No. 198
April 2018

Bits & Bytes

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

HPC

Lorenz Hüdepohl, Hermann Lederer, Sebastian Ohlmann, Markus Rampp, Klaus Reuter, Ingeborg Weidl

New HPC system Cobra

The Hydra supercomputer installed in 2012/2013 will be replaced in two phases. Cobra, the first phase of the follow-on system, has been installed and placed into operation. The system delivered by Intel and Atos is based on Intel Skylake processors (Xeon Gold 6148) with Intel Omni-Path Interconnect (100 Gb/s). Four islands with 636 compute nodes each have a full-fat-tree non-blocking topology, with a blocking factor of 1:8 among the islands. Two islands are equipped with 96 GB RAM per node, two islands with 192 GB RAM per node, and eight nodes provide 768 GB RAM. The batch system is Slurm as on Draco. In addition to the Cobra GPFS file systems /u and /ptmp, Hydra and Draco file systems are available, as well as the migrating file system /r. The system will be extended by a fifth island (636 nodes with 192 GB RAM each) in May 2018. The procurement of phase 2 will start in 2019.

Software organization and hierarchical environment modules

Introduction

With the advent of the new HPC system Cobra the software stack provided by the system and application groups at the MPCDF was reworked completely, including the organization of the environment modules, see the documentation below.

Previously, most of the software was compiled and installed interactively on an HPC system or Linux cluster, allowing MPCDF staff to select compilers, optimization flags, and similar factors individually for the system. As the list of supported compilers, libraries, tools, and application software has been growing tremendously in tandem with the number of systems during the last few years, an improved and more efficient work flow became necessary.

Background

Software packages are now built centrally and automatically by an instance of the Open Build Service which is a generic system for building binary packages from sources for different Linux distributions and architectures. This brings several key advantages for both users and MPCDF staff. First, packages can now be offered for a variety of combinations of compilers and libraries, e. g., MPI libraries, giving the users more safety and consistency when building their applications. Moreover, the new central approach implies a homogeneous software environment on different systems, making their usage more uniform to the user. For the software maintainer, building a package requires to write only one specification file used for all the combinations of architectures, compilers and libraries, thus reducing the effort dramatically.

Hierarchical module environment

To manage the plethora of software packages resulting from all the relevant combinations of compilers and MPI libraries, we have decided to organize the environment module system for accessing these packages in a natural hierarchical manner. Compilers (gcc, intel) are located on the uppermost level, depending libraries (e. g., MPI) on the second level, more depending libraries on a third level. This means that not all the modules are visible initially: only after loading a compiler module, will the modules depending on this become available. Similarly, loading an MPI module in addition will make the modules depending on the MPI library available.

Note, that by default after login, the Intel compiler, Intel MPI and Intel MKL module will be loaded. To start at the root of the environment modules hierarchy, issue '`module purge`'.

For example, the FFTW library compiled with the default Intel compiler and the default Intel MPI library can be loaded as follows: First, load the default Intel compiler module using the command

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

```
module load intel
```

second, the default Intel MPI module with

```
module load impi
```

and, finally, the FFTW module fitting exactly to the compiler and MPI library via

```
module load fftw-mpi
```

You may check using the command

```
module available
```

that after the first and second steps the depending environment modules become visible, in the present example impi and fftw-mpi. Moreover, note that the environment modules can be loaded via a single '`module load`' statement as long as the order given by the hierarchy is correct, e. g., '`module load intel impi fftw-mpi`'.

It is important to point out that a large fraction of the available software is not affected by the hierarchy, e. g., certain HPC applications, tools such as git or cmake, mathematical software (maple, matlab, mathematica), visualization software (visit, paraview, idl) are visible at the uppermost hierarchy. Note that a hierarchy exists for Python modules with the 'anaconda' module files on the top level.

In addition to the new Cobra HPC system, other systems are planned to be upgraded to the new software environment in the future, allowing users a more homogeneous and consistent access. Details on these upgrades will be announced separately.

