

# C2115

# Practical introduction to supercomputing

Lesson 10

Petr Kulhánek

[kulhanek@chemi.muni.cz](mailto:kulhanek@chemi.muni.cz)

National Centre for Biomolecular Research, Faculty of Science  
Masaryk University, Kamenice 5, CZ-62500 Brno

# Content

- **Data storages of MetaCentrum**
  - **Types of disk arrays and their use**
  - **Access rights (revision)**
  - **Access to data repositories**
    - **MetaCentrum**
    - **Local clusters (WOLF, ...)**
  - **Good practice**
    - **access rights**
    - **data visualization**
  - **Personal computers**
    - **Configurations, packages**

# Types and methods of use

## Types of repositories and their use:

- **local data storage (HDD, SSD)**
  - temporary job data
- **(remote) data storage (disk array)**
  - live data of jobs or solved projects
- **hierarchical data storage**
  - completed projects and backups

## Data throughput (qualitative):

**SSD** >> **HDD** > **disk arrays** >> **hierarchical data stores**

- Job data that is read sequentially can be read directly from the disk array.
- Job data that is created sequentially can be stored directly on the disk array.
- Other data (**vast majority of cases**) **must be copied to the local data storage before running the job**. Pay special attention to temporary files that programs access with random access. This data must again be created on local data repositories.
- We never use a hierarchical data storage as a direct data source for jobs. We first copy the data to disk data storage.

# Access rights

---

# Access rights

Access rights determine what operations user can perform on files or directories in the file system.

## Access rights:

<b>r</b>	read the file	list contents of directory
<b>w</b>	change file	change contents of directory (create or delete file or directory)
<b>x</b>	run the file	enter the directory

Each file or directory has a designated owner and user group. Access rights are listed separately for **file owner (u)**, **user group (g)** and **other users (o)**.

```
$ ls -l
```

```
  u  g  o  
drwxrwxr-x  3 kulhanek lcc  4096 2008-10-13 09:57 bin/  
drwx-----  2 kulhanek lcc  4096 2008-10-13 09:58 Desktop/  
-rw-rw-r--  1 kulhanek lcc  5858 2008-10-17 11:58 distance.cpp
```

↑ access rights

↑ owner (user) and user group (group)

↑ size (B)

↑ time of last change

↑ name of file or directory/

↑ type: file (-) or directory (d)

# Rights evaluation procedure

```
$ ls -l
  u  g  o
drwxrwxr-x  3 kulhanek lcc  4096 2008-10-13 09:57 bin/
drwx-----  2 kulhanek lcc  4096 2008-10-13 09:58 Desktop/
-rw-rw-r--  1 kulhanek lcc  5858 2008-10-17 11:58 distance.cpp
```

If the user is accessing a file or directory:

- 1) identical to the owner of the file, access is governed by the access rights of the owner
- 2) is a member of the group, access is governed by the access rights for the group
- 3) belongs to other users, access is governed by access rights for other users

order of evaluation of access rights to given entity (directory or file)

If a user accesses a file or directory specified by a path, the above rule is applied sequentially from the highest directory:

**/home/user/ test.txt**

order of evaluation of access rights

# Default setting and its change

## When creating a file or directory:

- owned by the user who creates the file or directory
- user access group is set to the primary group to which the file owner belongs at the time the file or directory is created or to the access group of the parent directory in the case of an active Set-Group-ID directive
- default access rights are derived from the mask set by the command **umask**

Some commands or applications may have a different default policy (e.g., ssh-keygen and access rights for the private key).

