Measuring Power Consumption and Emissions of Web Applications: Implementing SCI for the Web

Oliver Winks, Adam Newman, Son Ba Pham
Root & Branch Digital Ltd
oliver@rootandbranch.io, adam@rootandbranch.io, son@rootandbranch.io

November 2024

1 Introduction

The environmental impact of digital technology has become an increasingly urgent concern, particularly as the internet's energy demands continue to grow [citation needed]. As essential components of the digital economy, websites and web applications contribute significantly to global energy consumption and carbon emissions [citation needed]. However, accurately quantifying these impacts remains challenging due to the diverse ways they are used and the inherent complexity of web infrastructure.

In this paper, we present the **Cardamon Web Model** (henceforth CWM), named after the Cardamon project, an Innovate UK-funded initiative focused on developing tools to assess and reduce software-related emissions. The model is designed to estimate the power consumption and associated carbon emissions of a single webpage during a typical user interaction. To facilitate the estimation of entire web applications the model can be applied to a representative set of pages that capture the app's overall functionality.

The CWM builds upon the **Software Carbon Intensity** (SCI) specification [citation needed], taking a bottom-up approach to calculating energy consumption, rather than a top-down methodology like that of other models e.g. **Sustainable Web Design Model** (SWD) [citation needed]. This allows for a more granular analysis of device components that consume energy during web interactions, resulting in greater accuracy and actionable insights.

This paper is organised as follows: Section 2 **Software Carbon Intensity**; introduces the SCI specification. Section 3 **The Cardamon Web Model**; describes the CWM in general terms, outlining its theoretical framework and principles, relating it back to the SCI. Section 4 **Functions, Parameters & Assumptions**; describes the energy consumption functions, constants and assumptions that underpin key calculations. Section 5 **Reference Devices**; discusses reference devices used in the implementation. Section 6 **User Interaction**; introduces user interaction strategies. Finally, Section 7 explores **Future Work** and potential developments to enhance the model's scope and applicability.

2 Software Carbon Intensity

The CWM builds upon the principles of the SCI specification by applying it to the domain of web applications. The SCI is a ISO standard [citation needed] designed to provide a standardised methodology for assessing and comparing the environmental impact of software [1]. Equations (1 - 4) describes the specification in mathematical terms.

$$SCI = (O + M) \quad \text{per } R \tag{1}$$

where: O = operational emissions (i.e. use-phase).
M = embodied emissions (i.e. production and disposal phases).
R = functional unit (e.g. API call, user interaction, etc).

$$O = E \cdot I \tag{2}$$

where: E = energy consumed.
I = region specific carbon intensity.

$$M = TE \cdot TS \cdot RS \tag{3}$$

where: TE = total embodied emissions of the hardware.
TS = share of the total lifespan of the hardware reserved for use by the software.
RS = share of the total available resources of the hardware reserved for use by the software.

$$TS = \left(\frac{TiR}{EL}\right) \tag{4}$$

where: TiR = the length of time the hardware is reserved for use by the software.
EL = the expected lifespan of the equipment.

Throughout this paper we relate the CWM back to this specification by using the same terminology and mathematical notation.

3 The Cardamon Web Model

In this section we explore how the CWM extends the SCI specification, applying it to web applications.

We start by breaking a web application into three distinct sub-systems, modelling each individually to estimate their energy consumption. The sub-systems are as follows:

- **Front-end**; the end-user device.
- **Network**; the routers, switches etc used to connect the end-user device to the various servers used to deliver the website.
- **Back-end**; the computers and associated infrastructure used to host and deliver the content to the end-user device. These are usually computers within a data-centre.

Operational and embodied emissions are calculated for each sub-system during an interactivity test which is conducted on physical reference devices for the front end.
Equation 5 shows how we reformulate the SCI from equation 1 to include these high-level sub-systems.

$$SCI_{web} = \sum_{s \in \mathcal{S}} O_s + M_s \tag{5}$$

where: $\mathcal{S}$ = (f, n, b) = set of system components; front-end (f), network (n) and back-end (b).
O_s = operational emissions of sub-system s.
M_s = embodied emissions of sub-system s.

Let's look at these system components in more detail, starting with the front-end:

3.1 Front-End

To estimate the energy consumption associated with end-user devices, our model accounts for the diversity among end user devices by categorising them into distinct categories such as desktop, mobile, and others. Each category represents a broad classification of devices that share similar characteristics in terms of hardware and usage patterns.

For each category, we define a reference device that serves as a representative model (see section 5). These reference devices are further decomposed into their constituent hardware components, such as the CPU, network adapter, screen, and others. Each hardware component has an associated energy consumption function which takes as input a set of metrics specific to the component (CPU utilization, data transfer etc).

