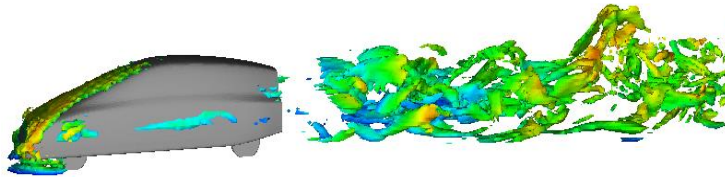


Programming Techniques for Supercomputers



HPC Services @ RRZE
University Erlangen-Nürnberg
Sommersemester 2020





- **Course website**
- **Login to RRZE's Emmy cluster**
- **Basic environment**
- **Some guidelines**
- **First Assignment**



- All slides, exercises, and miscellaneous material can be found on the course pages:

<http://tiny.cc/PTfS>

- Please use the **discussion forum** there if you have a question that might be of interest to your fellow students
- Please **enroll in the course** so you get all e-mail announcements



- **If you haven't done so already, send**
 - Your name
 - Your student ID number (Matrikelnummer)

to georg.hager@fau.de in order to get an RRZE account set up
- **RRZE accounts are different from your IdM accounts**
 - The account will show up under “Services” in your IdM portal
 - Set a password and **wait for a couple of hours** until the account becomes active



- Login to RRZE cluster front-end machines
 - `ssh ptfsXXXh@emmy.rrze.uni-erlangen.de`
 - `-X` enables X forwarding
 - Front-end nodes: `emmy1`, `emmy2`

```
$ ssh -X ptfsXXXh@emmy.rrze.uni-erlangen.de
Last login: Tue Apr 19 16:36:04 2016 from fau-eduroam-1-34.wlan.rrze.net
=====
Welcome at "Emmy", RRZE's IvyBridge/NEC cluster!
[...BLURB...]
=====
ptfsXXXh@emmy2:~$
```

- Works only from inside FAU (or w/ VPN)
<https://www.anleitungen.rrze.fau.de/internet-zugang/vpn/>
(German only)



Best practice: Use RRZE “dialog server”

- Login to `csnpc.rrze.uni-erlangen.de`
 - All necessary tools installed
 - Access to all hpc systems and most filesystems
 - Linux Desktop from Windows: Nomachine
<https://www.anleitungen.rrze.fau.de/hpc/dialogserver/>
 - Available from outside RRZE/FAU
 - Compile on Emmy frontends (see later)

```
$ ssh -X ptfsXXXh@csnpc.rrze.uni-erlangen.de
[...BLURB...]
ptfsXXXh@csnpc:~$ ssh -X ptfsXXXh@emmy.rrze.uni-erlangen.de
[...BLURB...]
ptfsXXXh@emmy2:~$
```



- **CIP Account registration:**
<https://account.cip.cs.fau.de/>
- **Using a CIP Pool from remote:**
<https://remote.cip.cs.fau.de/>
- **Start any kind of shell, e.g. konsole**
(All workstations in the tutorial room should run linux)
- **Login to RRZE cluster front-end machines**
 - `ssh ptfsXXXh@emmy.rrze.uni-erlangen.de`
 - `-x` enables X forwarding
 - Front-end nodes: `emmy1`, `emmy2`





- **Make compiler available for use:**
 - `module load intel64`
 - `icc` → Intel C compiler
 - `icpc` → Intel C++ compiler
 - `ifort` → Intel Fortran compiler

- **Recommended Intel compiler options**
 - `-O3` high optimization level
 - `-xHost` optimize for CPU the compiler is running on
 - `-fno-alias` assume no overlap between any arrays or elements
 - Other options (`-c`, `-o`, `-S`, `-L`, `-I`, `-l`,...) are the same as for GCC

- **Additional software**
 - `module available` → overview over all available software
 - `module list` → currently loaded modules
 - `module unload <modulename>` → unload module



- **Issue an interactive job**
 - `qsub -l nodes=1:ppn=40,walltime=01:00:00 -I`
 - Requests all 40 virtual cores of one node for one hour
 - `-I` requests an interactive login shell on the compute node
 - Always request a full node (ppn=40)!
- **The node is yours alone for the allocated time**
 - Compiling is possible on the cluster nodes, modules work

```
ptfsXXXh@emmy2:~$ qsub -l nodes=1:ppn=40,walltime=01:00:00 -I
[... BLURB ...]
qsub: waiting for job 585626.eadm to start
qsub: job 585626.eadm ready

Starting prologue... Tue Apr 19 16:29:11 CEST 2020
Master node: e0104
Kill all process from other users
Adjust oom killer config
Clearing buffers and caches on the nodes.
Power management available, enabling ondemand governor
End of prologue: Tue Apr 19 16:29:18 CEST 2020
ptfsXXXh@e0104:~$
```



