

Bits & Bytes

No.217, December 2024

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

High-performance Computing

AlphaFold3 available on *Raven*

Recently, Google DeepMind released the source code of AlphaFold3¹, shortly after Demis Hassabis and John Jumper from the same company had received half of the Nobel Prize in Chemistry 2024² for the development of AlphaFold2. AlphaFold3 extends the capabilities of AlphaFold2 by enabling predictions of interactions of biomolecules in addition to inferring their structure.

AlphaFold3 is now available on *Raven*, complementing installations of AlphaFold2 that have been provided and regularly updated since summer 2021. To get started, execute the command `module help alphafold/3.0.0` on *Raven* and follow the instructions.

An important difference to AlphaFold2 is that the AI model behind AlphaFold3 is not publicly available. Users must register with Google DeepMind and download their personal copy after approval. By default, for the software installation provided by MPCDF, the weights file ‘af3.bin’ has to be placed into the home directory at ‘~/alphafold_3_0_0/model/'. It is the responsibility of the user to comply with the terms of use of the AI model.

The MPCDF is interested in getting feedback from users on the usability and performance of AlphaFold3 on the A100 GPUs of *Raven*. The memory requirements of AlphaFold3 are higher than those of AlphaFold2, and therefore out-of-memory conditions are more likely to happen. The scripts provided by our installation try to mitigate this situation by enabling CUDA unified memory for the inference step, logically extending the GPU memory with host memory.

Klaus Reuter

Resource limits on the HPC machines

In order to maintain the responsiveness of the login nodes on the HPC machines, per-user resource limits were introduced on *Raven*³ and also on *Viper* earlier this year. The per-user limit is currently two cores on *raven01/02* and

viper01/02 and 6 cores on *raven03/04* and *viper03/04*, respectively. A hard memory limit is also enforced, which is 10% of the available memory on the first two login nodes of *Raven* and 50% of the available memory for the login nodes 3 and 4 of *Raven* and all login nodes of *Viper*. The following table summarizes these limits.

	raven01/02	raven03/04	viper01/02	viper03/04
cores	2	6	2	6
memory	50 GB	256 GB	256 GB	256 GB

As a consequence, this limits running multi-threaded or distributed jobs (which is the intention), but it may also affect the (parallel) performance of the build procedures of large HPC codes. Usually the builds are done in parallel with the build system spawning threads (`make -j` or `cmake --build . --parallel` are typical examples). The number of threads spawned should be limited in the build procedure to the number of cores available (2 or 6) by passing this number to the build command (`make -j 6` or `cmake --parallel 6`).

It is important to note that these resource limits also apply to CI jobs executed by a GitLab runner which has been launched by the user on the login nodes. Hence, also such runners should take the above mentioned resource limits into account when launching parallel builds, otherwise the build jobs may slow down significantly.

There are the following options to increase the build performance for such CI jobs:

- Move your GitLab runners which use local builds from *raven01/02* and *viper01/02* to *raven03/04* and *viper03/04* and restrict the build procedure’s parallelism to 6.
- You can keep your runners on the first two *Raven* nodes, but submit the build job into the “interactive” queue via the Slurm system. There you can use up to 8 cores for your job and hence use 8 threads for the build procedure.

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

¹<https://github.com/google-deepmind/alphafold3>

²<https://www.nobelprize.org/prizes/chemistry/2024/press-release/>

³<https://docs.mpcdf.mpg.de/doc/computing/raven-user-guide.html#resource-limits>

```
salloc --partition=interactive -n 1 --cpus-per-task=8 \
--time=00:20:00 --mem=32G srun <build command>
```

- Change your build tests to use the shared runners of our GitLab instance.

Tilman Dannert

HPC monitoring on *Viper*

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* supercomputer. Unlike *Raven* and previous HPC systems, *Viper* features AMD EPYC Genoa CPUs with Zen4 cores that have somewhat less-capable Performance Monitoring Units (PMUs). For instance, while on Intel-based CPUs, GFLOP rates can be obtained individually for each precision and vector width, only total GFLOP rates independent of the precision can be measured on the AMD Zen4 processors. Similarly, the support for uncore events such as the memory bandwidth is still limited, but expected to continuously improve with more recent kernel versions. Users can access their performance data under these limitations. We're working on improving the support for the *Viper* system over time.

Klaus Reuter

Routine transition to a new set of CI module images in 2025

Since late 2023 the MPCDF has been providing Docker images with software stacks that are installed in an essentially identical fashion on the HPC systems, enabling users to test their software consistently with various compiler and library toolchains on the GitLab shared CI runners. Interested readers can find the full announcement in Bits & Bytes No. 214, December 2023⁵.

We would like to remind the CI users of the strategy we are employing to tag and manage these CI images. Starting with the year 2025, the images tagged with '2024' will not receive any updates and will hence stay unchanged. At the same time, we will start with a new set of images

tagged '2025' (then identical to 'latest') that contain more recent software. Users can find up-to-date lists of the available images and the software therein here⁶. Please note that users do not need to take action unless they want to access more recent software stacks for their CI tests.

