Carbon measurement and energy attribution for processes and hardware devices in Linux

Aditya Manglik
ETH Zürich, Switzerland

`LinkedIn: linkedin.com/in/adityamanglik/`
`Email: amangli@ethz.ch`

FOSDEM-2024
3 February, 2024

Brief Introduction

Graduate student at ETH Zürich, Switzerland

Brief Introduction

Graduate student at ETH Zürich, Switzerland

Research at the intersection of computer architecture and operating systems

Outline

Background

- Energy sources in computing systems:

  Direct: DC input / USB / Ethernet

Background

- Energy sources in computing systems:
  - Direct: DC input / USB / Ethernet
  - Battery

Background

- Energy sources in computing systems:
  - Direct: DC input / USB / Ethernet
  - Battery
  - Energy harvesting

Background

- Energy sources in computing systems:
  - Direct: DC input / USB / Ethernet
  - Battery
  - Energy harvesting
- We want to use the ~~maximum~~ minimum amount of energy to perform computation

Background

- Energy sources in computing systems:
  - Direct: DC input / USB / Ethernet
  - Battery
  - Energy harvesting
- We want to use the ~~maximum~~ minimum amount of energy to perform computation
- Battery capacity is a major design constraint and UX aspect for any consumer device: cellphones and AR/VR headsets

Outline

Calculating Energy Consumption of Software

$$\text{Energy Consumption} = \text{Power} \times \text{Latency}$$

Calculating Energy Consumption of Software

Energy Consumption = Power $\times$ Latency

Power is determined by hardware

Calculating Energy Consumption of Software

Energy Consumption $=$ Power $\times$ Latency

Power is determined by hardware

Latency is determined by software

Calculating Energy Consumption of Software

Energy Consumption $=$ Power $\times$ Latency

Power is determined by hardware

Latency is determined by software

Programmers often optimize latency using well-established tools (e.g., perf) and metrics (e.g., CPU clock cycles)

Calculating Energy Consumption of Software

Energy Consumption $=$ Power $\times$ Latency

Power is determined by hardware

Latency is determined by software

Programmers often optimize latency using well-established tools (e.g., perf) and metrics (e.g., CPU clock cycles)

Question: Tools to measure application's energy?

Calculating Energy Consumption of Software

$$\text{Energy Consumption} = \text{Power} \times \text{Latency}$$

Calculating Energy Consumption of Software

Energy Consumption = Power × Latency

Power is reported by the CPU (e.g., RAPL interface)

Calculating Energy Consumption of Software

Energy Consumption $=$ Power $\times$ Latency

Power is reported by the CPU (e.g., RAPL interface)

Example: CPU $\approx$ 15 W

Calculating Energy Consumption of Software

Energy Consumption $=$ Power $\times$ Latency

Power is reported by the CPU (e.g., RAPL interface)

Example: CPU $\approx$ 15 W

Latency is determined by software

Calculating Energy Consumption of Software

Energy Consumption = Power $\times$ Latency

Power is reported by the CPU (e.g., RAPL interface)

Example: CPU $\approx$ 15 W

Latency is determined by software

Example: Application X $\approx$ 5 ms

Calculating Energy Consumption of Software

Energy Consumption = Power × Latency

Power is reported by the CPU (e.g., RAPL interface)

Example: CPU ≈ 15 W

Latency is determined by software

Example: Application X ≈ 5 ms

Energy Consumption = 15 W × 5 ms = 75 mJ

Calculating Energy Consumption of Software

Energy Consumption = Power $\times$ Latency

Power is reported by the CPU (e.g., RAPL interface)

Example: CPU $\approx$ 15 W

Latency is determined by software

Example: Application X $\approx$ 5 ms

Energy Consumption = 15 W $\times$ 5 ms = 75 mJ

Problem: Does not reflect ground reality!

