Software energy usage reporting

16.01.2019

Why?

- World is facing global warming, ecological and energy crisis. I believe that everyone should feel responsible and take action to help to solve the problems.
- Every watt counts and every energy unit (let it be Joules or kWh) spent is important. To start solving the problem we have to make the proble m visible first. The goal is to create open source tools and solutions that could be used by developers and integrated to CI for continues monitoring.

Example

- 1 W electricity consumer
- 8.8 kWh consumed per year
  - 50 km drive with electric car
- 15 kg of oil shale has to be burned
- 1 EUR costs the electricity
- 8.8 W solar panels needed to produce the same energy
  - Doesn’t take into account storage need due to day/night production and usage
- 15 EUR costs the PV-panels with installation
- 3W LED lamp 1h per day => 1 kWh / year

Plan

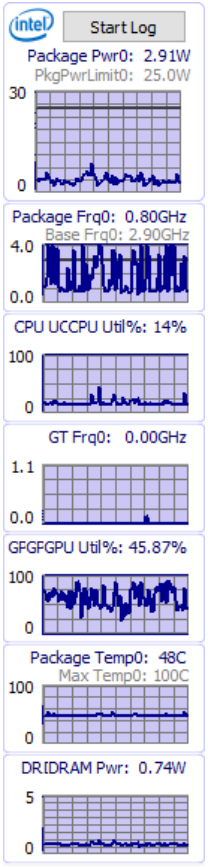
- Figure out what data and metrics we need to take account.
  - Figure out what tools are needed to measure the energy usage in different scenarios.
  - Create tools (software and hardware) to measure the energy usage.
  - Tools which could be integrated to automated and end to end testing.
  - Tools which could be integrate to clients (web and native) to collect statistics continuously and make the results visible in graphs.
  - Introduce web browser plugin and browser API proposal to capture energy used by web page in browser tab since start.
  - Tools to be open sourced.

Metrics

- Measure only software stack impact, don't take into account monitor or other external devices energy usage which we can't control
- Take operating system and other applications power usage as baseline and find out how much running application adds to the usage.
- We need to measure power in mW units scale.

Existing tools

- Intel® Power Gadget, measures Intel CPU + integrated GPU usage API to query values or cmd-line to automate logging to file. https://software.intel.com/en-us/articles/intel-power-gadget-20
- Use Windows battery report in mWh units to calculate power usage (might take time to run tests and get results in mW units) https://docs.microsoft.com/en-us/uwp/api/Windows.Devices.Power.BatteryReport
- Use external HW based power meter



HW tools

- There are expensive tools available on the market.
- We need power meters which could be connected to PC for automated data acquisition.
- DIY
  EVALSTPM32 http://www.st.com/content/st_com/en/products/evaluation-tools/solution-evaluation-tools/energy-and-smartgrid-solution-eval-boards/evalstpm32.html
  + Isolated 3.3V power
  + https://github.com/mzeitler/openstrom

Proposal

- Direct power measurement is not always available
- We can measure CPU and GPU relative usage
- We have to measure power usage for different CPU and GPU's and correlate it with relative CPU and GPU usage
- Build up reverse lookup table to get actual power usage from CPU and GPU information

Math

- https://en.wikipedia.org/wiki/CPU_power_dissipation
- $P_{CPU} = P_{leak} + CV^2 f \Rightarrow$
- $\begin{cases} E_{app} \approx \sum(b_{CPU} f t_{app} + c_{CPU} f^2 t_{app}) \\ N_{cycles} = t_{app} f \end{cases} \Rightarrow$
- $E_{app} \approx \sum(b_{CPU} N_{cycles} + c_{CPU} N_{cycles} f) \Rightarrow$
- $E_{app} \approx b_{CPU} \sum N_{cycles} + c_{CPU} \sum N_{cycles} f \Rightarrow$
- $E_{app} \approx b_{CPU} S_1 + c_{CPU} S_2$
- We need to collect CPU type (mapping => b_{CPU}, c_{CPU}), S_1 and S_2