Klaus Reuter, Tobias Melson

Checks for uninitialized variables disabled in latest Intel Fortran compiler

Many Fortran developers rely on correctness-checking capabilities of the compiler⁷, for example in their CI pipelines or other non-regression-checking and debugging strategies. The Intel Fortran compilers, for example, by using the option `-check`, can instrument an executable with various runtime checks⁸ including the commonly used array-bounds check (`-check bounds`), or uninitialized-variables check (`-check uninit`). With the latter option, however, the new Intel compiler, `ifx`, is producing false positives, in particular in combination with MPI, which is why Intel decided to disable the `-check uninit` option. Note also, that `-check all` effectively translates to `-check all,nouninit` in the latest `ifx` (2025) version⁹ which may be perceived as a silent relaxation of the overall strictness.

In general, we recommend to use the following set of options for enabling a restrictive set of runtime checks for Intel and GNU Fortran compilers, respectively, which includes an effective check for uninitialized variables by pre-setting variables with "nan" and catching resulting floating-point exceptions.

- `ifx -g -traceback -check all -fpe0 -init=arrays \ -init=snan`
- `gfortran -g -Wall -fcheck=all -finit-real=snan \ -ffpe-trap=invalid,zero,overflow`

Note, however, that some of these options can significantly increase the execution time of the generated executable and hence these should only be used for debugging and correctness checking.

Markus Rampp

A Shared AI System for Eleven MPIs

The demand for using and developing AI methods is significantly growing in many Max-Planck Institutes and the recent adoption of generative models has given another major push. MPCDF already supports many institutes

in these efforts through its AI support team, which also ensures the efficient availability of major AI frameworks on the MPG supercomputers.

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

⁵<https://docs.mpcdf.mpg.de/bnb/214.html#no-214-december-2023>

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

⁷<https://docs.mpcdf.mpg.de/doc/computing/software/debugging-tools.html#run-time-checks>

⁸<https://www.intel.com/content/www/us/en/docs/fortran-compiler/developer-guide-reference/2025-0/check.html>

⁹<https://www.intel.com/content/www/us/en/developer/articles/release-notes/oneapi-fortran-compiler-release-notes.html>

A key bottleneck is, however, the availability of high-end GPUs, which are needed for training and increasingly also for inference, particularly in large language models. Hence, eleven Max-Planck Institutes, coordinated by MPCDF, have joined forces to procure a GPU system. A total budget of over 6.5 million Euro has been raised, more than 50% of which contributed by the BAR. The system is being procured in two rounds – the first one has just finished and a system comprising 136 Nvidia H200 GPUs (configured in 17 nodes with 8 GPUs), 1 PB of fast NVMe storage and Infiniband interconnect will be installed in the second quarter of 2025. The second procurement round, potentially targeting even newer GPU generations, will take place mid 2025 and the final system will then feature well over 200 GPUs.

The system, available to the participating institutes (Fritz-Haber Institute of the MPG, MPI for Human Development, MPI for Human Cognitive and Brain Sciences, MPI for Informatics, MPI for Software Systems, MPI

for Sustainable Materials, MPI for Biochemistry, MPI for Polymer Research, MPI for Biogeochemistry, MPI for Geoanthropology, and MPI for Multidisciplinary Sciences) will not only provide an economical, shared resource, but also facilitate the exchange of experiences between the participating MPIs and beyond. It is anticipated to become a nucleus and hub for further collaborations among MPIs in the extremely rapidly growing field of AI.

The technical design of the machine allows for future expansion with additional funding. Interested groups can already now contribute to the procurement in 2025. In addition, we plan to organize regular AI roundtables, workshops and trainings, open also to groups not yet participating in the system.

If you are interested joining the discussions or even the investment, please contact support@mpcdf.mpg.de

Erwin Laure

Events

International HPC Summer School 2025

The International HPC Summer School (IHPCSS) 2025 will take place from July 6th to July 11th in Lisbon, Portugal. The series of these annual events started 2010 in Sicily, Italy and provides advanced HPC knowledge to computational scientists, focusing on postdocs and PhD students. Through the participation of Canada, the USA, South Africa, Japan, Australia and Europe a truly international group of highly motivated students is meeting each year.

Interested students and postdoctoral fellows should monitor the school's website (<https://ss25.ihpcss.org>) where registration opens on December 16th. School fees, travel, meals and housing will be covered for all accepted applicants through funds from the European Union and EuroHPC. For further information and application, please visit the website of the summer school.

Erwin Laure

Introduction to MPCDF services

The next issue of our semi-annual seminar series on the introduction to our services will be given on May 15th, 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.

Meet MPCDF

The next editions of our monthly online-seminar series “Meet MPCDF” are scheduled for

- February 6th, 15:30 “Introducing the Viper GPU system with AMD MI300A APUs”
- March 6th, 15:30 Topic to be announced
- April 3rd, 15:30 Topic to be announced

All announcements and material can be found on our training webpage¹⁰ and the “Meet MPCDF” invitations will be sent to the all-users mailing list.

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.de¹¹.

Tilman Dannert

¹⁰<https://www.mpcdf.mpg.de/services/training>

¹¹<mailto:training@mpcdf.mpg.de>