Calculation Model

- The model assumes linear power draw

Calculation Model

- The model assumes linear power draw

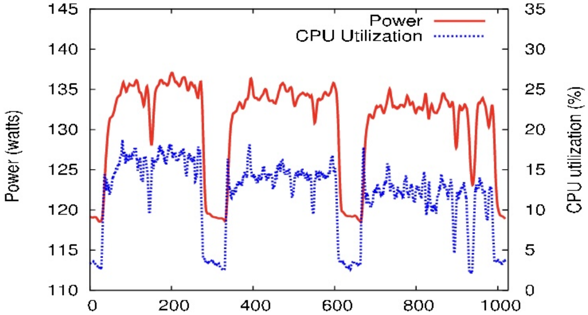


Figure: CPU Power Consumption over time

Calculation Model

- The model assumes linear power draw

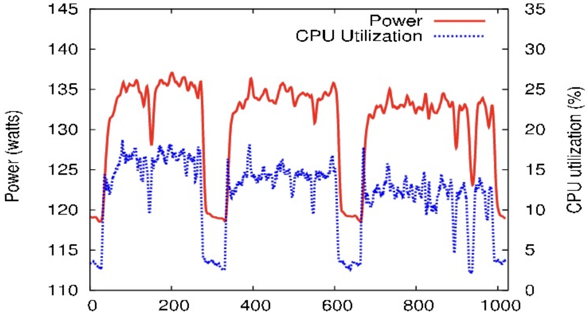


Figure: CPU Power Consumption over time

- **Limitation 1**: Power consumption (on y-axis) is not linear over time (on x-axis)

Ground Truth

- Platform-specific interfaces: RAPL is available only on Intel

Ground Truth

- Platform-specific interfaces: RAPL is available only on Intel
- AMD and ARM have different interfaces

Ground Truth

- Platform-specific interfaces: RAPL is available only on Intel
- AMD and ARM have different interfaces

- **Limitation 2**: We do not have uniform interfaces and formats needed to measure power reliably across different platforms

Calculation Model

- The model focuses on the CPU

Calculation Model

- The model focuses on the CPU
- **Limitation 3**: What about devices like memory (DRAM), screen, and network cards?

Calculation Model

- The model focuses on the CPU
- **Limitation 3**: What about devices like memory (DRAM), screen, and network cards?
- Experiments are contrary to assumptions, findings similar to Google [1]

  [1] Barroso, Luiz André, Urs Hölzle, and Parthasarathy Ranganathan. "The datacenter as a computer: Designing warehouse-scale machines." Synthesis Lectures on Computer Architecture 13.3 (2018): i-189.

Problem Summary

- We are *inaccurately* calculating only *a fraction* of the system's actual energy consumption!

Problem Summary

- We are *inaccurately* calculating only *a fraction* of the system's actual energy consumption!

- **Summary:** We cannot improve what we cannot measure.

Outline

Goal

Develop a framework to *accurately and reliably* measure the energy consumption of the applications on Linux

Goal

Develop a framework to *accurately and reliably* measure the energy consumption of the applications on Linux

Report the statistics to the

Goal

Develop a framework to *accurately and reliably* measure the energy consumption of the applications on Linux

Report the statistics to the

- End-users: In an easy-to-understand and useful format

Goal

Develop a framework to *accurately and reliably* measure the energy consumption of the applications on Linux

Report the statistics to the

- End-users: In an easy-to-understand and useful format
- Programmers: Via APIs that improve programmer actionability

Goal

Develop a framework to *accurately and reliably* measure the energy consumption of the applications on Linux

Report the statistics to the

- End-users: In an easy-to-understand and useful format
- Programmers: Via APIs that improve programmer actionability
- System Designers: To enable iterating over low-energy designs

Goal

- **Framework** = Models and Tools

Goal

- **Framework** = Models and Tools
- **Power models** = How we reason about and estimate a device's power draw over time

Goal

- **Framework** = Models and Tools
- **Power models** = How we reason about and estimate a device's power draw over time
- **Power models** are often not available or poorly understood for many devices, e.g., DRAM

