No. 204
August 2020

Bits & Bytes

Max Planck Computing & Data Facility (MPCDF)*
Gießenbachstraße 2, D-85748 Garching bei München

Two-factor authentication at the MPCDF

Andreas Schott, Amazigh Zerzour

Maybe some of you have heard of the break-ins into many German and European supercomputers in May (see hpcwire). Fortunately, the MPCDF was not affected and could continue without shutdown. This massive incident made clear that it is mandatory to use high standards for the security of the authentication process of users, and to improve the process, where needed. For this reason, the MPCDF has decided to make two-factor authentication (2FA) available for all regular users. This means that in addition to the kerberos password, a so-called one-time password (OTP) will be required for login.

For some years now, the sysadmin work at the MPCDF has already been secured by a centralized 2FA solution. The positive experience gained from this convinced us to set up 2FA for all users of the MPCDF now. To enable 2FA a device providing the second factor is required for each user. The primary option for that will be an app on the user's mobile phone assuming that most of our users have one. Alternatively, users can opt to use a hardware token, which the MPCDF will provide on request.

The MPCDF SelfService (selfservice.mpcdf.mpg.de) will as of September 2020 offer an entry to activate 2FA. Here you will be able to scan a QR code with the app on your mobile phone. The app or the alternative hardware token will display a 6-digit number, which changes every 30 seconds, as OTP, to be provided as a second password during login. Amazon, Dropbox and Google offer this same method already as so-called "2-step verification".

In case your OTP app is accidentally deleted from your phone or you forgot your hardware token at home, you can add support for SMS tokens by providing a mobile number. If you do not have your mobile at hand, you can also request an OTP via e-mail. Ideally, you configure at least one of these backup methods, SMS or e-mail, immediately after enabling 2FA in the SelfService portal.

Initially, 2FA will be supported only on a few machines and services, e. g. the SelfService portal. After all users are equipped with 2FA, it will become mandatory on all SSH gateway machines and on the HPC login nodes. Other MPCDF services like GitLab or DataShare are envisioned to follow.

With the introduction of 2FA the access to MPCDF resources will become significantly more secure, and privacy of your data will improve. As soon as all parts are ready for use, all users will be informed by e-mail.

High-performance Computing

Markus Rampp, Klaus Reuter, Hermann Lederer, Renate Dohmen

Max Planck supercomputer Raven

The first CPU segment of the new Max Planck supercomputer Raven has been delivered and is currently being installed at the MPCDF. This interim system will provide 514 compute nodes based on the Intel Xeon CascadeLake-AP processor (Xeon Platinum 9242) with 96 cores per node (4 "packages" with 24 cores each), and 384 GB of main memory (24 memory channels) per node. The CascadeLake microarchitecture is very similar to the SkyLake microarchitecture which MPCDF users are familiar with on the HPC system Cobra and many other clusters. The Raven nodes are interconnected with a Mellanox HDR InfiniBand network (100 Gbit/s) using a non-blocking fat-tree topology over all nodes. Raven will be operated with virtually the same operating system, application-software stack (including latest Intel and GCC compilers, Intel MPI, tools and libraries) and batch system (slurm) as Cobra, this means that users will experience a smooth transition which will essentially require only a recompilation of their codes and minor adaptations to the batch submit scripts. Early user operation on Raven is expected

*Tel.: +49(89) 3299-01, e-mail: benutzerberatung@mpcdf.mpg.de, URL: https://www.mpcdf.mpg.de/

Editor: Renate Dohmen

to start in September 2020. In the first half of 2021 this interim system will be replaced by the final CPU system which will be roughly twice as powerful and employs the upcoming Intel Xeon IceLake processor. In this time frame also 192 GPU nodes with 4 Nvidia Ampere GPUs each will become available.

New Cobra HPC stack

Early in July, in addition to an upgrade of the operating system, a completely new set of software modules was deployed on the Cobra HPC system. Major upgrades are:

- new OS: SLES12 SP5 (was: SLES12 SP4)
- new OmniPath software: opa 10.10.2.2.1 (was 10.10.1.0.36)
- new Nvidia drivers: 440.95.01 (was: 440.64.00)
- new slurm version: 20.02.3 (was: 18.08.8)
- new Intel compiler and library module defaults from Intel Parallel Studio 2020.1: intel/19.1.1 (was: 19.0.4), mkl/2020.1 (was: 2019.4), impi/2019.7 (was: 2019.4)
- new cuda module default: cuda/10.2 (was: cuda/10.1)

Although the old software stack can still be found on the file system, it is considered obsolete and none of the new modules refer to it. Even though old executables may still work, we strongly encourage users to re-compile their codes with the new stack of compilers, MPI, and dependent libraries.