Metrics for each component are gathered directly from the reference devices during an interactivity test in which a set of actions are performed on the web page. This aligns with the functional unit R in the SCI framework.

Interactivity tests are defined at the level of individual web pages rather than entire web applications. This design choice effectively makes an interaction within a single-page the fundamental building block of CWM. These blocks of interaction can be tested independently as single page measurements but also pieced together to model specific user journeys as well as entire websites and web applications by identifying representative pages and interactions that capture the app's overall functionality.

Once metrics have been gathered during the interactivity test we can use them as parameters to our component energy models to estimate the breakdown of system energy. We can then optionally use this information along with user analytics to scale the estimated energy consumption by the number of hits of that type originating from each device category, in each region - building a representative estimation of end user emissions.

Lets begin describing this method mathematically from the SCI specification.

$$SCI_f = O_f + M_f \tag{6}$$

where: O_f = total operational emissions across all end-user devices.
M_f = total embodied emissions across all end-user devices.

To apply the front end portion of the CWM to the SCI specification we must unpack both the operational and embodied emissions terms (O_f and M_f respectively).

3.1.1 Operational Emissions

Expanding the operational emissions term O_f to include each device category and scaling by the number of hits results in:

$$O_f = I_f \sum_{d \in \mathcal{D}} h_d (E_{f,d} + E_{i,d}) \tag{7}$$

where: I_f = average carbon intensity across all end-user devices.
h_d = hits originating from device category d over reporting period.
$E_{f,d}$ = energy consumption of the d'th category of user-device.
$E_{i,d}$ = idle energy consumption during the interactivity test.
$\mathcal{D}$ = set of user device categories (e.g. mobile, desktop).

Even in this form, the connection to the SCI specification is still very clear. The terms I and E from equation 2 are present but have been expanded in the case of energy consumption to include idle consumption E_i and active consumption E_f.

We can further expand these terms like so:

$$E_i = t \cdot e_i \tag{8}$$

where: t = elapsed time of the interactivity test.
e_i = idle energy consumption of the reference device for the category under consideration.

To expand the active energy consumption we must include the hardware components considered for each device category and the metrics gathered for those components during the interactivity test we can:

$$E_f = \sum_{c \in \mathcal{C}} f_c(\mathcal{M}_c) \tag{9}$$

where: $\mathcal{C}$ = set of hardware components $\{CPU, networkadaptor, ...\}$.
f_c = energy consumption function for component c.
$\mathcal{M}_c$ = metrics associated for component c.

User analytics often contains information about where, geographically, each hit to the web page originated from. This data can be used to calculate the average carbon intensity across all hits. However, in the absence of this user analytics data the global average carbon intensity can be used instead.

$$I_f = \sum_{r \in \mathcal{R}} q_r I_r \tag{10}$$

where: q = proportion of views originating from region r.
I = carbon intensity of region r during the test.
$\mathcal{R}$ = Set of all geographical regions.

3.1.2 Embodied Emissions

$$M_f = t \sum_{d \in \mathcal{D}} h_d m_d \tag{11}$$

where: t = elapsed time of the interactivity test.
h_d = hits originating from device category d over reporting period.
m_d = embodied emissions rate of end-user device category d.
$\mathcal{D}$ = set of user device categories (e.g. mobile, desktop).

Equation (11) shows how the total embodied emissions of end-user devices is calculated in the CWM. Our formulation differs from that of the SCI specification but is completely compatible with it when we expand the definition of m_d.

Using the terminology defined in equations 3 and 4 and making the assumption that during the interactivity test the device is used exclusively for that task we can expand m_d as follows:

$$m_d = TE_d \left(\frac{TiR_d}{EL_d} \right) RS \tag{12}$$

Under the assumption that the device is used exclusively for the purpose of rendering the web page during the interactivity test equation 12 simplifies to:

$$m_d = TE_d \left(\frac{TiR_d}{EL_d} \right) \tag{13}$$

It is important to note that the CWM in it's current form is likely to overestimate the embodied emissions of the front-end due to the above assumption. Modern devices are likely to be performing many other tasks whilst rendering a web page.

3.2 Network

Unlike the front-end and back-end subsystems, where constituent hardware components can be modelled using usage metrics, the network infrastructure presents unique challenges. This is due to the fact that most of the infrastructure is not accessible comprising of elements such as undersea cables, switches, routers, internet exchange points etc. Consequently, a top-down modelling approach is necessary to estimate energy consumption.

To model the energy consumption of the network infrastructure, we rely on estimates derived from academic research and established models. One such model is the Sustainable Web Design model, which provides an estimate of the energy consumed during data transfer over the global networking infrastructure. This model calculates energy consumption by dividing the total estimated energy consumption of the global networking infrastructure (310 TWh per year [citation needed]) by the total estimated data transfer of the internet (5.29 ZB per year [citation needed]). This results in an average energy consumption of:

$$0.059 = \left(\frac{310x10^9}{5.29x10^{12}} \right) \tag{14}$$

This figure aligns with estimates from other studies, such as the work by Hanna Pihkola et al. (2018) [citation needed], which estimated that the energy consumption per gigabyte would be less than 0.1 kWh by 2020.