Recommendations for using the Intel Compiler 2018

On the new HPC system Cobra – and in the near future also on Draco and on dedicated mid-range clusters – the latest versions of Intel compilers and libraries from the Parallel Studio XE 2018 have become the default. Specifically, for the frequently used Fortran, C, and C++ compilers (current version is 18.0) we'd like to point out a few changes with respect to earlier versions which might be relevant for MPCDF users:

1) The syntax of the frequently used option `-openmp` for enabling OpenMP directives in source code, which was marked deprecated for some time already, has finally been replaced by -qopenmp. Similarly, all -o... options were renamed -qo..., i. e. as of intel/18.0 compilers `-qopenmp` has to be used instead of `-openmp`, `-qopt-report` instead of `-opt-report`, etc.

2) While on previous generations of Intel CPUs the Intel compiler option -xhost always enabled the latest (and widest) SIMD instruction set (e. g. AVX, AVX2) this is no longer true on Skylake CPUs. When compiling with `-xhost` (or – somewhat counterintuitively – with `-xcore-avx-512`) the compiler does *not* use the 512-bit wide zmm registers by default, but takes a more conservative approach and uses the 256-bit wide ymm registers (AVX2). This can be checked by analysing the compiler optimization reports (use `-qopt-report=5` and/or the Intel advisor tool). In order to enable the zmm registers, the compiler option `-qopt-zmm-usage=high` (default is: low) has to be specified in addition. The underlying rationale is that using the zmm registers can be detrimental to the code's overall performance, since, for thermal reasons, it entails a significant downclocking of the CPU which is not always compensated by the larger SIMD width. The behaviour is code-specific and users are advised to compare the performance of their codes, using `-qopt-zmm-usage=high` and `-qopt-zmm-usage=low` (the default). As a general guideline, *'applications may benefit from more aggressive optimization with the additional option -qopt-zmm-usage=high if significant time is spent in vectorized loops, unless typical loop trip counts are small or the application is significantly memory bound. Applications with few vectorized loops or low trip counts may perform better with -qopt-zmm-usage=low'* (taken from software.intel.com).

3) As emphasized in a previous article and documented on the MPCDF web pages, Intel compilers tend to adopt increasingly more aggressive defaults for the optimization of floating-point semantics. This continues to be true for the intel/18.0 suite of compilers, and users are reminded to verify the accuracy of results by selecting more accurate floating point models, like, e. g. `-fp-model=strict` and/or comparing with other compilers, e. g. GCC.

For further details, and as a general recommendation, users are encouraged to check the release notes of the Intel C/C++ or Fortran compilers, and accompanying libraries like MKL and MPI.

Spark on Draco

Giuseppe Di Bernardo, John Alan Kennedy, Andreas Marek

Apache Spark is an open-source cluster-computing framework that supports big data and machine learning applications. To allow projects to explore the use of Spark, the MPCDF now provides the ability to run a Spark application on Draco via the Slurm batch system. This allows users to benefit from the scale of computing resources provided by Draco.

From the outside the application is a standard Slurm batch job. However, inside the job a stand-alone Spark cluster is created and the Spark application is executed on this cluster. This is enabled via a Spark module and helper script which creates the Spark cluster and is achievable in a few lines of a Slurm batch command script (see below for an example).

The GPFS (/ptmp/) file system is used for data storage, meaning that users can access their data easily and don't need to stage it to a separate system (e. g. The Hadoop Filesystem, HDFS).

This Slurm/GPFS based solution is aimed at ease of use and means that users do not need to deploy and manage a Spark/Hadoop cluster. The cluster is created on-the-fly as part of the job, where the Spark application appears as a standard batch job and is submitted and managed as such.

