No. 195
April 2017

Bits & Bytes

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

Draco cluster extended by large-memory nodes

Ingeborg Weidl

In January 2017, the DRACO cluster was expanded by 64 Intel 'Broadwell' processor (E5-2698v4, 2,2 GHz) based nodes that were purchased by the Fritz-Haber Institute. The new nodes each have a main memory of 256 GB (248 GB for the application) and 40 cores (80 logical cores in hyperthreading mode).

To use these nodes, please specify

```
#SBATCH --partition=broadwell
```

in your job script on Draco. For an MPI job without hyperthreading, please set:

```
#SBATCH --ntasks-per-node=40
```

or for an OpenMP job:

```
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=40
```

However, if your application needs the big memory of the 'Broadwell' nodes, you may use only 32 tasks per node as well. Below is an overview of the partitions (batch queues) that are currently available on Draco:

Partition	Processor type	Max. CPUs per node	Max. Memory per node	Max. Nodes
small	Haswell	16	60 GB	1
express	Haswell	32 / 64 in HT mode	120 GB	32
short	Haswell	32 / 64 in HT mode	120 GB	32
general	Haswell	32 / 64 in HT mode	120 GB	32
fat	Haswell	32 / 64 in HT mode	504 GB	4
fat01	Haswell	32 / 64 in HT mode	248 GB	1
gpu	Haswell	32 / 64 (host cpus)	120 GB	32
viz	Haswell	32 / 64 in HT mode	248 GB	4
broadwell	Broadwell	40 / 80 in HT mode	248 GB	32

A simple command-line client for the MPCDF DataShare (ownCloud) service

Klaus Reuter, Florian Kaiser

Since its introduction in 2015 the MPCDF DataShare service has become very popular when it comes to file sharing with internal and external collaborators. In addition to browser-based operation, client software for PCs and mobile devices is available. However, a lightweight solution for the users of the HPC systems and the Linux clusters at the MPCDF has yet been missing. E. g. in order to upload a (large) file from one of the HPC or cluster file systems to the MPCDF DataShare service the file had to be transferred first to the user's PC, from where it was synchronized with DataShare. This workflow is tedious, inefficient and sometimes even infeasible.

To make life easier we have developed a simple command-

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

Editor: Renate Dohmen

line client called pocli (Python ownCloud command line interface). It supports basic operations such as upload or download of single or multiple files, directory creation, and file or directory removal.

To get started use the commands:

```
$ module load datashare
$ ds --help
```

Help on individual commands is available as well, e. g.:

```
$ ds put --help
```

Two basic usage examples are described in the following.

Example 1: To upload a file to your DataShare space use the command

```
$ ds put file.tar
```

As a second step, you can then log in to the DataShare web portal, share the file with another DataShare user, or create a download link for your external collaborator.

Example 2: Let's assume that there's a file located at 'data/file.tar' in your DataShare space (owned by you, or shared by another DataShare user with you). To download the file to the current working directory, issue the command

```
$ ds get data/file.tar
```

Technically, pocli is written in Python (tested with versions ⟩=2.7) on top of the pyocclient library. At the first invocation of the 'ds' command a configuration file '~/.ocrc' is created. It is preconfigured for the MPCDF DataShare service, but can be edited and adapted to any ownCloud service. The 'ds' command asks for the password at each invocation.

The pocli package is released under the MIT license. Further details and the software itself are available at https://gitlab.mpcdf.mpg.de/mpcdf/pocli.

BagIt - command-line tool supporting the BagIt compound data format

John Alan Kennedy, Thomas Zastrow

When transporting many small files over a network, or preparing them to be archived, it is often best to compile all files and subdirectories into one large archive file. Traditionally, this can be achieved by using Tar and Zip archives. However, for such activities, container formats like BagIt have several advantages and can be used hand-in-hand with traditional archive tools like Tar: BagIt is a hierarchical file packaging format which consists of a data payload and metadata files which contain checksums on data objects as well as user-specific metadata for the collection. By using the BagIt format users can ensure that their data collections are self-describing and can be validated after transport.

