The GNOME™ Conference

GUADEC

Power Measurement & Attribution systems in GNOME

Adi M (IRC: carpediem)
reach.aditya.here+guadec@gmail.com

whoami

- Researcher in computer architecture and systems at ETH Zurich (with Prof. Onur Mutlu)
- GSoC 2018 student with **GNOME**
- Worked on implementing the power panel in GNOME-Usage
- Mentors: *Felipe Borges, Christian Kellner*

Section 1

Overview

- Defining the Problem Statement
  - Non-commercial users and developer impact
  - Enterprise user impact
  - How Power attribution solves these problems?
- Case Study: Windows Energy Estimation Engine (E3)
- Case Study: MAC OS Energy Impact
- Proposed System Architecture
- Bringing it all together; *GNOME-Usage*
- *Brainstorming*

Section 2

Problem Statement

Understanding desktop-user concern

- I have been using my computer for barely 30 minutes. Why is the battery already at 50%?

Understanding desktop-user concern

- I have been using my computer for barely 30 minutes. Why is the battery already at 50%?
- Why are the fans blasting at full speed, *I am just browsing the internet!*

Understanding desktop-user concern

- I have been using my computer for barely 30 minutes. Why is the battery already at 50%?
- Why are the fans blasting at full speed, *I am just browsing the internet!*
- Why is the battery at 87% when it was fully charged last night?

Understanding desktop-user concern

- I have been using my computer for barely 30 minutes. Why is the battery already at 50%?
- Why are the fans blasting at full speed, *I am just browsing the internet!*
- Why is the battery at 87% when it was fully charged last night?
- **Why is a process consuming much more energy than the amount of value I am deriving from it?**

Premise

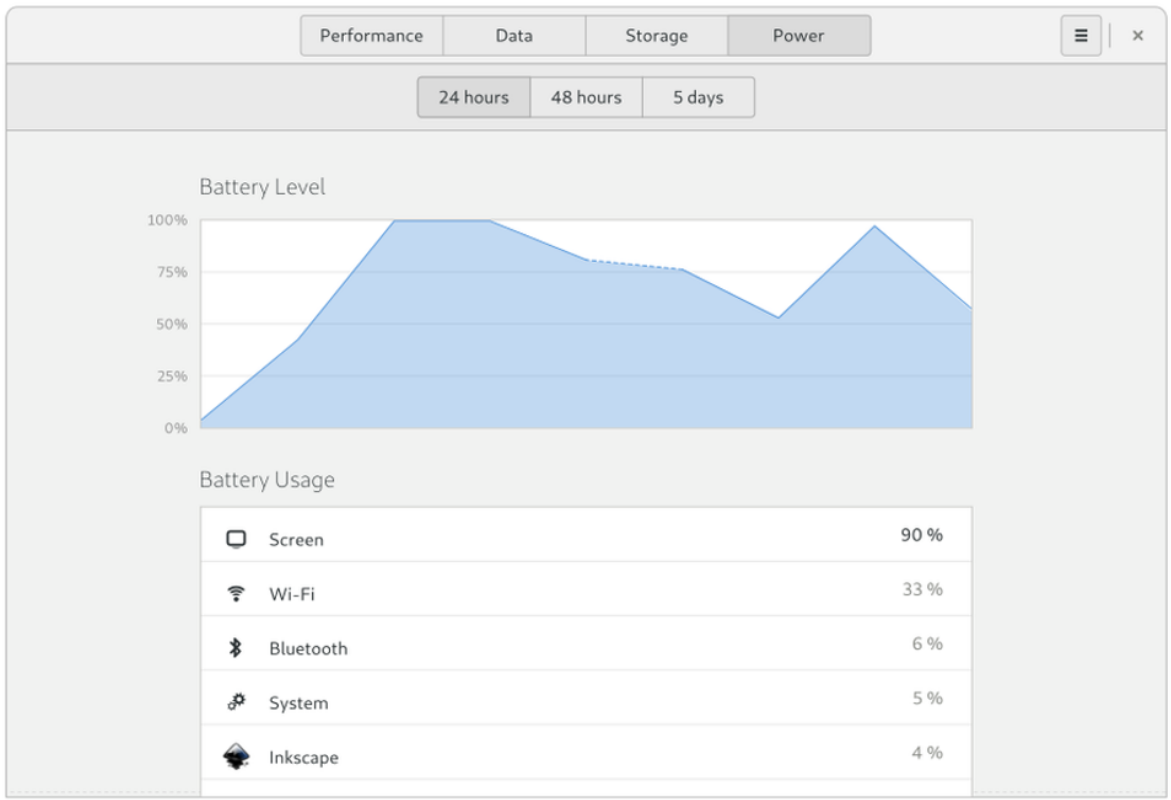


Figure: GNOME-Usage Mockup, Credits - Allan Day

Understanding developer use-cases

- Why is my application getting bug reports about draining the battery too fast?

