

Using the Linux Pool Computers of the TUHH Computer Center for OMNeT++

As the OMNeT++ simulations might take several hours, it is worthwhile to execute these on powerful machines which you do not use for your everyday tasks. The Linux pool computers in the TUHH Computer Center are suitable for this task. These are hardly used on weekends or during the night. In the following, we will demonstrate a possibility to run your simulations on these computers remotely via secure shell (SSH), i.e. you do not have to be in the computer pool but can work from home. However, when you are outside the university network, you have to either use VPN or first login to `ssh.rz.tu-harburg.de` before you can login to the Linux pool computers.

You can find a list of the computers including their hostnames and their respective configuration on the website of the computer center¹. Log in to a computer of your choice via SSH (PuTTY² is a suitable Windows tool for that) by using the address **<hostname>.rz.tu-harburg.de** (the hostname is something like IXXpYY where X and Y can be found on the computer center website).

- By default the pool computers use the *tcsh* shell. However, this **complete tutorial requires** the usage of the *bash* shell. Start it via the command `bash`.
→ **This is needed every time you log in again!**
- Step 1 and 2 have to be only done once. Step 3 has to be done when you change your project locally on your computer.

1. Installation of OMNeT++

- Add the following three lines to the end of your `~/.bashrc` file to change the compiler used and to add the **required tools to run the simulations via the command line** later on to the path. This has only to be done once.
 - `echo 'export PATH=/nfs/rzpool/gcc/gcc-4.8.2/bin:$HOME/omnetpp-5.2.1/bin:$PATH' >> ~/.bashrc`
 - `echo 'export LD_LIBRARY_PATH=/nfs/rzpool/gcc/gcc-4.8.2/lib64:$LD_LIBRARY_PATH' >> ~/.bashrc`
 - `echo 'export C_INCLUDE_PATH=/nfs/rzpool/gcc/gcc-4.8.2/include:$C_INCLUDE_PATH' >> ~/.bashrc`
- Afterwards, leave `bash` by issuing `exit` and start it again with `bash`, so that the environment variables are set correctly.

¹ <https://www.tuhh.de/rzt/studium/pools/linux-pools/hardware.html>

² <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

- Download and extract the OMNeT++ source code to your home directory. We provide an archive where necessary changes are already made to allow for successful compilation of OMNeT++ on the pool computers.
 - `cd ~`
 - `wget --content-disposition https://cloud.tuhh.de/index.php/s/yOj7GTfsJfDjh29/download`
 - `tar xvzf omnetpp-5.2.1.tar.gz`
 - `cd omnetpp-5.2.1`
- Prepare and compile OMNeT++. (Avoid building OMNeT++ with multiple cores in parallel, this might lead to problems.) Watch out that the system is case sensitive! Meaning, typing `MODE` or `mode` makes a difference.
 - `./configure`
 - `make MODE=release`

2. Compilation of the INET-Framework

- Download and extract the INET-Framework to your workspace. You should use INET version 3.4. We provide an archive where necessary changes are already made to allow for compilation on the pool computers.
 - `mkdir ~/workspace`
 - `cd ~/workspace`
 - `wget --content-disposition https://cloud.tuhh.de/index.php/s/x3GcXcANxFNHqMT/download`
 - `tar xvzf inet-18a0b55.tar.gz`
 - `cd inet`
- Compile the INET-Framework. Use the release mode for fastest simulations. (Avoid building INET with multiple cores in parallel, this might lead to problems.)
 - `make makefiles`
 - `make MODE=release`

3. Prepare your OMNeT++ Project

- In your OMNeT++ IDE, select your project and go to *Project* → *Properties* → *OMNeT++* → *Makemake*. Select your project folder and click *Export* (to be done only one time for your project).
- Copy the complete project folder to your workspace folder on the machine you are connected to. WinSCP³ is a suitable Windows tool for that. Transferring your project over GitLab is another option (see below).
- Prepare and compile your project:
 - `cd ~/workspace/<projectfolder>`
 - `make CONFIGNAME=gcc-release -f makemakefiles`
 - `make MODE=release`
- Afterwards your executable will be found in the subfolder *out/gcc-release/*. Additionally, you will see the executable created with your project name in the root of your project folder. Use the this for your simulations. It is referred to as *<executable>* in the following.