Hints for using Intel MPI across version updates

In addition to the regular update routine of the operating system and related software, the compiler and scientific library stacks of HPC systems need to be kept up-to-date in order to stay within the support windows of the providers, to roll out performance improvements and bug fixes, and to offer new features. Unfortunately, such updates are prone to cause job errors for user-compiled applications at run time, in particular when the versions of the Intel MPI library used during compile time, submit time, and run time differ as a result of a change of the corresponding environment module defaults that may occur during a scheduled maintenance.

Therefore we generally recommend to always explicitly specify software versions, e. g. currently on Cobra by using

```
module purge
module load intel/19.1.1 impi/2019.7 mkl/2020.1
```

interactively at compile time and, consistently, via the batch script at run time. On Cobra, and similarly on Draco, such a set of default modules is loaded automatically upon login (check 'module list') and within a batch job for convenience, but only the recommended explicit loading with a specification of the version ensures a consistent and robust behaviour across maintenances. Ideally, please recompile and test your codes with the announced new software stack already in advance of any maintenance, or do so now if you haven't done yet after the last Cobra maintenance on July 1st.

Since it is absolutely crucial to use the same version of the Intel MPI library for building and running an MPI application, the explicit specification of the versions for Intel compilers and MPI will be enforced in the future on MPCDF systems (starting with Raven), and no set of default modules will be loaded automatically anymore.

Slurm memory usage statistics and notes on interpretation

On the HPC system Cobra, additional information about the memory usage was added to the basic CPU and memory utilization reports which are appended to standard output of batch jobs. The slurm batch system records a "high water mark" of the memory usage during the lifetime of a job, individually for each MPI task (process). From this figure a maximum (MaxRSS) and an average (AveRSS) value is computed for all tasks. In addition to MaxRSS our memory utilization reports now also contain AveRSS. A large discrepancy between MaxRSS and AveRSS indicates that some tasks require more memory than others. In such situations the derived quantity "Maximum memory per node" (which we need to derive from MaxRSS by multiplying with the number of tasks per node) has to be cautiously interpreted as a conservative upper bound for the total memory requirements. With the help of the command sacct (cf. 'man sacct') users can query comprehensive statistics about finished jobs, including information about the location (node, task) of the memory "high water mark". Example:

```
$ sacct --units=M -j 2859791.0 \
    -o "AveRSS,MaxRSS,MaxRSSNode,MaxRSSTask"
    AveRSS     MaxRSS MaxRSSNode MaxRSSTask
---------- ---------- ---------- ----------
   229.50M    666.08M     co1040          0
```

Rclone - The Swiss army knife of cloud storage

John Alan Kennedy

Rclone is a command-line program to manage files on remote/cloud storage. Rclone has a rich set of features and supports over 40 cloud storage systems including ownCloud (Datashare), OpenStack Swift, as well as standard transfer protocols (HTTP, SFTP, FTP) and local filesystem. Rclone's ability to connect to many different storage services makes it a real Swiss army knife when it comes to moving and managing data. It is a very valuable tool for modern day researchers, whose data is often located in several different data silos.

Within Rclone each storage resource is configured as a so-called remote. Calling "rclone config" from the command line will open an interactive configuration session:

```
rclone config
e) Edit existing remote
n) New remote
d) Delete remote
r) Rename remote
c) Copy remote
s) Set configuration password
q) Quit config
e/n/d/r/c/s/q>
```

Witihin this session, remotes can be added and/or altered. Alternatively, you can call "rclone config" with a specific configuration option directly.

Once remotes are configured they may be accessed to list content

```
$ rclone ls remote:path
```

Data may be copied or moved between remote storage resources as follows:

```
$ rclone copy source:sourcepath dest:destpath
$ rclone move source:sourcepath dest:destpath
```

The actual data transfer runs through the Rclone client.

Additionally Rclone allows for data syncing (similar to rsync):

```
$ rclone sync source:path dest:path
```

This will sync the source to the destination, changing the destination only. Unchanged files will not be transferred and files at the destination may be deleted. Since this can cause data loss, always test first with the --dry-run flag to see exactly what would be copied and deleted.

Note: Be advised that "rclone sync" acts differently to rsync w.r.t. the creation of target dirs, rclone will not auto-create dirs on the target. For instance: **never** do "rclone sync somedir datashare". This will delete all the data in datashare – replacing it with the data in somedir (using --dry-run will help avoid such problems).

To use local storage simply omit the remote prefix and use the data path as usual.