Understanding developer use-cases

- Why is my application getting bug reports about draining the battery too fast?
- How do I quantify the power impact of my code?

Understanding developer use-cases

- Why is my application getting bug reports about draining the battery too fast?
- How do I quantify the power impact of my code?
- **Power attribution data gives developers the ability to see how and where the power is being consumed.**

Understanding developer use-cases

- Why is my application getting bug reports about draining the battery too fast?
- How do I quantify the power impact of my code?
- **Power attribution data gives developers the ability to see how and where the power is being consumed.**
- For example, a compute-intensive application should not be dominated by data movement costs which would show up DRAM energy!

Understanding developer use-cases

- Why is my application getting bug reports about draining the battery too fast?
- How do I quantify the power impact of my code?
- **Power attribution data gives developers the ability to see how and where the power is being consumed.**
- For example, a compute-intensive application should not be dominated by data movement costs which would show up DRAM energy!
- This also allows system administrators stronger control and easier ways to detect misbehaving applications

Understanding Enterprise concerns

- Energy costs of running servers represents 55% of the Total Cost of Ownership (TCO) in data centers (Reference)

Understanding Enterprise concerns

- Energy costs of running servers represents 55% of the Total Cost of Ownership (TCO) in data centers (Reference)
- Power attribution enables data-centers to monitor high energy-cost applications, and improve application scheduling across clusters for optimizing energy efficiency

Understanding Enterprise concerns

- Energy costs of running servers represents 55% of the Total Cost of Ownership (TCO) in data centers (Reference)
- Power attribution enables data-centers to monitor high energy-cost applications, and improve application scheduling across clusters for optimizing energy efficiency
- Theoretically, enterprise IT administrators could create scripts to collect periodic logs to analyze energy usage data from devices, and improve workload allocation across devices (RNNs for server-class CPUs, CNNs for GPUs, cloud apps etc.)

Limitations of the battery

- Moore's law doesn't apply to batteries!

Limitations of the battery

- Moore's law doesn't apply to batteries!
- Current battery technologies have been stagnant w.r.t charge capacity and density improvements.

Limitations of the battery

- Moore's law doesn't apply to batteries!
- Current battery technologies have been stagnant w.r.t charge capacity and density improvements.
- Power envelopes have emerged as the major constraint for any consumer-facing system = mobile devices, laptops, tablets, etc.

Section 3

Why Power attribution

How can power attribution fix these problems?

- **"You cannot improve what you cannot measure."**
- As developers, we need to be acutely aware of power consumption

How can power attribution fix these problems?

- **"You cannot improve what you cannot measure."**
- As developers, we need to be acutely aware of power consumption
- Despite immediate user impact, current power attribution infrastructure is not well developed (PowerTOP)

How can power attribution fix these problems?

- **"You cannot improve what you cannot measure."**
- As developers, we need to be acutely aware of power consumption
- Despite immediate user impact, current power attribution infrastructure is not well developed (PowerTOP)
- Windows, MAC OS, Android have closed the gap

How can power attribution fix these problems?

- **"You cannot improve what you cannot measure."**
- As developers, we need to be acutely aware of power consumption
- Despite immediate user impact, current power attribution infrastructure is not well developed (PowerTOP)
- Windows, MAC OS, Android have closed the gap
- Linux solution still awaited despite maximum server deployment

Section 4

Challenges

Hardware is hard

- Processors run at 3 GHz+; within one second, millions of events happen. **Humans run at 1 Hz +- !**

Hardware is hard

- Processors run at 3 GHz+; within one second, millions of events happen. **Humans run at 1 Hz +- !**
- Determining individual process and hardware power costs, and interpreting the data for human time-frames is hard!

Hardware is hard

- Processors run at 3 GHz+; within one second, millions of events happen. **Humans run at 1 Hz +- !**
- Determining individual process and hardware power costs, and interpreting the data for human time-frames is hard!
- Power is a hardware concept, and most applications use multiple hardware devices simultaneously (CPU, GPU, Display, RAM, SSD, Ethernet/WiFi etc.)

