COMS6156
Justin Hudis

Final Project Proposal

The topic I would like to investigate for my final project continues to be smartphone lifetime extension with a focus on upcycling. Specifically, I would like to explore upcycling from the perspective of a "lazy" consumer. Most research into upcycling assumes that millions of old phones will be collected by providers and hooked up to compute clusters in a data center somewhere. While this may be the most efficient way to aggregate the compute power of discarded phones, it relies on phone providers taking the lead on a cause they have not proven to be invested in. Moreover, a very sizable portion of consumers already leave their old phones to rot in a drawer rather than turn them into their provider. Imagine, instead, if those consumers kept their phones plugged in, installed some app which utilized the hardware, and kept it running indefinitely. I intend to implement and evaluate such an app.

Similar apps already exist, but they don't fully utilize the remaining compute power of an old smartphone. Any app currently on the market which promises to pay users in exchange for phone resources is only, in fact, taking advantage of their internet connection. That is, the customers of companies which produce these apps can route their traffic through the connected phones, allowing them to avoid unfair geographic IP bans which enforce government censorship or commercial restrictions. Countless carbon emissions go into manufacturing CPUs and RAM sticks for smartphones, so we should try to extract as much as we can from those components instead of generating more emissions to build new devices for computational power.

The reason for choosing a consumer-centric approach is that it is the easiest and fastest to adopt. It may be decades before academia solves scalable upcycling, providers decide it's worth trying, and they finally implement it. In the meantime, more and more phones are lost to junk drawers and landfills, and older phones become useless even for upcycling as their parts degrade. A more immediate solution is needed, and thus the responsibility rests in the hands of consumers.

Consumers have two drives: money and ease. They won't do anything if it doesn't make them money and they won't do anything if it is at all difficult. To address the former, my app could theoretically sell the compute power it harnesses and pay the consumers a fraction of the revenue. Regarding the latter, one must consider any "frictions" which may impede a lazy consumer from adopting my solution:

- While much prior research has chosen to replace the default OS in old phones for ease of manipulation, this would be a difficult operation for the average consumer. That is why I have chosen to develop an app that any consumer could simply install and start making money with in only a few clicks.
- Charging a phone when it is at full battery makes the battery degrade faster and wastes electricity, incurring excess charges for the consumer and harming the environment. To solve this problem, researchers purchase smart chargers for each of their phones which only charge the phones when necessary and even optimize for times of day when electricity is greener or cheaper in their area. However, a lazy consumer will not want to buy hardware to adopt my solution, so at best we can encourage but not require the use of a smart charger.

- In the data center model of phone upcycling, it is assumed that someone will be employed to replace phone batteries when they inevitably malfunction. That said, replacing a phone's battery is generally a difficult task and again requires purchasing hardware. One must assume that a lazy consumer will completely give up on their phone once the battery stops working. Thus, we can only assume that this app will run for around a year, i.e. the average time a battery lasts beyond the time the average consumer discards their phone. A more accurate estimate can be found through further research.
- This app must support very old mobile OS versions and not rely on any features specific to a new or unique OS. If a consumer tries to open my app and it crashes, they'll give up trying to use it.

I assume that the typical consumer has a charger for their old phone, a spare outlet, a Wi-Fi router, and a new phone they've replaced the old one with. Given this, my solution will be implemented using the following technologies: four 2017 Motorola Moto E4 Plus phones, one 2023 Samsung Galaxy S23 phone, a Wi-Fi router, five standard chargers, a power meter, a power strip, an Android app, and a DigitalOcean Droplet VM. The Moto phones represent the old phones of four people. The Samsung phone represents a consumer's new phone, which they will use to monitor the status of their old phones. All the phones will be connected to the internet over Wi-Fi. The old phones will be charged continuously using a single power strip which will be monitored by the power meter. The phones will all be running the same app which can be in either "monitor" or "worker" mode, corresponding to the new and old phones, respectively. The VM will be a command center for the workers and will update the monitors on their status. It will also be used to simulate requests that in practice would be made by a customer of my app.

Note that while the average user will likely only have one phone to monitor at a time, it is possible they have two functioning old phones or perhaps are a parent managing an entire household of old phones. Moreover, it is good software design practice to make the application extensible and I will need this feature for my own testing. Also, while the monitor side of the app will allow consumers to check their old phones' statuses whenever they'd like, it should generate very few notifications so that the consumer is not annoyed into muting or uninstalling the app. They will only be notified (via push notifications) when an old phone is no longer reachable, either because it crashed, lost internet connection, or ran out of battery.

My app will facilitate the worker phones to run a task whenever it is requested to, e.g. test if a number is prime or serve a webpage. With a network of phones constantly responding to rapid requests, this could allow a customer to perform a larger task, e.g. search for new prime numbers or host a website. Similar functionality can be seen with Google Cloud Functions, which makes it a good baseline.

The two goals of this app are to enable consumer profits and help the environment. To that end, I will run my phones for a week and compare the cost and CCI (Computational Carbon Intensity) of my phones to Cloud Functions of equivalent RAM and CPU in a similar manner to Switzer et al. from my first presentation.

COMS6156
Justin Hudis

When analyzing costs, I will ensure that there is enough profit for both the consumer and the theoretical app developers and managers. While constructing an exact business model would be out of scope for this project, it is worth noting that developers would need to be paid to design and maintain such an app, the command center server would need to be purchased and kept running, marketers would need to be hired to find customers, etc. Further, my app must be so economically efficient that customers would be able to provision its resources for a lower cost than conventional cloud resources, otherwise adoption would be slow.

There are some limitations to my experimental design not already mentioned. Realistically, the old phones should be in separate homes monitored by separate new phones, but this would be difficult to implement and not particularly illuminating. The old phones could also be of different models to demonstrate versatility, but heterogeneity in upcycling has already been investigated and is not the focus of this research, not to mention the additional cost. Lastly, I have limited the scope of my research to Android phones since they are the easiest to procure and develop for.