted.
- **Use the batch system from outside the container.** For safety, Slurm client tools are not available inside the container. Submit and manage jobs (`sbatch`, `srn`, `squeue`) from a regular login shell; the agent can then wait for the results and analyse the output files.
- **Keep an eye on cost and quotas.** Agents can issue many API calls in a short time. Be aware of the usage limits and billing associated with your chosen provider and account.

⁶<https://apptainer.org/>

Availability and support

The AI agents container is available now on MPCDF systems. It adds a valuable layer of protection, but — as noted above — it is not a guarantee of security: you remain fully responsible for the agents and workloads you

run, and for assessing the associated security and compliance risks. Questions, feedback, and problem reports are welcome via the MPCDF helpdesk.

Klaus Reuter, Andreas Marek

HPC-Cloud

Standard cloud images

The MPCDF HPC-Cloud provides flexible compute and storage resources to Max Planck Institutes. Compute typically takes the form of a virtual server running the Linux operating system (OS). Rather than installing the OS onto a blank disk as one might with a personal computer, the lifecycle of a virtual server begins with a disk *image* containing a pre-installed Linux distribution. Data written to the server's disk is stored separately from the image, which is itself immutable. This technical approach allows new servers to be launched quickly, and also conserves space in the backend storage system.

The cloud team provides a set of standard images based on feedback from project admins, while taking into account the support timelines of the respective distributions. Each one is based on the upstream “cloud-ready” image, to which a limited set of site-specific customizations is applied. The full details can be found in the image-builder project⁷.

Ubuntu 26.04 is the latest addition, joining supported releases of **Debian**, **AlmaLinux**, and **openSUSE Leap**. A complete list can be found in the documentation⁸.

Images contain not only primary data, but also meta-data which determines certain aspects of the servers's virtual hardware. On this subject we would like to highlight an important recent change: The virtual storage controller is now set to type *virtio-scsi*, which supports up to 255 block volumes. A side effect of this change is that volumes appear as */dev/sdX*, with persistent path */dev/disk/by-id/scsi-OQEMU_QEMU_HARDDISK_UUID* where *UUID* is the OpenStack ID of the volume. Thus, it may be necessary to adjust automation scripts to use the correct paths on newly-launched servers. No changes are required on existing servers.

Brian Standley

S3-IAM - Self Service user management for object storage

The CEPH Squid release introduces Self Service, multi-tenant, functionality via Identity and Access Management (IAM) accounts. IAM accounts allow projects to independently manage their resources including users, groups, roles, policies, and buckets; using an API interface mod-

eled after AWS IAM. IAM provides a hierarchy of accounts within a project; a root account for the project admins and end user accounts for individual users.

- Root account: The admin responsible for the Account. Manages resources within a specific project/IAM account, using clients such as the AWS CLI and the IAM API endpoint.
- S3 end users: Operate within the confines of permissions granted by the root account.

The root account can create and manage users, groups, roles, and permissions within the IAM account. This allows multiple users to gain access to a project or specific buckets within a project. However, all buckets and data are owned by the root account.

Operations that the IAM root account can perform without the intervention of a CEPH admin include:

- Create, modify, and delete users
- Manage account users' access and secret keys
- Manage IAM policies
- Manage IAM user policies
- Manage IAM groups
- Create, modify, and delete OIDC providers
- Create, modify, and delete notification topics
- Create, modify and delete quotas on buckets

This allows the root account to fully manage the resources and easily provide solutions for numerous use-cases.

Common use-cases:

1. Providing a bucket per user: Multiple users can be created with policies set to allow dedicated read/write access to a single or multiple buckets. Additionally quotas can be set on each bucket for fine grained resource usage.
2. Read/write and read-only users: Within a project, or even for specific buckets, read/write and read-only users can be created allowing control over who generates and who consumes the data.

Projects supporting IAM can now be created upon request. Existing projects may be migrated to IAM. However, this process is irreversible and requires additional planning. Ideally we suggest starting with a new project.

