No. 193
August 2016

Bits & Bytes

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

DRACO as an extension of HYDRA

Ingeborg Weidl, Hermann Lederer

The DRACO system was delivered in May and put into production in July 2016 as an extension of the HYDRA supercomputer. Through this added compute resource, the long waiting times on HYDRA shall be reduced, and HPC needs of recently established theory departments in institutes in Frankfurt, Hamburg and Stuttgart shall be satisfied.

DRACO consists of 879 compute nodes with Intel Haswell processors E5-2698v3 (at 2.3 GHz), with 32 cores and 128 GB of main memory per node. Four of the nodes contain 512 GB RAM, and 106 nodes are equipped with GPU accelerator cards of type GTX 980. The system has a total of about 28.000 cores, 116 TB RAM and a peak performance of about 1 PetaFlop/s. There are additional four login nodes and eight I/O nodes that serve the additional 1.5 PetaByte of disk storage. InfiniBand FDR14 is the same interconnect type as on HYDRA, but with a different blocking factor. The software stack on DRACO is as similar as possible to that on HYDRA, with two exceptions: For the batch system, the open source system SLURM is used, and for inter-node communication Intel MPI.

With respect to the total job mix, DRACO with its many 'compute islands' with non-blocking full fat tree interconnect up to 1024 cores (and blocking factor 1:8 between islands) shall be used for running the smaller batch jobs up to 32 nodes, while HYDRA (with a non-blocking full fat tree interconnect up to 1800 nodes (36.000 cores) in the 'large' island) will be reserved for the larger batch jobs.

Detailed information about the extension system DRACO can be found on http://www.mpcdf.mpg.de/services/computing/draco.

VM Backups

Florian Kaiser

Virtual machines are secured against individual hardware failures by redundancy (servers) respective RAID (storage). However – just like a normal server – by default there is no backup in case of accidental deletion of data inside the VM or catastrophic failure of the whole cluster and/or storage. Ultimately, preventive measures are up to the administrator of the VM. While the probability of a catastrophic failure is low, we encourage you to plan for this to ensure that no important data is lost.

Due to many different usage types of the virtual hosting environment, unfortunately there exists no one-size-fits-all solution. Depending on the use case, there are multiple options for backup available, from manual (e. g. 'mysqldump && rsync') to redundant services (e. g. MySQL Master-Slave replication), TSM Backup to tape (geo-redundant copies, potentially long restore times) and Snapshot-Backups inside VMware (only feasible for smallish VMs, VM is frozen for a short time while the snapshot is taken).

Please contact the MPCDF support to discuss individual, best-fitting solutions for your VMs.

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

TeD-T: The Term Definition Tool

Thomas Zastrow

TeD-T, the 'Term Definition Tool', is a web application for creating and managing interdisciplinary scientific terminology. It allows to define for every term a bunch of definitions in a hierarchical order.

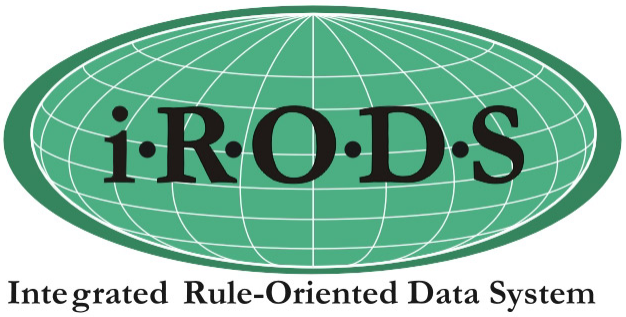
TeD-T was originally put in place to support the DFT Working Group of the Research Data Alliance (RDA). The DFT task was to describe a basic, abstract data organization model to enable better understanding within and between communities. Due to the open nature of the RDA project, everybody is invited to participate in the further development of the DFT terminology.

TeD-T was developed, implemented and is hosted at the MPCDF (go to website). Starting as a pure installation of the Mediawiki software, soon it was clear that the group would need a more formalized tool for creating and editing terms. Therefore, the Mediawiki software was extended by the Semantic Mediawiki Extension. A complex data structure was implemented to fit the needs for the different user groups. Over the time, DFT members filled up the TeD-T with ca. 350 terms, a lot of them with more than one definition. The content of the TeD-T Tool is also available as (static) PDF file (ca. 200 pages), HTML and RDF output (download overview).

iRODS

Raphael Ritz

The Integrated Rule-Oriented Data System (iRODS) is open source data management software used by research organizations and government agencies worldwide. It virtualizes data storage resources, so users can take control of their data, regardless of where and on what device the data is stored. As data volumes grow and data services become more complex, iRODS is increasingly important in data management. It provides plug-in support for microservices, storage resources, drivers, and databases as well as extensive documentation, training and support services.



Benefits of iRODS include (i) the enabling of data discovery using a metadata catalog that describes every file, every directory, and every storage resource in the data grid; (ii) automation of data workflows, with a rule engine that permits any action to be initiated by any trigger on any server or client in the grid; (iii) secure collaboration, so users only need to log in to their home grid to access data hosted on a remote grid; (iv) data virtualization, allowing access to distributed storage assets under a unified namespace, and freeing organizations from getting locked in to single-vendor storage solutions. Sustainability of the iRODS technology is enhanced through the iRODS Consortium.

