Estimating the Carbon Footprint of Personal Computing

Aditya Manglik
ETH Zürich, Switzerland

Email: amangli@ethz.ch

Linux Foundation Energy Summit-2023
2 June, 2023

Brief Introduction

Graduate student at ETH Zürich, Switzerland

Brief Introduction

Graduate student at ETH Zürich, Switzerland

Focus on research at the intersection of computer architecture and operating systems

Outline

- Background
- Problem
- Goal
- Relevant Work
  - Windows
  - macOS
  - Android
  - iOS
  - PowerTOP
- System Design
- End Product
- Conclusion

Outline

- Background
- Problem
- Goal
- Relevant Work
  - Windows
  - macOS
  - Android
  - iOS
  - PowerTOP
- System Design
- End Product
- Conclusion

Background

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

Background

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

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

Background

Carbon Footprint $=$ Energy Consumption $\times$ Composition

Energy Consumption $=$ Power $\times$ Latency

Composition depends on multiple factors, including geography, availability, and cost

Energy Consumption

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

Energy Consumption

Energy Consumption $=$ Power $\times$ Latency

Power dissipation is determined by hardware

Energy Consumption

Energy Consumption = Power $\times$ Latency

Power dissipation is determined by hardware

Example: CPUs: 55 - 150 W, GPUs: 50-300 W

Energy Consumption

Energy Consumption $=$ Power $\times$ Latency

Power dissipation is determined by hardware

Example: CPUs: 55 - 150 W, GPUs: 50-300 W

Latency is determined by hardware and software

Energy Consumption

Energy Consumption = Power × Latency

Power dissipation is determined by hardware

Example: CPUs: 55 - 150 W, GPUs: 50-300 W

Latency is determined by hardware and software

Typically, only latency is optimized by programmers
because it is ***measurable*[1]*, and observable by end-users***

[1] CPU clock cycles

Outline

Background

Problem

Goal

Relevant Work

- Windows
- macOS
- Android
- iOS
- PowerTOP

System Design

End Product

Conclusion

Software power attribution

State of consumer operating systems

Software power attribution

State of consumer operating systems

- Windows: Kernel-supported tracing (Windows E3)

Software power attribution

State of consumer operating systems

- Windows: Kernel-supported tracing (Windows E3)
- Mac OS: Kernel-supported tracing (Activity Monitor)

Software power attribution

State of consumer operating systems

- Windows: Kernel-supported tracing (Windows E3)
- Mac OS: Kernel-supported tracing (Activity Monitor)
- Android: Kernel-supported tracing (Battery subsystem)

Software power attribution

State of consumer operating systems

- Windows: Kernel-supported tracing (Windows E3)
- Mac OS: Kernel-supported tracing (Activity Monitor)
- Android: Kernel-supported tracing (Battery subsystem)
- iOS: Kernel-supported tracing (Battery usage)

Software power attribution

State of consumer operating systems

- Windows: Kernel-supported tracing (Windows E3)
- Mac OS: Kernel-supported tracing (Activity Monitor)
- Android: Kernel-supported tracing (Battery subsystem)
- iOS: Kernel-supported tracing (Battery usage)
- Linux: ??

Problem

Why software power attribution matters?

Consumers wish they had *infinite* battery capacity

Problem

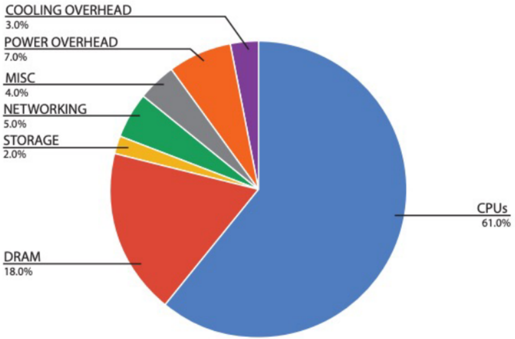
Why software power attribution matters?

Consumers wish they had *infinite* battery capacity

Battery capacity is a major design constraint and *UX factor* for any consumer device: mobiles, laptops, VR headsets [1]

[1] Cole, Wesley, A. Will Frazier, and Chad Augustine. Cost projections for utility-scale battery storage: 2021 update. No. NREL/TP-6A20-79236. National Renewable Energy Lab.(NREL), Golden, CO (United States), 2021.

Hardware power attribution



Representative breakdown for per-component power dissipation.

Source: 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.

