1. build process

build.sh:

```
#!/usr/bin/bash
spack load intel-oneapi-mkl@2024.0.0
spack load openblas
spack load openmpi@5.0.3/ccc6ge6
spack load zlib

export MPICC=mpicc
export MPICXX=mpicxx
export CXXFLAGS="-Ofast -Wall -march=znver2 -mtune=native -mfma -ffast-math
-mieee-fp"
export CFLAGS="-Ofast -Wall -march=znver2 -mtune=native -mfma -ffast-math
-mieee-fp"

make clean
make -j 8

cd rayleighTaylor3d
make clean
make -j 8
```

config.mk:

```
# OpenLB build configuration
#
# This file sets up the necessary build flags for compiling OpenLB with
# the GNU C++ compiler and sequential execution. For more complex setups
# edit this file or consult the example configs provided in `config/`.
#
# Basic usage:
#  - Edit variables to fit desired configuration
#  - Run `make clean; make` to clean up any previous artifacts and compile
the dependencies
#  - Switch to example directory, e.g. `examples/laminar/poiseuille2d`
#  - Run `make`
#  - Start the simulation using `./poiseuille2d`

# Compiler to use for C++ files, change to `mpic++` when using OpenMPI and
GCC
CXX             := mpicxx
```

```
# Compiler to use for C files (used for emebedded dependencies)
CC              := mpicc

# Suggested optimized build flags for GCC, consult `config/` for further
examples
CXXFLAGS        := -Ofast -Wall -march=znver2 -mtune=native -mfma -ffast-math
-mieee-fp
# Uncomment to add debug symbols and enable runtime asserts
#CXXFLAGS        += -g -DOLB_DEBUG

# OpenLB requires support for C++17
# works in:
#  * gcc 9 or later      (https://gcc.gnu.org/projects/cxx-status.html#cxx17)
#  * icc 19.0 or later
(https://software.intel.com/en-us/articles/c17-features-supported-by-intel-c-
compiler)
#  * clang 7 or later  (https://clang.llvm.org/cxx_status.html#cxx17)
CXXFLAGS        += -std=c++17 -fPIC

# optional linker flags
LDFLAGS         :=
-L/opt/.spack/opt/spack/linux-ubuntu22.04-zen2/gcc-11.4.0/gcc-14.1.0-affchq7w
2n7kmuuu2jbzsjfmge3fcy6y/lib64
-Wl,-rpath,/opt/.spack/opt/spack/linux-ubuntu22.04-zen2/gcc-11.4.0/gcc-14.1.0
-affchq7w2n7kmuuu2jbzsjfmge3fcy6y/lib64

# Parallelization mode, must be one of: OFF, MPI, OMP, HYBRID
# Note that for MPI and HYBRID the compiler also needs to be adapted.
# See e.g. `config/cpu_gcc_openmpi.mk`
PARALLEL_MODE   := HYBRID

# optional MPI and OpenMP flags
MPIFLAGS        :=
OMPFLAGS        := -fopenmp

# Options: CPU_SISD, CPU_SIMD, GPU_CUDA
# Both CPU_SIMD and GPU_CUDA require system-specific adjustment of compiler
flags.
# See e.g. `config/cpu_simd_intel_mpi.mk` or `config/gpu_only.mk` for
examples.
# CPU_SISD must always be present.
```

