MAX PLANCK
COMPUTING & DATA FACILITY
MPCDF

Bits & Bytes

No.214, December 2023

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

HPC Software News

CUDA-aware OpenMPI on *Raven*

MPCDF provides CUDA-aware OpenMPI on *Raven* based on different compilers and CUDA versions. The complete list can be inspected by running `find-module openmpi_gpu`. Below, we highlight some relevant combinations of compiler and CUDA modules that can be used with the `openmpi_gpu/4.1` module.

GCC-based CUDA-aware OpenMPI builds are available after loading `gcc/11 cuda/11.6` or `gcc/12 cuda/12.1`. Recently, a CUDA-aware OpenMPI module has been added which works with the CUDA version and the compilers provided by the Nvidia SDK. To access it, the modules `nvhpcsdk/23 cuda/11.8-nvhpcsdk` must be loaded.

Tobias Melson, Tilman Dannert

GPU-accelerated VASP

With the deployment of CUDA-aware OpenMPI for Nvidia compilers (nvhpcsdk, see above) MPCDF provides GPU-accelerated builds of the VASP[1] software package for atomic-scale materials modelling from first principles. Currently, a `vasp-gpu/6.4.2` module is available on *Raven* and selected institute clusters. Note that MPCDF does not hold a license for VASP. Individual users have to bring in their own license (via MPCDF helpdesk) in order to be enabled for using VASP at MPCDF.

Markus Rampp

Intel oneAPI: transition from ifort to ifx

The transition to the new LLVM-based compilers in the Intel oneAPI package is progressing. Already in the currently installed module `intel/2023.1.0.x`, `icx` and `icpx` are the default compilers for C and C++, respectively, replacing the "classic" compilers `icc` and `icpc`. As the next step, MPCDF will follow Intel's recommendation to set `ifx` together with its MPI wrapper `mpiifx` as the default Fortran compiler in the upcoming intel module corresponding to the oneAPI release 2024.0. The "classic" Fortran compiler `ifort` will still be present for some time, but should be considered deprecated, because its development had effectively been frozen some time ago.

Users are advised to adjust all Fortran builds to use the new `ifx` compiler. A porting guide[2] exists with detailed information on this transition. Further support is provided at the MPCDF Helpdesk[3].

Tobias Melson, Markus Rampp

Module Software Stacks for Continuous Integration Pipelines on MPCDF GitLab Shared Cloud Runners

In order to provide the developers of HPC applications with a familiar and comprehensive software environment also within GitLab-based continuous integration (CI) pipelines, the MPCDF is maintaining special Docker images. These images use environment modules to make software accessible, very similar to how software is handled on the HPC systems[4], and they can be used on the Shared Cloud Runners[5] offered by the MPCDF. Hence, e.g. build scripts would work on both the HPC systems and the CI cloud environment in a consistent way.

This article introduces a redesign of the module-enabled Docker image infrastructure which is eventually going

*Editors: Dr. Renate Dohmen & Dr. Markus Rampp, MPCDF
[1] https://www.vasp.at/
[2] https://www.intel.com/content/www/us/en/developer/articles/guide/porting-guide-for-ifort-to-ifx.html
[3] https://helpdesk.mpcdf.mpg.de/
[4] https://docs.mpcdf.mpg.de/doc/computing/software/environment-modules.html
[5] https://docs.mpcdf.mpg.de/doc/data/gitlab/gitlabrunners.html

to replace the currently used `module-image`. Documentation[6] and a software list[7] for the current `module-image` are available online.

Introducing a novel module-enabled Docker image infrastructure

To address some limitations of the `module-image` we have developed a more flexible infrastructure composed of various Docker images, each of which provides a toolchain based on a *single* combination of compiler and MPI variant. The access to the software is implemented via environment modules. Up-to-date lists of the images together with lists of the software contained are documented in GitLab[8]. Currently, the images are based on *openSUSE Leap 15.5* which is largely compatible with the SLES 15 operating system used on many HPC clusters at MPCDF.

As indicated by its tag, each image only contains a single toolchain, namely one compiler with optionally one MPI library plus a selection of widely used additional libraries. The initial list of software can be extended upon request. Arbitrary further software from the official OpenSUSE repos may be installed by the users individually, if necessary.

Users are encouraged to migrate to the new CI images soon, report potential issues and request additional software modules via the helpdesk, if necessary. Essentially, the `module-image` can simply be replaced in the user's `gitlab-ci.yml` file with one of the new images that provides the desired software stack for the respective CI job. Please note that the modules inside the images will be updated and extended regularly, similarly to the software modules updated on the HPC systems.