Outline

- Background
- Problem
- Goal
- Relevant Work
  - Windows
  - macOS
  - Android
  - iOS
  - PowerTOP
- System Design
- End Product
- Conclusion

Goal

Develop a framework to *reliably* determine the energy consumption of any process executing on a Linux system

Goal

Develop a framework to *reliably* determine the energy consumption of any process executing on a Linux system

Report the statistics to the

Goal

Develop a framework to *reliably* determine the energy consumption of any process executing on a Linux system

Report the statistics to the

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

Goal

Develop a framework to *reliably* determine the energy consumption of any process executing on a Linux system

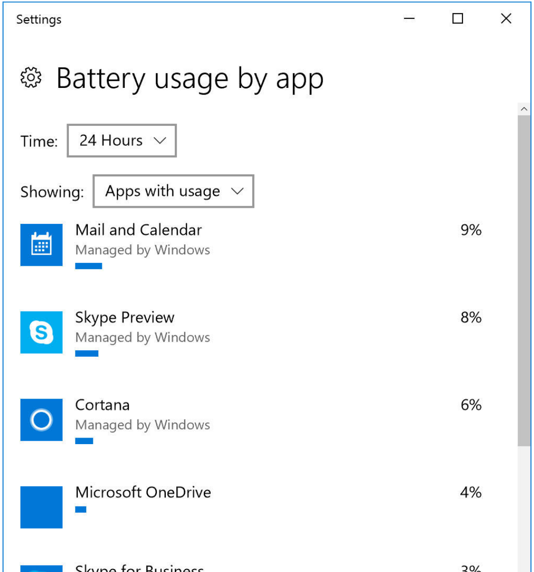
Report the statistics to the

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

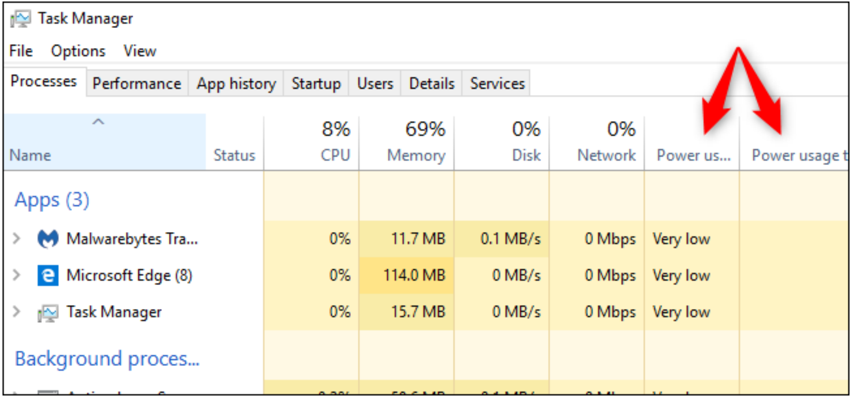
Outline

- Background
- Problem
- Goal
- Relevant Work
  - Windows
  - macOS
  - Android
  - iOS
  - PowerTOP
- System Design
- End Product
- Conclusion

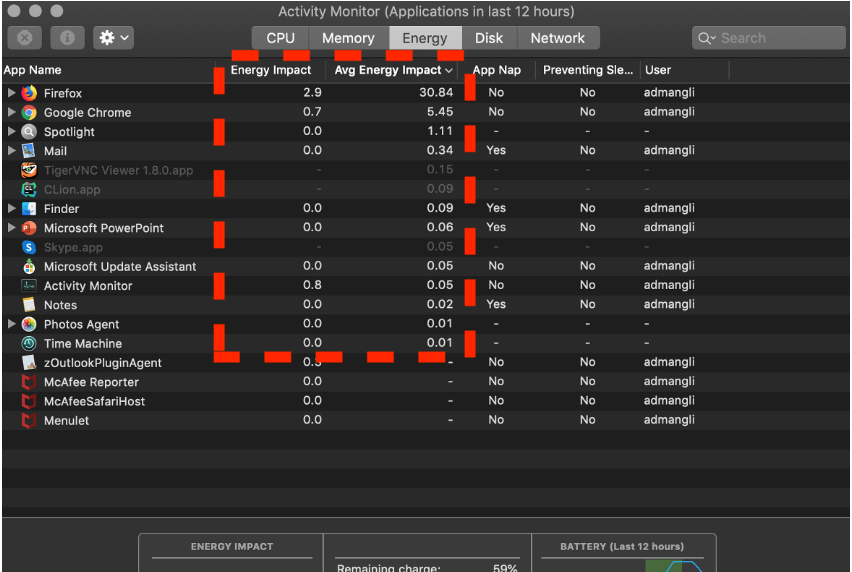
Windows Energy Estimation Engine (E3)