## Change:

- of file owner can only be made by superuser (command **chown**)
- of user access groups can be made by the file owner to the groups to which he/she belongs, or by superuser to any group (command **chgrp**)
- of access rights can be made by the owner of the file or superuser (command **chmod**)
- of mask can be done by the user with the command **umask**, for permanent effect it is necessary to insert the command into `~/.bashrc` file

# User and group identity

**User identity** and his **groups** can be found by the command **id**:

```
[kulhanek@wolf01 ~]$ id
uid=18773(kulhanek) gid=2001(lcc) groups=2001(lcc),2027(kulhanek),2030(compchem)
```

user login name and its numerical representation

primary group users to which the user belongs and its numerical representation

user groups to which the user is assigned and their numeric representation

Assigning a user to primary and other groups can be **changed only by superuser**.

**Users assigned to groups** can be listed with the command **getent**:

```
[kulhanek@wolf ~]$ getent group compchem
compchem:*:2030:408530z,409282aa,acechova,aderim12,ailar,akprmf, .... (shortened)
```

group name

numerical representation

a comma-separated list of users (logins) in the group

Command **getent** can also be used for other queries, e.g., to list all system users (**getent passwd**).



# Change of access rights

File and directory access rights can be changed by the file owner or superuser by command **chmod**.

```
$ chmod permissions file1 [file2 ...]
```

```
  u   g   o  
  ---  
drwxrwxr-x
```

## Access rights:

<b>r</b>	read the file	list the contents of the directory
<b>w</b>	modify the file	change the contents of the directory
<b>x</b>	run the file	enter the directory
<b>X</b>	sets the right to run a file that already has this right in another group rules and always for directory (usable for recursive change of rights)	

## Groups of rights :

<b>u</b>	owner (user)
<b>g</b>	user group (group)
<b>o</b>	others (other)
<b>a</b>	all (all), the law applies to u, g, o

## Example:

```
$ chmod u+x,g-w file
```

Adds (+) execution rights to the owner and removes the (-) writing rights for the group

# Change of access rights

Access rights in octal (octal) notation:

u g o  
drwxrwxr-x

0xyz

zero (prefix of octal notation)

total sum of octal values for individual rights in a given group

Right	Octal value
r	4
w	2
x	1

Examples:

rwxrwxr-x      0775  
r---w---x      0421  
rwxr-x---      0750

# Group change

User group for files and directories can be changed by the owner or superuser by command **chgrp**. The owner can only use the groups to which he/she belongs (can be found out with the command **id**).

```
$ chgrp group_name file1 [file2 ...]
```

```
[kulhanek@wolf01 ~]$ id
```

```
uid=18773(kulhanek) gid=2001(lcc) groups=2001(lcc),2027(kulhanek),2030(compchem)
```

```
[kulhanek@wolf01 ~]$ ls -ld Documents/
```

```
drwxr-xr-x 9 kulhanek lcc 4096 Feb 16 2012 Documents/
```

```
[kulhanek@wolf01 ~]$ chgrp compchem Documents/
```

 group change

```
[kulhanek@wolf01 ~]$ ls -ld Documents/
```

```
drwxr-xr-x 9 kulhanek compchem 4096 Feb 16 2012 Documents/
```

# Setting mask

Default access rights are set using the mask set by the command **umask**. Current mask settings can be found by the command `umask` without any argument. (Documentation: `man bash`, SHELL BUILTIN COMMANDS)

## Default access rights for:

files are 0666

directories are 0777

**Mask** indicates access rights that are **removed from default rights** before they are used to set access rights to the file or directory being created.

e.g., mask 0027 leads to the following access rights:

for file 0640

for directory 0750

The mask can be changed with the command **umask** pasted to the end of `~/.bashrc` file or by a setting executed by a command **ams-config** (Infinity environment).

# Overview of commands

## *File system (access rights):*

<b>id</b>	lists the user groups, displays the primary group
<b>getent</b>	lists information about users, user groups, and other information
<b>umask</b>	default access rights for newly created files or directories
<b>chmod</b>	changes access rights to a file or directory
<b>chgrp</b>	changes the user access group for files or directories
<b>chown</b>	changes the owner of the file or directory

# Access to data storages

---

# Access from MetaCentrum

These are volumes mounted with the NFS protocol with the expected behavior of the access policy (standard POSIX rights) because MetaCentrum environment has a uniform set of user accounts on all nodes.