Goal

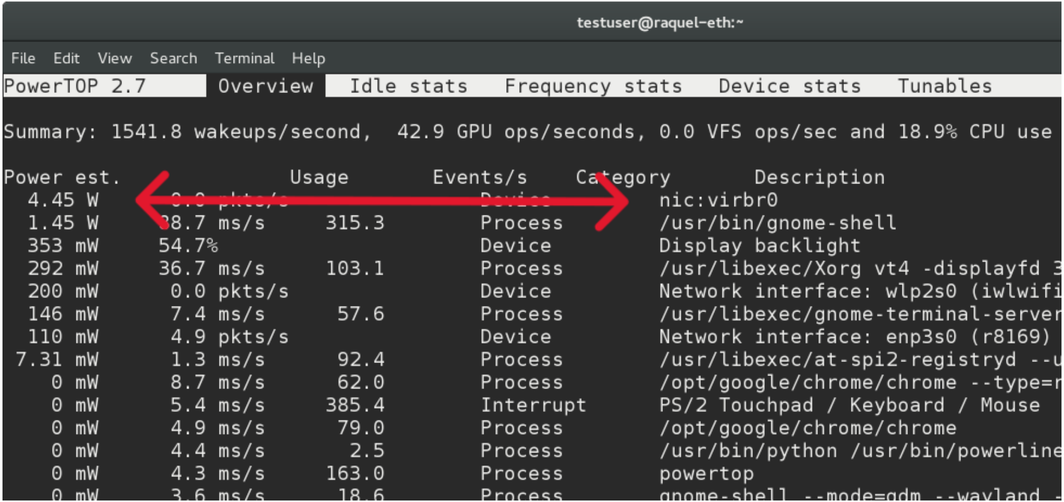
- **Framework** = Models and Tools
- **Power models** = How we reason about and estimate a device's power draw over time
- **Power models** are often not available or poorly understood for many devices, e.g., DRAM
- **Tools** can be built to accurately calculate power using the models, e.g., nvidia-smi

Goal

- **Framework** = Models and Tools
- **Power models** = How we reason about and estimate a device's power draw over time
- **Power models** are often not available or poorly understood for many devices, e.g., DRAM
- **Tools** can be built to accurately calculate power using the models, e.g., nvidia-smi
- **Takeaway:** We need accurate models and reliable tools to calculate energy consumption

Outline

PowerTOP



testuser@raquel-eth:~
File Edit View Search Terminal Help
PowerTOP 2.7 Overview Idle stats Frequency stats Device stats Tunables
Summary: 1541.8 wakeups/second, 42.9 GPU ops/seconds, 0.0 VFS ops/sec and 18.9% CPU use
Power est. Usage Events/s Category Description
4.45 W 0.0 pkts/s Device nic:virbr0
1.45 W 38.7 ms/s 315.3 Process /usr/bin/gnome-shell
353 mW 54.7% Device Display backlight
292 mW 36.7 ms/s 103.1 Process /usr/libexec/Xorg vt4 -displayfd 3
200 mW 0.0 pkts/s Device Network interface: wlp2s0 (iwlwifi
146 mW 7.4 ms/s 57.6 Process /usr/libexec/gnome-terminal-server
110 mW 4.9 pkts/s Device Network interface: enp3s0 (r8169)
7.31 mW 1.3 ms/s 92.4 Process /usr/libexec/at-spi2-registryd --u
0 mW 8.7 ms/s 62.0 Process /opt/google/chrome/chrome --type=r
0 mW 5.4 ms/s 385.4 Interrupt PS/2 Touchpad / Keyboard / Mouse
0 mW 4.9 ms/s 79.0 Process /opt/google/chrome/chrome
0 mW 4.4 ms/s 2.5 Process /usr/bin/python /usr/bin/powerline
0 mW 4.3 ms/s 163.0 Process powertop
0 mW 3.6 ms/s 18.6 Process gnome-shell --mode=gdm --wayland -

PowerTOP

It is possible to use Powertop to view the "power estimate" of a process/device/interrupt/timer.

PowerTOP

It is possible to use Powertop to view the "power estimate" of a process/device/interrupt/timer.

Challenges:

1. Power estimate is a discrete-time event. Energy consumption is a continuous process with a higher correlation to battery drain.

PowerTOP

It is possible to use Powertop to view the "power estimate" of a process/device/interrupt/timer.

Challenges:

1. Power estimate is a discrete-time event. Energy consumption is a continuous process with a higher correlation to battery drain.
2. Vendor-specific implementation

PowerTOP

It is possible to use Powertop to view the "power estimate" of a process/device/interrupt/timer.

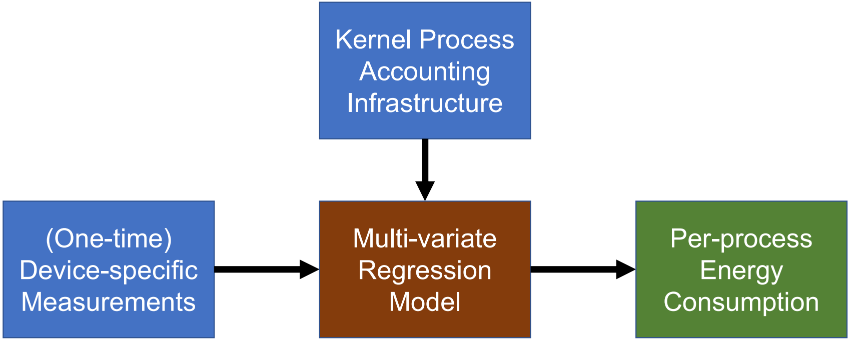
Challenges:

1. Power estimate is a discrete-time event. Energy consumption is a continuous process with a higher correlation to battery drain.
2. Vendor-specific implementation
3. Actionability of this data for end-users and programmers
   Process X consumes 1.45 Watts. What should the programmer do to optimize it?