We acknowledge the complexity of network emissions estimations and the limitations of an 'emissions per byte' approach. In the absence of an accepted standard for network emissions we chose to adopt this approach rather than ignore network emissions completely. We will continue to align our method with the latest research in this space as it evolves.

3.3 Back-End

The back-end subsystem refers to the servers and associated infrastructure responsible for hosting and delivering web content to end-user devices. Estimating the energy consumption and carbon emissions of the back-end can be approached in two ways:

The first approach mirrors the methodology used for the front-end. This involves conducting an interactivity test on the front-end while simultaneously recording metrics from the back-end infrastructure. The same energy consumption functions described in Section 3.1 are applied, with the exception of screen energy consumption, as servers typically do not have screens attached. However, this approach requires the application to be tested in a private development or test environment to avoid noise from external requests. For accurate results, the test environment must closely resemble the production environment, which may not always be feasible due to technical constraints. Additionally, this approach requires specialised tools capable of measuring the energy consumption of various back-end processes during the interactivity test. One such tool is Cardamon Core [citation needed].

The second approach, which is described in detail in this section, involves using metrics gathered from the production environment over the period of interest. Most cloud platforms provide details about the servers provisioned (e.g. instance type, number of virtual CPUs) and record basic metrics such as CPU utilisation and network traffic. These metrics can be used to estimate energy consumption using various component models (see later sections). Importantly, this approach is not limited to cloud platforms. As long as metrics are being gathered, the same methodology can also be applied to on-premises data centres. Many modern on-premises setups include monitoring tools that track server performance and resource usage, enabling similar estimations of energy consumption and carbon emissions. Unlike the front-end approach, this method relies on historical data spanning the entire reporting period, which is typically several weeks or months, and accounts for all activity on the site during that time.

Before we detail the model we must first address the fact that servers are often shared between multiple guests. In the case of dedicated hardware, the number of other guests is simply assumed to be zero. However, in most cases, servers are shared among multiple residents through virtualisation. Virtualisation allows a single server to divide its resources (CPU, memory, storage, etc.) among it's residents. When allocating resources to individual guests, the total resources available to the server are reserved and divided accordingly. For example, RAM and storage are often divided into gigabyte chunks, while CPUs are divided into virtual CPUs (vCPUs). Each vCPU is roughly equivalent to a single CPU thread. For instance, Intel's Xeon Gold 5418Y processor has 24 cores, with each core supporting two independent threads. This means the processor is capable of supporting 48 vCPUs.

It is not always known how many other guests a server is shared with but it can be assumed that a server is shared with other residence by looking at statistics for average server utilisation in typical data centres [citation needed]. Armed with this information we can now begin describing the backend model starting with CPU utilisation:

3.3.1 Server CPU Utilisation

$$U_{\lambda g} = \left(\frac{N-K}{N}\right) A_g \tag{15}$$

$$U_{\lambda s} = \left(\frac{K}{N}\right) A_s \tag{16}$$

$$U_t = U_{\lambda g} + U_{\lambda s} \tag{17}$$

where: $U_{\lambda g}$ = guests proportion of the servers total CPU utilisation.
$U_{\lambda s}$ = your proportion of the servers CPU utilisation.
U_t = total server CPU utilisation.
A_g = average server utilisation in data centre.
A_s = average utilisation of your instance.
N = number of vCPU the reference server can accommodate.
K = number of vCPU your instance has.

3.3.2 Server CPU Power

$$P_{cpu} = \overbrace{\lambda_u f_{cpu}(U_t)}^{active} - \overbrace{f_{cpu}(0)}^{idle} \tag{18}$$

where: P_{cpu} = active CPU power consumption that you are responsible for.
λ_u = U_s/U_t ; proportion of CPU utilization you are responsible for.
f_{cpu} = CPU power curve.

3.3.3 Server Network Adaptor Power

$$P_{net} = \left(\frac{D_t}{D_{tr}}\right) P^a_{net} \tag{19}$$

where: P_{net} = power consumption of network adaptor during data transfer.
D_t = number of bytes transferred.
D_{tr} = data transfer rate in bytes per second.
P^a_{net} = power consumption of network adaptor during transfer.

3.3.4 Server Idle Power

$$P_{ser} = \lambda_r P^i_{ser} + P_{cpu} + P_{net} \tag{20}$$

where: λ_r = K/N ; proportion of resources (number of vcpu) allocated to you on the server.
P^i_{ser} = idle power consumption of the server (this includes the cpu and network adaptor idle power consumption).

