Cloud Computing, Fall 2019

Take-home final exam

Please carefully read the following notes:

1. Some questions are open. Please try your best to give your answers based on your understanding.
2. You are welcome to search the web to obtain the background knowledge. However, you **CANNOT** copy & paste answers directly from the sources or from other students. Please use your own words. Any significant duplicates will result in 0 for that question.
4. You will need to include the references (papers or URLs) you use for your answers.
5. Try to make your answers concise (typically in 1 page, no more than 2 pages per question). Do not include irrelevant information.

Question 1. We have discussed Spark in the class. Message Passing Interface (MPI) is another method for parallel computing infrastructures such as high performance computing (HPC). Please discuss the major features distinguishing Spark and MPI, including the following aspects: *performance, easiness of algorithm design and programming, and fault tolerance*.

Question 2. Apache Flink was developed for stream processing. You can check flink.apache.org or other resources to find the related information. Similarly, Spark has been used for stream processing as well (which was not discussed in the class).
(1) Please briefly describe how Spark and Flink stream processing works.
(2) Compare the fault-tolerance methods used by Flink and Spark and describe the differences if there is any.

Question 3. Phishing/spamming emails contain false information or advertisements and try to take advantage of careless email receivers. It is challenging to protect internet users from spamming and phishing if spammers use cloud computing resources for spamming.
(1) If spammers use Amazon EC2 for spamming, what are the major challenges in preventing this type of spamming?
(2) If this problem is already addressed by Amazon, please describe the Amazon's method. If you think the problem is still not well addressed, can you briefly describe your idea to detect and prevent spamming/phishing?

Question 4. The cloud + mobile model is the important paradigm behind many mobile applications. The typical examples are google search/maps accessed from mobile phones. In this paradigm, the data and compute intensive components run in the cloud, while the presentation and user-interaction components run in the phone. Assume we have a facial recognition app that has the following steps: capture image → preprocess image → extract features from images → run feature matching algorithm →retrieve information of the most matched persons from the person database →present the information to the user. You need to make decision on placing the components on the cloud or the client side.

If you don't have the background knowledge of this application, you may read some web resources to understand it first. After you fully understand what happens in the workflow, please describe your designs of (1) the cloud-intensive (using more cloud resources), (2) cloud-mobile mix, and (3) mobile-intensive schemes (using mostly or even all mobile resources), respectively. Discuss the pros and cons of each solution in terms of the factors: storage costs, communication costs, response time, and required mobile-phone's processing capability.

Question 5: We have discussed Intel SGX for secure cloud computing. You can check the slides and the tutorial https://eprint.iacr.org/2016/086.pdf to refresh your understanding. An SGX application typically has two parts: the one running in the SGX enclave with confidentiality and integrity guarantees, and the one running in untrusted memory area. If you are asked to design an SGX-based single machine ***wordcount*** program to securely process 1GB data, consider the following problems: (1) the enclave memory is limited, much less than 1GB. Should you be concerned about this issue? If so, how do you handle it? (2) How do you handle the integrity of the part of your program running in untrusted area? (3) briefly describe your design of the entire program.