A simple example of a Spark wordcount application follows:

1. First a shared secret is created to ensure that a user's cluster is only accessible by the user themselves (this only needs to be done once on a Draco login node).

```
$ module load jdk
$ module load spark
$ spark-create-secure-setup
```

2. Next a Slurm batch script is created. In this example case a simple Python-based wordcount is distributed across a 3-node Spark cluster (spark-wordcount.cmd).

```
#!/bin/bash
#SBATCH -N 3
#SBATCH -t 00:30:00
#SBATCH --mem 20000
#SBATCH --ntasks-per-node 4
#SBATCH --cpus-per-task 5
#SBATCH -p express

module load jdk
module load anaconda
module load spark

spark-start
echo $MASTER
spark-submit --total-executor-cores 60 \
--executor-memory 5G \
/u/<username>/Spark/example-wordcount.py \
file:///u/<username>/Spark/Data/big-text.txt
```

The example-wordcount.py is an example application taken from the Spark distribution.

3. The batch script is submitted to the Slurm system as usual

```
$ sbatch spark-wordcount.cmd
```

4. Once the job is completed the output can be retrieved (for instance from /ptmp).

For more information see MPCDF's corresponding web page Spark on Draco.

Note: To date the evaluation of Spark has mainly focused on PySpark (the Spark Python API). However, Java and/or Scala applications are expected to run without problems and will be evaluated further in the coming months.

Acknowledgement: We would like to thank Josko Plazonic from Princeton University for his contribution during the initial phase of this project.

Major Upgrade of DataShare Service

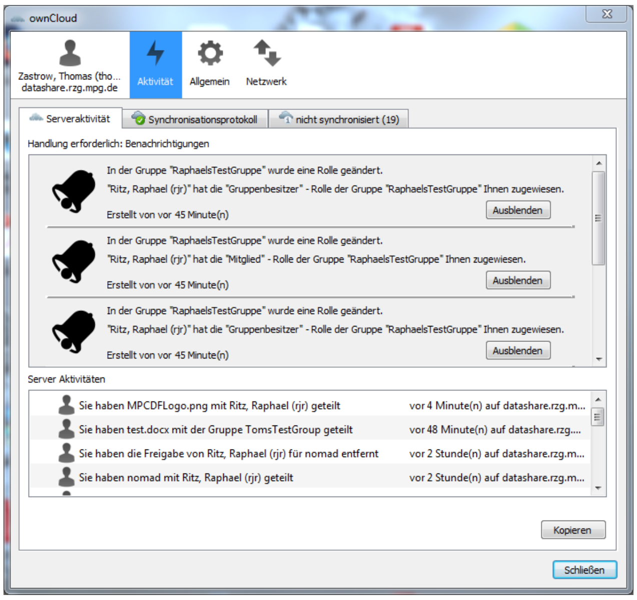
Thomas Zastrow

The latest DataShare upgrade, which took place in March, came with a large number of changes and enhancements. In this article we will highlight those which are most relevant/interesting to users. Every DataShare user is invited to test and make use of the new features and applications and report back findings, ideas, critics etc. to the MPCDF helpdesk (support@mpcdf.mpg.de). Please note that some of the new features and applications are still marked as beta versions and we can only offer limited support for them.

New clients

To be able to take advantage of the new features available on the server as well as of numerous bug fixes, the desktop sync clients should be updated to the newest version. Clients older than 2.2.4 are no longer supported and will not work. You can download the newest version of the

DataShare clients here. Any other ownCloud compatible client, including ownCloud's own clients, will also work with the DataShare service.



Handlung erforderlich: Benachrichtigungen

In der Gruppe "RaphaelsTestGruppe" wurde eine Rolle geändert.
"Ritz, Raphael (rjr)" hat die "Gruppenbesitzer" - Rolle der Gruppe "RaphaelsTestGruppe" Ihnen zugewiesen.
Erstellt von vor 45 Minute(n)

In der Gruppe "RaphaelsTestGruppe" wurde eine Rolle geändert.
"Ritz, Raphael (rjr)" hat die "Mitglied" - Rolle der Gruppe "RaphaelsTestGruppe" Ihnen zugewiesen.
Erstellt von vor 45 Minute(n)