On the Draco supercomputer, the bagit tool may be enabled and usage information gained by using the following commands:

```
module load bagit
bagit --help
man bagit
```

The bagit tool may then be used to create or validate BagIt packages. In its simplest form, the bagit command takes a directory as parameter and creates a BagIt container in this directory. In doing this, BagIt will move all the data in the directory into a subdirectory 'data' and create additional files with metadata in the base directory of the BagIt package. Note: The bagit tool does NOT create a separate package/file, rather it transforms the directory passed as an argument into a BagIt format.

As an example – taking a bagItTest directory containing several data files:

```
bagItTest/
|-- Data1.hdf5
|-- Data2.hdf5
‘-- Data3.hdf5
```

Creating a BagIt package from this directory:

```
bagit bagItTest
```

Results in the following BagIt package which contains the data payload and metadata files.

```
bagItTest/
|-- bag-info.txt
|-- bagit.txt
|-- data
|    |-- Data1.hdf5
|    |-- Data2.hdf5
|    ‘-- Data3.hdf5
|-- manifest-md5.txt
‘-- tagmanifest-md5.txt
```

By default md5 checksums of the data files are created and recorded in the manifest-md5.txt file. In addition command line options allow users to choose alternate checksum algorithms (sha1, sha256, sha512).

It is also possible to specify some specific metadata manually on the command line or via config files (run 'bagit help' command for more infomation). For example, the following parameter adds a metadata entry 'Contact Name' to the metadata file bag-info.txt:

```
bagit --contact-name 'John Smith' bagItTest
```

Note: These metadata entries can also be set in a config file ~/.bagit.cfg, see the bagit man page for details.

Once the BagIt container has been created, the directory can be zipped or tared for archiving or transport. Exporting the new BagIt container out of Draco is easy with the help of MPCDF's DataShare service (see the article 'A simple command-line client for the MPCDF DataShare (ownCloud) service' in this Bits&Bytes edition). Thanks to the new datashare client, this is achievable in a few commands:

```
bagit bagItTest
tar zcf bagItTest.tgz bagItTest
ds put bagItTest.tgz
```

Once the BagIt package is available in datashare it can be shared as usual with external collaborators etc.

The bagit tool can also be used to validate bagit style packages. This can be achieved by using the following command:

```
bagit --validate bagItTest
```

The validation process will ensure that the checksums recorded in the BagIt metadata and the actual checksums of the data files within the BagIt package are consistent.

Archiving your data with HPSS

Manuel Panea-Doblado

The MPCDF provides an archiving solution for all registered users. If you have data that you want to keep for a long time, but you want to save space on your local storage, you can store the data in MPCDF's archive.

Data stored in the archive is written first to a large disk system. At a later time, the system will move the data from disk to tape (two copies are made on two different tapes). If you want to read your data back, the system will retrieve it automatically from tape. This implies some delay of typically between 1 and 10 minutes before you start getting data, plus the time it takes to deliver the data, which depends on the amount of data you want to read.

To access the archive, you have to establish an scp, sftp or rsync connection to the machine archive.rzg.mpg.de and log in with your MPCDF (Kerberos) user id and password.

1. If you have a Linux/Unix machine, then scp, sftp and rsync are usually installed by default. Many file managers, like the ones included with Gnome, KDE and others, also have an option to connect to an SSH server.

2. If you have a Windows machine, you can install an scp or sftp client, for example WinSCP (`http://winscp.net`) or FileZilla (`http://filezilla-project.org`).

Once you have established the connection, you will be in the folder '/ghi/r/⟨initial⟩/⟨userid⟩' (for example: /ghi/r/s/smith). You can create subfolders there at your convenience and transfer the data from your local system.

As a special case: if you are a user of one of MPCDF's HPC systems Hydra or Draco, then '/ghi/r' is available on the login nodes, so you can simply use 'cp', 'mv' etc. instead of 'scp' to transfer data from '/u' or '/ptmp' to the archive.

A very important point to keep in mind is the size of files: The archive system can only work efficiently with large files. The recommended size is between about 1 GB and about 500 GB per file. If you have many small files, please pack them first together into a large file with a suitable tool like 'tar' or 'zip'. There is a limit of 100,000 files per user, so if you try to archive many small files you will hit that limit very quickly and you will not be able to store any more data.

