



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

Singularity: containers for HPC

Christian Guggenberger, Mykola Petrov

Singularity (<http://singularity.lbl.gov/>) is an open-source project which enables containers to be deployed in high-performance scientific computing environments. Through Singularity, scientists, developers and system administrators have the possibility to work with full control of their environments, including operating system, application software and dependencies. This environment can then be easily distributed and executed on other platforms in the form of containers.

The goals of Singularity are:

- mobility of compute – the ability to create and maintain a workflow, within a container, which can later be executed on different hosts, Linux clusters and cloud services;
- reproducibility – provide a unique representation of all the software and data files within a container, thus enabling the preservation and validation of data and scientific results;
- user freedom – within a container a user can install applications and dependencies for workflows without impacting the host operation system;

- security – Singularity blocks privilege escalation within and outside of the container;
- support of HPC resources – Singularity natively supports InfiniBand, existing cluster filesystems (e.g. GPFS), and works seamlessly with resource managers (e.g. SLURM, SGE, etc.). This is possible because running a Singularity container is just like running any other command on the host HPC system.

Singularity utilizes container images, however, numerous other formats are also supported (docker images, tar.gz, cpio etc). The mechanisms to obtain the images include the standard URI format (i. e. http(s) download) as well as downloads from Docker and Singularity image registries.

In order to provide Singularity for researchers of the Max Planck Society, the MPCDF has deployed Singularity on several resources. The Singularity software is now available on the Draco HPC cluster as well as on several other systems. Please contact MPCDF helpdesk, if you want to use Singularity on Draco. More information, including example use cases and problems that Singularity can solve, can be found on the MPCDF [software home page](#).

Transferring data to and from MPCDF

John Alan Kennedy

In this article we will outline the main tools which are in general use for data transfers at the MPCDF. Since each data transfer case is different we will break the tools down into three categories.

- Large-scale data transfers
- Medium and small-scale data transfers
- Sharing of small datasets (files)

Large-scale data transfers

For large-scale data transfers (often in the multi-TB range) the MPCDF has made good experience using the Globus Online and bbcp tools.

Globus Online: Globus Online (Globus.org) is a free service which allows users to move large volumes of data in simple and reliable manner. In general Globus Online requires sites to set up a Globus server for data transfers,

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

however, individual users can also install a personal client to enable them to move data to/from Globus servers from a user login node or a laptop.

The MPCDF supports the deployment of Globus Online servers for specific projects and Globus Online personal clients for individual users. This means that a Globus Online solution can be found on a project and/or user level – enabling data to be transferred between external sites and the MPCDF.

bbcp: bbcp is a peer-to-peer command-line tool which is invoked at both source and target to enable fast and reliable data transfers. bbcp is made available on MPCDF clusters via the modules environment.

To enable bbcp:

```
module load bbcp
```

To get help info:

```
bbcp --help
```

Transferring medium and small data sets

For transfers of smaller data sets, in the GBs range, rsync and scp are perfectly suitable. The rsync and scp tools can be used for transferring small data sets between ssh enabled servers. Both tools are universally available and well documented, making them ideal candidates for data transfers. The rsync tool is a natural fit when syncing data and can improve data transfer speeds by simply avoiding transferring data that has not changed.

Exposing data via MPCDF DataShare

The MPCDF DataShare service provides MPCDF users with a web-based sync-and-share service. This service allows users to upload and share data with external collaborators, or to simply upload it for later download from an external site (or home PC/laptop/tablet). This makes the DataShare service a perfect fit for exposing and sharing smaller data sets, or single files (documents). Users can simply upload their data from their cluster and share it via DataShare.

To enable this service for users of the HPC and Linux clusters a command-line client called `pocli` (Python ownCloud command-line interface) was developed. The `pocli` client supports basic operations such as upload or download of single or multiple files, directory creation, and file or directory removal. For more information about the `pocli` client see: [Bits&Bytes No. 195](#).

Conclusion

Many tools exist to support the various data transfer use-cases. Here we have outlined three major use-cases and highlighted tools which help in each case. Understanding how these tools relate to each other and which is the best fit for user data transfers often involves some hands-on evaluation and testing.

As always the MPCDF support is available to answer questions – please submit a helpdesk ticket in cases where you feel support and/or advice is needed when transferring data. For more information about data transfer to/from MPCDF see [MPCDF's home page](#).

DataShare via WebDAV

Thomas Zastrow

The MPCDF DataShare service for data sharing and synchronization can be used in several ways. The easiest way is the web interface (<https://datashare.mpcdf.mpg.de>), which can be used to upload/download files, create folders etc. Another way is via the desktop clients, which provide a permanent data synchronization between the DataShare server and the personal desktop.

