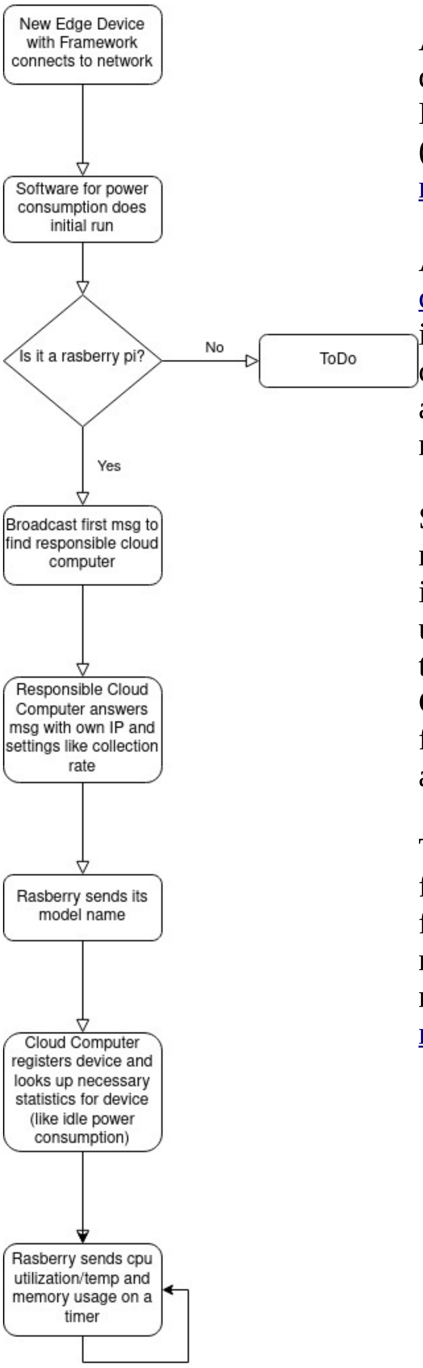
Energy Consumption Tracking on Edge Devices

New Discoveries:

The basic first was to measure the cpu frequency and using the cpu utilization to estimate the power consumption. Basically researching the average power consumption for a given processor and just applying that value to the time a processor is running vs when its idle.
The big problem is that there are a lot more factors outside of cpu frequency that affect the energy consumption of a microprocessor, especially ones with more capability like as rasberry pi.

There are two sources I found (https://www.pidramble.com/wiki/benchmarks/power-consumption, https://raspi.tv/2018/how-much-power-does-raspberry-pi-3b-use-power-measurements), where individuals measured different versions of the rasberry pi and got widely varying results based upon the implementation.



Additionally for the rasberry pi the linux system I use doesn't collect the frequency the processor is running at. Normally it can be found in the *proc/cpuinfo file* (https://serverfault.com/questions/179481/how-to-mearsure-the-average-cpu-frequency-in-linux).

Also according to this paper (http://www.jmest.org/wp-content/uploads/JMESTN42352081.pdf), that looks at instruction level events and compares them to power consumption, it makes sense to choose a variety of statistics and compare them to the power consumption, to built a model.

So the current idea is to collect a variety of metrics regarding cpu utilization, wait times for I/O operations, interrupts and also idle time. Then these metrics can be used in tandem with energy measurements to build a model to accurately estimate the consumption of a given device. Currently I am building a general approach, but we can focus, as you mentioned, on a certain application, once we are happy with basic model/implementation.

The energy measurements I'll be using will be the two former mentioned ones. Additionally I will look out for further sources. But the end goal is to get a multi-meter and measure the power consumption myself to built an accurate model, like here (https://github.com/David00/rpi-power-monitor).

Current Status:

Currently I implemented most of the flow diagram. My rasberry pi is sending an initial udp message with its serial number. The server is receiving that message. The pi is collecting information over a given period of time and then sends it to the server. Also I made a list of all rasberry pies with their architecture and their power consumption under various loads. All my code, tables, sources and diagrams can be found at the github link at the bottom.

Whats missing from flow diagram:

The server has to register the rasberry in a database, so it can accurately estimate the power consumption for the given model, according to my table. Some small life improvements like remotely activating the pi program. After this step is done I will resume doing the same for the arduino and esp32. I made a ticket system I will link at the bottom, where you can track my progress.