GPU usage options

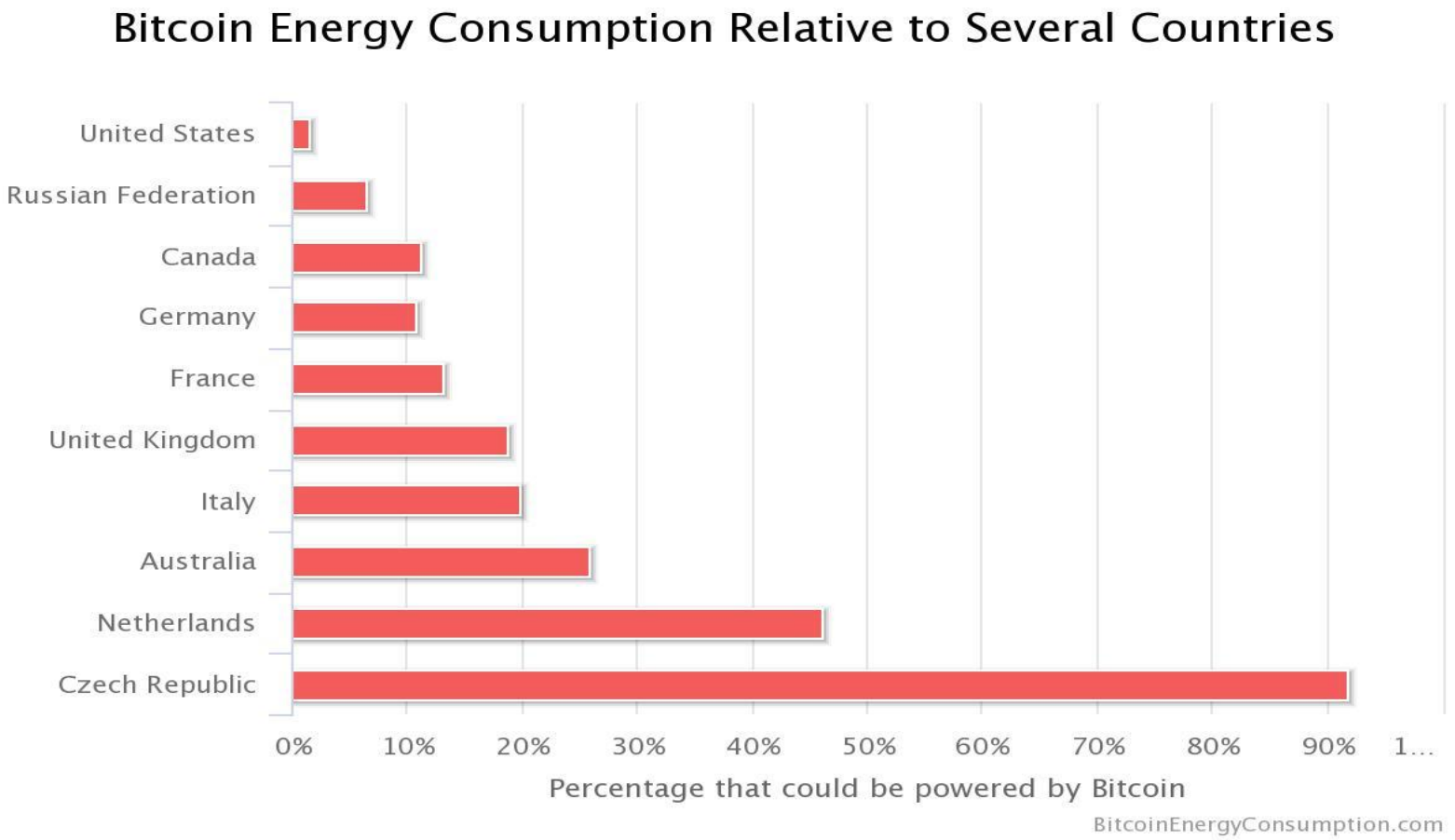
- Browsing Performance Counters https://msdn.microsoft.com/en-us/library/windows/desktop/aa371886(v=vs.85).aspx
- D3DKMTQueryStatistics_T https://forum.sysinternals.com/d3dkmtquerystatistics-t-buggy-on-multigpu-system_topic32543.html
- GPU relative usage available in these tools
  - Win10 ver 1709 Task Manager adds relative GPU usage stats, but not power usage
  - SysInternals Process Explorer

CPU usage

- Lots of options to query CPU usage per process in cycles and CPU time units.
- Need to query CPU frequency too, to calculate actual power usage from CPU usage