For more details about MPCDF's archive system, see `http://www.mpcdf.mpg.de/services/data/backup-archive/archives`.

Visualizing data from molecular simulations with VisIt

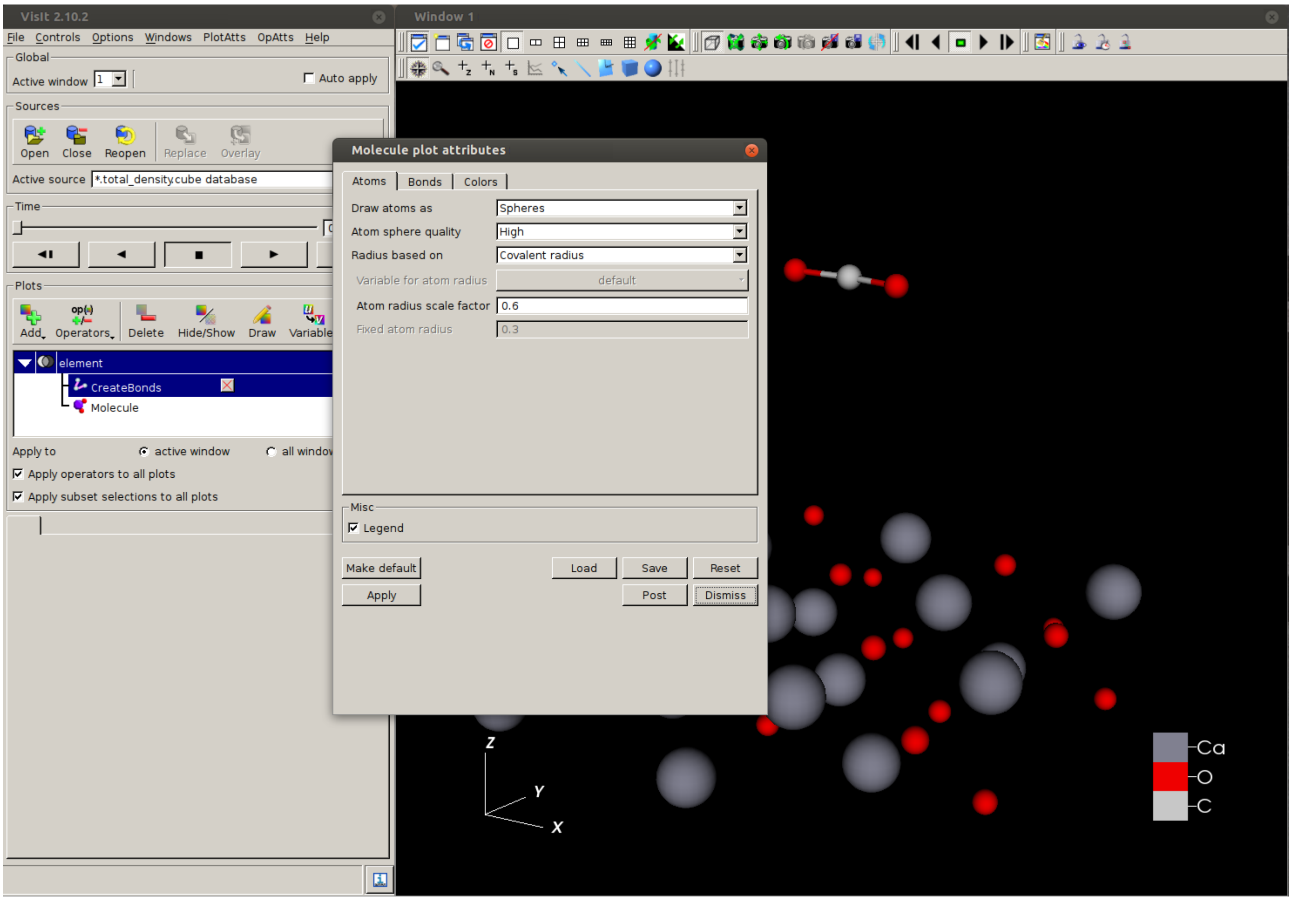
Michele Compostella, Markus Rampp

The MPCDF visualization team has recently employed the VisIt software tool for analyzing and visualizing data from molecular simulations in various scientific contexts. VisIt has been successfully used as a major workhorse for analyzing data from astrophysics, plasma physics, computational fluid dynamics and other fields of research (cf. www.mpcdf.mpg.de/services/visualization/rzgprojects).

