Recitation 7: MapReduce

Pre-Recitation

1. What are the performance goals of MapReduce (both the programming model + its implementation)?

    - MapReduce created a way, through functional programming style, to automate the parallelization of data processing across multiple machines.
    - It was to do so in an extremely fault tolerant manner.
        - worker failures
        - master failures
    - It was to do so in an extremely scalable way as well.
        - Able to process terabytes of data across thousands of machines

2. How was MapReduce implemented at Google to meet those goals?

    - MapReduce used a restricted progamming model to make it much easier to distribute the work while maintaining fault tolerance.
    - MapReduce also was implemented in a way that reduced the amount of data sent across networks.
    - MapReduce will use idle machines to duplicate computations across multiple machines.

3. Why was MapReduce implemented in this way?

    - The restircted programming model enforces constraints that allow different pieces of information to be processed seperately, making parallelization of computaion much easier.
    - Google engineers found that networks are extremely slow and tend to be a bottleneck, so its best to optimize in a way that relies the least on the network.
    - By duplicating computations across many other machines that otherwise would be idle, we reduce the impact that slow machines have, thus clearing bottlenecks caused by a slow machine.