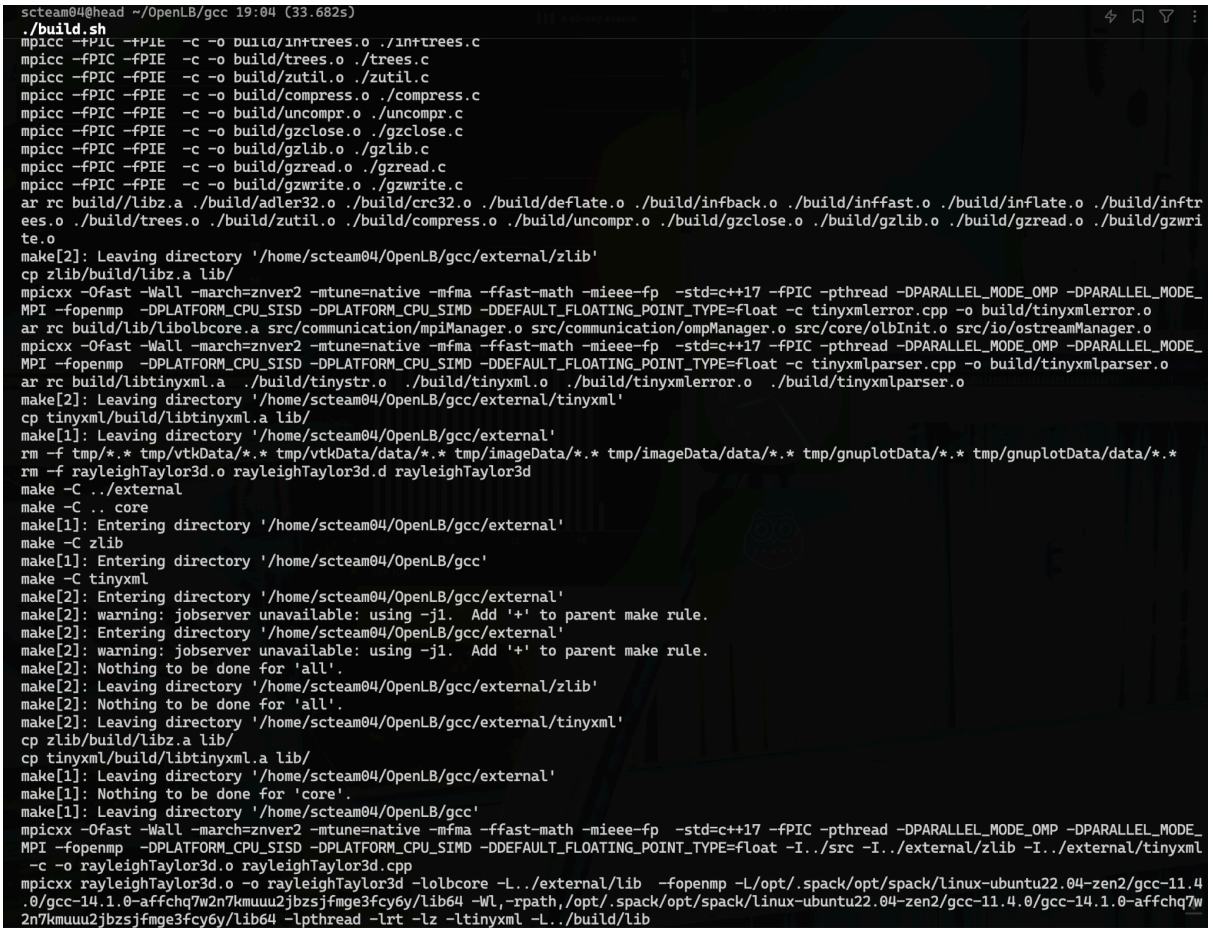
```
PLATFORMS        := CPU_SISD CPU_SIMD


# Fundamental arithmetic data type
# Common options are float or double
FLOATING_POINT_TYPE := float


# Any entries are passed to the compiler as `-DFEATURE_*` declarations
# Used to enable some alternative code paths and dependencies
FEATURES         :=


# Set to OFF if libz and tinyxml are provided by the system (optional)
USE_EMBEDDED_DEPENDENCIES := ON
```

2. build screenshot



3. running

```
[OmpManager] Sucessfully initialized, numThreads=1
[ThreadPool] Sucessfully initialized, numThreads=1
[prepareGeometry] Prepare Geometry ...
[SuperGeometry3D] cleaned 0 inner boundary voxel(s)
[SuperGeometryStatistics3D] updated
[SuperGeometry3D] the model is correct!
[CuboidGeometry3D] ---Cuboid Stucture Statistics---
[CuboidGeometry3D]  Number of Cuboids:  16
[CuboidGeometry3D]  Delta  (min):              1
[CuboidGeometry3D]         (max):              1
[CuboidGeometry3D]  Ratio  (min):              0.5
[CuboidGeometry3D]         (max):              2
[CuboidGeometry3D]  Nodes  (min):              500000
[CuboidGeometry3D]         (max):              500000
[CuboidGeometry3D]  Weight (min):              500000
[CuboidGeometry3D]         (max):              500000
[CuboidGeometry3D] --------------------------------
[SuperGeometryStatistics3D] materialNumber=1; count=3960000; minPhysR=(0,1,0); maxPhysR=(199,99,199)
[SuperGeometryStatistics3D] materialNumber=2; count=3960000; minPhysR=(0,100,0); maxPhysR=(199,198,199)
[SuperGeometryStatistics3D] materialNumber=3; count=40000; minPhysR=(0,0,0); maxPhysR=(199,0,199)
[SuperGeometryStatistics3D] materialNumber=4; count=40000; minPhysR=(0,199,0); maxPhysR=(199,199,199)
[SuperGeometryStatistics3D] countTotal[1e6]=8
[prepareGeometry] Prepare Geometry ... OK
[prepareLattice] Prepare Lattice ...
[prepareLattice] Prepare Lattice ... OK
[main] starting simulation...
```