Politecnico di Milano

Middleware Technologies for Distributed Systems

09th of September 2015

Rules:

- **You are not allowed to use books, notes, or other material.**
- **You can answer in Italian or English.**
- **Total time for the test: 1.5 hours.**
- **Check the back of this page for any additional exercises.**

---

1. Implement in **Java** a PhotoBook class that organizes a (finite) set of photos and uploads them to a remote host so that they can be shared with others. Suppose a Photo class exists, together with a class SharingSvc that provides the following method:

   void upload(String bookTitle, Photo[] photos, String[] captions).

   Class PhotoBook provides a constructor that specifies the maximum number of photos in the book and the book's title. It also provides the following methods:

   int add(Photo p) : this method adds a new photo in the book's first empty slot and returns the slot number. It suspends the caller if the book is full.

   void setCaption(int slot, String caption) : this method sets the caption for the photo in the given slot.

   void upload(SharingSvc svc): this method uploads the entire book to the given sharing service and clears the book, leaving it ready to be filled again. Since the uploading operation may take a long time, method upload must operate asynchronously w.r.t. the caller (i.e., using a separate thread).

2. In **TinyOS** write a module MaxRead that uses and implements the Read<uint16_t> interface. When requested to read a value from its clients, it reads the sensor wired to the Read<uint16_t> interface it uses 10 times (each read being interleaved from the previous one by 20ms) and returns the maximum value read. The following interfaces may help you:

```
interface Timer<precision_tag> {
    command void startPeriodic(uint32_t dt);
    command void startOneShot(uint32_t dt);
    command void stop();
    event void fired();
}

interface Read<val_t> {
    command error_t read();
    event void readDone(error_t err, val_t val);
}
```

3. Write the **OpenMP** code (the minimal fragment of code you need) to count the hamming distance between two arrays of integers data1 and data2 (of size NUM). Note: The hamming distance of two arrays is the number of positions at which the two arrays differ, i.e., the number of times the two arrays have different values in the same position.

4. Imagine you are developing a **REST** API for performing bank operations, and that you have identified each bank account as a single resource. Now imagine that you want to allow a developer to transfer money from one account to another. This implies an adjustment on both the transfer's source account and on the transfer's destination account. How would you design this using REST? Take into account that a bank transfer needs to be an atomic operation. How can atomic (and more in general batch) operations be designed in REST?

5. Mobiphone is a mobile telephone company. It is interested in providing its customers with detailed access to call records and to billing information. Assume that each phone call generates a tuple of the kind <date, sourceNum, destNum, length, cost>. Date provides the moment in which the phone call started, in the format DD:MM:YYYY-HH:MM:SS. sourceNum and destNum are the phone numbers of who made and who received the call respectively. Cost is the cost in euros of the phone call.

   1. Given a specific phone number, we want to provide the aggregate costs for the last three months as well as the ongoing month's partial. For example, if the request is made on September 9th we want to be able to provide the totals for June, July, and August, as well as the ongoing partial for September. Describe the **Hadoop** job(s) needed to produce this information.

   2. Given a specific phone number, we want to calculate the average amount of money spent per month (in 2015) by that customer calling people that are on the same mobile network, as well as the average amount of money spent per month (in 2015) by that same customer calling people that are NOT on the same network. People that are on Mobiphone's network all have phone numbers starting with "333". Describe the **Hadoop** job(s) needed to produce this information.

   NOTE: you do not have to write code; simply describe the dataset structures and what you would do in the various job(s) steps.