In der Gruppe "RaphaelsTestGruppe" wurde eine Rolle geändert.
"Ritz, Raphael (rjr)" hat die "Gruppenbesitzer" - Rolle der Gruppe "RaphaelsTestGruppe" Ihnen zugewiesen.
Erstellt von vor 45 Minute(n)

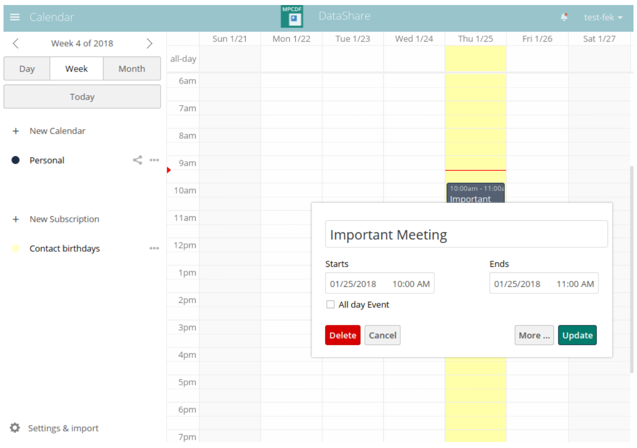
Server Aktivitäten

Sie haben MPCDFLogo.png mit Ritz, Raphael (rjr) geteilt	vor 4 Minute(n) auf datashare.rzg.m...
Sie haben test.docx mit der Gruppe TomsTestGroup geteilt	vor 48 Minute(n) auf datashare.rzg....
Sie haben die Freigabe von Ritz, Raphael (rjr) für nomad entfernt	vor 2 Stunde(n) auf datashare.rzg.m...
Sie haben nomad mit Ritz, Raphael (rjr) geteilt	vor 2 Stunde(n) auf datashare.rzg.m...

Figure 1. The new client under Windows

Calendar (beta)

The long awaited calendar app has been integrated into the DataShare service. You can now manage personal or team calendars. Calendars can be accessed via the web interface (upper left corner) or common PIM suites like Mozilla Thunderbird as well as on mobile devices. Click here for further information on the calendar app.



Important Meeting
Starts 01/25/2018 10:00 AM
Ends 01/25/2018 11:00 AM
All day Event

Figure 2. Calendar application in DataShare

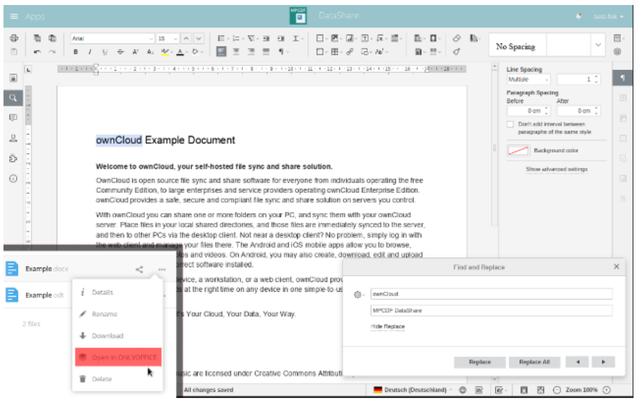
Docs (beta)

The DataShare service now integrates a collaborative online office suite, similar to Google Docs. You can edit documents, presentations and spreadsheets online directly in the browser. You can share these documents with other DataShare users and work together at the same time on these shared documents. An integrated chat function makes it easy to collaborate with co-editors.

You can create a new document under the '+' sign in the top left of the web interface or upload an existing document. Microsoft Office OOXML documents (docx, pptx, xslx) are supported natively. Other formats (odt etc.) will be converted automatically. A simple click on an office document in the web interface offers the document for downloading again: if you want to edit a document online, open it via the documents menu entry 'Open in ONLYOFFICE'.

Be careful not to edit a document at the same time online in DataShare and elsewhere, for example in Microsoft Office on your desktop. Any changes to an online edited document are only saved back to DataShare after the last online editing window has been closed.

Further information can be found here.



ownCloud Example Document

Welcome to ownCloud, your self-hosted file sync and share solution.