## Access point:

**`/storage/<NAME>/home/$USER`**

New files and directories are created with the owner and group derived from the login name and the primary effective user group on that node. The default access rights are then set according to the mask set by the command `umask` on a given node.

If the data storage is inaccessible or non-functional and no failure has been reported, contact user support of MetaCentrum via [meta@cesnet.cz](mailto:meta@cesnet.cz)

# Access from MetaCentrum

```
[kulhanek@onyx ~]$ id
uid=18773(kulhanek) gid=10056(kulhanek) groups=10000(meta),221(ncbr),
10056(kulhanek),10086(strcmu),20138(storage)
[kulhanek@onyx ~]$ pwd
/storage/brno2/home/kulhanek
[kulhanek@onyx ~]$ ls -l
total 31392
drwxr-xr-x  5 kulhanek meta          4096 Jul 10  2012 00.Scripts
drwxr-xr-x 16 kulhanek meta          4096 Feb 17  2015 03.projects
drwxr-xr-x  3 kulhanek meta           23 Jun  1  2013 04
-rw-r----- 1 kulhanek meta        10191 Oct 14  2014 1UZV_3H2O_cutoff.xyz
-rw-r----- 1 kulhanek kulhanek     183 Jun  8  2015 add_users_to_infloc
....

[kulhanek@onyx ~]$ mkdir test
[kulhanek@onyx ~]$ umask
0027
[kulhanek@onyx ~]$ ls -ld test
drwxr-x--- 2 kulhanek kulhanek 6 Jan 31 16:07 test
```



# Local clusters (WOLF, ...)

Access is possible with a valid Kerberos ticket from realm META. Disk arrays are mounted using the NFS protocol with a special access rights mapping. The rights are verified on the NFS server side from which the volume is mounted.

**Access point (identical as in MetaCentrum):**

**`/storage/<NAME>/home/$USER`**

New files and directories are created with the owner and group derived from the login name and the user's primary group from the server from which the volume is mounted (derived from Kerberos principal). However, the default access rights are set according to the mask set by the command `umask` on the local machine.

If the data storage connection on our local clusters is inoperative and no outage has been reported to MetaCentrum, contact the LCC group user support via [support@lcc.ncbr.muni.cz](mailto:support@lcc.ncbr.muni.cz)


# Local clusters (WOLF, ...), cont.

```
[kulhanek@wolf01 ~]$ id
uid=9703(kulhanek) gid=2027(kulhanek) groups=2001(lcc),
2027(kulhanek),2029(rmarek),2030(compchem)
```

```
[kulhanek@wolf01 ~]$ pst brno2
```

```
[kulhanek@wolf01 kulhanek]$ pwd
/storage/brno2/home/kulhanek
```

```
[kulhanek@wolf01 kulhanek]$ ls -ld test
drwxr-x--- 2 kulhanek@META kulhanek@META 6 Jan 31 16:07 test
```



mapping to special names of users and groups, which are created dynamically, these are names from MetaCentrum space terminated by @META (provided by metanfs4d service)

```
[kulhanek@wolf01 kulhanek]$ chgrp ncbr@META test
```

```
[kulhanek@wolf01 kulhanek]$ ls -ld test
drwxr-x--- 2 kulhanek@META ncbr@META 6 Jan 31 16:07 test
```

