Project: Mobile Based MapReduce
Name: Jake Sanders(jacobs22@uw.edu), Pingyang He(pingyh@uw.edu)

Introduction:

MapReduce is getting more and more popular since it is fully scalable, and it can process lots of data in parallel on hundreds of machines. Especially in this big data era, the demand for processing a large size of data is increasing. Meanwhile, another popular product, smartphone, also has a huge market. In 2013, there are about 4 billion active smartphones on this planet. However, consumer mobile devices largely spend their time idle, waiting for calls/text messages to arrive or for the user to unlock the device. Much of the time, this is the correct action, given that limited battery life is a primary consideration for many mobile users. However, there are significant periods of time, where the mobile device is both hooked in to external power and has access to wifi, where the computational power of the device could be exploited but currently is not. In a mobile marketplace which is incredibly hostile to up-front app costs, leveraging these exploitable cycles to offset costs presents opportunities to both mobile users as well as interested data miners.

To that end, we would like to construct a MapReduce framework where Map and Reduce tasks are processed on Android devices.

Related work:

MMR(http://mason.gmu.edu/~mhassanb/mmr.pdf) allows mobile users to use nearby public cloud to increase the performance of the intensive applications on their mobile phones. However, the MapReduce work is assigned to users' computers instead of their mobile phones. Also, MMR is designed to improve response time of mobile apps for end users. On the other hand, our framework is designed to provide computing power and mapreduce functions with lower costs to companies or organizations who do not want to pay for expensive servers and build up their own MapReduce framework.

Hyrax(http://reports-archive.adm.cs.cmu.edu/anon/2009/CMU-CS-09-164.pdf) is similar, it allows mobile apps to utilize the distributed resources remotely.

MapReduce System over Heterogenous mobile devices (http://dl.acm.org/citation.cfm?id=1694312) has a similar design. However, we aim to provide a library instead of building app directly, so any app can use our framework and potentially earn profit from it. Also, we are targeting on Android platform instead of iOS, and they have very different event handling mechanism.

Misco(http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=889F6AA97A36737166AB185A5F9B905C?doi=10.1.1.175.8435&rep=rep1&type=pdf) also uses similar mechanism. However, the design goal of Misco is to provide a powerful programming abstract to allow distributed mobile app development without the need to deal with underlying problems of distributed computing.

What we're doing:

In this project, we will design a MapReduce framework that does not use a large amount

of server or PCs. Instead, our design uses smart phones as MapReduce workers. We will borrow some computing powers from users' cell phones when the phones are idle or currently not using a lot of CPU time (for eg, reading books or browsing facebook). To achieve that, we set up a central server that is used for coordination works the nodes. When a smart phone is ready (e.g. an app is open, phone is on AC power), it will connect to the server and ask for work. Then, the central server will send a job (either Map or Reduce) along with the associated data (small chunk) to the phone. Once the job is done, the phone will send the result back to the server and ask for more work. On the server side, it contains a bunch of MapReduce tasks along with their data and a list of connected smart phones. It will dispatch Map job to a set of smartphones, collecting intermediate key value pairs from these phones, and send the Reduce job to another set(no necessary to be exclusive with the previous set) of smart phones. Later, the server will collect the final result from these smartphones.

This framework could potentially benefit the smart phone users, app developers, as well as the framework users. Any company or organization can use our framework to do MapReduce work with very low cost. We provide this framework as a library so that any app developers can integrate our work into their apps. The app maker will earn money based on the CPU time the app contributed. App makers can build apps that provide movies, music and books for good price, since this can attract more users spend more time on their apps, and then earn more dollars from it. We can also allow users directly sell CPU time to us if they want. Similar to app makers, the amount of payment could be based on the amount of work completed.

In order to evaluate the merits of our design, we could compare the relative performance of our system to, for instance, a single PC performing the same tasks on Hadoop. However, since the primary goal of the system is to take advantage of unused processor cycles rather than to perform jobs quickly, we will also have to evaluate how efficiently these tasks can be performed (e.g. how much power the phone consumes vs. the compensation for the customer), and if the system even makes sense from that standpoint.