Robert Hish, Florian Kaiser, John Alan Kennedy

⁷<https://gitlab.mpcdf.mpg.de/mpcdf/cloud/image-builder>

⁸<https://docs.mpcdf.mpg.de/doc/cloud/technical/compute.html#images>

News & Events

Securing MPCDF account application form against bots

Lately, it turned out that the form to apply for an account at MPCDF⁹ was not sufficiently protected against being filled by bots. In order to prevent such fake applications triggering a huge amount of E-mails to the account approvers we implemented several means to protect the form against misuse by bots. In addition to some purely technical hurdles for bots, the E-mail address of the applicant now requires a confirmation, before the application is finally sent to the approver, which establishes a two-step mechanism already in the application process.

Andreas Schott

New CoE projects

EuroHPC has just finished the evaluation for a new round of HPC Centres of Excellence (CoEs). This time, two flavours of CoEs are envisaged, so-called “community CoEs” focussing on fostering the uptake and efficient usage of European HPC applications, and “lighthouse” CoEs focussing on improving and further developing important HPC applications. MPCDF has been part of four proposals (two community and two lighthouse ones) in the areas of biomolecular modeling and fusion plasma simulations, and all four of them have been proposed for funding and are now in contract preparations with a planned start early 2027. Specifically, the long-running CoE “BioExcel”¹⁰ will enter a new phase as “community” CoE and MPCDF will work alongside the MPI for Multidisciplinary Sciences on supporting the GROMACS community. The GROMACS code¹¹ itself will be the focus of a new “lighthouse” CoE, where MPCDF will work together with KTH, BSC and FZJ on improving and extending GROMACS. In the area of fusion plasma simulations, the existing Plasma-PEPSC CoE¹² will continue as a “community” CoE and MPCDF will work together with the Max Planck Institute for Plasma Physics (IPP) on their GENE code¹³ family. Another IPP code, JOEREK¹⁴, will be the focus of a new lighthouse CoE.

Erwin Laure

MPCDF at Garching Campus Open Doors (October 3)

MPCDF will open its doors for the general public at the *Open Day Campus Garching*¹⁵ at October 3, 10:00-

17:00. We will provide short talks, posters about scientific high-performance computing, data science, and artificial intelligence and offer the opportunity to have a peek into the machine hall. The program¹⁶ is targeted at the general public, but we, at MPCDF, always appreciate exchange with our friends and expert users who might take the opportunity of the campus event to meet in person with MPCDF staff.

Friederike Neu

MPCDF Hands-On Cloud Computing Workshop

On 13-14 October 2026, MPCDF organizes a Cloud Hands-on Workshop. This workshop is designed for system administrators, cloud practitioners, and DevOps engineers and there is no prior experience required. We will guide you from the basics to a solid, working understanding of cloud infrastructure. If you already use an HPC-Cloud project from the MPCDF, you'll also come along with new insights and ideas for your service planning. The workshop will be conducted online and we are going to provide a limited number of cloud projects for the time of the workshop. For more details and registration, please visit the event homepage¹⁷

Fabio Baruffa

Agentic AI for Science: An on-site workshop at MPCDF, 7-8 October 2026

AI agents are reshaping the way scientific research is conducted. Unlike traditional AI tools, agents can plan, reason across multiple sources, and carry complex tasks through many steps autonomously, from literature review and data analysis to code development and simulation workflows. To explore the current state and future potential of this technology, MPCDF is organising a 2-day in-person workshop bringing together researchers, research software engineers, and industry partners from across the Max Planck Society.

The first day will combine hands-on tutorials with real-world experiences from MPG researchers, offering a practical introduction to AI agents alongside talks from groups already using them in their workflows. The second day will feature contributions from leading industry partners, showcasing their latest solutions and discussing how agentic AI is being adopted across the research landscape.