Outline

- Background
- Problem
- Goal
- Current Tools
  - PowerTOP
- System Design
- End Product
- Conclusion

System Design



Device-Specific Measurements

Goal: Determine regression parameters

Device-Specific Measurements

Goal: Determine regression parameters

Algorithm:

Device-Specific Measurements

Goal: Determine regression parameters

Algorithm:

1. Minimize system load by turning off all devices

Device-Specific Measurements

Goal: Determine regression parameters

Algorithm:

1. Minimize system load by turning off all devices
2. Measure battery drain rate over multiple intervals

Device-Specific Measurements

Goal: Determine regression parameters

Algorithm:

1. Minimize system load by turning off all devices
2. Measure battery drain rate over multiple intervals
3. Turn on a single target device

Device-Specific Measurements

Goal: Determine regression parameters

Algorithm:

1. Minimize system load by turning off all devices
2. Measure battery drain rate over multiple intervals
3. Turn on a single target device
4. Sweep target device parameters from low to high while measuring battery drain

Device-Specific Measurements

Goal: Determine regression parameters

Algorithm:

1. Minimize system load by turning off all devices
2. Measure battery drain rate over multiple intervals
3. Turn on a single target device
4. Sweep target device parameters from low to high while measuring battery drain
5. Turn off target device or set parameter to low

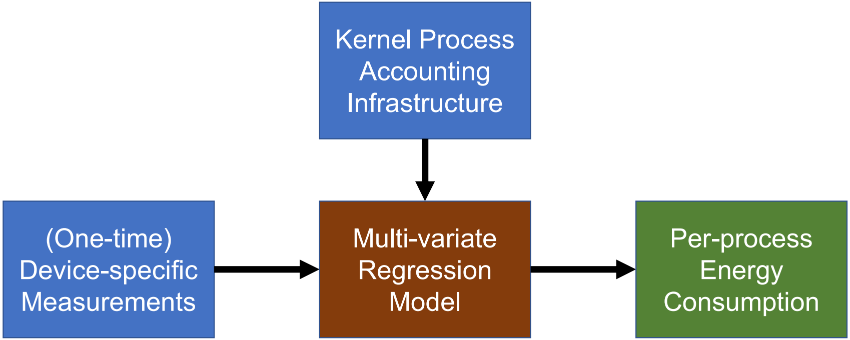
Device-Specific Measurements

Goal: Determine regression parameters

Algorithm:

1. Minimize system load by turning off all devices
2. Measure battery drain rate over multiple intervals
3. Turn on a single target device
4. Sweep target device parameters from low to high while measuring battery drain
5. Turn off target device or set parameter to low
6. Repeat step 3-5 for all target devices

System Design



Kernel Process Accounting Infrastructure

Goal: Determine regression inputs

Kernel Process Accounting Infrastructure

Goal: Determine regression inputs

Method:

Kernel Process Accounting Infrastructure

Goal: Determine regression inputs

Method:

- Poll the process accounting infrastructure to determine CPU time allocation, network activity, open file handles, memory, disk usage, network, and screen wakeups.

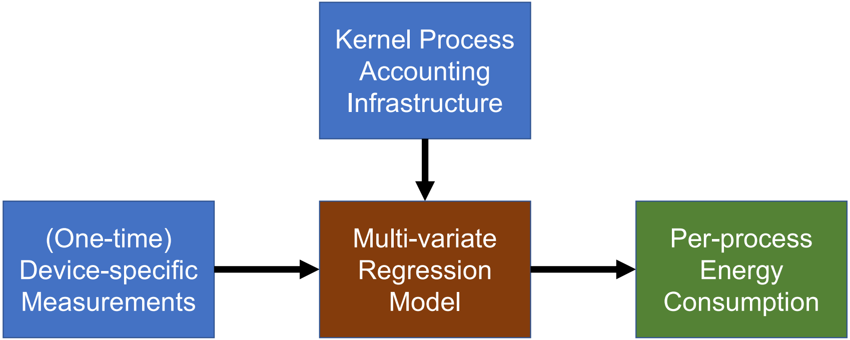
Kernel Process Accounting Infrastructure

Goal: Determine regression inputs

Method:

- Poll the process accounting infrastructure to determine CPU time allocation, network activity, open file handles, memory, disk usage, network, and screen wakeups.
- Input the measured values in the regression model to predict energy consumption

System Design



Challenge: System Design

- Estimated value (All models are wrong, but some are useful.)