Github:
https://github.com/OscarLange/EnergyConsumptionEdgeDevice
https://github.com/OscarLange/EnergyConsumptionFramework-RasberryPi
https://github.com/OscarLange/EnergyConsumptionFramework-Server
https://github.com/OscarLange/EnergyConsumptionFramework-Arduino

Tickets:
https://app.clickup.com/36706807/v/b/li/174663701?pr=54794232

Power measurement model:

Nomenclature

Low level => Architectural Level, Logic Gates, Transistor Level
High Level => Instruction Level, Functional Level
Cost => Energy consumed by an instruction
Inter instruction => Time between two instructions

Energy Consumption Measurements

Measuring power draw:

- Precision resistor connected to the cpu core, memory power line
- Multi-meter connected to the entire system

Measuring power consuming actions:

- Instructions
- Inter-Instruction Costs
- Architecture (Number of registers, size of operation operands, CISC/RISC)
- Memory model and usage (e.g. Cache Misses)
- IO operations
- Interrupts

Model:

Statistical => needs large amount of data (mostly regression analysis, cpu/controller as blackbox)

Non-statistical => Incredibly complex relations between actions and power draw, especially for higher level analysis

Different Examples:

Power Analysis of Embedded Software: (https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.51.6929&rep=rep1&type=pdf)

Motiviation:
Architecture is complex and there are many co-factors such as cache misses (bottom-up is not accurate)
Embedded system users often don't have the information to all underlying architecture

Basic Formular:

Energy Consumption = Average Power Consumption * Time

Average Power Consumption = Average Current * Supply Voltage

Method:
Instruction Level Modeling

- Instruction Pipeline
- Interrupts (Cache miss)
- Measure instruction cost in loop and use as average
  - => Assumption only 1 instruction concurrently
    - => Measurements even hold with staggered pipeline
- Cycle based
- Miss-aligned memory read
- Interinstruction cost negligent for the used system but might be important for other architectures
- buffer stalls (pre-fetch)
- Cache miss and stalls => 3% cost
- Memory modeling => Page switch, need for dynamic profiling

Tipps for myself:
Top-Down Approach (measure instruction sequences, group them)
Short programs can be looped to obtain accurate measurements
Memory measurement needs dynamic modeling as its hard to estimate

Cycle-Accurate Energy Measurement and Characterization With a Case Study of the ARM7TDMI https://ieeexplore.ieee.org/document/994992:

Using charged capacitors and switching them energy is measured
=> To much effort in the electrical engineering aspect
=> rather use a multimeter

Measurements Analysis of the Software-Related Power Consumption in Microprocessors https://www.researchgate.net/publication/220408258_Measurements_analysis_of_the_software-related_power_consumption_in_microprocessors:

Method:

Two categories of models

Physical measurement based

Simulation based models

Linear regression (Size of operand => Energy Consumption)

Tipps for me:

Energy sensitive factors are independent on the actual instruction and can be added

Instructions using the same hardware (add alu, registers of certain size)

=> shift amount, register operand, immediate operand

=> number of used registers

Energy consumption estimation in embedded systems https://www.researchgate.net/publication/3094221_Energy_Consumption_Estimation_in_Embedded_Systems

Method:

Logging program

Cycles instead of instructions

Polynomial expression

External Ram, A/D converter, Microcontroller => each measured independently

Cycles, read/write of memory

Tipps for me:

Main driver is the energy consumption from read/write access

An Accurate Instruction-Level Energy Estimation
Model and Tool for Embedded Systems
https://www.researchgate.net/publication/260304082_An_Accurate_Instruction-Level_Energy_Estimation_Model_and_Tool_for_Embedded_Systems

Method:
Considers cpu, memory controller and sram
instruction opcode, shift operations, register bit flips, instruction weight, mem access
embedded model parameters in instruction level profiler
simulation model difficult for microcontroller (no good simulation model e.g. SPICE)
black box approach
1-ohm resistor at power supply pin
disable not used parts
split program in init and main loop which is analysed
Energy => fetch, decode, execute, static

Memory:
    Flash read and write (mostly read as write is done in offline phase)
CPU:
    base energy, interinstruction, pipeline stall
    depends on type of instruction
    change in input signal as hamming way distance
    instruction word weight => 1
    shift operations
    register bank bit flips
Static energy consumption:
    Can be ignored as it is constant over time and added at the end

regression model