Schedule:

Date	Activity	Status
10/31-11/6	Be familiar with algorithm, design the data structure for the implementation of serial version. Keep the design suitable for parallelization	complete
11/07-11/13	Complete the serial version of lid-driven cavity model	complete
11/14-11/20	Complete the serial version of data center air-conditioning system	complete
11/21-11/24	parallelize the serial version of lid-driven cavity with CUDA	complete
11/25-11/27	parallelize the serial version of lid-driven cavity with OpenMPI	complete
11/28-11/30	test the correctness of lid-driven cavity with CUDA	complete
12/1-12/3	test the correctness of lid-driven cavity with OpenMPI	complete
12/4-12/6	Test performance with different number of processors with CUDA	complete
12/7-12/9	Test performance with different mesh size and numbers of processors with OpenMPI	complete
12/10-12/13	Complete the poster and report	complete

Milestone status:

Instead of directly building the complex geometry of data center, we have firstly completed the serial version of CFD simulation of “lid-driven cavity”. With the mesh of 128*128 cells, the serial version takes 99.9s to complete the simulation of the steady airflow patterns within the cavity after 10s. The correctness of the serial version has been validated by comparing the results with Matlab version.

After analyzing the code, we’ve realized the iterations of computing pressure solved by Poisson equation with Gauss-Seidel method have accounted for the major part of the runtime. Therefore, for the following weeks, we will implement CUDA to get the expected speedups by parallelizing Gauss-Seidel method. Meanwhile, we will also use sub-domain decomposition to parallelize the base domain using MPI.

Revised goals and deliverables:

We are a little bit behind our proposed schedule as we should start our first parallel version by the end of last week. However, we should be able to compare our parallel version with serial version as well as simulation with ANSYS in the end. Instead of having a demo during poster session, we will show speedup graph of the approaches we have implemented and some analysis as well as some visualizations.

a) Speedup graphs of two or possibly more parallel approaches
b) Analysis of the parallel approaches with speedup benefits and bottlenecks
c) Simulation visualization and animation

Potential issue:

a) When using CUDA, the memcpy between cpu and gpu may cause unsatisfactory speedup
b) When using OpenMPI, the communication between subdomains may cause unsatisfactory speedup.