In the context of molecular simulations VisIt has proven particularly useful for handling periodic structures, e. g. from electronic structure calculations or from 'ab-inito' molecular dynamics simulations of solids, thus complementing well-known visualization tools like VMD, PyMOL for analyzing (bio-)molecular systems.

The examples shown in the following sketch the visualization approaches we have taken for 'ab-initio' molecular dynamics simulations in the framework of density-functional theory (DFT) performed by the Fritz-Haber Institute of the Max Planck Society with the FHI-aims code, as well as time-dependent DFT (TDDFT) simulations with the OCTOPUS code performed by the Max Planck Institute for the Structure and Dynamics of Matter (Department A. Rubio).

VisIt provides internal data model support, several plotting functions and many operators to display atomic characteristics and molecular data efficiently and without difficulty. Moreover, numerous features for data exploration are also available. Using the 'Molecule' plot (see Figure 1), it is possible to render an input data file (in one of the common formats, e. g. CUBE or VTK) containing atoms and bonds with spheres and lines or cylinders.


VisIt 2.10.2
File Controls Options Windows PlotAtts OpAtts Help
Global
Sources
Open Close Reopen Replace Overlay
Time
Plots
Add Operators Delete Hide/Show Draw Variable
element
CreateBonds
Molecule
Window 1
Molecule plot attributes
Atoms Bonds Colors
Misc
Make default Load Save Reset
Apply Post Dismiss
Z
Y
X
Ca
O
C

Active window 1 ☐ Auto apply

Active source *.total_density.cube database

Apply to ● active window ○ all windows

☑ Apply operators to all plots

☑ Apply subset selections to all plots

Draw atoms as: Spheres

Atom sphere quality: High

Radius based on: Covalent radius

Variable for atom radius: default

Atom radius scale factor: 0.6

Fixed atom radius: 0.3

☑ Legend

Fig. 1: Options available for the 'Molecule' plot of the VisIt visualization tool

Several options to customize the visualization are available: atoms can be drawn with different 3D geometries (spheres, polygons, etc.), characterizing the atomic radius with numerous properties and with various render quality; bonds can be represented using cylinders or lines with different styles and shading attributes. It is also possible to modify the colouring options for the atoms and bonds, adapting the colour scheme to every demand.

Moreover, the 'Molecule' plot can be combined with additional standard plots provided by VisIt, as demonstrated in Figure 2 for a scattering process simulated with the OCTOPUS code at the MPI for the Structure and Dynamics of Matter. The image shows the volume rendering of the three-dimensional electron localization functions around a C2H4 molecule some time after the scattering of a proton off the molecule.

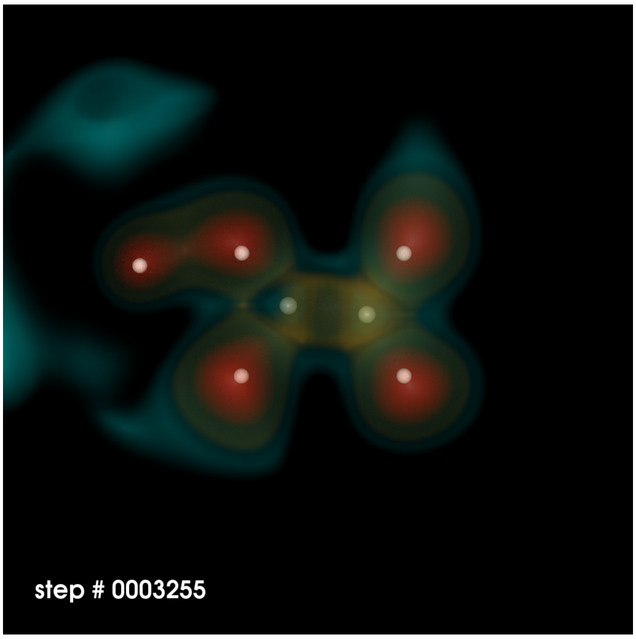


Fig. 2: Scattering of a proton off a C2H4 molecule

It is also possible to combine several plotting windows within a single canvas in order to create a mosaic of images that illustrates the scattering process from several points of view. This is the case, for example, for Figure 3, where the effect of the propagation of a laser pulse through a molecule of Azulene (C10H8) is studied.