4. Running your Simulation

- To prevent your simulation from being terminated when logging out, use *screen*⁴.
 - `screen` (to create a new screen and attach to it)
 - `CTRL + A + D` (to detach from a screen you are currently attached to)
 - `screen -ls` (to list all screens currently active)
 - `screen -r <identifier>` (to reattach to a screen with a certain identifier, which you get from `screen -ls`)
- Do not forget to use *bash* as your shell in *screen*, as *screen* starts the standard shell again.
- You should use the local file system instead of your home directory to save your result files. The home file might not have enough storage space to save all the files created during the simulation, which can be several gigabytes. Get the path of your personal temporary directory and add it to your *omnetpp.ini* file. Do not forget to save the result files to your local machine, as the files in the temporary folder are deleted after seven days.
 - `echo "result-dir=$TMP" >> ./omnetpp.ini`
 - `cd $TMP` (to go to your temporary directory)

³ <https://winscp.net>

⁴ Try "`man screen`" or for example <https://www.rackaid.com/blog/linux-screen-tutorial-and-how-to/>.

- Display configurations and number of runs. On Linux executables usually do not have any file name extension (no .exe or similar).
 - `./<executable> -a`
- Run the simulation with a certain configuration and certain run numbers, e.g. “0..5”. Leaving out the “-r <runs>” parameter executes all runs present. Replace `<inet_src>` with the path to the src folder within your inet project. Replace `<inifile>` with the path to your omnetpp.ini Replace `<configname>` with the configuration from your ini-file you want to run (e.g. General).
 - `./<executable> -u Cmdenv -c <configname> -r <runs> -n .:<inet_src> <inifile>`
- To simulate multiple runs in parallel, execute the following command. Replace `<N>` (no white space) by the number of cores present in the machine you are connected to. Replace `<inet_src>` with the path to the src folder within your inet project. Replace `<inifile>` with the path to your omnetpp.ini
 - `opp_runall -j<N> ./<executable> -u Cmdenv -c <configname> -r <runs> -n .:<inet_src> <inifile>`
- Example commands would be
 - `./ProjectABC -a`
 - `./ProjectABC -u Cmdenv -c General -r 0..5 -n ../../inet/src/ omnetpp.ini`
 - `opp_runall -j3 ./ProjectABC -u Cmdenv -c General -r 0..5 -n ../../inet/src/ omnetpp.ini`

Using Git on the pool computers

- The easiest is to create a new ssh-key on the pool computers and use that:
 - `ssh-keygen -t rsa -b 4096 -C "rz-pool"`
- Display your public key:
 - `cat ~/.ssh/id_rsa.pub`
- Copy your public key to the Gitlab (all what is shown as output of the previous command): <https://collaborating.tuhh.de/profile/keys>
- Clone your project:
 - `cd ~/workspace`
 - `git clone git@collaborating.tuhh.de:<username>/<projectname>.git`

Frequent problems

- If you have questions, please make first sure, that you **followed every step** in the given order. Unfortunately most problems arise, if people didn't **read our guide properly**.
- Make sure, that you took our prepared omnetpp and inet files (from the TUHH-Cloud as noted above). We made adaption to the config files for them to work on the pool computers.
- Make sure the **folder structure is identical** on your laptop, where you exported your project and on the pool computers so that **relative paths match**. In case you change folder names or move projects after compilation, it might be necessary to recompile.
- Most frequent problems:
 - Are you in the bash shell? Find that out by typing `echo $0`
 - Is the OMNeT++ directory in your path? Find that out by typing `echo $PATH`
 - Different folder structure → see above. We recommend that your project and inet are in the same subdirectory. Something like:
`workspace/inet` **and** `workspace/your-project`