Hardware is hard

- Processors run at 3 GHz+; within one second, millions of events happen. **Humans run at 1 Hz +- !**
- Determining individual process and hardware power costs, and interpreting the data for human time-frames is hard!
- Power is a hardware concept, and most applications use multiple hardware devices simultaneously (CPU, GPU, Display, RAM, SSD, Ethernet/WiFi etc.)
- Hardware chips do not measure/expose individual wattage information

Hardware is hard

- Processors run at 3 GHz+; within one second, millions of events happen. **Humans run at 1 Hz +- !**
- Determining individual process and hardware power costs, and interpreting the data for human time-frames is hard!
- Power is a hardware concept, and most applications use multiple hardware devices simultaneously (CPU, GPU, Display, RAM, SSD, Ethernet/WiFi etc.)
- Hardware chips do not measure/expose individual wattage information
- Reliable values available include Processor wattage (post Nehalem) and available battery charge (via ACPI/UPower)

Hardware devices

- CPU: Cores vs Clock, P-states vs C-states
- GPU: thousands of cores + high-bandwidth memories
- I/O Peripherals: USB devices are polled every 5 ms
- Display: Backlight can brighten/darken your day
- Network Adaptors: Ethernet, WiFi pings
- Disk: HDD writes are cached for bulk ops
- RAM: Till 2016, Macs could only use maximum 16 GB RAM due to DDR3 power requirements (Reference)

Ideal solution

- *Determine time spent on CPU, I/O and Memory. Multiply by nominal power for these components.*

Ideal solution

- *Determine time spent on CPU, I/O and Memory. Multiply by nominal power for these components.*
- *Time spent = Ideally one should do an accounting update on task switch, along with all the other housekeeping. There's a* ***process accounting*** *infrastructure which perhaps could be adapted for this.*

Ideal solution

- *Determine time spent on CPU, I/O and Memory. Multiply by nominal power for these components.*
- *Time spent = Ideally one should do an accounting update on task switch, along with all the other housekeeping. There's a* ***process accounting*** *infrastructure which perhaps could be adapted for this.*
- Heisenberg's uncertainty principle in action: High frequency updates will cause lots of additional wake-ups, which will badly skew what you're measuring. Ex, CPU states.

Ideal solution

- *Determine time spent on CPU, I/O and Memory. Multiply by nominal power for these components.*
- *Time spent = Ideally one should do an accounting update on task switch, along with all the other housekeeping. There's a **process accounting** infrastructure which perhaps could be adapted for this.*
- Heisenberg's uncertainty principle in action: High frequency updates will cause lots of additional wake-ups, which will badly skew what you're measuring. Ex, CPU states.
- ACPI / RAPL / Manufacturer data-sheets = Conflicting data sources. **How to align and make sense?**

Ideal solution

- *Determine time spent on CPU, I/O and Memory. Multiply by nominal power for these components.*
- *Time spent = Ideally one should do an accounting update on task switch, along with all the other housekeeping. There's a **process accounting** infrastructure which perhaps could be adapted for this.*
- Heisenberg's uncertainty principle in action: High frequency updates will cause lots of additional wake-ups, which will badly skew what you're measuring. Ex, CPU states.
- ACPI / RAPL / Manufacturer data-sheets = Conflicting data sources. **How to align and make sense?**
- OEMs: Collect data from devices running your software

Section 5

Case Studies

Windows: Energy Estimation Engine

- The Energy Estimation Engine (E3) service runs on ALL Windows devices. This *lightweight* service attributes energy consumption to hardware components and applications.

Windows: Energy Estimation Engine

- The Energy Estimation Engine (E3) service runs on ALL Windows devices. This *lightweight* service attributes energy consumption to hardware components and applications.
- Microsoft engineers have stated that software-based power attribution provides about 85% accuracy compared to a 98% accuracy rate from systems equipped with dedicated current and voltage monitors, like the Microsoft Surface.

