Energy Awareness in Mobile Applications

Kayla Wheeler
CSCI 460
Montana State University
Fall 2019

Introduction:

In everyday technology, one of the main features of a device people look at is the battery life. Battery life is also one of the first things that people will complain about as the device ages. As more and more features that require extra energy and hardware space are added to each new model of phones, battery life will always remain a concern for the user.

Apple estimates that one battery will last 400 charges, which is around two years' worth of charges. And as the battery gets closer to the 400 charges, the amount of charges needed within a day will increase. After the battery has reached that 400th charge, the battery will only be able to hold 80 percent of its original capacity. Apple has come out with ways of maximizing battery life and lifespan, which include turning off background activity in certain apps, and putting the device in the low power mode.

Samsung is also in the same boat as Apple. A Samsung lithium-ion battery will last around 2-3 years. But up until the Samsung Galaxy s5, Samsung devices had a bit of an upper hand on Apple devices. They had a removable battery, that could be replaced easily if it was starting to go bad. Samsung also came out with a power saving mode to reduce the battery consumption until the device was able to be charged.

But, shouldn't there be a better way to preserve battery life then having to have the device and the apps run with half the features that it provides? Having applications and operating systems work together on an adaptation that has the application dynamically modify their activity to preserve energy, the operating system can then use the adaptation to have longer battery life on devices.

This paper will dive into a process called process offloading and how beneficial it is to a system.

Details of the Experiment:

Looking at a study done at Free University of Bozen-Bolzano in Italy, they analyzed the energy consumption of image processing applications by offloading the heavy computations to a remote environment. To analyze the energy consumption, they used two benchmarks: matrix multiplication and image filtering. They used different loads to simulate various real-world image processing and to show the energy spent with heavier sized computational jobs.

Matrix multiplication is where the input of the application is a text file with two matrices. If offloading does occur, all the bytes of the file are sent over the Internet. The system then reads the input file from the phone storage, and then converts the matrices from a text format into numerical two dimensional arrays where the dot product multiplication can be applied. The output is then a two-dimensional array of integers written to another text file stored on the phone.

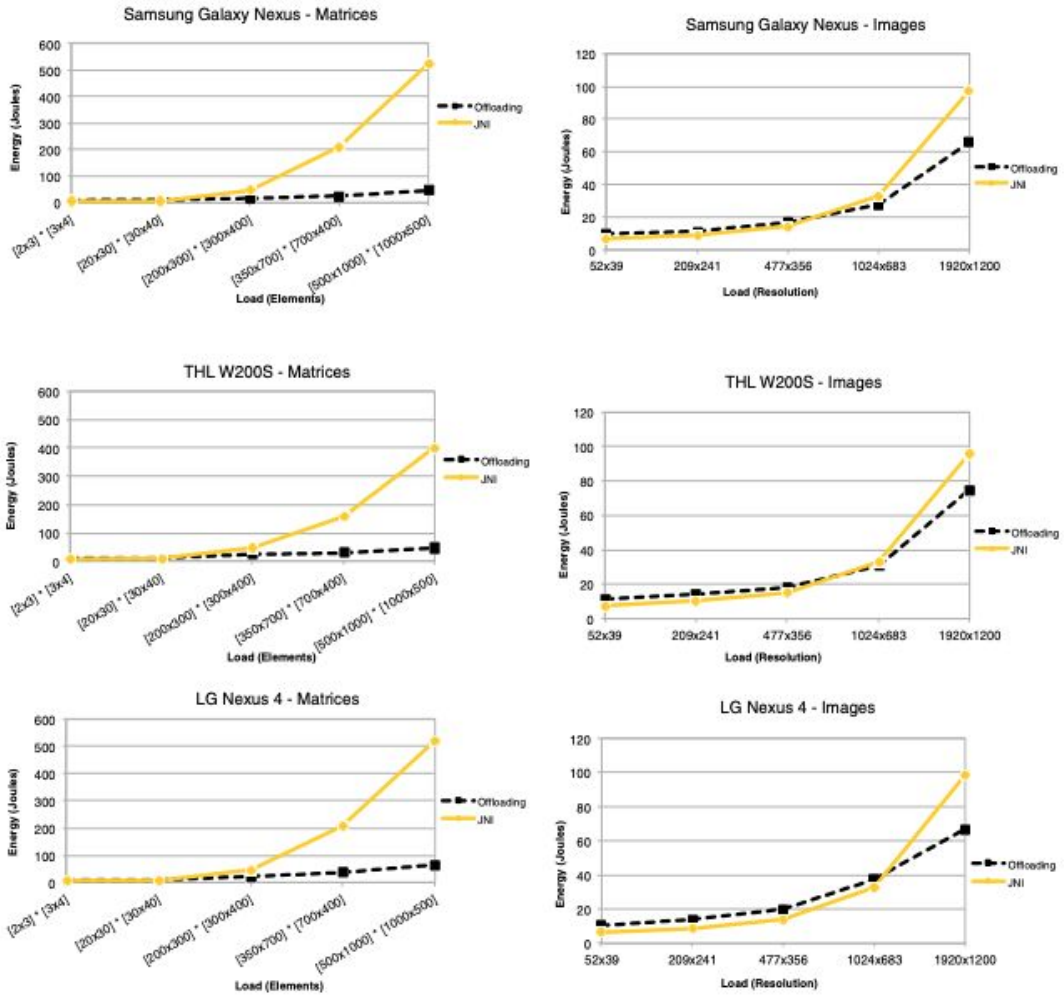
Image filtering is where the input of the application is an image file in PNG, JPG, or JPEG. If offloading occurs, then all the bytes of the file are sent over the Internet. The server reads the input file and applies the same algorithm with the same settings for filtering. The resulting image file is then stored on the phone.