If the combination of a web interface and a desktop client is not sufficient, or possible, a third alternative exists in the form of WebDAV. Through the WebDAV protocol, any DataShare account can be mounted as a network drive into your local machine. On most operating systems, this works without installing additional software.

At first, you have to login to the DataShare web interface. In the bottom left corner of the screen, you can see a link to 'Settings'. Opening the Settings, you can copy the shown WebDAV link to your clipboard

(<https://datashare.rzg.mpg.de/remote.php/webdav/>).

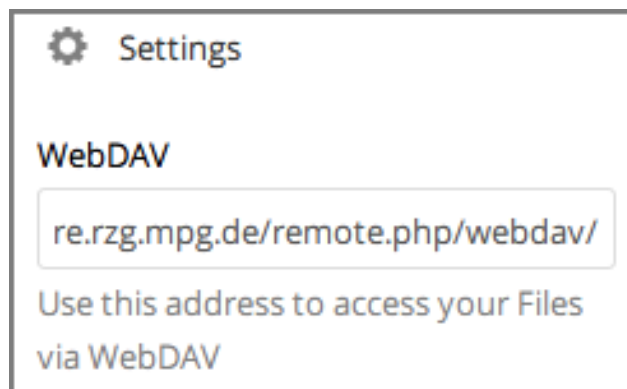
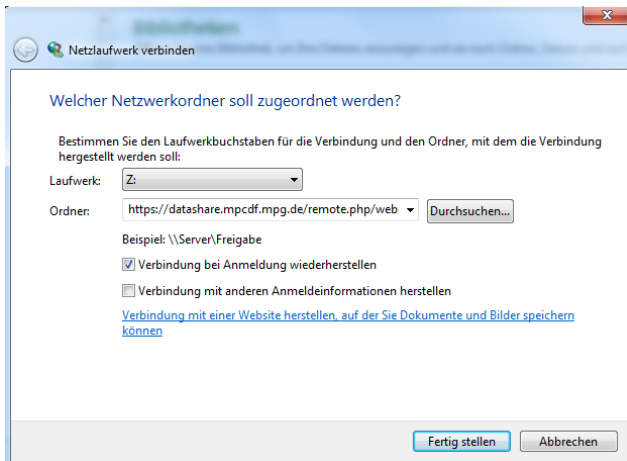


Fig. 1: WebDAV Settings in DataShare

Connecting a Windows 7 machine

Open the Windows Explorer and right click on 'Network'. Choose 'Netzlaufwerk verbinden'. After entering your personal credentials in the following window, your DataShare account is mounted as a netdrive and you can find it in the Explorer under 'Computer'.



Linux

Under Linux, there are two ways of accessing a WebDAV share. First, most modern file managers like Dolphin under KDE or Files under Gnome offer the possibility to connect a WebDAV-based network share. Second, a WebDAV share can be mounted on the command line via the 'davfs2' package:

```
apt-get install davfs2
```

2. Now the 'root' user can mount your DataShare account – choose an existing mountpoint as last parameter,

```
sudo mount -t davfs https://datashare.rzg.mpg.de/remote.php/webdav/ /media/datashare/
```

3. It is also possible to add the DataShare account via `/etc/fstab`. After adding a line:

```
https://datashare.rzg.mpg.de/remote.php/webdav/ /media/datashare davfs user,rw,noauto 0 0
```

the root user can mount the WebDAV share into the given mountpoint:

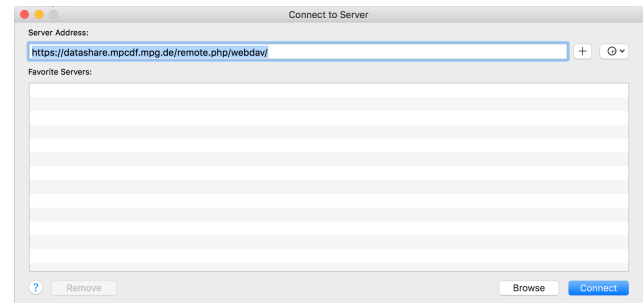
```
sudo mount /media/datashare
```

Remarks

The WebDAV protocol was never meant to transport large amounts of data. Additionally, the handling of a large

Apple Mac

In the Mac's Finder, go to 'Go / Connect to server'. In the popup window, enter the DataShare URL:



In the next dialog, enter your credentials and click 'Connect as registered user'.

1. Depending on your distribution, install the 'dav2fs' package. For instance, on Ubuntu or other Debian derivatives:

the command will ask you for your DataShare credentials:

number of (small) files can be difficult with WebDAV. In these cases, the MPCDF offers some other solutions for data transport (cf. [Bits&Bytes article on 'data transfer methods'](#) in this issue).