- Modern CPUs can adjust their own clock speed depending on some conditions (“**Turbo Mode**” etc.)
 - # of active cores
 - Temperature
 - ???
- **Accurate and reproducible benchmarking** requires a **constant clock speed**
- **Emmy** allows you to set the clock speed at job submission

```
ptfsXXXh@emmy2:~$ qsub -l nodes=1:ppn=40:f2.2,walltime=01:00:00 -I
[...]

Starting prologue... Tue Apr 19 16:29:11 CEST 2020
Master node: e0104
Kill all process from other users
Adjust oom killer config
Clearing buffers and caches on the nodes.
Power management available, setting specified frequency of 2200000 using
performance governor
End of prologue: Tue Apr 19 16:29:18 CEST 2020
ptfsXXXh@e0104:~$
```



- Interactive jobs are inadequate for “production runs”
 - Parameter studies
 - Long runs
- Substitute `-I` by the name of a **shell script**:

```
$ qsub -l nodes=1:ppn=40:f2.2,walltime=01:00:00 script.sh
```

```
#!/bin/bash
#
# the script runs in $HOME, so
# change to correct directory (i.e. the directory
# from which the job was submitted)
cd $PBS_O_WORKDIR
# start executable
./a.out
```



- **Remember: Performance** $P = \frac{W}{T_{wall}}$

W = work

T_{wall} = “wallclock time”, elapsed time

- **Accurate time measurement is important!**
 - Very short periods are difficult to measure
 - Measure **at least for 100 ms**

- **Example code in**
~unrz55/GettingStarted

```
#include "timing.h"
int main(int argc, char *argv[])
{
    double wcTime, wcTimeStart, wcTimeEnd;
    wcTimeStart = getTimeStamp();

    /* PUT YOUR CODE HERE */

    wcTimeEnd = getTimeStamp();
    wcTime = wcTimeEnd - wcTimeStart;
    printf("Walltime: %.3lf s\n", wcTime);

    return 0;
}
```



- Do not run benchmarks on the frontend nodes, as multiple programs and users interact there
- You may do test runs, e.g., compilation tests and verification on frontends, but also on the compute nodes (via an interactive job)
- For obtaining lots of results, write your own scripts and execute them via the batch system
- Check your results for plausibility (Cool! My code runs @ petaflop/s!)



- Getting Started Guide for your work with the cluster:
 - <https://moodle.rrze.uni-erlangen.de/mod/page/view.php?id=7900>
- Linux tutorial for n00bs:
 - <https://ryanstutorials.net/linuxtutorial/>
- Intel processor details (just one example):
 - [https://en.wikipedia.org/wiki/Ivy_Bridge_\(microarchitecture\)](https://en.wikipedia.org/wiki/Ivy_Bridge_(microarchitecture))
- Confused about all those CPU code names?
 - https://en.wikipedia.org/wiki/Intel_Core
- Emmy cluster details:
 - <https://www.anleitungen.rrze.fau.de/hpc/emmy-cluster/>
- Blogs by RRZE HPC staff:
 - <https://www.blogs.fau.de/hager/>
 - <https://www.blogs.fau.de/zeiser/>



- **First assignment is released today**
 - Deadline for Assignment 0: May 7 at 8 AM
- **Future assignments are released on Thursdays**

- **Report submission:**
 - Deadline: next Thursday morning at 8 AM(!)
 - Upload a single file in moodle
 - PDF report or
 - compressed archive including a PDF report and supporting material
 - Grading will be done based on PDF report
 - NOT required, but necessary to get bonus

- **Problem?**
 - Ask during Monday Q&A
 - Use moodle forum



- **Report must include**
 - Specific answers to questions/tasks mentioned in assignments
 - Explanation on how you arrived at your answer
 - Description of the steps you took to measure performance/timings/etc.
 - Documentation of used compiler switches, used frequencies and anything necessary to reproduce your results by someone else

- **Complete sentences**

- **Never forget units!**

- **When using plots for explanation: Label axes!**



- **Presentation of solution to previous assignment**
- **Presentation of current assignment**
- **Possibility to ask questions**



- **Hands-on introduction into benchmarking practices on our clusters and with LIKWID:**

<https://fau.zoom.us/j/95648105106?pwd=ODIQZUZGbVZ3U3Y4UXBBc2c5cTB0QT09>

- Thursday April 30, 16:15-17:45
 - Thursday May 7, 16:15-17:45
-
- **Not obligatory, but highly advised to take part.**
-
- **Recordings will be provided afterwards.**