Windows: Energy Estimation Engine

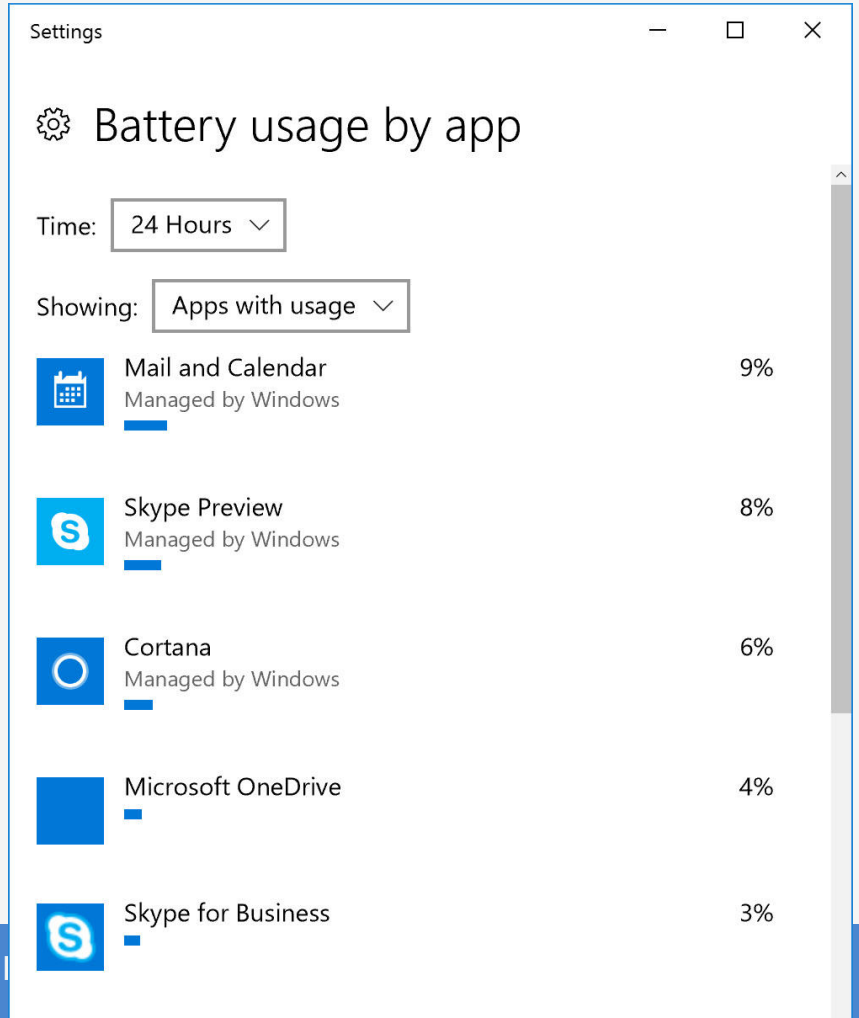
- The Energy Estimation Engine (E3) service runs on ALL Windows devices. This *lightweight* service attributes energy consumption to hardware components and applications.
- Microsoft engineers have stated that software-based power attribution provides about 85% accuracy compared to a 98% accuracy rate from systems equipped with dedicated current and voltage monitors, like the Microsoft Surface.
- Microsoft also claims that they prioritize data from devices with dedicated chips while developing the software-based power models.

Windows: Energy Estimation Engine

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

E3 front-end: Battery usage

This breakdown can be observed via the Settings app in Windows.
The interesting observation here is that they do not report hardware device attribution, rather only for processes.



Task Manager Front-end

The Task Manager shows per-process power impact, for short-term (first column) and over long term (second column).

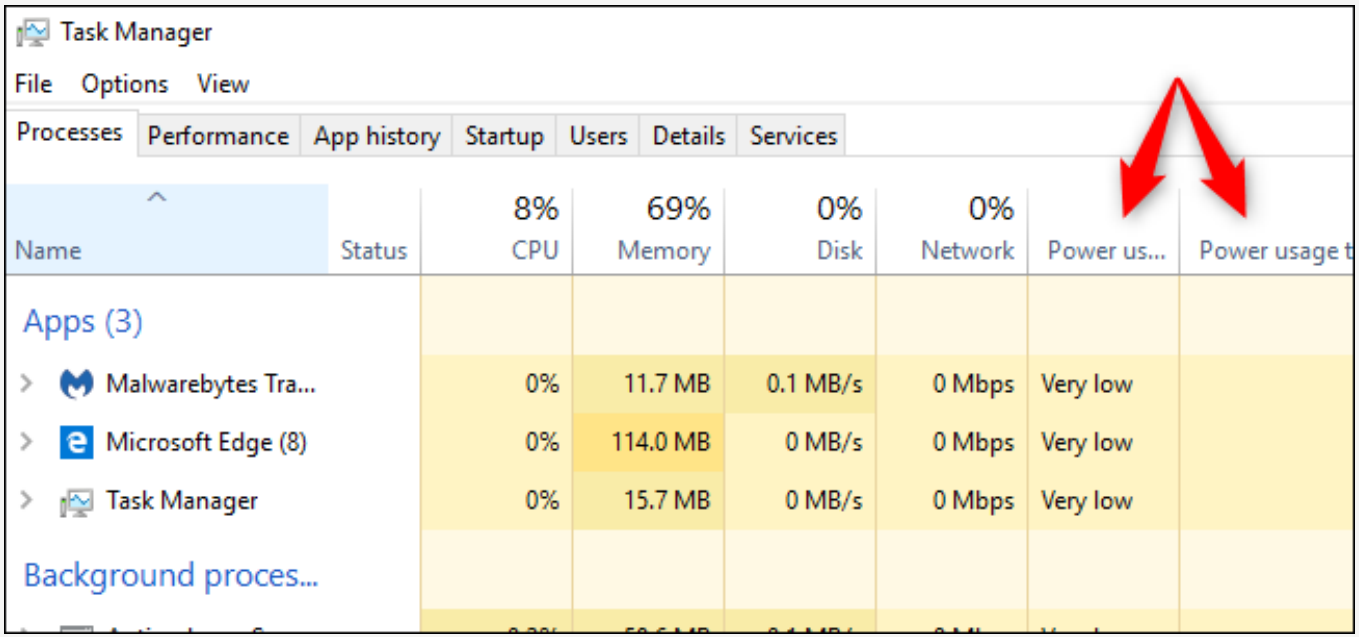


Figure: Observation: No absolute numbers are presented, only relative terms such as *Low usage, Very High usage etc.*

