Hessisches Kompetenzzentrum
für Hochleistungsrechnen

HKHLR–How To
run ANSYS CFX 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 CFX on a Linux machine - Command Line

To use ANSYS CFX on the Lichtenberg cluster (as well as on other Linux machines with *environment modules*), a module of a proper version of the ANSYS package 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 CFX is started in the command-line in interactive mode simply by using the command

```
cfx5solve
```

In interactive mode, the user is able to adjust numerical simulations and to set parameters of a computational problem. Important options of the CFX solver which are mostly applied in batch mode can be found in a short manual (press `q` to exit):

```
cfx5solve -help | less
```

To run ANSYS CFX for performing parallel computations in batch mode on the cluster, the following command-line options of the `cfx5solve` command must be specified:

- `-def <input_file>` to specify the input file of the solver for a single configuration simulation;
- `-parallel` to run ANSYS CFX in parallel mode (necesssary if you use -n or -C option in your batch script);
- `-start-method <parallel_method>` to specify one of parallel methods used by the CFX solver;

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

- `-par-dist <hostlist>` to specify the list of names of computational nodes (or hosts);
- `-batch` to run the ANSYS CFX solver in batch mode.

The file specified by the `-def` option describes a computational problem (mesh, boundary/initial conditions, settings of the solver, and so on). This file is prepared by the user in advance by using the pre-processing tool `cfx5pre` (a definition file) or is obtained from other computations done by ANSYS CFX (a file of results). On the Lichtenberg cluster the `"Intel MPI Distributed Parallel"` method is recommended to be specified for the `-start-method` option. The `-par-dist` option is used to specify the list of names of nodes (hosts) and corresponding numbers of cores (tasks) of each node applied for performing numerical simulations (e.g. `host1*N1,host2*N2`). This list must be generated at runtime automatically, as nodes are assigned by the batch system (see below).

As an example, the following command is applied to start ANSYS CFX for performing computations, where `inputfile.def` is the input file of the CFX solver, the environment variable `MYHOSTLIST` stores the string of the names of the computational nodes:

```
cfx5solve -def inputfile.def \
        -parallel \
        -start-method "Intel MPI Distributed Parallel" \
        -par-dist "$MYHOSTLIST" \
        -batch
```

In addition to the important command-line options described above, some options are worth mentioning, but they are not mandatory:

- `-double` to use double-precision floating point format (it can be specified directly in `cfx5pre`);
- `-fullname` and `-name` to specify the name of the output files (instead of the standard ones);
- `-continue-from-file <result_file>` to use initial values and to continue the run history from the specified file of results (restart of computations);
- `-baket <elapsed_time_frequency>` to write a backup file with a given frequency;
- `-max-elapsed-time <elapsed_time>` to set the maximum time for running the CFX solver.

In addition, the CFX solver can be applied for numerical simulation of problems of fluid-structure interaction (FSI problems). In this case, the useful options of CFX can be the following: `-mfx-run-mode`, `-ansys-input`, `-ansys-restart`. More information can be found in the ANSYS manual.

Submitting a job: Batch script

To run ANSYS computations on the Lichtenberg cluster, the `cfx5solve` command together with its options must be specified in a so-called batch script. An example of the batch script for submitting computing jobs to the scheduler SLURM is shown in listing 1.

The directory where the batch script is located is called a working directory by default, although there is an option used for setting a particular directory as the working directory.

The command to submit the batch script, with a user-defined name `<batch_script>` (whitespaces are not allowed), to the scheduler SLURM is:

```
sbatch <batch_script>
```

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. 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 one processor, 2GB on Lichtenberg Cluster).
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 and/or for one step over time. Corresponding data can be found in the output file of the CFX solver. Note that computations with a shorter wall-clock time can be started by SLURM faster. **After your time limit, a restart based on a finishied time step (or iteration) with ANSYS CFX is possible. To do that in a proper way with the scheduler SLURM, the use of job arrays is recommended. Please feel free to contact us, if you are not sure how to write the bash-script for that cases.**
6. The number of computational tasks (or cores) specified by the `-n` option should be chosen carefully. On the one hand, more cores allow to run numerical simulations of larger problems and to perform computations faster. On the other hand, there is the maximum number of cores, when further increase make the job inefficient. This maximum highly depends on a problem and can be determined by performing tests.
7. The module of the ANSYS package loaded for performing computations must correspond to the version of ANSYS used for preparing numerical simulations.

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

1. Names of all input/output files as well as directories 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 definition file or the file of results obtained from the previous computations is used as the input file of multiple numerical simulations (specified by the `-def` option), it is reasonable to store the file in a separate directory and to provide its name with the absolute path. Each numerical simulation is performed in a separate directory, where the output is generated (see the previous item). If the definition file or the file of results is located in the working directory, it is enough to specify only its name. Again: Please feel free to contact us, if you are not sure how write to the bash-script for that cases.
4. It is highly 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).

```bash
#!/bin/bash

#SBATCH -J jobcfx_test
#SBATCH -e jobcfx_test.err.%j
#SBATCH -o jobcfx_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 line of names of computational nodes (or hosts).
MYHOSTLIST=$(srun hostname | sort | \
               uniq -c | \
               awk '{print $2 "*" $1}' | \
               paste -sd, -)

echo $MYHOSTLIST

#Run Ansys CFX.
cfx5solve -def inputfile.def \
        -parallel \
        -start-method "Intel MPI Distributed Parallel" \
        -par-dist "$MYHOSTLIST" \
        -batch
```

Listing 1: The example of the batch script for running parallel computations on the Lichtenberg cluster performed by ANSYS CFX using 16 cores of nodes of phase 1. The variable `MYHOSTLIST` generated automatically contains the names of the computational nodes. The input file, `inputfile.def`, must be created in advance (the name is user-defined). The script can be adapted for other parallel computations performed by ANSYS CFX.

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.