# Local clusters (WOLF, ...), cont.

```
[kulhanek@wolf01 ~]$ id
uid=9703(kulhanek) gid=2027(kulhanek) groups=2001(lcc),
2027(kulhanek),2029(rmarek),2030(compchem)
[kulhanek@wolf01 kulhanek]$ umask 0077
[kulhanek@wolf01 kulhanek]$ mkdir test2
[kulhanek@wolf01 kulhanek]$ ls -ld test2
drwx----- 2 kulhanek@META meta@META 6 Jan 31 2016 test2
```



set according to wolf01



set according to the NFS server

# Command pst

Command **pst** comes from the module **meta-storages** of Infinity environment. The command is available on all our local clusters and in MetaCentrum if the user has activated Infinity environment. The command supports TAB autocomplete. If you have another login name in MetaCentrum, **pst** command will not work and you will need to enter the storage path manually.

```
[kulhanek@wolf ~]$ pst
Usage: pst <domain>
  Data storages:
    budejovice1 brno1-cerit brno2 brno3-cerit brno6 brno7-cerit
    brno8 brno9-ceitec plzen1 prahal
  HSM storages:
    brno5-archive brno10-ceitec-hsm jihlava2-archive
    plzen2-archive projects

kulhanek@wolf ~]$ pst brno3-cerit
bash: cd: /storage/brno3-cerit/home/kulhanek: Permission denied

[kulhanek@wolf ~]$ kinit
Password for kulhanek@META:

[kulhanek@wolf ~]$ pst brno3-cerit

[kulhanek@wolf kulhanek]$ pwd
/storage/brno3-cerit/home/kulhanek
```

# Exercise 1

1. How many standard disk storages are in MetaCentrum?
2. How many hierarchical data storages are in MetaCentrum?
3. Log in to any front end of the MetaCentrum, and then create a file test.txt which will contain three lines of text. What data storage did you create the file on?
4. Create kerberos ticket on your workstation and mount the data store using the command pst.
5. Verify that the file contains the text that you created.
6. Provide the name of the NFS server that serves the data storage (use df command).

# Good practice (access rights)

If you require **restrictive access to data** (i.e., access to data provided only for you or your co-workers), then follow these rules:

## User soloist:

set on all machines: **umask 0077**

(only the owner can work with data - read, create)

the primary group can be arbitrary

## User working in a group:

create a new user group in MetaCentrum

(contact MetaCentrum support via [meta@cesnet.cz](mailto:meta@cesnet.cz), state the reason for setting up the group, the name proposed, and the list of users who should belong to it, at the same time request the change of the primary group to the created group, for all its members)