The Set Up

PowerTutor, an open-source measurement application, was used to track the energy that the app consumed. The application used was an Android app that had offloading capabilities which was then tested on three different Android devices, the LG Nexus 4, Samsung Galaxy Nexus, and the THL 220W. The server-side application or better known as the remote environment, was set up using Java, and produced the same equivalent algorithms when the devices ran the algorithms on their operating system. The phones used the Wi-Fi capabilities to be able to establish a connection with the external infrastructure to be able to send and receive the needed files.They ran over 900 readings, all with different workloads to see how well process offloading would work.

Results:

The figure below shows the summary of the energy consumption of each test device for each benchmark.



From the graphs, it was concluded that offloading is not always the best case, especially with smaller loads. Offloading to a remote server over a Wi-Fi connection took the same amount of energy, calculated in Joules, as it would to just execute the process on the device. However, when the load was of larger size, offloading was very beneficial. Since the processor is free of the instructions with the heavier load, the amount of battery drained is lower because the device does not have to handle that big of a load.

Offloading the matrix multiplication is some cases, could save up to 800% of the energy consumed and the image filtering could save up to 30% of the energy consumed, which in the long run could save the device's battery, if the processes

are in the load size that saved that much energy consumption. Looking at the overall idea of offloading, it might not be beneficial in the long run. If applications had a certain load size they would start offloading at, it would be beneficial then, but having every app do that, with all different load sizes, it wouldn't be beneficial in every case since load sizes depend on each user.

Also, the results found in this study are highly variable. Since low-level kernel processes have precedence over application processes, each test done in this experiment has a different volume of energy drained from the battery. The low-level kernel processes running in the background may need some CPU usage at the same time that the application was running. Also, if there was noise in the radio waves of other hardware that interfered with the Wi-Fi connection, the rate of exchange between the remote device and the phone was highly variable as well.

Recommendations:

Based on this study, as well as other studies done at other universities around the world, I think that incorporating a few ideas together could potentially yield better results. Having energy-aware API's is a good starting point in application design. Having certain energy states defined in the application, like if the battery reaches a certain temperature or percentage, and then setting the application to a certain energy state when an energy event is triggered would allow applications to be more energy-conscious as a user is running the application. To be able to have energy-aware API's there does need to be an energy profiler that allows the developer to be able to have an energy estimator, and state collector of the battery.

Another addition the applications could have would be a way to determine what services are highly required, required, and optional. This would allow the operating system and application to stop using certain parts of the application or operating system that could possibly be consuming energy where it is not critical for it to be running at the time. Having applications be able to determine what to use and what not to use at a certain energy state would allow the application to be able to be more energy efficient. Since the operating system accounts for nearly half of the device's energy consumption, being able to turn off what is not

necessary for the application to run at the time, would be able to save the device energy once it reaches the low energy state.

Ontop of having the API's and the way to determine what is necessary to run, the application could then also offload processes to the remote server as another option in the states of the application. Since it did save energy consumption in the higher loads, having the application be able to have the threshold of the load limit where offloading is more beneficial to the system. That way, once it hits that heavy load, the system doesn't have to go into a low energy state in able to process the request on the application, instead it can send it to a remote server and still be able to run in a non-low energy state.