Windows Task Manager Integration

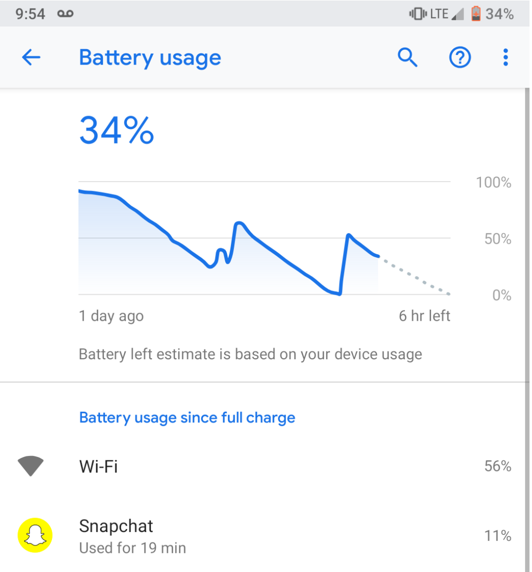


macOS



App Name	Energy Impact	Avg Energy Impact	App Nap	Preventing Sle...	User
Firefox	2.9	30.84	No	No	admangli
Google Chrome	0.7	5.45	No	No	admangli
Spotlight	0.0	1.11	-	-	-
Mail	0.0	0.34	Yes	No	admangli
TigerVNC Viewer 1.8.0.app	-	0.15	-	-	-
CLion.app	-	0.09	-	-	-
Finder	0.0	0.09	Yes	No	admangli
Microsoft PowerPoint	0.0	0.06	Yes	No	admangli
Skype.app	-	0.05	-	-	-
Microsoft Update Assistant	0.0	0.05	No	No	admangli
Activity Monitor	0.8	0.05	No	No	admangli
Notes	0.0	0.02	Yes	No	admangli
Photos Agent	0.0	0.01	-	-	-
Time Machine	0.0	0.01	-	-	-
zOutlookPluginAgent	0.3	-	No	No	admangli
McAfee Reporter	0.0	-	No	No	admangli
McAfeeSafariHost	0.0	-	No	No	admangli
Menulet	0.0	-	No	No	admangli

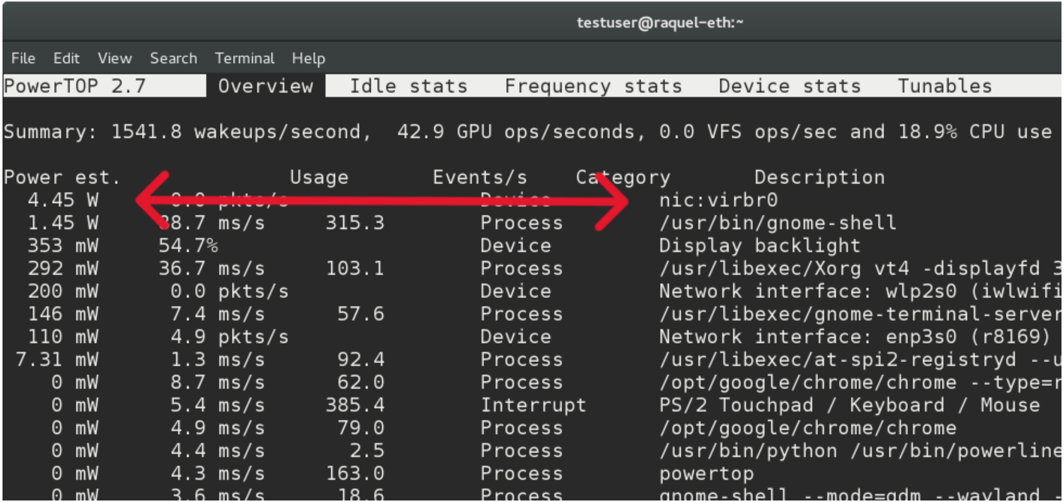
Android


9:54
LTE 34%
Battery usage
34%
100%
50%
0%
1 day ago
6 hr left
Battery left estimate is based on your device usage
Battery usage since full charge
Wi-Fi
56%
Snapchat
Used for 19 min
11%

iOS

Vodafone IN
12:15 PM
44%
Settings
Battery
refresh, automatic downloads, and some visual effects are reduced or turned off.
Battery Percentage
Show percentage of battery remaining in the status bar.
BATTERY USAGE
Last 24 Hours
Last 7 Days
Music
13 min on screen – 1.5 hrs background
55%
WhatsApp
53 min on screen – 42 min background
20%
Home & Lock Screen
34 min on screen
6%
App Store

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: The solution?