OwnCloud is open source file sync and share software for everyone from individuals operating the free Community Edition, to large enterprises and service providers operating ownCloud Enterprise Edition. ownCloud provides a safe, secure and compliant file sync and share solution on servers you control.

With ownCloud you can share one or more folders on your PC, and sync them with your ownCloud server. Place files in your local shared directories, and those files are immediately synced to the server, and then to other PCs via the desktop client. Not near a desktop client? No problem, simply log in with the web client and manage your files there. The Android and iOS mobile apps allow you to browse, ...s and videos. On Android, you may also create, download, edit and upload ...rect software installed.

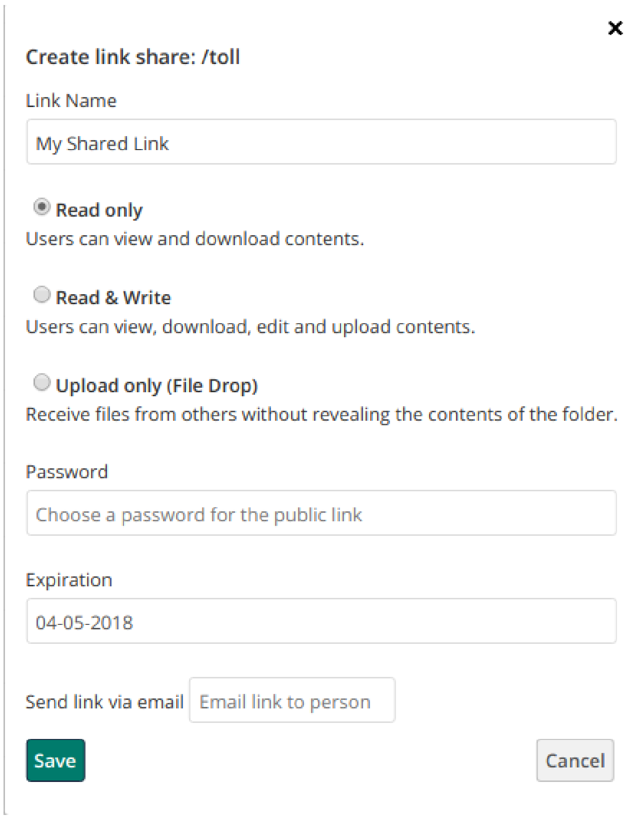
ownCloud
MPCDF DataShare

Figure 3. Online office functionality

Better Sharing

Sharing via public links has been completely overhauled. You can now generate multiple links for any DataShare object and assign different permissions to every single link. Any shared link can have an 'Expiration date' and DataShare will deactivate the share on that day.

The previous 'Files Drop' functionality has been integrated into the sharing dialog (option 'Upload only (File drop)'). The old 'File Drop' app is no longer available and 'Files Drop' links ('/apps/files_drop/xxx') are not working anymore. Further information on the sharing functionality is available here.



Create link share: /toll

Link Name: My Shared Link

● **Read only**
Users can view and download contents.

○ **Read & Write**
Users can view, download, edit and upload contents.

○ **Upload only (File Drop)**
Receive files from others without revealing the contents of the folder.

Password: Choose a password for the public link

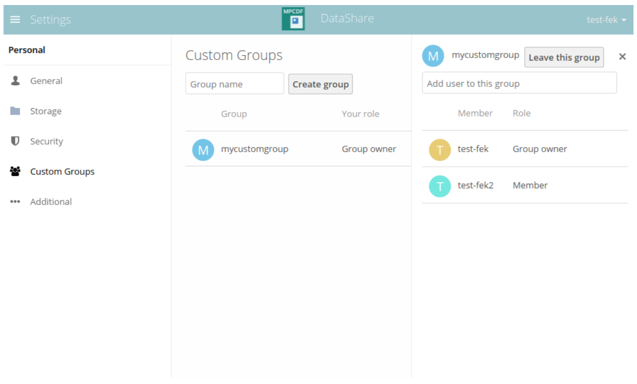
Expiration: 04-05-2018

Send link via email: Email link to person

Figure 4. New sharing dialogue

Custom groups

Another long awaited feature is the ability to create groups of users. Until now, this was only possible for admins: with the DataShare update, every user can now create groups. You can find a menu entry 'Custom groups' under the 'Settings' menu (upper right corner). Please try to assign specific group names to avoid potential collisions with already existing groups. Custom groups are only visible to the creator and group members and can be used to share files or folders. More information on custom groups can be found here.



