First, let’s start with a group exercise (15 minutes)

MapReduce is based on list mapping techniques from functional programming. “Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. The run-time system takes care of the details of partitioning the input data, scheduling the program’s execution across a set of machines, handling machine failures, and managing the required inter-machine communication. This allows programmers without any experience with parallel and distributed systems to easily utilize the resources of a large distributed system”.[1]

Answer the following questions: 1) What is the programmer’s role in running MapReduce?

2) (True or False?) A single physical machine can only be assigned a single task, either a mapper or a reducer.

3) How does the interaction happen between the mappers and reducers in the system?

   Interaction among Mappers?

   Interaction among Reducers?

   Interaction between Mappers and Reducers?

4) What is the role of the master?

5) Why do completed map tasks need to be re-executed on failure, while completed reduce tasks do not?

6) What happens if the master fails?