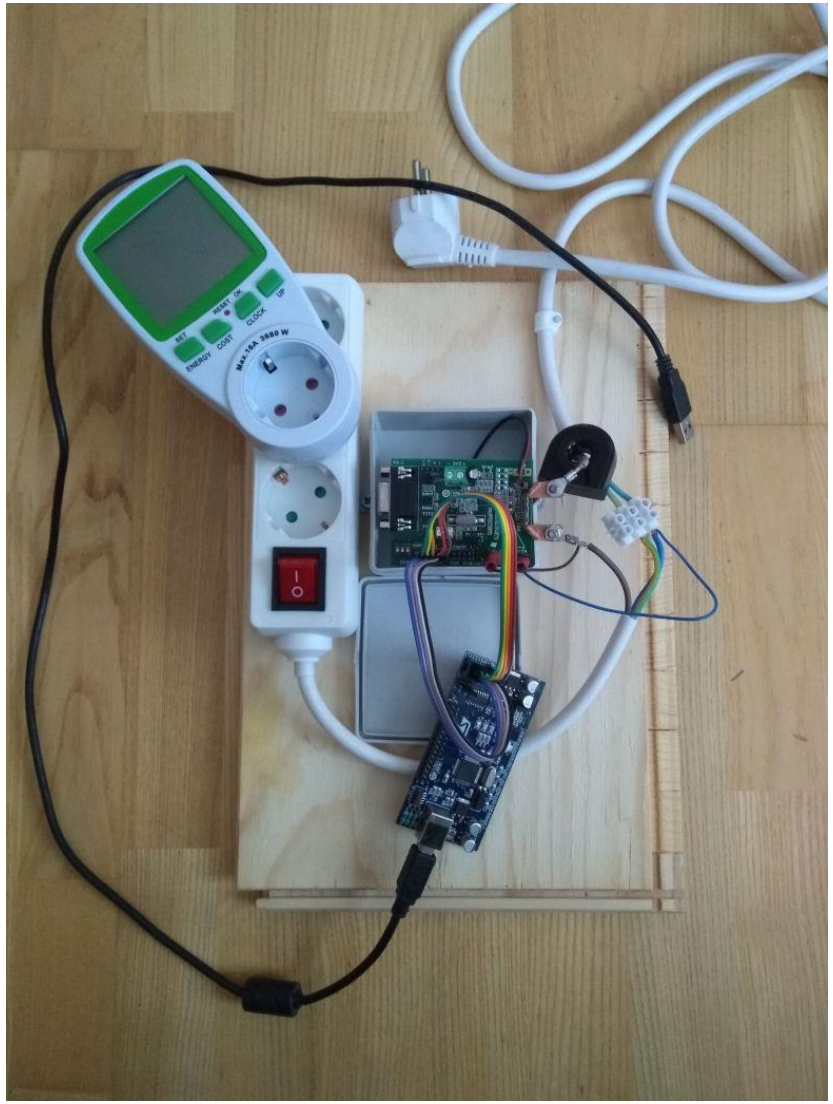
Bitcoin energy usage

- 61 TWh/year as of 23rd of April 2018



Lessons learned

- With IPG it is easier to save each test logs to separate file and read summary from log
- IPG causes video driver instability and PC restart is needed to use some software
- With EVALSTPM32 don't assume that shunt sensor is connected to L-line and not to N.
- HW built but SW not complete yet ->



TODO

- Automate measurement taking
- Complete HW
- Take more measurements with different CPU/GPU combinations
- Expose energy usage measurement results via performance counters

Sub projects

Calibration

- Tool for
  - CPU, GPU and other components fingerprinting
  - Using external HW for measuring correlation coefficients on different machines and platforms

Measurement library

- API for OS, process, thread level(Windows, Mac OSX, Chrome plugin)
- External tool(Windows, Mac OSX)
- Investigate PAPI (http://icl.cs.utk.edu/papi/index.html) PCM (https://github.com/opcm/pcm) To use and contribute?

Reporting Telemetry

- CSV, JSON, XML, TXT etc

API available

	OS	Process	Thread
Windows	GetProcessorSystemCycleTime()	QueryProcessCycleTime()	QueryThreadCycleTime()
	CallNtPowerInformation()		

Results

- PAPI direct usage no-go as Windows support was removed long time ago
- MS Joulemeter, integrated into Visual Studio
- Open PCM – Intel provided library for reading power values directly from CPU if supported. Requires kernel level driver and admin rights
- Open Hardware Monitor – uses WinRing0.sys driver and admin rights to get access to low level CPU registers
- psutil – multi platform Python library
- gperftool – Google provided performance monitoring tools