E3 Architecture

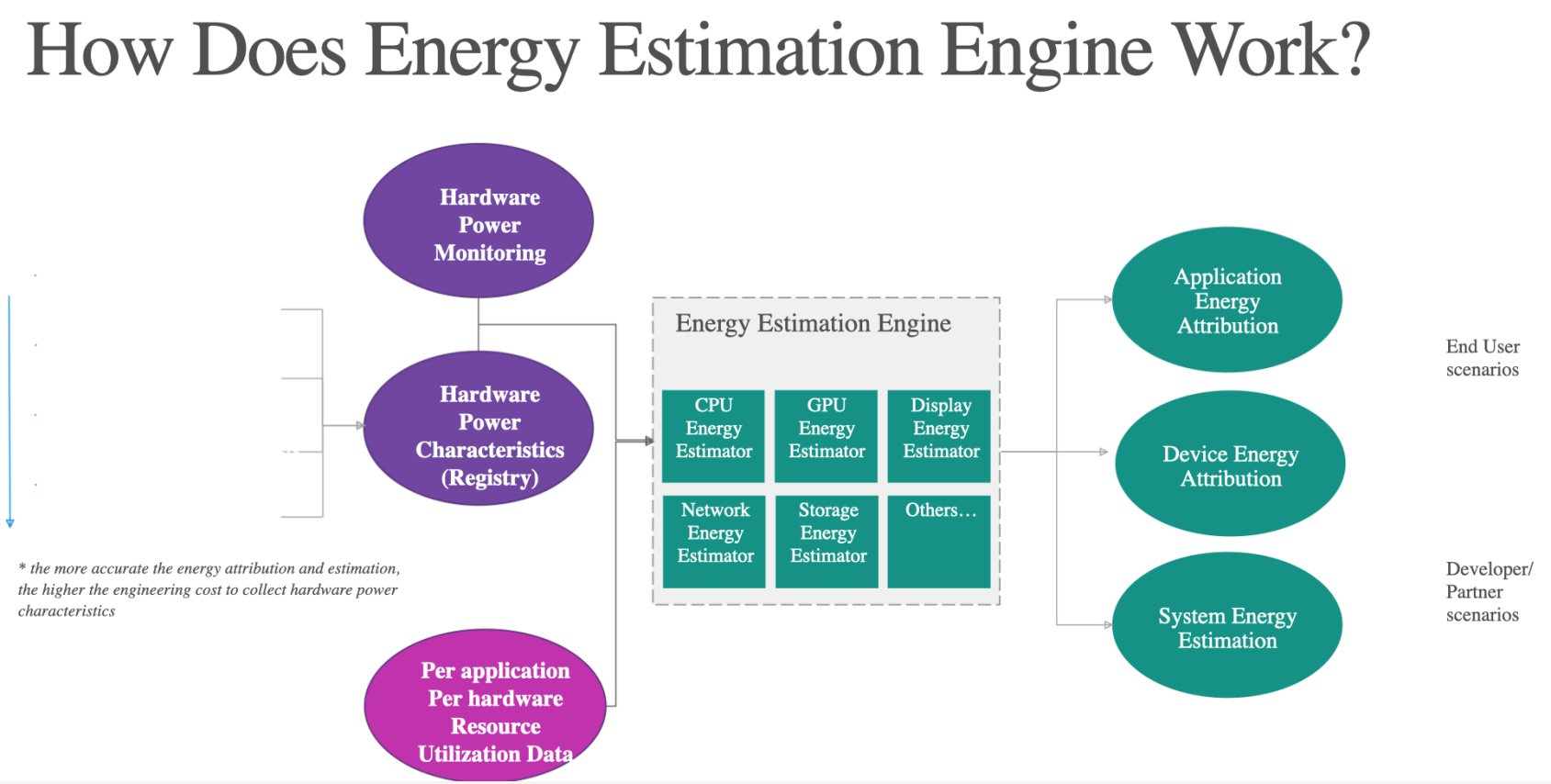


Figure: Source link: Microsoft presentation to hardware vendors

Back-end

- Power profiles: Windows has separate power profiles for individual hardware devices like network, disks etc. Further, profiles specialize for Laptops, Tablets, Phones devices etc.