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

PowerTOP: The solution?

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

Problems:

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

PowerTOP: The solution?

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

Problems:

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: The solution?

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

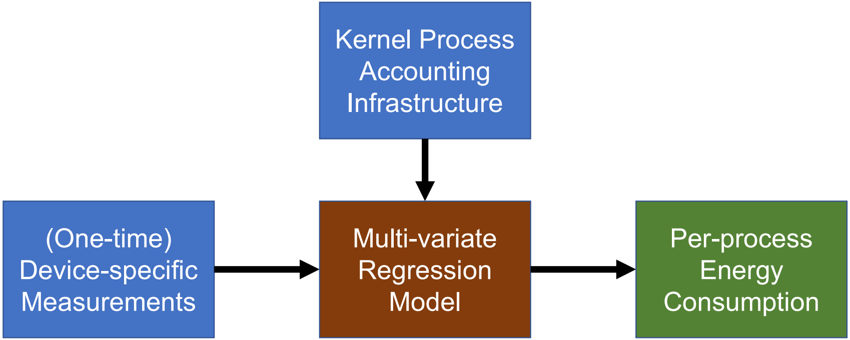
Problems:

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
- Relevant Work
  - Windows
  - macOS
  - Android
  - iOS
  - PowerTOP
- System Design
- End Product
- Conclusion

System Design



Device-Specific Measurements

Goal: Determine regression parameters

Device-Specific Measurements

Goal: Determine regression parameters

Method:

Device-Specific Measurements

Goal: Determine regression parameters

Method:

- Minimize system load by turning off all devices

Device-Specific Measurements

Goal: Determine regression parameters

Method:

- Minimize system load by turning off all devices
- Measure battery drain rate over multiple intervals

Device-Specific Measurements

Goal: Determine regression parameters

Method:

- Minimize system load by turning off all devices
- Measure battery drain rate over multiple intervals
- Turn on target device

Device-Specific Measurements

Goal: Determine regression parameters

Method:

- Minimize system load by turning off all devices
- Measure battery drain rate over multiple intervals
- Turn on target device
- Sweep target device parameters while measuring battery drain

Device-Specific Measurements

Goal: Determine regression parameters

Method:

- Minimize system load by turning off all devices
- Measure battery drain rate over multiple intervals
- Turn on target device
- Sweep target device parameters while measuring battery drain
- Train multivariate model

Device-Specific Measurements

Goal: Determine regression parameters

Method:

- Minimize system load by turning off all devices
- Measure battery drain rate over multiple intervals
- Turn on target device
- Sweep target device parameters while measuring battery drain
- Train multivariate model
- Repeat for all target devices

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:

- For each running process, poll the process accounting infrastructure to determine CPU time allocation, network activity, open file handles, memory usage, and screen wakeups.

Kernel Process Accounting Infrastructure

Goal: Determine regression inputs

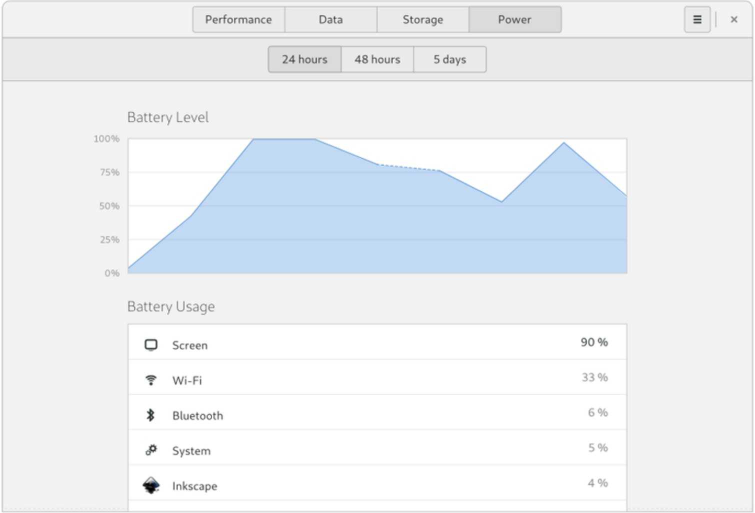
Method:

- For each running process, poll the process accounting infrastructure to determine CPU time allocation, network activity, open file handles, memory usage, and screen wakeups.
- Input the measured values in the regression model to predict energy consumption estimate for each process.