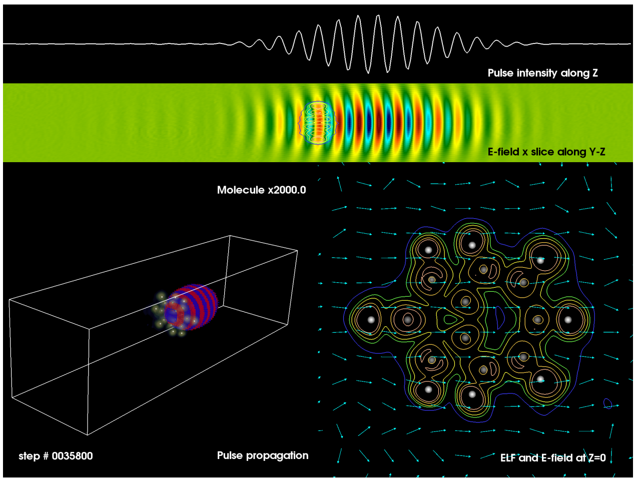


Fig. 3: Effect of a laser pulse on an Azulene molecule

Here, given the complexity of the physical event and the multi-scale nature of the problem, each plot focuses on a different aspect of the scattering process: the top panels display the intensity of the electric field of the laser pulse along the propagation axis (upper panel) and a plane perpendicular to the molecule (lower panel), while the bottom views provide a three-dimensional illustration of the problem (left) and the effect of the electric field on the electron localization functions in the plane of the molecule (right).

VisIt also provides operators specifically created for the visualization of molecular data. The 'CreateBonds' operator, for example, is used to create bonds between selected atoms or between species within a given distance. The 'Replicate' operator, on the other hand, is extremely useful allowing one to duplicate the same atomic structure several time in space.

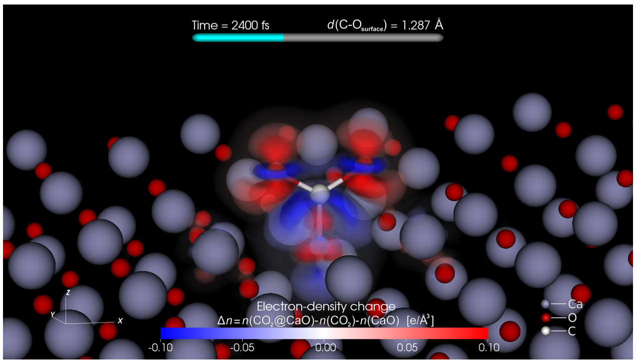

Time = 2400 fs
d(C-O$_{surface}$) = 1.287 Å
Electron-density change
$\Delta n = n(CO_2@CaO) - n(CO_2) - n(CaO)$ [e/Å^3]
-0.10
-0.05
0.00
0.05
0.10
Ca
O
C
z
y
x

Fig. 4: Interaction of a CO2 molecule with a CaO surface

Figure 4 combines these operators in order to visualize the reaction of a CO2 molecule with a CaO (001) surface from data generated at the Fritz-Haber Institute with the FHI-aims code for the NOMAD repository. In this case, the atomic structure characterizing the CaO surface is replicated around the original volume and the adsorption of a single CO2 molecule is followed in time. The change in the electron density of the system caused by the interaction of the molecule with the surface is rendered in three dimensions with different colour shades corresponding to different values. When the distance between the Carbon and the Oxygen atoms is lower than a certain threshold, cylinders representing atomic bounds are drawn, connecting the two species and emphasizing the moment when the adsorption process is completed. To view the full video follow this link.

These are only some of the capabilities offered by VisIt for the representation of data from molecular dynamics simulations. Additional information and options are described at this link.