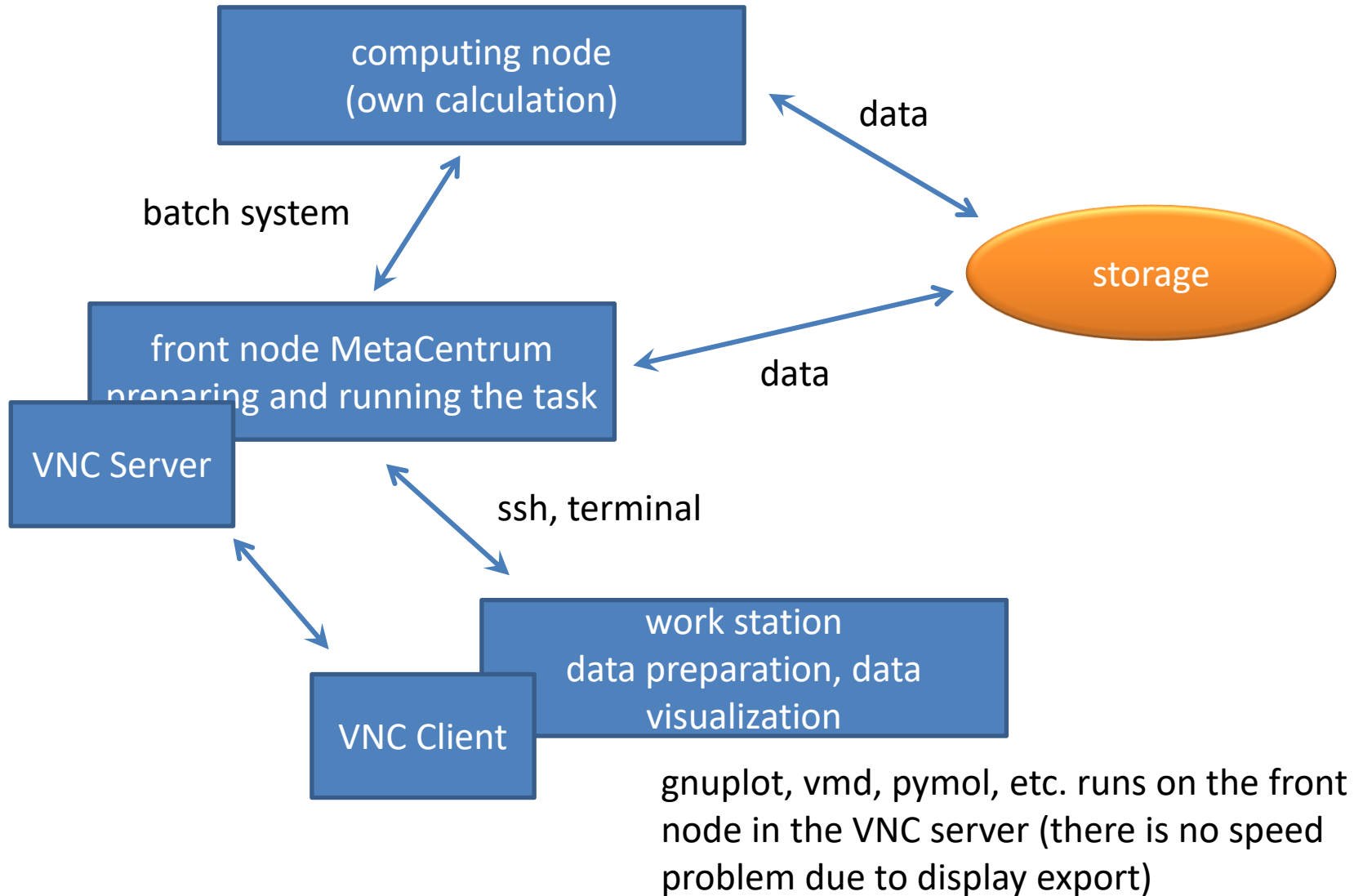
group members must have this group activated as primary group in MetaCentrum