Challenge: System Design

- Estimated value (All models are wrong, but some are useful.)
- Accuracy and Bias trade-off: Accurate models generate larger systemic load that biases observations

Challenge: Data Collection

- There are millions of devices, and billions of ICs inside these devices. The power estimates can range across 2-3 orders of magnitude. How can we develop **accurate & reliable** power models across this diversity?

Challenge: Data Collection

- There are millions of devices, and billions of ICs inside these devices. The power estimates can range across 2-3 orders of magnitude. How can we develop **accurate & reliable** power models across this diversity?
- **Privacy concern: Should users share this data to a "centralized" server?**

Challenge: Validation of Correctness

- There is often significant difference between estimated values (from the model) and actual values (**ground truth**)

Challenge: Validation of Correctness

- There is often significant difference between estimated values (from the model) and actual values (**ground truth**)
- How to identify regressions from **ground truth** without hardware modifications?

Carbon emissions of software

$$\text{Carbon Footprint} = \text{Energy Consumption} \times \text{Energy Composition}$$

Carbon emissions of software

$$\text{Carbon Footprint} = \text{Energy Consumption} \times \text{Energy Composition}$$

$$\text{Energy Consumption} = \text{Power} \times \text{Latency}$$

Carbon emissions of software

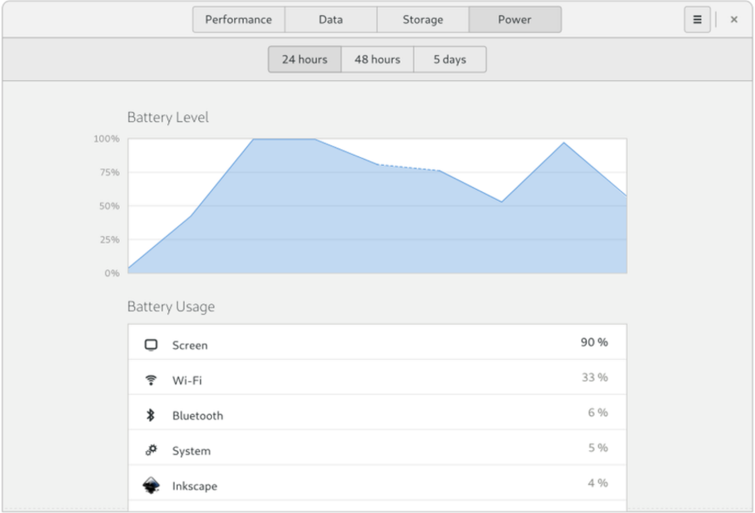
Carbon Footprint = Energy Consumption × Energy Composition

Energy Consumption = Power × Latency

Energy Composition depends on multiple factors, including geography, time of availability, and cost

Outline

End-users


Performance
Data
Storage
Power
24 hours
48 hours
5 days
Battery Level
100%
75%
50%
25%
0%
Battery Usage
Screen
90 %
Wi-Fi
33 %
Bluetooth
6 %
System
5 %
Inkscape
4 %

Programmers

Expose API for programmers: Indicate devices with high energy consumption to allow backtracing to code

Example use-case: Energy-efficient code optimization suggestions in the coding platform

System Designers

Expose API for system designers to enable better carbon accounting practices with clear scope identification

Example use-case: Develop better tools to explore the design space of performance vs energy vs carbon efficiency

Outline

- Background
- Problem
- Goal
- Current Tools
  - PowerTOP
- System Design
- End Product
- Conclusion

Key Takeaways

- **We cannot improve what we cannot measure.**

Key Takeaways

- **We cannot improve what we cannot measure.**
- **Non-CPU system components can dominate the overall energy consumption.**

Thank you!

Feedback/Collaboration ?