Custom Groups

Group	Your role
mycustomgroup	Group owner

mycustomgroup

Member	Role
test-fek	Group owner
test-fek2	Member

Figure 5. Custom groups in DataShare

Misc

Improved security

User accounts will now be locked for a short period of time after multiple failed login attempts.

Notifications

Notifications from DataShare are now collected in one place and displayed in the browser as well as in the desktop clients. Watch for the little bell in the upper right corner of the web interface!

SSH security considerations

Michael Kachel

SSH Basics

SSH connections comprise three main phases which are handled by three corresponding parts of the SSH v2 protocol: transport, user authentication and connection. The main actions evolving in the first two parts can nicely be watched with:

```
$ ssh -vvv <user>@<host>
```

It comprises of:

(1) Key exchange

As soon as client and server have sent each other their identification strings, the so-called *key exchange* phase starts. During this phase client and server negotiate the crucial components of the subsequent secure communication supported by both sides. The most important components being: *key exchange method* (i. e. flavour of Diffie-Hellman), *public key algorithm*, *symmetric encryption algorithms* and *hash algorithms*.

After this is done, the Diffie-Hellman algorithm (DH) enters and does the smart work of 'creating' a common symmetric key, which is subsequently used for data encryption.

The user sees one important aspect of this phase right at the beginning of an SSH session: If a client connects to an SSH server for the first time, the client asks immediately to accept the fingerprint – i. e. the hash of the server's *public host key*. It is important to verify the validity of the fingerprint, because the authenticity of the server is bound to it (see SSH fingerprints in DNS later this article).

At the end of the key exchange phase, both client and server have created the *same symmetric key* (aka: *shared secret*) and can continue to communicate in encrypted fashion.

(2) User authentication

After the *shared secret* is created, the authentication of the client (human or machine) follows and is done over the already secured channel. The authentication can happen on basis of a keyword, an SSH key, gssapi, OTP token or other mechanisms made additionally available by PAM (pluggable authentication module).

(3) Data encryption

The subsequent actual data transfer is secured by the session-bound symmetric key, that was negotiated beforehand, and by a chosen *hash algorithm* in order to detect tamperings during transmission.

Encryption components and quality

There are four main components: (a) Key exchange algorithms, (b) key types, (c) symmetric ciphers and (d) hash algorithms.

(a) Key exchange algorithms

Basically two kinds of algorithms are used: Either *Diffie-Hellman* or *Ecliptic Curve Diffie-Hellman* (ecdh). Client and server determine which common kex algorithms are available and agree on one. To check your client, just issue:

```
$ ssh -Q kex
```

To see what a remote host offers you might issue

```
$ ssh -G <hostname> | grep kex
```

If your ssh does not support these options, you should upgrade your client. Furthermore you can and should disable weak and deprecated versions like diffie-hellman-group1-sha1 via the ssh-config files.

(b) Key types and quality

On current Linux/Unix systems four 'key suites' are operated: DSA, RSA, ECDSA and Ed25519. You can check the available suites via:

```
$ ssh -Q key
```

DSA is deprecated and should be avoided, while RSA – probably the most common SSH key type – can still be used if the key size is at least 2048bits, preferably 3072. For further information about current key suites and for more detailed information in general see the detailed web article.