Outline

- Background
- Problem
- Goal
- Relevant Work
  - Windows
  - macOS
  - Android
  - iOS
  - PowerTOP
- System Design
- End Product
- Conclusion

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

Upstream the kernel module

Expose API for programmers

Example use-case: Energy-based optimization suggestions in IDE

Outline

- Background
- Problem
- Goal
- Relevant Work
  - Windows
  - macOS
  - Android
  - iOS
  - PowerTOP
- System Design
- End Product
- Conclusion

Conclusion

A useful feature is missing from the Linux kernel

This module empowers both end-users and programmers with realistic carbon footprint of their applications

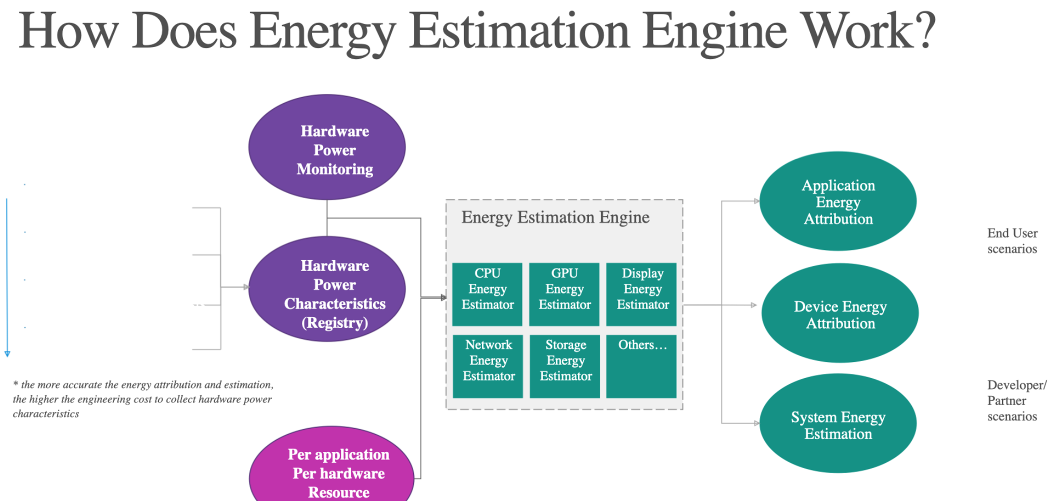
Questions/Suggestions ?

amangli@ethz.ch

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

Extended Discussion

Windows Energy Estimation Engine (E3) System Design



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

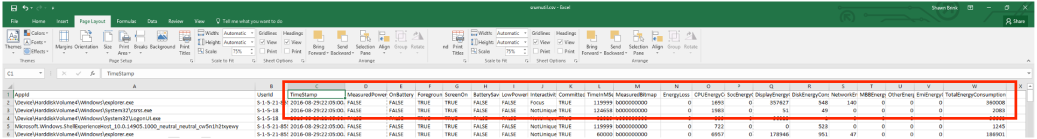
- The Energy Estimation Engine (E3) service runs on all Windows devices and attributes energy consumption to individual hardware components and applications.
- Why software-based attribution: Few PCs in the market have such dedicated chips: According to reports, 99% of current devices in market lack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



AppId	UserId	TimeStamp	MeasuredPower	OnBattery	Foregroun	ScreenOn	BatterySav	LowPowerI	Interactivit	Committed	TimeInMSe	MeasuredBitmap	EnergyLoss	CPUEnergyC	SocEnergyC	DisplayEnergyC	DiskEnergyCons	NetworkEn	MBBEnergy	OtherEnerg	EmiEnergyC	TotalEnergyConsumption
\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
\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
\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]
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
\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

[2] Stretch goal

System design goals

The framework should be:

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

[2] 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

[2] 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[2]

[2] 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[2]
- Reliable: Repeat experiments should yield *similar* results

[2] 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.