Hessisches Kompetenzzentrum
für Hochleistungsrechnen

# HKHLR – How To
# run ANSYS-Fluent on an HPC-Cluster

Version of 29th of March 2019

## Preliminary remarks

This guide describes the steps for running parallel computations performed by ANSYS on the Lichtenberg cluster[1]. In this case a computational problem is prepared in advance by means of an application with the graphical user interface (GUI), while computations on the cluster are performed only in batch mode without using the GUI.

If this is the first time you want to run a job on Lichtenberg Cluster, please visit an introductory course!

ANSYS is a licensed software, please respect the license-agreement for the use of ANSYS at TU Darmstadt:
`www.hrz.tu-darmstadt.de/software/uebersicht_1/campuslizenzen/ansys`

## ANSYS Fluent on a Linux machine - Command Line

To use ANSYS Fluent on the Lichtenberg cluster, a module of a proper version must be found and loaded:

```
module avail ansys
module load ansys/<VERSION>
```

**Attention: It is highly recommended to prepare a problem and to perform computations by using the same version of the ANSYS package (it must be checked before computations).**

After loading the module, the ANSYS manual with detailed information is available by the command

```
anshelp
```

ANSYS Fluent is run in the command-line in interactive mode by the command

```
fluent
```

In this mode the ANSYS GUI is run, and the user can adjust some parameters of numerical simulation, although it is better to prepare everything on a local machine. Important command-line options of Fluent can be found in a short manual (press `q` to close this manual):

```
fluent -help | less
```

To run ANSYS Fluent for performing computations in batch mode, the `fluent` command is followed by a general specification of the solver, which consists of the type of a problem (`2d` for 2D case and `3d` for 3D) and of the type of floating-point format (float is set by default and `dp` is added to apply double-precision floating-point format). For example, a 3D problem with double-precision format is specified as `3ddp`.

To run Fluent for performing parallel computations in batch mode on the Lichtenberg cluster, the following command-line options must be specified (in addition to the specification of the problem):

- `-g` to run in batch mode (without GUI);
- `-mpi=intel` to specify Intel MPI;

[1] This guide should work for other clusters with the scheduler SLURM, too

- `-ssh -pinfiniband` to specify the internode connection;
- `-t<N>` to specify the total number of cores (tasks) used for computing;
- `-cnf=<hostsfile>` to specify a file of the names of the nodes for computing;
- `-i <journalfile>` to specify the journal file (the input file);
- `> <outputfile>` to specify the output file of a batch execution.

The file of the names of the computational nodes (`<hostsfile>`) is generated automatically (see the batch script at the end). The journal file (`<journalfile>`) must be created by the user before computing. The output file (`<outputfile>`) is generated during computations by Fluent. The names of the journal and output files are user-defined and must **not** contain any whitespaces.

As an example, the following command is applied to start ANSYS Fluent for performing computations with double-precision of a 3D problem in batch mode with 16 cores:

```
fluent 3ddp \
      -g \
      -mpi=intel -ssh -pinfiniband \
      -t16 \
      -cnf=hosts.file \
      -i fluent.jou \
      > fluent.out
```

Here `host.names` is the file with the names of the computational nodes (or the hosts) discussed below, `fluent.jou` is the journal file, and `fluent.out` the output file. The journal and output files are given without paths and must be located in the same directory where the `fluent` command is executed (this directory is called the working directory). Otherwise, these names must be specified together with their paths. How to use this command for computing on the cluster is discussed at the end of this manual.

## Case and data files

To save a problem prepared for computations by ANSYS Fluent and to read it again, a case file and a data file are applied. Both files are generated in Fluent and have the standard extensions `.cas` and `.dat` (for the case file and the data file, respectively). While the case file contains general information on the problem (including settings), the data file stores fields (initial conditions or results of computations).

Both files can be read/write by using a single command or by using two separate commands. If the files have their standard extensions (`.cas` and `.dat`), it is enough for Fluent to specify only the (first) parts of their names without extensions. It is also possible to read/write both files simultaneously by one command. In this case, they must have their standard extensions and the same first part of the name.

## Journal file

The journal file contains commands executed by ANSYS Fluent and structures of a scheme language, which is beyond the present guide. Any command of Fluent can be copied into the journal file from the console integrated in Fluent (available in the GUI). The journal file is primarily used to read/write the case and data files, i.e. to load the problem and to save results.

A simple version of the journal file for a problem of a steady flow consists of the following commands:

```
file/read-case-data inputfile
solve/iterate 10
file/write-case-data outputfile10
exit
```

While executing the above commands, Fluent reads the initial case and data files, `inputfile.cas` and `inputfile.dat`, by the single command, perform $10$ iterations, write results to the output case and data files, `outputfile10.cas` and `outputfile10.dat`, by the single command, and exit.