Back-end

- 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 etc.

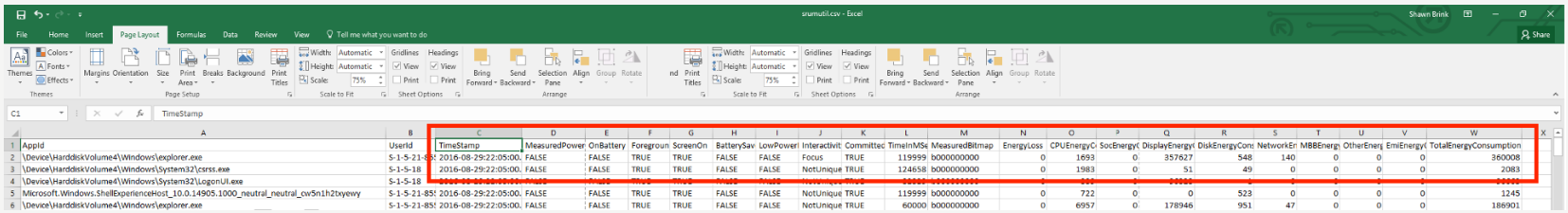


Figure: Statistics recorded by Windows E3

macOS statistics

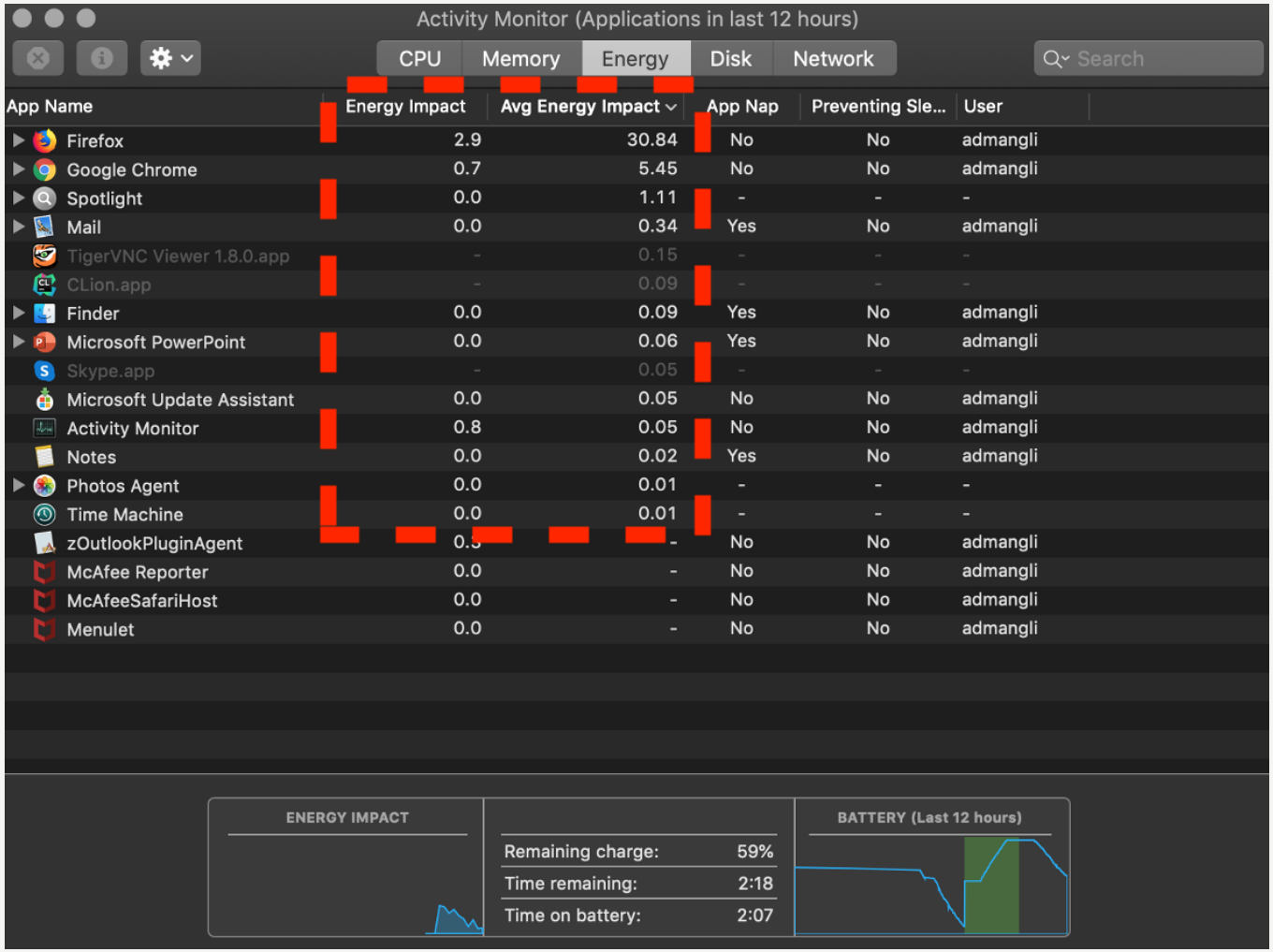


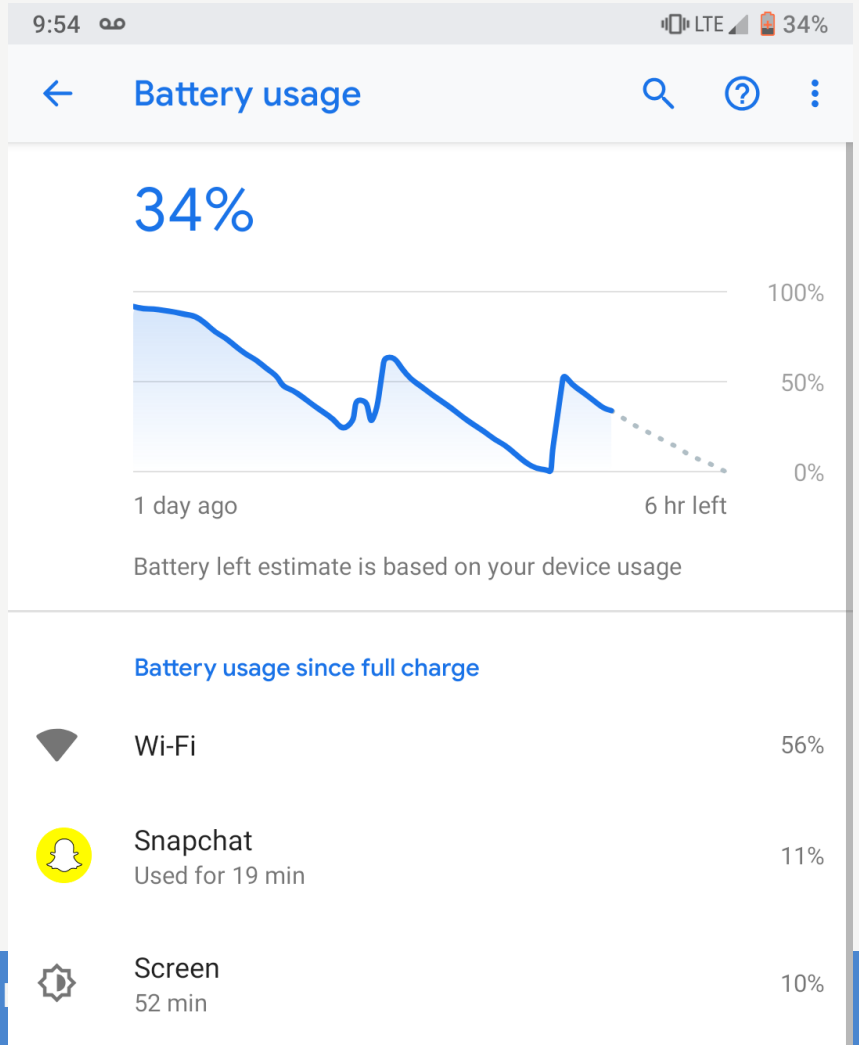
Figure: Activity Monitor displays process-relative power impact

macOS: Energy Impact

- The Energy panel debuted in Activity Monitor approximately 6 years ago
- The panel displays "Energy Impact" of each open app based on a number of factors including CPU usage, network traffic, disk activity, Interrupts and more.
- The higher the number, the more impact an app has on battery power (maximum observed around 780 during stress tests).
- Similar to Windows, MAC OS also attributes power only to processes, not individual hardware devices
- *Details are sparse, but I strongly suspect that MAC devices have dedicated chips for power measurement*

Android

Android has stringent power envelopes, and power statistics predate at least v2.3 GingerBread! Interestingly, android **attributes power to both hardware and software!**



Section 6

System Architecture

Back-end

- Data collection- Huge variation in each device, billions of devices, order of magnitude differences in same configurations - **Data needed to make models!**

Back-end

- Data collection- Huge variation in each device, billions of devices, order of magnitude differences in same configurations - **Data needed to make models!**
- Develop power models for each device: *Gaussian distributions?*

Back-end

- Data collection- Huge variation in each device, billions of devices, order of magnitude differences in same configurations - **Data needed to make models!**
- Develop power models for each device: *Gaussian distributions?*
- Incorporate these models into a multi-variate regression model with current battery charge as one of the inputs

Back-end

- Data collection- Huge variation in each device, billions of devices, order of magnitude differences in same configurations - **Data needed to make models!**
- Develop power models for each device: *Gaussian distributions?*
- Incorporate these models into a multi-variate regression model with current battery charge as one of the inputs
- Can also be interpreted as a variant of the **Multi-armed Bandit Problem**

Data Collection?