System and software upgrade on the HPC extension system Draco

Ingeborg Weidl, Christian Guggenberger, Alexis Huxley, Markus Rampp, Klaus Reuter

In June 2017, the **operating system** of the HPC extension system Draco was upgraded from SLES11 to SLES12. This implied that all applications had to be relinked. Module software packages like GROMACS, NAMD, etc. were updated or relinked by the application support group of the MPCDF, and new module defaults were implemented for compilers, MPI, numerical libraries and applications. The upgrade of the operating system also required an upgrade of the SLURM batch system from slurm-15.08 to slurm-16.05. The queue configuration was not changed except some minor modifications of the memory limits of the large-memory nodes.

The most relevant **changes in the module defaults** are as follows:

- intel/17.0(default) was: intel/16.0
- gcc/6.3(default) was: gcc/5.4
- impi/2017.3(default) was: impi/5.1.3

- cuda/8.0(default) was: cuda/7.5
- mkl/2017(default) was: mkl/11.3
- fftw/3.3.6(default) was: fftw/3.3.4
- elpa/2017.05(default) was: elpa/2016.11
- magma/2.2.0(default) was: magma/2.0.2

Users are advised to recompile their application using the new default compilers and libraries.

Please note that the **Python environment modules** were restructured for reasons of clarity and maintainability, e. g. Anaconda Python 2 is now provided by the module 'anaconda/2/4.3.1' (in contrast to the previous naming 'anaconda/2_4.3.1'). Moreover, after loading an 'anaconda' environment module, additional depending Python packages become available via environment modules. See the output of 'module avail' for a list of additional Python packages.

MPCDF's GitLab now supports Continuous Integration & Delivery

Raphael Ritz, Florian Kaiser, Thomas Zastrow

In software development [Continuous Integration](#) is the practice of integrating code into a shared repository and building or testing each change automatically, as early as possible – usually several times a day. [Continuous Delivery](#) in addition ensures that the software can be released to production at any time, often by automatically pushing changes to a staging system. Continuous Deployment goes even further and pushes changes to production automatically.

The benefits of adopting **Continuous Integration** include: (i) Detection of errors as quickly as possible, (ii) Possibility to fix them while your changes are fresh in your mind, (iii) Reduction of integration problems, as smaller problems are easier to digest, (iv) Less compound problems, (v) Allowing teams to develop code faster and with more confidence.

Continuous Delivery on the other hand ensures that every change to the system is releasable. It lowers the risk of each release and makes releases 'boring'. Developers

can get fast feedback on what their users care about.

The MPCDF through its [GitLab service](#) now supports Continuous Integration and Delivery. To get started it is strongly recommended to read the [Quick Start documentation](#) first. One important concept to understand is that of a [Runner](#), as that's where the build will be performed and where your test suite will be executed – should you have one. In general, you are required to provide your own runner(s), however, for minor jobs and for initial testing the MPCDF provides a *shared runner*.

Some test suites and build processes produce output that can be shared with collaborators on the web, the MPCDF's GitLab supports this functionality through [GitLab Pages](#).

In order to enable **Continuous Integration** or **Pages** for a specific GitLab project go to the Settings section of that project and then choose Pipelines or Pages, respectively, and follow the instructions there. If you have questions or need help, please turn to gitlab@mpcdf.mpg.de.

Events

Tilman Dannert, Hermann Lederer, Raphael Ritz

HPC Training Workshop at the MPCDF

The MPCDF is organizing a workshop for teaching advanced HPC practices. It will take place from October 17th to 19th at the MPCDF in Garching. During the first two half-days, experts from the MPCDF will give lectures on several advanced HPC techniques like software engineering, Continuous Integration, debugging and profiling serial and parallel codes. The second part of the workshop will be dedicated to hands-on sessions on the users' codes. For the lecture part there are still some seats left, so feel free to send an e-mail to hpcworkshop-17@mpcdf.mpg.de to register.

HPC Summer School 2017

This year's International HPC Summer School on Challenges in Computational Sciences took place in Boulder, Colorado, United States, from June 25th to 30th, hosted by XSEDE on the campus of the University of Colorado. This unique expense-paid event brought together 80 excellent PhD students and postdocs from institutions in

Canada, Europe, Japan and the US to participate for one week in an exciting summer school with a sophisticated program, covering many disciplines and programming techniques. In addition to the diversified program, intensive mentoring and networking made this again a world-wide unique and highly successful event. The 2018 HPC Summer School is expected to take place in July in a location in Europe, with a call opening in Dec/Jan (see <http://www.ihpcss.org>).

Envri+ Summer School 2017

The Research Data Alliance Europe (coordinated by the MPCDF) together with the [ENVRiplus Cluster](#) organized a [Summer School on Data Management and Data Science](#). The school took place in Espoo, Finland and ran from July 12th to 16th, 2017. Topics ranged from organizing and using a proper data repository via data registration, typing and collection building all the way to data analysis using notebooks. A recent [blog post by one of the participants](#) provides a nice summary of many of the aspects covered.