To limit the individual growth of these Docker images over time, we will put the following tagging-and-purging strategy in place: Essentially, all images are tagged using 'latest' and/or the calendar year. In the course of a year, say 2023, the images tagged with 'latest' and the year ('2023') are identical and receive regular updates and additions of software. With the beginning of the new year, all images tagged with the previous year stay unchanged (frozen). The newly created images for 2024, say, will start out in early January again in a slim state and will be tagged 'latest'. Users can then choose to migrate to the more recent images (tagged '2024' and 'latest' in our example) or stick with the older (but static!) images (tagged '2023') for a while. In case a user opts for using the tag 'latest', please be warned that the software environment will change at the beginning of each year.

Announcing legacy status and later discontinuation of the `module-image`

The current `module-image` has been provided for several years now to provide easy access to the familiar modules environment also from within GitLab CI jobs. As the image and infrastructure are based on the outdated CentOS 7, the `module-image` will be considered legacy after the test phase of the new Docker image infrastructure and will then not be updated any more. Ultimately, the image will have to be removed (to be announced in due time).

Tobias Melson, Klaus Reuter

Compressed Portable Conda Environments for HPC Systems

Introduction and Motivation

The Conda package manager and the related workflows have become an accepted standard when it comes to distributing scientific software for easy installation by end users. Using `conda`, complex software environments can be defined by means of simple descriptive `environment.yml` files. On MPCDF systems, users may use Conda environments, but without support from MPCDF for the software therein.

Once installed, large Conda environments can easily amount to several 100k individual (small) files. On the local file systems of a laptop or PC this is typically not an issue. However, in particular on the large shared parallel file systems of HPC systems the vast amount of small files may cause issues, as these file systems are optimized for other scenarios. Inode exhaustion and heavy load due to (millions of) file opens, short reads, and closes happening during the startup phase of Python jobs from the different users on the system are only two examples.

Move Conda environments into compressed image files

MPCDF developed the new open-source tool *Condainer*, which adresses these issues by moving Conda environments into compressed squashfs images, reducing the number of files stored directly on the host file system by orders of magnitude. Condainer images are standalone and portable: They can be copied between different systems, improving reproducibility and reusability of proven-to-work software environments. In particular, they sidestep the integration of a specific `conda` executable into the user's `.bashrc` file, which often causes issues and is orthogonal to the module-based software environments provided on HPC systems.

[6] https://docs.mpcdf.mpg.de/doc/data/gitlab/gitlabrunners.html
[7] https://mpcdf.pages.mpcdf.de/module-image/modules.list
[8] https://mpcdf.pages.mpcdf.de/ci-module-image/

Technically, Condainer uses a Python basis from Miniforge (which is a free alternative to Miniconda) and then installs the user-defined software stack from the usual `environment.yml` file. Package dependency resolution and installation are extremely fast thanks to the `mamba` package manager (an optimized replacement for `conda`). As a second step, Condainer creates a compressed squashfs image file from the staging installation, before it deletes the latter to save disk space. Subsequently, the compressed image is mounted (using `squashfuse`) at the very same directory, providing the full Conda environment to the user who can `activate` or `deactivate` it, just as usual. Moreover, Condainer provides functionality to run executables from the Conda environment directly and transparently, without the need to explicitly mount and unmount the image.

Please note that the squashfs images used by Condainer are not "containers" in the strict terminology of Docker, Apptainer, or alike. With Condainer, there is no process isolation or similar, rather Condainer is an easy-to-use and highly efficient wrapper around the building, compressing, mounting, and unmounting of Conda environments on top of compressed image files.

Basic usage examples

Build a compressed environment

Follow along the following commands once in order to build a compressed image of a Conda environment that is defined in 'environment.yml':

```
# on MPCDF systems, e.g. Raven:
module load condainer
# create specific project directory:
mkdir my_cnd_env && cd my_cnd_env
# initialize project directory with a skeleton:
cnd init
ls
# edit the 'environment.yml' example file,
#  or copy your own file here
# build the environment and compressed image:
cnd build
ls
```

Activate a compressed environment

After building, you can activate the environment for your current shell session, similar to plain Conda or a Python virtual environment:

```
source activate
```

Please note that `source activate` will only work with bourne shells (e.g. `bash` or `zsh`), not with the older C shells and korn shells.

Alternatively, run an executable from a compressed environment directly

In case you do not want to activate the environment, you can run individual executables from the environment directly, e.g.

```
cnd exec -- python3
```

The `cnd` command supports the flag `--directory` to specify a certain Condainer project directory, allowing for arbitrary current working directories.