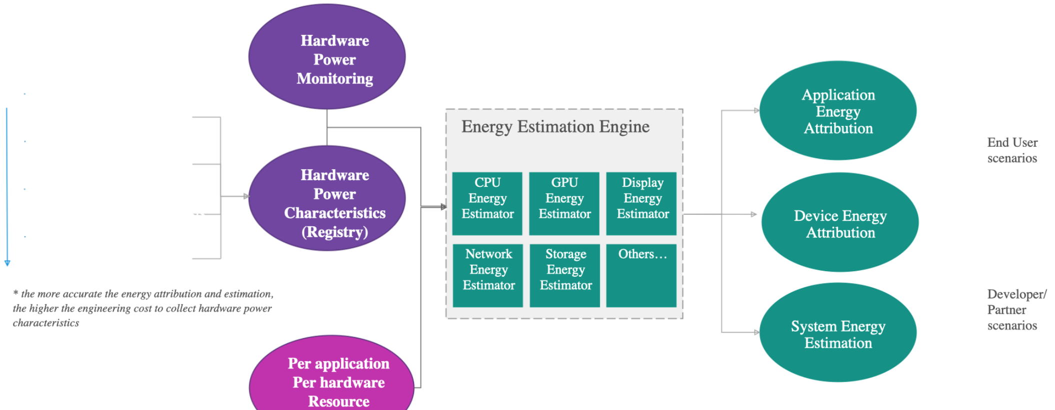
https://www.linkedin.com/in/adityamanglik/

amangli@student.ethz.ch

Extended Discussion

Windows Energy Estimation Engine (E3) System Design

How Does Energy Estimation Engine Work?



Reverse Engineering Windows' Energy Estimation Engine: back-end

- The Energy Estimation Engine (E3) service runs on all Windows devices and attributes energy consumption to individual hardware components and applications.

Reverse Engineering Windows' Energy Estimation Engine: back-end

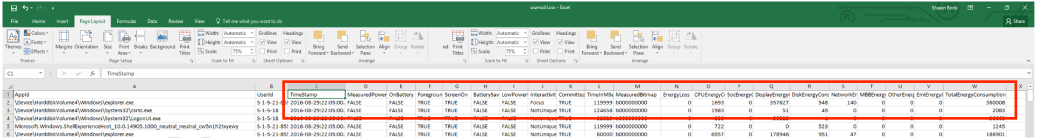
- The Energy Estimation Engine (E3) service runs on all Windows devices and attributes energy consumption to individual hardware components and applications.
- Why software-based attribution: Few PCs in the market have such dedicated chips: According to reports, 99% of current devices in market lack dedicated current and voltage monitors.

Reverse Engineering Windows' Energy Estimation Engine: back-end

- The Energy Estimation Engine (E3) service runs on all Windows devices and attributes energy consumption to individual hardware components and applications.
- Why software-based attribution: Few PCs in the market have such dedicated chips: According to reports, 99% of current devices in market lack dedicated current and voltage monitors.
- Software-based power attribution provides about 85% accuracy compared to a 98% accuracy rate from systems equipped with dedicated current and voltage monitors (e.g., Microsoft Surface)
- Microsoft claims that they prioritize data from devices with dedicated chips while developing the software-based power

E3 System Design

- Power profiles: Windows has separate power profiles for individual hardware devices like network, disks etc. Further, profiles specialize for Laptops, Tablets, Phones devices etc.
- The following data columns can be observed in the E3 Service Report (shown below): ScreenOnEnergy, CPUEnergy, SoCEnergy, DisplayEnergy, DiskEnergy, MBBEnergy, NetworkEnergy, EmiEnergy, and many more.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	AppId	UserId	TimeStamp	MeasuredPower	OnBattery	Foregroun	ScreenOn	BatterySav	LowPowerI	Interactivit	Committed	TimeInMSe	MeasuredBitmap	EnergyLoss	CPUEnergyC	SocEnergy(	DisplayEnergy(	DiskEnergyCons	NetworkEr	MBBEnergy	OtherEnerg	EmiEnergy(	TotalEnergyConsumption
2	\Device\HarddiskVolume4\Windows\explorer.exe	S-1-5-21-85	2016-08-29:22:05:00.	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	Focus	TRUE	119999	b000000000	0	1693	0	357627	548	140	0	0	0	360008
3	\Device\HarddiskVolume4\Windows\System32\csrss.exe	S-1-5-18	2016-08-29:22:05:00.	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	NotUnique	TRUE	124658	b000000000	0	1983	0	51	49	0	0	0	0	2083
4	\Device\HarddiskVolume4\Windows\System32\LogonUI.exe	S-1-5-18	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
5	Microsoft.Windows.ShellExperienceHost_10.0.14905.1000_neutral_neutral_cw5n1h2txyewy	S-1-5-21-85	2016-08-29:22:05:00.	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	NotUnique	TRUE	119999	b000000000	0	722	0	0	523	0	0	0	0	1245
6	\Device\HarddiskVolume4\Windows\explorer.exe	S-1-5-21-85	2016-08-29:22:05:00.	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	NotUnique	TRUE	60000	b000000000	0	6957	0	178946	951	47	0	0	0	186901

