a little
RESEARCH
https://www.alittleresearch.com.au

Power Consumption of the Raspberry Pi

Nicholas Paul Sheppard

First prepared 21 June 2019. Revised 31 March 2020.

Abstract

This report reviews published research related to the power consumption of the Raspberry Pi single-board computer. Section 1 reviews tools and techniques for measuring the power consumption of a Pi. Section 2 then reviews strategies by which the power consumption of a Pi may be reduced.

1 Measuring Power Consumption

The overall power consumption of any computer can of course be measured using a power meter attached to the power supply of the computer. A *power model*, however, allows more sophisticated analysis by relating the overall power consumption of the computer to the utilisation of each individual component within the computer. An accurate power model allows developers to estimate the power consumed by each component within the computer, and to therefore tune the usage of each component so as to minimise the computer's overall power consumption.

Fabian Kaup and colleagues published power models for the Raspberry Pi Model B in 2014 [7] and Raspberry Pi 2 and Raspberry Pi 3 in 2018 [8] (the latter paper also includes power models for Cubictruck and Odroid devices). They construct each model by connecting the computer to a power meter, measuring its power consumption in response to a series of synthetic loads designed to utilise each component at different levels, then using standard statistical techniques to derive a function that describes the overall power consumption in terms of the utilisation of each component.

The models derived by Kaup and colleagues take the form of a linear combination of the energy consumed by the CPU and the two network interfaces (Ethernet and WiFi):

$$P_{pi} = P_{idle} + P_{CPU}(u) + \sum_{if} (P_{if,idle} + P_{if,up}(r) + P_{if,down}(r))$$

where u is the CPU utilisation $(0.0 - 1.0)$, r is the rate of network traffic (bits per second) and if represents each of the network interfaces. The constants P_{idle} and $P_{if,idle}$ represent the power consumed while the Pi and each network interface is idle, respectively, while the function $P_{CPU}(u)$ represents the power consumed by the CPU at utilisation u and $P_{if,up}(r)$ and $P_{if,down}(r)$ represent the power consumed by the network interface if while uploading (i.e. transmitting) and downloading (i.e. receiving) data at rate r, respectively. Kaup and colleagues give values for each of these constants and functions for each computer that they tested in their 2018 paper, and report that their model calculated values between 2.5% and 9.0% of the actual value in testing with video streaming software.

Luca Ardito and Marco Torchiano published another power model for the Raspberry Pi 2 Model B in 2018 [3]. They use much the same approach as Kaup and colleagues, but assume that all of the power functions are linear, so that they can be derived by linear regression from the experimental data. Observing that the energy consumption of the Ethernet devices seems to change signifcantly at around 50 Mb/s, however, they model $P_{eth,up}(r)$ and $P_{eth,down}(r)$ as two-part linear functions (Kaup and colleagues use quadratic functions). They report that their model calculated values within about 2.5% of the actual value in testing with several kinds of software.

Rather than build a power model in terms of the components of the computer, the same hardware configuration can be used to measure the power consumption of the computer as a whole while performing various tasks of interest. Girish Bekaroo and Aditya Santokhee published measurements of the Raspberry Pi 2 Model B while performing tasks such starting up, copying a file, viewing web pages, and so on, in 2016 [5]. While they do not explicitly derive a power model, it is easy to see how one might combine such measurements with knowledge of the usage pattern of a Pi in order to estimate its overall power consumption depending on the number of times it is restarted, the number of files copied, and so on.

2 Minimising Power Consumption

Given an accurate power model of a computer of the sort discussed above, it should in principle be possible to optimise the computer's power consumption by optimising the use of each component according to the model (for example, choosing the most energy-efficient network interface for the expected load or avoiding power-hungry applications) but doing so might require quite complex and difficult analysis of the individual application for which the computer is to be used.

Some general strategies for reducing the energy consumption of the Raspberry Pi have been published between 2016 and 2019, each focusing on a different application and taking a different approach. These will be described in turn in the following.