Combining these ideas into one application could be a tricky thing to do, and maybe not even be possible, but these three ideas seemed to be the most promising in the future of energy aware applications, and seemed to yield the best results.

Another idea to solve the ever growing battery demand is changing the battery itself. Operating systems on devices will always be changing to run faster and better than the last version, so shouldn't the battery be doing the same thing as well?

Lithium-ion is so commonly used, that it would be hard to replace it when it is in everything. The way to make the lithium-ion batteries more efficient is by having solid-state lithium-ion batteries. The difference between the solid state and regular lithium-ion batteries is that lithium-ion batteries use a liquid electrolytic solution where as the solid-state batteries use a solid electrolyte. The electrolytes in batteries serve as a chemical medium that allows the flow of electrical charge from the cathode to the anode. This is what gives the device it's flow of energy. Solid state also has higher energy densities, making it hold a longer charge, faster recharging times and is safer than lithium ion batteries. The solid state batteries charge in seven minutes, making it ideal for anyone who needs a fast charge on their device without having the time to let it charge. They also have a life cycle of 10 years, which is far superior than the 2 years that most phones offer today.

Cell phones have yet to incorporate these batteries into their devices for a couple reasons. Cost and dendrites are the biggest problems with solid state batteries. Dendrite is a build up of lithium metal that can grow throughout the battery. This reduces the capacity of the battery, and could even short circuit the battery which could potentially cause a fire.

Researchers from Columbia University have potentially found a way to stop the growth of dendrites in the solid state batteries. Having a 5 to 10nm layer of boron nitride isolates the lithium metal and the ionic conductor, which prevents the dendrite build up, and uses in a small amount of liquid electrolyte. As researchers get closer to fixing the dendrite problem as well as finding a way to lower the cost of building these batteries, these batteries could be better for mobile devices for their overall capacity, life span and charging speed.

Conclusion:

While researching this topic, it is evident that not much research has been done on how to make the applications we use everyday more efficient. Adapting an offloading process in applications can save the device energy, if the load is high enough to be beneficial to allow the processor the free space from the all of the instructions the higher load requires. The next best option that had the best results was having an energy-aware API implemented in applications that could potentially monitor and reduce their battery usage while using the application.

With mobile devices being the most important and biggest platform, the capabilities of devices continues to grow. But managing the energy consumption without diminishing the user's experience is one of the bigger challenges in mobile computing. User's want all the capabilities and features of their device without having to give anything up all while wanting the best battery life possible.

As software continues to grow and continues to add in features that require more and more battery to run, the typical battery capacity in a phone is not holding up for more and more users. For now, the best solution we have is the lower power modes that Apple and Androids have already in place in their operating systems. Until a better battery is introduced or until a better process to saving energy in applications has been developed and researched, the best thing app developers

can do is follow the energy efficiency guides released for iOS and Android apps to make their apps run at the best efficiency they can to have the best user experience with the application.

References:

Chen, Hui, et al. “Anole: A Case for Energy-Aware Mobile Application Design.” *IEEE Explore*, 2012, ieeexplore.ieee.org/document/6337485.

Corral, Luis, et al. “Analysis of Offloading as an Approach for Energy-Aware Applications on Android OS: A Case Study on Image Processing.” *Research Gate*, 2019, www.researchgate.net/profile/Alberto_Sillitti/publication/278701758_Analysis_of_Offloading_as_an_Approach_for_Energy-Aware_Applications_on_Android_OS_A_Case_Study_on_Image_Processing/links/57c695d208ae28c01d4de6da/Analysis-of-Offloading-as-an-Approach-for-Energy-Aware-Applications-on-Android-OS-A-Case-Study-on-Image-Processing.pdf.

Flinn, Jason, and M. Satyanarayanan. “Energy-Aware Adaptation for Mobile Applications.” *Carnegie Mellon University*, 1999, www.cs.cmu.edu/~satya/docdir/flinn-sosp-1999.pdf.

Noble, Brian D, et al. “A Programming Interface for Application-Aware Adaptation in Mobile Computing.” *Carnegie Mellon University*, 1995, www.cs.cmu.edu/~satya/docdir/noble-usenix-1995.pdf.

Vallina-Rodriquez, Narseo, and Jon Crowcroft. “Energy Management Techniques in Modern Mobile Handsets.” *UC Berkeley*, 2012, www1.icsi.berkeley.edu/~narseo/papers/ieee-energymgmtsurvey.pdf.