Limitations

As the squashfs fuse mounts are specific to an individual compute node, Condainer currently (v0.1.8) does not support multi-node batch jobs.

Availability

The software including its documentation is freely available via the MPCDF gitlab[9]. Moreover, it is provided via the environment module `condainer` on the *Raven* HPC system, and will be offered on more systems in the near future.

Klaus Reuter

New Features in the HPC-Cloud

After commissioning the initial set of compute and storage resources in 2021[10], deploying the GPU- and NVMe-focused extension[11] earlier this year, and rolling-out integrated object storage[12], MPCDF has recently deployed several new features to better support the diverse technical requirements of current and future projects:

Expanded menu of flavors and images

Projects now have access by default to more combinations of vCPUs and memory, known as *flavors* within the cloud environment, so that virtual machines can be sized to match the application. In practice, MPCDF now offers flavors up to 24 vCPUs and 64 GB of memory, subject to certain boundary conditions which ensure they can be efficiently mapped to a physical compute node. As before, even larger flavors as well as local SSD-, GPU-, and NVMe-enabled versions of the default flavors can be created on request.

Several new virtual machine templates, known as *images*

[9] https://gitlab.mpcdf.mpg.de/mpcdf/condainer
[10] https://docs.mpcdf.mpg.de/bnb/207.html#hpc-cloud
[11] https://docs.mpcdf.mpg.de/bnb/212.html#mpcdf-hpc-cloud
[12] https://docs.mpcdf.mpg.de/bnb/213.html#hpc-cloud-object-storage

within the cloud environment, have been prepared, including AlmaLinux as well as newer releases of CentOS Stream, Debian, and openSUSE Leap. Commercial operating systems such as Red Hat Enterprise Linux and SUSE Linux Enterprise Server are also available on a BYOL (bring your own license) basis.

SSD-based block volumes

There is now an SSD-based block volume type *CephSSD* representing a middle-ground option between highly-performant local SSDs and the highly-flexible and scalable network-based HDD storage associated with the default volume type. While I/O performance cannot be guaranteed in a shared-resource environment, one can expect a roughly 2X speedup in terms of small I/O operations per second and large transfer bandwidth, as well as a significant reduction in latency.

To evaluate whether the new volume type is a good option for your project, please contact the cloud enabling team via the helpdesk[13]. As a tip, existing block volumes can be migrated between types *online*, making it relatively simple to test an already-deployed application.

In addition to the new volume type, all Ceph-based system disks, i.e. the OS root of the VMs not hosted on local SSDs, have been transparently migrated to an SSD pool "for free", so that routine tasks such as software installation and updates complete more quickly.

Automated domain name service

Hostnames are now automatically generated for most devices attached to the public or local cloud networks, including virtual machines and floating IP addresses. The system works like this:

1. Each virtual machine is assigned a hostname of the following form:

   `VM_NAME.PROJECT_NAME.hpccloud.mpg.de`

   If the name of the virtual machine is invalid according to the requirements of DNS, then a unique hostname based on the fixed IP address will be substituted automatically.

2. Each floating IP is assigned a hostname of the following form:

   `FIP_DESCRIPTION.PROJECT_NAME.hpccloud.mpg.de`

   If the description field is empty or invalid, then a unique hostname based on the floating IP address will be substituted automatically.

3. Hostnames are synchronized with the MPCDF DNS servers every five minutes. For devices on the public cloud network, both forward and reverse entries are propagated to the global DNS, whereas on local networks only the forward (i.e. hostname->IP address) entries are published.

Thus, within the framework described above it is possible to deploy and configure many applications on the HPC-Cloud without tracking individual IP addresses.

Shared filesystem service

The HPC-Cloud now implements shared filesystem-as-a-service (FSaaS), also known as OpenStack Manila or simply *Share* within the cloud dashboard, as an alternative to Nexus-Posix project directories. The two technologies are complementary, with differing strengths and weaknesses summarized in Table 1:

The key advantages of Manila-based shares are that they can be provisioned quickly by project admins and, being logically isolated from other systems, can easily handle arbitrary UIDs and GIDs including root and/or service users. On the other hand, Nexus-Posix as an MPCDF-administered filesystem supports secure interoperation with *Raven* and other systems as well as built-in backups. To evaluate filesystem-as-a-service for your project please get in touch with the cloud enabling team.

More details about these new features can be found in the technical documentation[14].