Immanuel Amirtharaj and colleagues [2] focus on saving energy through a technique known as *duty-cycling*, in which the device is powered on only when necessary. The efficiency of this technique depends on the efficiency with which the device can be started and shut down, so Amirtharaj and colleagues investigate methods by which the power consumed by each of these phases can be reduced. They use a hardware configuration similar to that described in Section 1 to estimate the power consumption of each unit during the startup and shutdown phases of a Raspberry Pi 3 and Raspberry Pi Zero, with results indicating that the power consumption of each can be reduced up to around 50% by disabling units that are unnecessary for the application (the actual reduction will obviously vary according to the application and the units that it requires). They additionally describe a technique called *Pallex* that reduces the startup time (and therefore the power consumption of duty-cycling) by allowing some user programs to begin executing while the startup procedure is still in progress; they say that Pallex reduced the power consumed by duty-cycling by about 30% for the Pi 3 and 9% for the Pi Zero. Further background can be found in Immanuel Amirtharaj's Masters thesis [1].

Fabián Astudillo-Salinas and colleagues [4] report on the reduction in power consumption of a Raspberry Pi 3 after various alterations to its hardware and software configuration. They first observe that the Raspbian Lite and Picore distributions result in slightly lower power consumption than the full-featured Raspbian and Pidora distributions. Disabling the HDMI controller reduced energy consumption by about 8% and disconnecting the Ethernet cable reduced it by about 19%; while reducing the SDRAM clock frequency reduced power consumption by about 10%, reducing the GPU clock frequency reduced it by about 7%, and reducing the CPU clock frequency reduced it about 5%.

Javier Corral-García and colleagues [6] investigate the impact of twenty-five different optimisations applied to C and C++ code on the performance of a Raspberry Pi 3 computer, as well as the impact of the optimisations applied by the GCC compiler. They say that the run-time (and therefore power consumption, though they did not directly measure this) of a program can be significantly reduced by seventeen out of the twenty-five manual optimisations, and that eight of the optimisations significantly reduced run-time even compared to GCC's optimisations. Removing repeated function calls within a loop, for example, reduced the run-time of their test program by 77% (even after compiler optimisations). Many of these optimisations would be well-known to proficient programmers, however, so the benefit of applying them may vary considerably according to the quality of the original code.

3 Conclusion

Experiments conducted by the researchers cited in this report suggest that the power consumption of a Pi may be significantly reduced by relatively simple strategies such as disabling units not required by the application, installing only the minimum necessary software, and applying well-known techniques for improving the run-time of code. The actual reductions would obviously vary significantly with the nature of the application.

For more complex analysis, Fabian Kaup and colleagues have demonstrated a procedure for building a power model of the Raspberry Pi and other computers, and published power models for the three versions of the Raspberry Pi released between 2012 and the present time. These power models can in principle be used in combination with a profiler or similar tool to tune the power consumption of a Pi though the analysis involved may be quite complex.

References

[1] I. Amirtharaj. Energy measurement and profiling of Internet of Things devices. Master's thesis, Computer Engineering, Sanata Clara University, 27 June 2018.

[2] I. Amirtharaj, T. Groot, and B. Dezfouli. Profiling and improving the duty-cycling performance of Linux-based IoT devices. *Journal of Ambient Intelligence and Humanized Computing*, 2019.

[3] L. Ardito and M. Torchiano. Creating and evaluating a software power model for Linux single board computers. In *Proceedings of the 6th International Workshop on Green and Sustainable Software*, pages 1–8, Gothenburg, Sweden, 27 May 2018.

[4] F. Astudillo-Salinas, D. Barrera-Salamea, A. Vázquez-Rodas, and L. Solano-Quinde. Minimizing the power consumption in Raspberry Pi to use as a remote WSN gateway. In *2016 8th*

IEEE Latin-American Conference on Communications (LATINCOM), Medellin, Colombia, 15-17 Noveber 2016.

[5] G. Bekaroo and A. Santokhee. Power consumption of the Raspberry Pi: A comparative analysis. In *2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech)*, Balaclava, Mauritius, 3-6 August 2016.

[6] J. Corral-García, J.-L. González-Sánchez, and M.-Á. Pérez-Toledano. Evaluation of strategies for the development of efficient code for Raspberry Pi devices. *Sensors*, 18, 2018.

[7] F. Kaup, P. Gottschling, and D. Hauseer. PowerPi: Measuring and modeling the power consumption of the Raspberry Pi. In *39th Annual IEEE Conference on Local Computer Networks*, pages 236–243, Edmonton, Alberta, 8-11 September 2014.

[8] F. Kaup, S. Hacker, E. Mentzendorff, C. Meurisch, and D. Hausheer. Energy models for NFV and service provisioning on fog nodes. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, Taipei, Taiwan, 23-27 April 2018.