**Attention: If the output case/data files exist, Fluent will ask for rewriting them. The above journal file does not contain commands to respond to the question. That is why, it is recommended to clean the working directory from old output files before starting new computations.**

A more advanced version of the journal file is obtained by adding several commands:

```
file/read-case-data inputfile
parallel/timer/reset
solve/iterate 10
file/write-case-data outputfile10
solve/iterate 10
file/write-case-data outputfile20
parallel/timer/usage
report/system/proc-stats
report/system/sys-stats
report/system/time-stats
exit
```

This file contains the commands for writing intermediate results (after $10$ iterations) and for writing addition statistical information on the solution process to the output file generated by Fluent (e.g. elapsed time, performance, required memory).

All the names of the input/output files given in the above journal files are user-defined. If these files are located in the current working directory (where Fluent is run), it is enough to specify only their names without the absolute/relative paths. Both journal files are shown as examples, and the user should write own journal files. Moreover, both files were written for the problem of a steady flow.

**Attention: It is recommended to test the journal file in the Fluent console available in the GUI on a local machine before performing parallel computations.**

## Submitting a job: Batch script

To run parallel computations on the Lichtenberg cluster, the `fluent` command with its options must be specified in a so-called batch script, which is a text file. An example of the batch script for submitting the job to the scheduler SLURM is shown in listing 1. The command to submit the batch script to SLURM:

```
sbatch <batch_script>
```

where the name `<batch_script>` is user-defined (whitespaces are not allowed) The directory where the batch script is submitted is called the working directory. If all input/output file are located in this directory, it is enough to specify only their names without the paths to these files.

The batch script must include the settings carefully specified and described below.

1. The name of the job is specified by the `-J` option (whitespaces are not allowed).

2. The name of the standard output file and the name of the file of errors are specified by the options `-o` and `-e`, respectively (whitespaces are not allowed).

3. The required memory specified by `--mem-per-cpu` is an estimated value. In case of memory issue, the value can be increased until a certain limit (e.g. the maximum of memory available for a processor).

4. The type of computational nodes specified by `-C` is limited by `avx` for phase 1 (Sandy-Bridge nodes) and `avx2` for phase 2 (Haswell nodes).

5. The wall-clock time specified by `-t` should be estimated carefully. It is recommended to perform tests for estimating the average time required for one iteration or for one step over time. The corresponding average time can be found in the output file (generated by Fluent and specified by the option of the `fluent` command), if one of the following commands are added to the journal file:

   ```
   parallel/timer/usage
   report/system/time-stats
   ```

   The number of iterations and/or the number of steps over time must be set so that computations have performed within the wall-clock time. Note that computations with a shorter wall-clock time can be started by SLURM faster.

6. The number of cores for computing specified by the `-n` option should be chosen carefully. On the one hand, many cores allow to run numerical simulation of large problems and to perform computations faster. On the other hand, there is the maximum number of cores, when further increase does not influence the performance significantly. This maximum depends on a problem and should be determined by performing short tests.

7. The module of the ANSYS package loaded for performing computations on the cluster must correspond to the version of ANSYS used in preparing numerical simulation.

To avoid problems while running computations, the important recommendations must be followed.

1. Names of directories/files of all input/output files used in computations must not contain whitespaces.

2. One directory should be used as the working directory only for one run of computations to avoid rewriting output files. In other words, only one job should be submitted in one directory, where output files are generated.

3. If the case and data files are used as the input files of multiple numerical simulations, it is reasonable to store them in a separate directory and to specify their names together with the absolute paths in the journal file.

4. It is recommended to specify all necessary loaded modules directly in the batch script and **not** to use other files including configuration files (e.g. the `.bashrc` file).

5. It is not recommended to use GPGPU due to long time for waiting in a queue (the number of nodes with GPUs is very limited). The inclusion of the support of GPGPU does not give considerable effect on the performance.

```
#!/bin/bash

#SBATCH -J jobfluent_test
#SBATCH -e jobfluent_test.err.%j
#SBATCH -o jobfluent_test.out.%j
#SBATCH --mem-per-cpu=1650
#SBATCH -C avx
#SBATCH -t 00:10:00
#SBATCH -n 16

module purge
module load ansys/17.2

#Generate file of names of computational hosts.
MYFILEHOSTS="hosts.$SLURM_JOB_ID"
srun hostname | sort > $MYFILEHOSTS

#Wait for finishing.
wait

#Run Ansys Fluent.
fluent 3ddp \
      -g \
      -mpi=intel -ssh -pinfiniband \
      -t$SLURM_NTASKS \
      -cnf=$MYFILEHOSTS \
      -i fluent.jou \
      > fluent.out
```

Listing 1: The example of the batch script for running parallel computations on the Lichtenberg cluster performed by ANSYS Fluent with using 16 cores of nodes of phase 1. The file of the names of the computational nodes is generated automatically (its name is stored in the `MYFILEHOST` variable). The journal file, `fluent.jou`, must be prepared in advance. The input case and data files specified in the journal file must be prepared before computing. The output file, `fluent.out`, is generated by Fluent during computations. The names of the journal and output files are user-defined, must **not** contain whitespaces, and are located in the working directory. The script can be adapted for other parallel computations performed by ANSYS Fluent.

### Remark

The present “How To” is based on the experience obtained in the HKHLR group. Any questions, suggestions and possible improvements are welcome as well as the feedback, if the guide is useful. Please, contact us by email: staff@hpc-hessen.de.