John Alan Kennedy, Brian Standley, Maximiliano Geier

The Robin cluster

The Remote Visualization Service at MPCDF has recently been expanded with a new cluster called *Robin*. *Robin* is the first compute cluster of MPCDF in the HPC-Cloud and its resources are available to all users with an access to the HPC systems (i.e. *Cobra* and *Raven*).

One of the main advantages of the *Robin* cluster is its flexibility: new nodes can be easily deployed in the HPC-Cloud, automatically configured and added to the Slurm cluster, allowing for a convenient scaling of the compute resources on *Robin* depending on the current demand.

Robin uses Slurm as a job scheduler and it can currently host up to 20 CPU sessions and 12 GPU sessions, concurrently. Access to the cluster is restricted via our Remote Visualization Service web interface, so that users are not allowed to connect directly via ssh to the login or compute nodes of *Robin*.

Each session on *Robin* provides 12 virtual CPUs and 64 GB of RAM, with GPU sessions having access to a shared Nvidia A30 GPU (up to 4 sessions can share a single GPU). *Robin* mounts the *Raven* file systems, providing access to all the software and data available on the *Raven* cluster, including the user's home directory and ptmp folder. A runtime of up to 7 days is currently allowed, with a plan to increase to up to 28 days of maximum runtime in the future, but users are encouraged to stop their sessions once their calculations are completed and should be aware that long-running jobs can be killed in case of maintenance of the cluster.

[13] https://helpdesk.mpcdf.mpg.de
[14] https://docs.mpcdf.mpg.de/doc/cloud/technical/

Service	Technology	Protocol	External access	Typical scale	Lifetime	Provisioning
FSaaS	CephFS,	NFS or native CephFS	*none*	100 GB. . . 50 TB	*arbitrary*	self-provisioned via dashboard/API
Nexus-Posix	IBM Storage Scale (GPFS)	NFS	*Raven*, *Robin*, GO-Nexus	10 TB . . . 100 TB, or larger	months to years	on request via helpdesk ticket

Table 1: Comparison of Nexus-Posix and the new filesystem-as-a-service (FSaaS)

Users requesting GPU sessions are encouraged to limit the memory used by their code to roughly 1/4 of the available GPU memory (~6 GB out of the 24 GB available), in order to avoid disrupting the calculations of other users sharing the same GPU. This is particularly important for Machine Learning software (e.g. Tensorflow, Pytorch) that can allocate the entire available GPU memory for a single process.

Robin is designed to provide a single solution for the remote visualization needs of future HPC clusters at MPCDF: the filesystem of new clusters (like the upcoming *Viper*) can be made available on *Robin*, providing easy access to software and data without the need of a dedicated installation of the Remote Visualization Service on each cluster.

Users interested in using the Remote Visualization Service on *Robin* are reminded to initialize their sessions on the cluster once (before submitting their first session), as described in our documentation[15].

Michele Compostella

News & Events

AMD-GPU development workshop

In preparation for the new supercomputer *Viper*[16] of the MPG with AMD MI300A GPUs to be installed in 2024, the MPCDF in collaboration with AMD offered an online course on AMD Instinct GPU architecture and the corresponding ROCm software ecosystem, including the tools to develop or port HPC or AI applications to AMD GPUs. The workshop was held as an online event, spanning three afternoons on November 28-30, 2023. The workshop material can be found on the MPCDF training website[17].

Tilman Dannert

Meet MPCDF

The monthly online-seminar series "Meet MPCDF" skips the talk in January 2024. The next edition will take place on February 1st, 2024 with a talk on "ScaLAPACK and ELPA: how to diagonalize really large dense matrices" given by Petr Karpov from the MPCDF. Subsequent dates will be March 7th and April 4th (topics to be announced). All announcements and material can be found on our training webpage[18].

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

Tilman Dannert

RDA Deutschland Tagung 2024

The German chapter of the Research Data Alliance[19] will have its next conference in Potsdam, February 20-21, 2024. This year's focus is on legal, administrative and organizational topics concerning research data management in Germany and Europe. The early registration deadline is January 12, 2024. Further details including the program are available from https://indico.desy.de/event/42727/. As in previous years, MPCDF is contributing to the organization of the event.

Raphael Ritz

[15] https://docs.mpcdf.mpg.de/doc/visualization/index.html
[16] https://docs.mpcdf.mpg.de/bnb/212.html#new-supercomputer-of-the-mpg-cobra-successor
[17] https://www.mpcdf.mpg.de/services/training
[18] https://www.mpcdf.mpg.de/services/training
[19] https://www.rda-deutschland.de/