4 Functions, Parameters & Assumptions

4.1 Functions

In section 3.1.1 and 3.3 we introduced the concept of energy consumption functions for various pieces of hardware. This section details the functions for all hardware components currently considered in the CWM.

4.1.1 CPU

CPU energy consumption is estimated using the Boavizta power consumption profile model [citation needed]. This model estimates power consumption using the following logarithmic function:

$$P(u) = a \cdot \ln(b \cdot (u + c)) + d \tag{21}$$

where: $P(u)$ = Power consumption (W) at utilization u (percentage).
a, b, c, d = Coefficients specific to CPU architecture and TDP.

This power consumption profile captures the non-linear relationship between CPU utilisation and its power consumption. The coefficients a, b, c and d define the shape of the power curve.

We obtain these coefficents by using non-linear regression against example solutions provided by the TEADS Thermal Design Power (TDP) scaling factor table [citation needed]:

TDP scaling factors				
u	0%	10%	50%	100%
$P(u)$	0.12	0.32	0.75	1.02

4.1.2 Network Adapter

In progress

4.1.3 Screen

Screen energy depends very much upon screen technology. For instance, LED screen energy depends upon the colour being rendered [citation needed]]whereas LCD screens colour makes neglible difference [citation needed].
So to calculate screen energy we need to make some assumptions based on the device type (mobile or desktop) what is the likelihood that screen is LED or LCD.
For LCD screens we use a baseline energy draw of ¿¿¿¿¿¿.
For LED screens we calculate the colour profile by taking snapshots of the rendered pages and forming a timeline covering the duration of the interactivity test. Each snapshot is analysed pixel by pixel to calculate an average pixel colour in RGB format. We then take each of the RGB values (which range from 0 to 255) and use them estimate screen energy as cited in ¿¿¿¿¿¿¿¿¿¿ [citation needed].

¡insert formula¿
Then, based on the device type we use statistical data to form LED:LCD weightings of X:Y for mobile [citation needed] and X:Y for desktop [citation needed] and apply these weightings to the energy calculations detailed above to form as single screen energy estimation.

¡insert formula¿
We then additionally apply the carbon intensity of the screen location as per the SCI.

¡insert formula¿

5 Reference Devices

Throughout this model reference devices are used to provide answers to questions related to hardware when details about the hardware are not known either because they can't be known - in the case of end user devices - or the information is not made readily available - in the case of servers running within a data-centre.

In the case where the hardware is known these reference devices are not required and data relating to the known hardware can be substituted in the CWM.

This section describes how these reference devices are chosen and how resources are mapped to the reference devices.

5.1 End-User Desktop

In progress

5.2 End-User Mobile

In progress

5.3 Servers

In progress

6 User Interaction

As described earlier, user interactions are the fundamental building blocks of the CWM. When a human (or bot) visits any web page or uses any application they are interacting with it. These user interactions can be anything from a simple page load, to adding items to a basket, processing a purchase, reading an article, watching a video, playing a game. We can think of the whole internet as a series of user interactions. Given that a user interaction is the fundamental unit of CWM, being able to run an interactivity test, we can do all sorts of interesting things. We will highlight two main use cases.

6.1 Estimating User Journeys

We can script (or record in tools like Chrome developer tools) re-playable user interactions, we call this an interactivity test. We can replay these scripted behaviours and use CWM to estimate energy and emissions of the specific user journeys. Thus, giving us the ability to consistently replay scenarios and optimise. Recognising key user journeys within sites and applications and optimising them gives development teams a great opportunity to reduce emissions. This aligns with SCI 'per R' functional unit.

6.2 Estimating Total Website Emissions

Given representative user analytics data we can estimate total emissions of all end user activity. To do this we identify representative user journeys and implement them as scriptable tests, running them across a representative sample of pages from a website (or sub section of a site). This is useful for the purpose of reporting end user emissions (which fall under Scope 3 in GHG protocol) [citation needed] but the total numbers obtained are not optimisable because the numbers are dependent on user activity which is out of our control. However, we can prioritise the user journeys (based on overall emissions) and use the approach in 6.1 to optimise as per SCI.

7 Future Work

References

[1] https://sci.greensoftware.foundation

[2] https://www.datacenterdynamics.com/en/news/global-data-center-electricity-use-to-double-by-2026-report

[3] https://sustainablewebdesign.org/estimating-digital-emissions/

[4] https://www.adriel.com/glossary/what-is-a-good-new-vs-returning-visitor-ratio

[5] https://www.statista.com/statistics/267018/global-market-share-held-by-pc-vendors/.

[6] Evaluating the Energy Consumption of Mobile Data Transfer—From Technology Development to Consumer Behaviour and Life Cycle Thinking