Forrest Smith
Vincent Weaver
ECE571
5/11/18

VMWOS v. Raspbian, a Performance and Energy Comparison

Introduction

The Raspberry Pi has a dynamic range of uses: it can be used as a primary computer, a home security device, a network hub for a server, and countless others. Since the Pi and its hardware are somewhat well documented, it is possible to develop custom software for the Pi outside of already existing operating systems like Raspbian or NOOBS. VMWOS is a custom operating system designed by Vincent Weaver for the Raspberry Pi. VMWOS will be benchmarked along side Raspbian in order to examine performance and energy differences between the two operating systems.

Testing will be performed using two custom benchmarks, one to stress the CPU and the other to stress memory. A run-length encoding benchmark was developed for the CPU end, and a preexisting benchmark from ECE598 was used for memory testing. The benchmarks both yielded surprising results: VMWOS tends to be more power efficient and perform faster for the developed benchmarks. The remainder of this report will: summarize some related work to this project, provide full details related to the experimental setup and programs used, list the results in full detail and provide analysis, and suggest some future work to be done on the project.

Related Work

In [1], Bekaroo and Santokhee investigate the power consumption of five different machines, one of which is a Raspberry Pi 2 Model B. They include measurements for loads connected to the network and offline, and they generally found that network loads used more power on the Pi.

He, Segee and Weaver compare GPU rendering and software rendering on a Raspberry Pi 2B+ in [2]. Their data was obtained using perf and a USB Power Gauge. [3] and [4] are non-scholarly related worked. [3] is a release article for the Pi 3 that gives mostly performance results but has one section on power draw. [4] is an article comparing the power usage of the Pi 2 and Pi 3 for both single core and multi-threaded tests. The article was written with both performance and power concerns in mind.

Experimental Setup

All measurements were performed using a Raspberry Pi Model 3B, which comes equipped with a quad-core 1.2GHz Broadcom BCM2837 and 1GB of DRAM. Version 9 of VMWOS was used; version 8 of Raspbian with kernel 4.9.35-v7+ was used. A Watts-Up? Pro power meter was used to take power measurements; the meter provides readings at a frequency of 1Hz. In order to interface with the meter, a utility called watts-up was downloaded from pyrovski's Github account. The utility was capable of printing the power measurements to terminal, so it's output was simply piped to a file for storage.

Both of the benchmarks that were used were custom and have been provided in the attached code. The first benchmark was intended to stress the CPU by using a compression algorithm, and the second was created for the purpose of stressing the Pi's memory. Initially, an attempt was made to use traditional benchmark slike bzip2, gzip, Stream, ot JPEG compression. Porting traditional benchmarks to VMWOS proved to be very challenging because VMWOS lacks the library support needed for most of the benchmarks. Custom libraries would need to be developed in order to support these benchmarks, and that was not in the scope of the project.

The compression algorithm that was ultimately chosen was run-length encoding because it was possible to implement it on VMWOS with the available library support. Run-length encoding is a form of losesless compression. It stores data in the form of “runs”, which are sequences where the same value appears consecutively. The “runs” are stored as the character and a count for how many characters were repeated. A simply example is compressing the string “RRRGGGBBB”; the run-length encoding is “3R3G3B”. The algorithm was first developed and tested using simply inputs like the on above, but once it was working, the input was changed to the dictionary. The dictionary file used was called “american-english” and was taken from the /usr/share/dict folder of a 16.04 LTS Ubuntu install. The benchmark was set to process the dictionary 250 time on both Raspbian and VMWOS. The number of runs to make on the dictionary was experimentally determined; the goal was to have each program run for at least 30 seconds in order to get at least 30 readings from the Watts-Up? Pro meter.

The memory benchmark that was chosen came from homework 7 in ECE598, Advanced Operating System; the benchmark is a 64-bit memory setting benchmark. The code for the custom 64-bit memory set was provided by Avery Rossow from his homework 7; it simply sets a specified memory location to a user passed value by writing 8 bytes at a time. The benchmark was tested to ensure it was writing to memory properly with a small amount and was then scaled up. The benchmark for testing was run to allocate/free a 16MB of chunk memory, 1024 times.

Results / Analysis

Both benchmarks were ran four times each on both VMWOS and Raspbian in order to ensure consistency in results. The averaged results can be seen in Table 1 below.

Benchmark	**Average Time**	**Average Power**	**Average Energy**	**Average Energy Delay**
Raspbian-Mem	143.142s	393.875W	56.38kJ	8.071MJ*s
Raspbian-RLE	121.574s	335.975W	40.85kJ	4.966MJ*s
VMWOS-Mem	32.871s	52.1W	1712.6J	56.295kJ*s
VMWOS-RLE	34.781s	53.4W	1857.3J	64.6kJ*s

Table 1. Average power and energy metrics for each benchmark

For both, benchmarks VMWOS seems to excel in nearly every way. Both benchmarks complete almost four times faster on VMWOS than on Raspbian. In addition, Raspbian is significantly more power and energy hungry than VMWOS in all cases. Raspbian consumed over 7 times more power than VMWOS for the memory benchmark and over 6 times more for the run-length encoding benchmark. Power data from a single run of each benchmark has been provided in Figure 1 below.