Figure: Data dump from E3 CLI

System design goals

The framework should be:

Accurate: Eliminate anomalous values

[1] Stretch goal

System design goals

The framework should be:

- Accurate: Eliminate anomalous values
- Portable: Able to function across different hardware vendors[2]

[1]Stretch goal

System design goals

The framework should be:

- Accurate: Eliminate anomalous values
- Portable: Able to function across different hardware vendors[2]
- Independent of *extra* measurement devices

[1] Stretch goal

System design goals

The framework should be:

- Accurate: Eliminate anomalous values
- Portable: Able to function across different hardware vendors[2]
- Independent of *extra* measurement devices
- Transparent: Should not induce *any* load on the target system[1]

---

[1] Stretch goal

System design goals

The framework should be:

- Accurate: Eliminate anomalous values
- Portable: Able to function across different hardware vendors[2]
- Independent of *extra* measurement devices
- Transparent: Should not induce *any* load on the target system[1]
- Reliable: Repeat experiments should yield *similar* results

[1]Stretch goal

Design Optimizations

Central information store to overcome randomness?

- Overcoming variation in values: Collect data across systems to create a database

Design Optimizations

Central information store to overcome randomness?

- Overcoming variation in values: Collect data across systems to create a database
- Privacy challenges: can we do better?

Design Considerations

- Reliable: Co-executing processes significantly influence power.
  Solution: Energy consumption should be roughly similar.

Design Considerations

- Reliable: Co-executing processes significantly influence power.
  Solution: Energy consumption should be roughly similar.
- Accuracy:
  - Challenging to isolate individual contributions as many processes use multiple hardware devices simultaneously (CPU, GPU, Display, RAM, SSD, Ethernet/WiFi).

Design Considerations

- Reliable: Co-executing processes significantly influence power.
  Solution: Energy consumption should be roughly similar.
- Accuracy:
  - Challenging to isolate individual contributions as many processes use multiple hardware devices simultaneously (CPU, GPU, Display, RAM, SSD, Ethernet/WiFi).
  - Hardware devices do not measure/expose individual power draw.

Design Considerations

- Reliable: Co-executing processes significantly influence power.
  Solution: Energy consumption should be roughly similar.
- Accuracy:
  - Challenging to isolate individual contributions as many processes use multiple hardware devices simultaneously (CPU, GPU, Display, RAM, SSD, Ethernet/WiFi).
  - Hardware devices do not measure/expose individual power draw.
  - Significant variation in data-sheets across devices and vendors.

Design Considerations

- Reliable: Co-executing processes significantly influence power.
  Solution: Energy consumption should be roughly similar.
- Accuracy:
  - Challenging to isolate individual contributions as many processes use multiple hardware devices simultaneously (CPU, GPU, Display, RAM, SSD, Ethernet/WiFi).
  - Hardware devices do not measure/expose individual power draw.
  - Significant variation in data-sheets across devices and vendors.
  - Reliable values: CPU perf counters (RAPL?) and current battery charge (ACPI?)

Different hardware devices

- CPU: Dominant factor, P-states vs C-states, interfaces (Intel RAPL)
- GPU: periodic bursts of large power draw
- RAM: Increasing DRAM capacity is challenging due to refresh power draw (Reference)
- I/O Peripherals: USB devices are polled every 5 ms
- Display: Often the most consistent drain
- Network Adaptors: Ethernet, WiFi ping frequency
- Disk: SSD, HDD writes are cached for bulk ops

Design Considerations

- Hardware requirements: Cannot rely on external power monitors
- Transparency: Polling for values induces load on the target system
- Able to function across different hardware vendor APIs
- Actionability of data: Reporting hardware power values is "futile" because hardware is difficult to change, but processes might be optimized.