primary group on other machines can be arbitrary

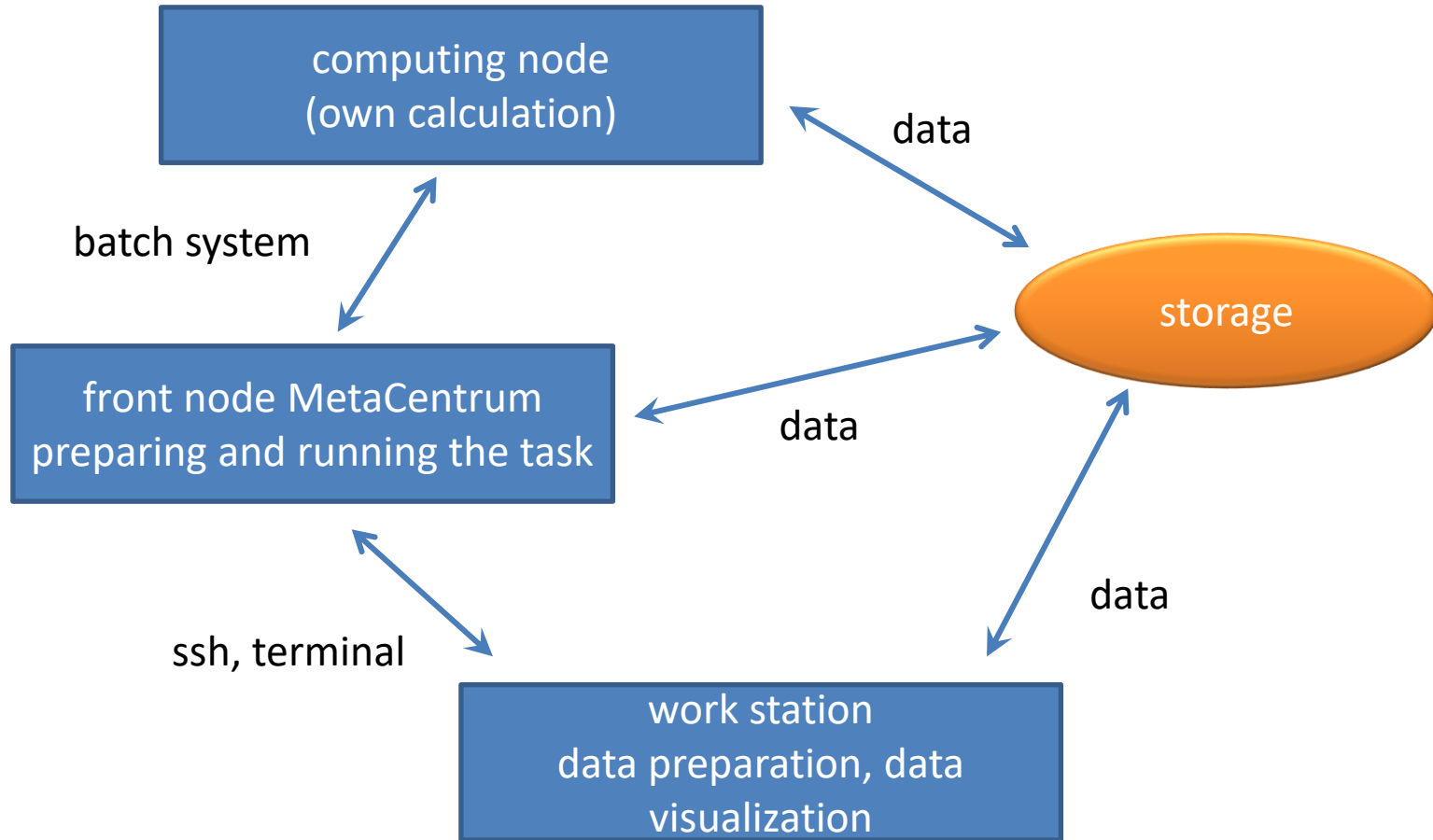
set on all machines: **umask 0027**

(only the owner can work with data - read, create, the group can only read)

# Good practice (data visualization)



# Good practice (data visualization)



gnuplot, vmd, pymol, etc. runs locally on remote data (however, there is no problem with the display speed due to the export of the display)



# Installation via packages

---

Tested for Ubuntu 18.04 LTS

# Workflow

- 1) Activation **public repository of NCBR packages**. The procedure is given at <https://einfra.ncbr.muni.cz> in the section "User Support >> Ubuntu" and the CEITEC MU/NCBR PUBLIC repository, select Ubuntu 18.04 LTS. Repository is **activated only once**.

- 2) Installation of package for connecting storage devices of MetaCentrum (choose default setting):

```
$ sudo apt-get install ncbr-metanfs4-krb5i-metastorages
```

- 3) Package installation remctl-client

```
$ sudo apt-get install ncbr-metanfs4-keytab
```

- 4) Creating a private ticket **krb5.keytab** for Kerberos:

```
$ gen-metanfs4-keytab
```

```
# Contacting KDC server for a keytab ...
```

```
# Moving the keytab to /etc/krb5.keytab ...
```

```
OK
```

- 5) **Restart computer.**

**See MetaCentrum documentation for details:**

Connect data storages to your own workstation via NFSv4

# Exercise 2

1. Install connection of MetaCentrum data storages in your installation of Ubuntu server.
2. Mount the data storage where you have placed the test.txt file. Verify the contents of the file.