The MPCDF has experience with iRODS most notably through its contributions to the EUDAT project. Its safe replication service B2Safe uses iRODS in its default implementation. If you want to know more about iRODS either because you are confronted in one of your projects with an iRODS-based data grid or if you are just curious, feel free to get in touch with us – preferably via e-mail to the author.

iRODS is released under a BSD license. Packaged binaries and source code are available from the iRODS web site.

Uberftp - A powerful command line tool for gridftp based data management

John Alan Kennedy

With gridftp, as with the bbcp transfer tool, the data transfer over wide-area networks is improved by providing optional encryption and a speed up through the use of multiple parallel transfer streams/channels. When interacting with gridftp-based storage most projects use globus-url-copy, which is available as part of the globus package. A nice additional/alternative tool is Uberftp. Uberftp is a command-line tool which provides both interactive and non-interactive access to gridftp-based storage resources. In this article we will focus on the interactive use-case, for which Uberftp provides an intuitive and user-friendly interface and numerous commands for data management and file transfer.

To start an interactive Uberftp (gridftp) session simply generate a grid proxy and then connect to your gridftp server using Uberftp.

```
grid-proxy-init
uberftp <my-gridftp-server.mpcdf.mpg.de>
```

Data Management Commands:

Uberftp provides a rich set of commands for managing data, several of which are listed in the box below.

```
ls    - list remote dir contents
pwd   - print remote working dir
mkdir - mkdir remote dir
cd    - change remote working dir
rm    - remove a file
rmdir - remove a dir
chmod - change file/dir mode
```

Equivalent commands for managing the local resource are prefixed with 'l' e. g. 'lls' to list local dir contents.

File transfer Commands:

Simple commands are also provided to move single or multiple files between clients and servers (note: server-server transfers are also possible).

```
get  - get files from remote resource
mget - get multiple files from remote resource
put  - put file to remote resource
mput - put multiple files to remote resource
```

A full list of commands can be found by typing 'help' and detailed info about each command can be found by typing 'help ⟨command⟩'.

Example session:

An example of an interactive session is shown below. A user logs in to a gridftp server, creates a 'Test' dir, moves into this dir, lists the local files for transfer, and finally uses put and mput to copy files to the server.

```
[testuser@laptop ~]$ uberftp <my-gridftp-server.mpcdf.mpg.de>
220 GridFTP Server 9.4 (gcc64, 1450284869-85) [Globus Toolkit 6.0.1449889199] ready.
230 User testuser logged in.
UberFTP (2.8)> pwd
/home/testuser
UberFTP (2.8)> mkdir Test
UberFTP (2.8)> cd Test
UberFTP (2.8)> lls 1mb_test*
-r--------  testuser testuser        1048576  Jul 19 10:36  1mb_testfile_n2
-r--------  testuser testuser        1048576  Mar 16 14:06  1mb_testfile_n1
-r--------  testuser testuser        1048576  Jul 19 10:36  1mb_testfile_n5
-r--------  testuser testuser        1048576  Jul 19 10:36  1mb_testfile_n4
-r--------  testuser testuser        1048576  Jul 19 10:36  1mb_testfile_n3
UberFTP (2.8)> put 1mb_testfile_n2
1mb_testfile_n2: 1048576 bytes in 0.150461 Seconds (6.646 MB/s)
UberFTP (2.8)> mput 1mb_testfile_*
1mb_testfile_n2: 1048576 bytes in 0.133748 Seconds (7.477 MB/s)
1mb_testfile_n1: 1048576 bytes in 0.150529 Seconds (6.643 MB/s)
1mb_testfile_n5: 1048576 bytes in 0.143220 Seconds (6.982 MB/s)
1mb_testfile_n4: 1048576 bytes in 0.148138 Seconds (6.750 MB/s)
1mb_testfile_n3: 1048576 bytes in 0.135282 Seconds (7.392 MB/s)
```

This example session shows how Uberftp makes data management simple and speedy. Uberftp offers many more options, for instance commands to manage transfer protection levels, which we'll cover next.

Data encryption:

A common misconception about gridftp-based data transfers is the belief that the data is encrypted while being transferred. By default this is not the case, however a user can enable this. While using Uberftp you can set the protection level to ensure data is encrypted and/or integrity checked. The interactive session in the box below shows how to check and configure the protection level in Uberftp. Note: for globus-url-copy command line options are also available to enable data protection.

```
UberFTP (2.8)> prot
Protection set to Clear.

UberFTP (2.8)> help prot
This command configures the level of security on the data channel after
data channel authentication has completed. Clear means that the data will
not be protected. Safe means that the data will be integrity protected
meaning that altered data will be detected. Confidential means that the data
will be unreadable to third parties. Private mode means the data will be
confidential and safe.

Usage: prot [C|S|E|P]
C  Set protection level to clear.
S  Set protection level to safe.
E  Set protection level to confidential.
P  Set protection level to private.

UberFTP (2.8)> prot P
Protection set to Private.
```

Conclusion:

Uberftp is intuitive, easy to use, and can increase user productivity. It provides a simple and yet powerful interactive command-line interface to gridftp-based storage and is well worth evaluation for projects which use gridftp.