(c) symmetric ciphers and (d) hash algorithms

Symmetric ciphers like aes256, chacha20 et al. do the bulk work of traffic encryption:

```
$ ssh -Q cipher
```

Hash algorithms assure data integrity; prefer the 'etm' ones:

```
$ ssh -Q mac
```

SSH fingerprint records for DNS (SSHFP)

In order to strengthen the server authenticity the MPCDF is putting more and more SSHFP (rdata_44) records into DNS. Thus DNS acts as third party in order to confirm that the server's fingerprint is in DNS and has not changed. This is especially useful for gateway systems. A dig or nslookup confirm existing entries, e. g.

```
$ nslookup -type=SSHFP gateafs.rzg.mpg.de
```

To make your client check the DNS for fingerprints, you should adapt the global or personal SSH config file:

```
Host *
   VerifyHostKeyDNS ask
```

The effect on login with the fully qualified hostname:

```
$ ssh <user>@gateafs.rzg.mpg.de
Matching host key fingerprint found in DNS.
```

In the future, with DNSSEC a short hostname may be sufficient.

Recommendations

– Try to restrict allowed kex, cipher and hash algorithms to non-deprecated ones. For details look into specific advisories covering different operating systems and versions, e. g. Stribika_SSH_Wiki or Infosec/-Mozilla_recommendations.

– Encourage your site admins to configure SSH server and client configurations to disable deprecated and insecure methods.

– The higher the security needs of a system, the more strict your clients should operate.

– Reinforce authentication reliability for sensitive hosts by introducing a second authentication factor.

– Hints for the user authentication:

- Ensure strong passwords (preferably enforced by your identity management system);
- For ssh keys: ensure strong keys; delete dsa-keys, check the others, e. g.:

```
$ ssh-keygen -l -f ~/.ssh/id_rsa
```

  256 bits for the new ed25519 key is in terms of cryptographical strength roughly equivalent to the 2048 bit of the RSA key. If you encounter RSA keys with 1024 or even 768 bit, please dispose of them.

- Always put a strong password on your private key. Do you use one? Test it:

```
ssh-keygen -y <privKey>
```

If you're not prompted to provide your password, then you have none :-(
– Finally a tip unrelated to security: thanks to Intel's and AMD's integration of standard AES instructions into microprocessor codes you can enhance the speed of large scp transfers significantly by forcing a fast cipher, e. g.

```
$ scp -c aes128-ctr huge_src_file dst
```

Announcements

Klaus Desinger, Renate Dohmen and Markus Rampp

Mail scanning outsourced to DFN

For the last 15 years we had been scanning incoming E-mail for Spam and malware. This task was now outsourced to our Internet provider 'Deutsches Forschungsnetz' (DFN) who service almost all German universities and public research facilities. DFN's scanning is also based on the spamassassin software, so you probably didn't notice any difference. As before suspicious E-mails are tagged with an additional header line 'X-Spam-Level: ' and a number of asterisks, denoting the 'spammyness' of the mail. We recommend to configure your mail reader to direct mails with more than 6 asterisks to a spam folder. Mails with more than ten spam asterisks or those definitely containing malware are rejected by DFN.

Advanced HPC optimization workshop 2018, first announcement

In autumn 2018, the MPCDF, in collaboration with Intel, will host an advanced training workshop for HPC application developers in the Max Planck Society. The workshop will take place at the MPCDF in Garching during two days in the October/November time frame (final dates to be announced). In the course of the first day, HPC application and software experts of the MPCDF and Intel will give presentations about tools and strategies for performance profiling and optimization. The second day is dedicated to a *code camp* which provides the opportunity for application developers to tackle specific performance issues in their codes together with HPC experts from the MPCDF and from Intel.

Capacity is limited to approx. 30 participants, and about 5 proposals for code optimization projects can be accepted. Registration for the workshop will be opened in the course of the next few weeks (to be announced via the hpc-users mailing list of the MPCDF). Application developers who are interested in actively participating in the code camp will be asked to provide a short description of the particular problem to be addressed during the workshop and to prepare representative code and setups in advance.

Expressions of interest and indication of topics which are of particular interest are welcome and should be directed at hpcworkshop-18@mpcdf.mpg.de.