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

Data Collection?

- There are billions of devices, and tens of billions of ICs inside these devices. The power estimates can range across 2-3 orders of magnitude. How do we develop **accurate & reliable** power models across this range?
- Windows performs data mining across ALL devices for developing the power models. These models enable reliable per-component estimates, with constant fine-tuning.

Data Collection?

- There are billions of devices, and tens of billions of ICs inside these devices. The power estimates can range across 2-3 orders of magnitude. How do we develop **accurate & reliable** power models across this range?
- Windows performs data mining across ALL devices for developing the power models. These models enable reliable per-component estimates, with constant fine-tuning.
- **Privacy concern: Should users share this data?**
  What can be the challenges here? How else can we obtain this data (across billions of devices, millions of ICs and thousands of OEM/IHV)?

Front-end

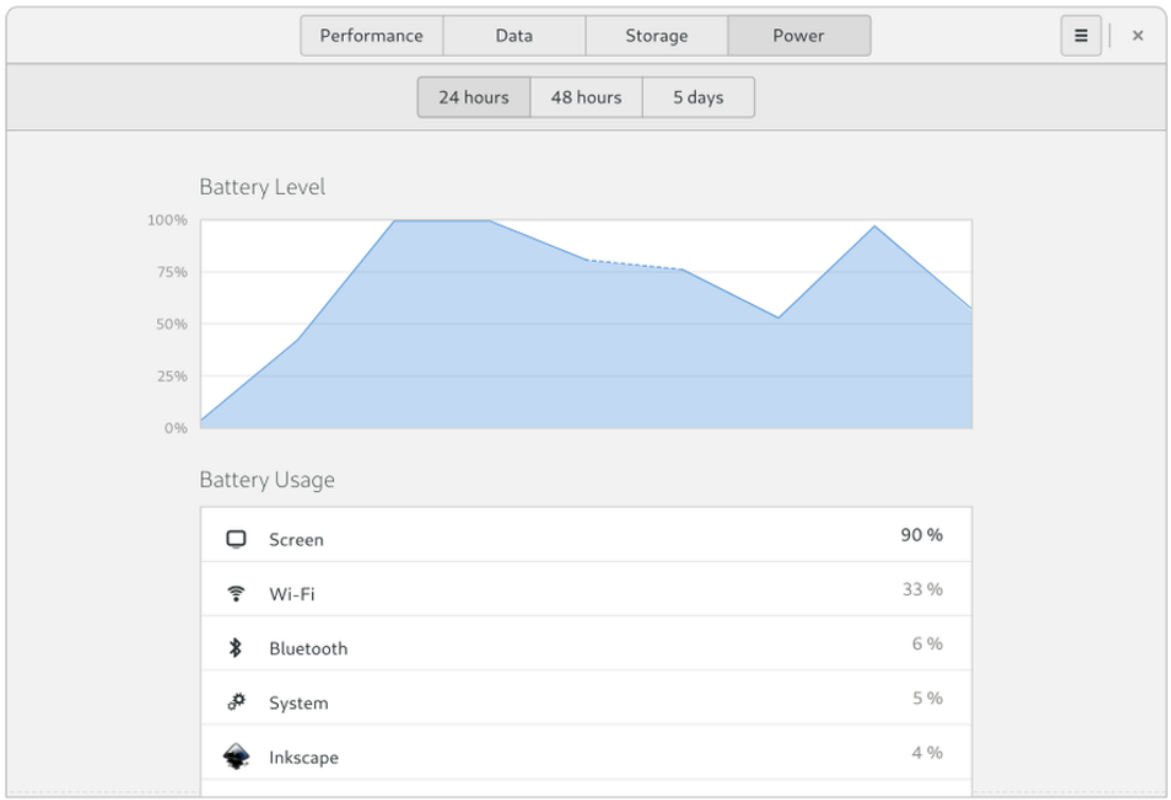


Figure: GNOME-Usage Mockup, Credits - Allan Day

Section 7

End

Questions?

Shout-out: Felipe Borges, Christian Kellner (gicmo)

Please reach out for questions via:

`reach.aditya.here+guadec@gmail.com`