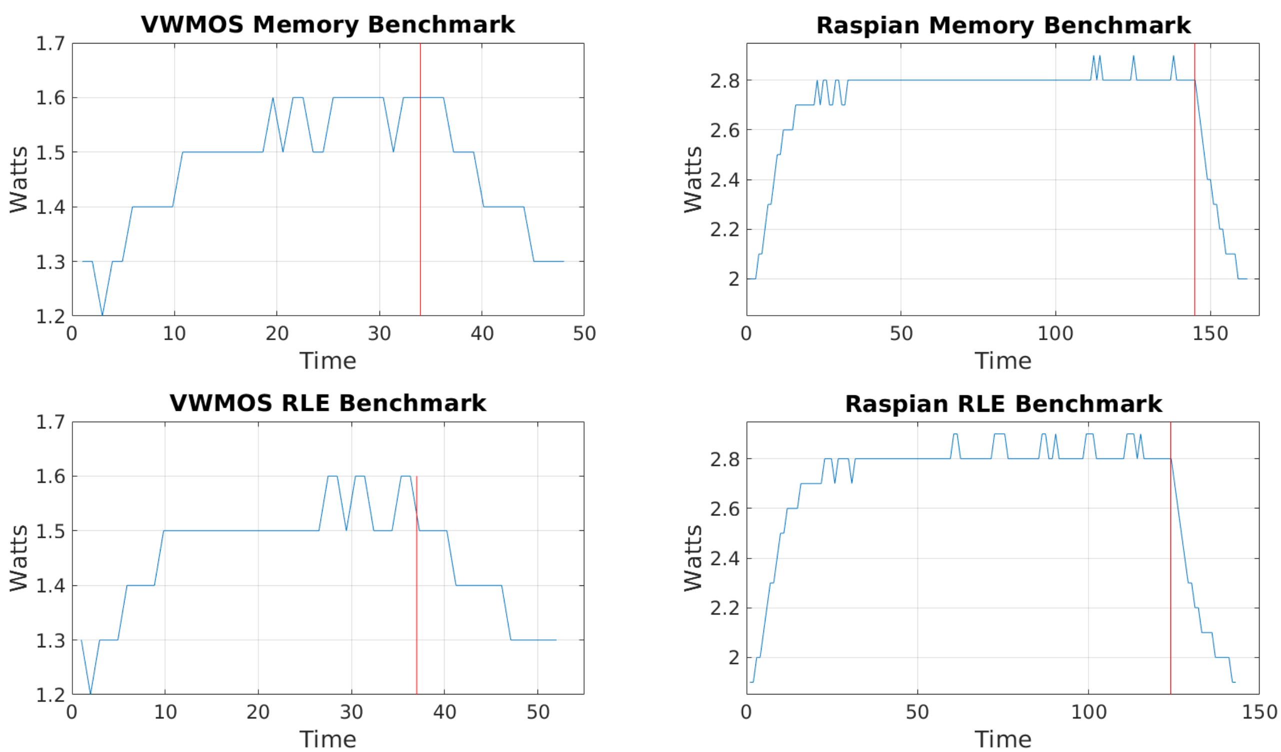

Figure 1: Power consumption examples for each benchmark

All the graphs in Figure 1 have been marked with a red line to indicate the time when the benchmark completes. The run-length encoding and memory benchmarks share similar power trends across operating systems. Raspbian tends to idle around 2.0W slightly higher than the 1.3W from VMWOS. As Raspbian runs programs, its power consumption ramps up as much as a watt more to around 3W, where as VMWOS tops out around 1.7W. Figure 2 provides a comparison of the benchmarks’ power data for each operating system.

