Data Intensive Computing - Review Questions 2

Deadline: September 11, 2020

1. **1 point.** Imagine you are working in a big company, and your company is planning to launch the next big Blogging platform. Tomorrow morning you go to your office and see the following mail from your CEO regarding a new work. How do you answer to this email? Hint: use MapReduce to solve it, and explain what you do in Map and Reduce phases.

   ```
   Dear <Your Name>,
   ```

   ```
   As you know we are building a blogging platform, and I need some statistics.  I need to
   find out, across all blogs ever written on our blogging system, how many times 1 character
   words occur (e.g., "a", "I"), How many times two character words occur (e.g., "be", "is"),
   and so on.  I know its a really big job.  I am going on a vacation for one week, and its
   really important that I've this when I return.  Good luck.
   ```

   ```
   Regards,
   ```

   ```
   The CEO
   ```

2. **1 point.** What is the problem of skewed data in MapReduce? (when the key distribution is skewed)

3. **1 point.** Briefly explain the differences between Map-side join and Reduce-side join in Map-Reduce?

4. **1 point.** Explain briefly why the following code does not work correctly on a cluster of computers. How can we fix it?

   ```scala
   val uni = sc.parallelize(Seq(("SICS", 1), ("KTH", 2)))
   uni.foreach(println)
   ```

5. **1 point.** Assume you are reading the file `campus.txt` from HDFS with the following format:

   ```
   SICS CSL
   KTH CSC
   UCL NET
   SICS DNA
   ...
   ```

   Draw the lineage graph for the following code and explain how Spark uses the lineage graph to handle failures.

   ```scala
   val file = sc.textFile("hdfs://campus.txt")
   val pairs = file.map(x => (x.split(" ")(0), x.split(" ")(1)))
   val groups = pairs.groupByKey()
   val uni = sc.parallelize(Seq(("SICS", 1), ("KTH", 2)))
   val joins = groups.join(uni)
   val sics = joins.filter(x => x.contins("SICS"))
   val list = sics.map(x => x._2)
   vali result = list.count
   ```