**Project2: Distributed computation competition on Raspberry Pi network (60%, due Nov 25)**

*Goal: give students hand-on experience with various aspects of mobile distributed systems: scalability, fault-tolerance and transparency.*

Each team is asked to design and develop a distributed computation framework, which consists two key components: (i) communication protocol that discovers helper nodes in the network (Ethernet or wireless) and monitors their operational status (load and link quality), and (ii) algorithm that divides a relatively large task (e.g., sorting list of numbers) into the appropriate size for helper nodes and coordinates with those helper nodes to complete the tasks. In the design, each team needs to address the following issues:

* Scalability, nodes are joining and leaving dynamically
* Load-balancing, nodes’ capability may change over time, therefore the algorithm need to monitor the performance metrics of a node
* Fault-tolerance, helper nodes may experience bad connectivity, therefore the algorithm may need to reschedule the incomplete task to others

For this project, we will test the following two tasks:

* Sort a large set of N random character+integers

(Example: K2342, w421, o9221, p999, k91)

* Compute the sum of all numbers that have the same prefix Char (case insensitive)

At week 14, I will collect all Raspberry Pi and connect them in a network. Some of them will be connected with Ethernet. I will periodically kill your client process; prepare your algorithm to address this problem.

Each team will put their algorithm in a competition. Each algorithm will be timed for its execution, the algorithm solved the problems correctly in minimum time is the Winner. The Winner of this challenge will receive a Bonus token. In the demo sessions, each team must share with the class some insights of their implementation. (**70% of Project 2**)

Test data (sample\_data.txt) is provided on myCourses. Each group is encouraged to test out your algorithm in the computer lab with 3-5 machines (to simulate the final environment). I will setup a network of Raspberry Pi for final test run before the demo week.

As part of the group project, each team needs to prepare a technical report, which discusses their design and development. Please use what you learned from paper reading challenge to prepare the technical report. Another Bonus token is given to group, which came up with a well-thought design described in report. The report is worth **30%** **of Project 2**. The report should consist the following elements:

* Design of the communication protocol and load distribution algorithm
* Implementation ideas; any special enhancement to improve the performance
* Evaluation results, for example, how your algorithm performance under different number of nodes
* A few words on future improvement

**Library**

Below link is for ssh utility (to run remote scripts) from Java

<http://www.jcraft.com/jsch/>

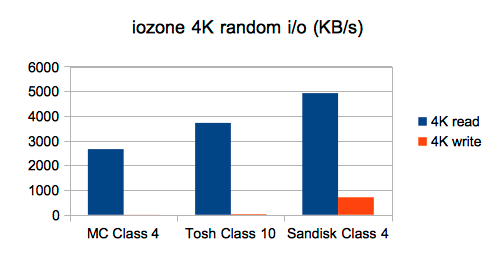
Example -

<http://stackoverflow.com/questions/16468475/sending-commands-to-remote-server-through-ssh-by-java-with-jsch>

The SCP command is very simple.

$> scp file1…fileN IP\_OF\_HOST:/PATH\_TO\_YOUR\_FOLDER

Please consider writing your own scp script for delivering the client program to other pi before the program start.



Source: https://blogs.oracle.com/blogsagainbynight/entry/sd\_card\_speed\_vs\_java

**Grading**:

Implementation (70%)

0-20% minimum code and/or no distributed aspects

21-50% reasonable code with distributed aspects; e.g., parallel computation

51-70% very tidy, well-organized code and well use of OO feature (if necessary)

Report (30%)

0-10% poorly structured and unable to identify key points

11-20% reasonable writing and identify some key points

21-30% good and concise writing (straight to the point) and identify all key points

My advice is start early!