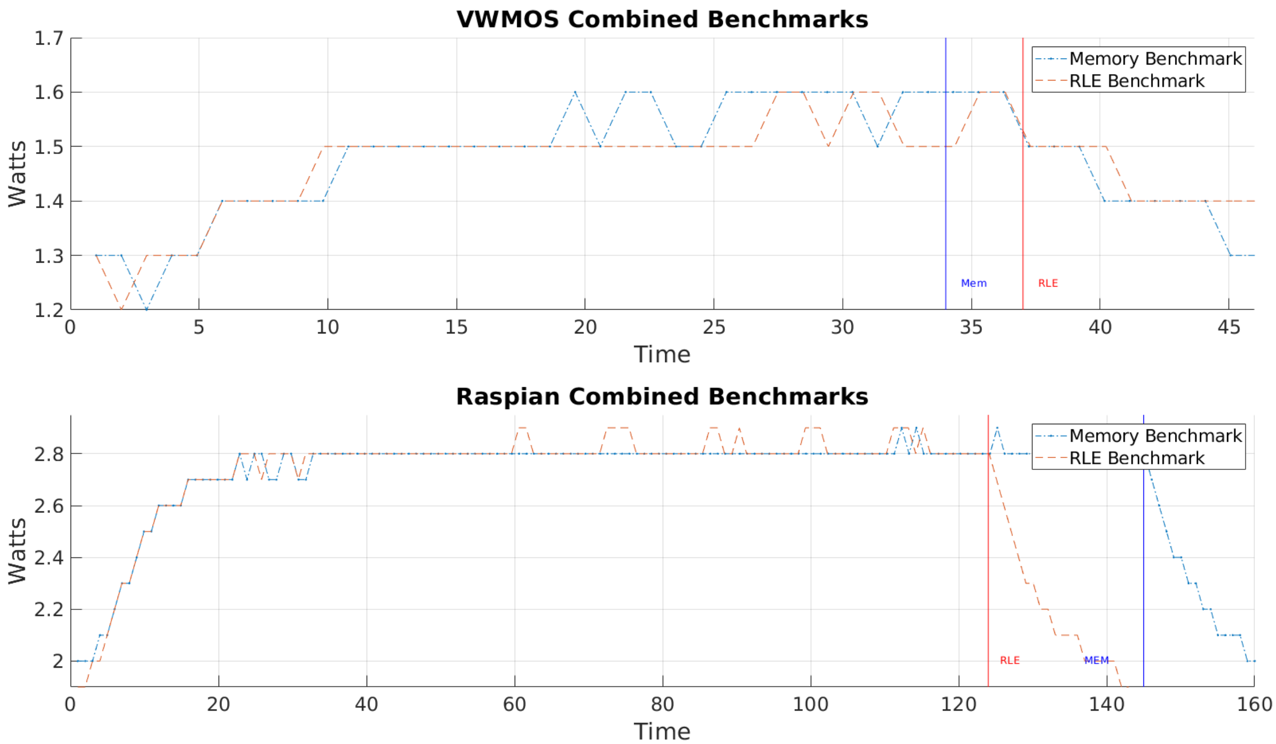

Figure 2. Combined power results for each operating system

As far as power is concerned, both benchmarks seem to stress each operating system about the same amount with concerns to power. VMWOS idles around 1.2W and slowly rises to 1.6W for both benchmarks; Raspbian idles around 2W and slowly rises to around 2.9W for each benchmark. VMWOS runs the memory benchmark quicker than the run-length encoding benchmark, and Raspbian actually has the opposite behavior.

The most surprising result is how much better VMWOS performs with regard to both power and performance. VMWOS is a much more simple operating system than Raspbian; it does not take full advantage of all the hardware on the Pi nor does it implement full virtual memory. Raspbian is constantly interfacing with Ethernet/Wifi, Bluetooth, and USB ports that VMWOS does not access. Raspbian also implements fully functioning simultaneous multiprocessing, which VMWOS does not currently support. The large overhead present in Raspbian is where the discrepancy in power consumption stems from; however, the difference in performance between the two operating systems is still a puzzling result.

Future Work / Conclusion

Adding more benchmarks would improve the project. In addition, trying to provide library support on VMWOS for more traditional benchmarks like bzip2 or stream would likely give more reliable performance performance results. Another future consideration would be to rerun the benchmarks in Raspbian while observing the frequency of each core. It is possible Raspbian is scaling down the processing speed of the benchmarks due to heat or other workloads. VMWOS would not suffer from this because it does not perform frequency scaling.

VMWOS and Raspbian have been compared using a run-length encoding and memory benchmark. It was found that VMWOS tends to be more power and energy efficient while also being faster than Raspbian for the benchmarks that were designed.

Related Work

[1] G. Bekaroo and A. Santokhee, "Power consumption of the Raspberry Pi: A comparative analysis," 2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech), Balaclava, 2016, pp. 361-366.

[2] Q. He, B. Segee and V. Weaver, "Raspberry Pi 2 B+ GPU Power, Performance, and Energy Implications," 2016 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, 2016, pp. 163-167.

[3] The MagPi Magazine. "Raspberry Pi 3B+ Specs and Benchmarks" The Magpi Magazine. raspberrypi.org 17 Mar. 2018. Web. 16 April 2018

[4] RasPi.TV. "How Much Power Does Raspberry Pi3B Use? How Fast Is It Compared to Pi2B?" raspi.tv 26 Feb. 2016. Web. 16 April 2018