⁹<https://selfservice.mpcdf.mpg.de/index.php?r=registration>

¹⁰<https://bioexcel.eu/>

¹¹<https://www.gromacs.org/>

¹²<https://plasma-pepsc.eu/>

¹³<https://www.genecode.org/>

¹⁴<https://www.jorek.eu/>

¹⁵<https://forschungscampus-garching.de/>

¹⁶<https://www.mpcdf.mpg.de/opendoors2026>

¹⁷<https://plan.events.mpg.de/event/806/>

The workshop is open to all researchers across the Max Planck Society and aims to build a shared understanding of where agentic AI can genuinely add value, what the current limitations are, and how the community can approach adoption responsibly. Further details and registration are available on the event homepage¹⁸.

Piero Coronica

Workshop on OpenMP Offloading with AMD GPUs, 27-28 October 2026

This online workshop, lead-organized by HLRS¹⁹ and AMD in collaboration with MPCDF, teaches the usage of OpenMP for GPU programming with a focus on the AMD MI300A APU which, for example, is employed in the HPC system *Viper-GPU* at MPCDF. Target audience are beginners in GPU programming having already some basic knowledge of parallelization with OpenMP on CPUs. After this course participants will have learned the basics to confidently start porting applications from a CPU-only system to systems with discrete GPU accelerators or APUs like the MI300A. Further details and registration can be found on the corresponding MPCDF event webpage²⁰.

Tilman Dannert

Introduction to MPCDF Services

The next introductory course will take place online on October 15, 2026, from 14:00 to 16:30. It is designed to familiarize new users with the MPCDF compute and data services. No registration is required; interested users can simply join via the Zoom link²¹, which is also published on our website. Please note that the link only becomes active at the start of the course.

Klaus Reuter

Seminar on Kokkos by Christian Trott

On September 9th, MPCDF will host a seminar talk on *Kokkos: The Why, the Who and the How* by Christian Trott, who is a co-leader of the Kokkos core team at Sandia National Laboratories and a co-Chair of the ISO C++ committees library working group (LWG). The Kokkos C++ Performance Portability Ecosystem²² is a popular, production level solution for writing modern C++ applications in a hardware agnostic way, and has been

widely adopted for writing performance-portable high-performance-computing applications, especially across GPU platforms.

The seminar will be given on September 9th, 10:00 in the main lecture hall of building D2, with the option of online participation via zoom (see the MPCDF event webpage²³). The speaker will be available for extended discussions after the talk.

Erwin Laure, Markus Rampp

IT4Science Days

The next *MPG DV-Treffen* will take place during the *IT4Science Days 2026* together with colleagues from other German research organizations performing organizations, specifically from the Helmholtz Association and the Fraunhofer Society. This year, we will meet in Göttingen from Monday, September 28th, through Thursday, October 1st. The main meeting starts on Tuesday around noon. The focus of this year's meeting will be on *Artificial Intelligence* in all its facets, but expect a wide spectrum of further contributions.

More information, including the program as well as the registration form is available from the website of the meeting²⁴.

Raphael Ritz

MPG Research Data Management (RDM) Conference 2026

The next *MPG Research Data Management (RDM) conference* will be held November 3-5, 2026, in Potsdam Golm preceded by an RDM introduction session on Monday, November 2, 2026. As with the RDM workshops of recent years, a special focus will be on the research support services of the hosting institutes (MPI of Colloids and Interfaces supported by the MPI of Molecular Plant Physiology and the Albert Einstein Institute). For the first time, the language of this event will be English. Participation in the conference is free of charge but limited to MPG members.

More information, including the program as well as the registration form is available from the website of the meeting²⁵.

Raphael Ritz

¹⁸<https://plan.events.mpg.de/e/ai-agents-for-science>

¹⁹<https://www.hlrs.de/>

²⁰<https://www.mpcdf.mpg.de/events/46269/2825>

²¹<https://mpcdf-mpg-de.zoom-x.de/j/69500447868?pwd=5fQ3xerZMaMDh77215EaPyPVm4MGnM.1>

²²<https://kokkos.org/about/overview/>

²³<https://www.mpcdf.mpg.de/events/46503/2825>

²⁴<https://plan.events.mpg.de/event/670/>

²⁵<https://rdm.mpl.mpg.de/mpdl-services/events/rdm-conference-2026/>