Several remotes can be configured (using different protocols), allowing you to easily move data between services. The example below shows remotes configured to connect to aws-s3, the MPCDF DataHub and Datashare services and an OpenStack Swift instance.

```
Current remotes:
Name                 Type
====                 ====
aws                  s3
datahub              sftp
datashare            webdav
openstack            swift
```

Once this configuration is setup data can be easily moved between Datashare, Datahub and local storage as well as any cloud-based storage (in this case swift and s3).

Rclone is a Go program and can be installed as a single binary file. For more information and to download Rclone please see the official Rclone website: https://rclone.org/.

Some notes on safe configurations: When a remote is configured in Rclone the remote password is saved, in obscured mode, in the Rclone configuration file. To secure the passwords you can create a password for the rclone configuration itself. When "rclone config" is called from the command line, you will see several options. If you select "s" you can set a configuration password (see below).

```
$ rclone config

s) Set configuration password

e/n/d/r/c/s/q> s
Your configuration is not encrypted.
If you add a password, you will protect your
login information to cloud services.
a) Add Password
q) Quit to main menu
```

Once a secure configuration file has been created you will need to provide a password each time you start an Rclone session. In addition when using owncloud/datashare an app password can be generated within data share and used on a client-by-client basis. This brings additional security and we would advise you to ALWAYS create a secure configuration file.

One final recommendation: Before moving very large data between remotes, check capacitites and network bandwidth. For more information about Rclone visit the project's homepage: https://rclone.org/

ELPA eigensolvers further enhanced

Andreas Marek, Hermann Lederer

During the last year the ELPA eigensolver library has been enhanced in two areas.

First, in a collaboration with the MPI for Dynamics of Complex Technical Systems, extensions have been developed to allow not only for the treatment of symmetric matrices, but also of real-valued skew-symmetric matrices. This feature is especially important for applications solving the Bethe-Salpeter Eigenvalue Problem. This additional functionality has been provided with ELPA release 2019.11. For more details see C. Penke et al., Parallel Computing 96, 102639 (2020).

Second, the most recent ELPA release 2020.5 now includes a GPU version based on the two-stage solver elpa2. This development has been achieved through a collaboration with the Duke University in Durham, NC and with Nvidia. For most cases the performance of this new GPU version is superior to that based on the one-stage solver elpa1.

The open-source ELPA library can be downloaded from https://elpa.mpcdf.mpg.de.

News & Events

Tilman Dannert, Hermann Lederer

Advanced HPC Workshop for MPG users (MPCDF, November 2020)

From November 23rd to 26th the third of our Advanced HPC Workshops will take place as pre-announced in the previous issue (No. 203) of Bits&Bytes. Due to the uncertainties of the Corona situation, only remote participation can be offered for this workshop. The online workshop will start around noon on Monday, with two and a half days of lectures given by High-performance-computing (HPC) application experts of the MPCDF, Intel, and Nvidia, followed by a day of hands-on sessions for a number of selected projects on Thursday. If you are interested in advanced topics of HPC debugging, profiling and optimization, please register by October 31st on the following webpage: https://www.mpcdf.mpg.de/cgibin-secure/hpc-workshop/register.pl. If, in addition to the lectures, you would like to work in a small group on a "bring-in" code on Thursday, 26th of October, please indicate this, by the end of September, via the same registration form.

To meet the focus of this workshop, we require participants to bring along a certain familiarity and experience with HPC programming. Beginners in HPC programming are referred to existing training programs (cf. https://www.mpcdf.mpg.de/services/computing/training), as well as to a newly developed series of introductory workshops on using HPC facilities at the MPCDF which will start late in 2020 (details to be announced at the MPCDF webpage).

Further EU support for NOMAD CoE

The Novel Materials Discovery (NOMAD) Centre of Excellence (CoE) has received a new round of Horizon 2020 funding of 5 million euros starting in October 2020 with a duration of 3 years. The prediction of novel materials with specific desirable properties is an important goal. Such materials can have immense impact on the environment and on society, e. g. on energy, transport, IT, medical-device sectors and much more. Currently, however, precisely predicting complex materials is computationally infeasible.

The mission of the NOMAD CoE is to develop a new level of materials modelling by exploiting HPC, including upcoming exascale technologies, and extreme-scale data. The international project with 12 partners is coordinated by the Fritz Haber Institute of the Max Planck Society, and the MPCDF will make contributions especially in the pillar exascale codes to the work package exascale DFT by further extending the ELPA eigensolver library into the exascale regime.

NOMAD offers several open positions for master and PhD students as well as Postdocs at several high-level institutions in Europe, see https://nomad-coe.eu/open_positions. For more details, please see https://nomad-coe.eu/.