Towards Energy Monitoring in Visual Processing Pipelines

Hareesh Veekanchery

Reconfigurable Computing Systems Lab, Electronic & Electrical Engineering, Trinity College Dublin, Ireland

Helios Lyons

Disguise Systems Ltd, London, United Kingdom

Philip Coulam-Jones

Disguise Systems Ltd, London, United Kingdom

Shreejith Shanker

Reconfigurable Computing Systems Lab, Electronic & Electrical Engineering, Trinity College Dublin, Ireland

Written for presentation at the

SMPTE 2025 Media and Technology Summit

Abstract. *With over 80% of global internet traffic attributed to video content, and tentpole movies averaging 2,840 tonnes of CO2 emissions, the media industry faces urgent sustainability challenges that will intensify with the increased adoption of AI in production workflows. The industry is embracing techniques such as Virtual Production (VP), and its wider adoption is expected to reduce the energy demands and carbon footprint. However, there are no existing standards or tools available to developers of visual processing algorithms, techniques and systems to assess the energy footprint of existing workflows in post-production and VP, or to guide the development of new algorithms and tools that are optimised for energy consumption. The main contribution of this work is to outline frameworks for monitoring the energy consumption of existing video processing pipelines using a set of software and hardware tools, and thus establish a standardised method to perform energy consumption measurements/profiling at runtime. Two different approaches for energy monitoring are presented to* *understand the current power requirements of standard VP hardware. The first approach uses deployable out-of-band monitoring interfaces for real-time monitoring and capacity planning. The second approach builds on profiling techniques to characterise the accuracy of on-device and on-chip power measurements, developing an invasive scheme to characterise the energy costs of mapping existing VP tasks to specific resources (CPU or GPU). Together, they enable run-time monitoring and granular characterisation to aid with energy-aware development and deployment of post-production and VP workflows.*

Keywords. Energy efficiency, Virtual Production, Energy monitoring

Introduction

The media industry faces critical sustainability challenges. Digital media production and content delivery now accounts for over 80% of global internet traffic, while tentpole movies generate an average of 2,840 tonnes of CO_2 emissions [1]. New technological developments and increased adoption of AI-powered tools in content creation, production, and distribution could lead to multiplier growth in energy consumption and carbon emissions. Whilst Virtual Production (VP) techniques could reduce energy demands and carbon footprint [2, 3], no existing standards or tools are available to developers or creators to assess the energy footprint of existing digital workflows (post-production) or the digital synthesis techniques used in VP.

Existing profiling tools from manufacturers like NVIDIA SMI and Intel SoC Watch, along with cross-platform solutions such as RAPL, can estimate energy costs but only for specific hardware components. VP workflows present unique challenges as they involve complex functions executed across multiple computing elements such as CPUs, GPUs, ASICs, and FPGAs, to achieve optimal performance. Estimating the energy cost of end-to-end VP systems becomes even more complex when accounting for data storage, network interconnects, and display systems, especially in the context of high-resolution video data. This multi-component architecture makes comprehensive energy assessment difficult with current single-hardware profiling approaches.

The lack of standardised protocols for measuring energy consumption across VP ecosystems further compounds these challenges. Current methodologies, where they exist, are typically bespoke solutions tailored to individual systems, preventing industry-wide adoption and comparison. This work addresses these limitations by outlining standardizable pipelines and techniques for the broader media technology landscape. We present two complementary approaches: first, leveraging existing server management systems commonly found in media processing workstations for production-ready monitoring; second, developing invasive sensing schemes for precise benchmarking of new system configurations and algorithms. The data collected from these methods enables energy-aware workflow automation and granular monitoring tools that can estimate task-level energy consumption. This framework provides developers with the tools needed to validate whether new systems improve upon existing solutions and encourages greater focus on sustainability in future workflow development.

Additionally, the lack of a standardised protocol/framework for measuring the energy consumption of Virtual Production tools across the wider VP ecosystem is a limiting factor for understanding the energy overheads of current systems, and for the development of new energy-optimised workflows. Currently, these methodologies, where they exist, can be very bespoke and specialised to individual systems. This work outlines standardisable pipelines and techniques that can be adopted across the wider media technology landscape in order to encourage a more energy-conscious approach to developing new technologies and workflows. We outline two approaches, one which explores existing server management systems that are commonly part of media processing workstations, and secondly, an invasive sensing scheme for benchmarking new system configurations and algorithms. We show that the data collected from these methods can be used to develop energy-aware automation in workflows and for developing granular monitoring tools that can estimate task-level energy consumption to help develop energy-optimised workflows and tools. We believe that an adoptable flow, such as the one outlined in this work, will drive more developers to validate whether their new systems are greener than previous versions and ensure that more effort is put towards sustainability for new workflows in the future. The specific contributions of this work are outlined below:

- Developing a site-level monitoring framework for capturing energy consumption of media servers and systems on-site
- An architecture for an invasive hardware-based monitoring system for calibrating on-chip and on-device sensors and for developing energy-optimised algorithms
- Characterisation of existing workflows for virtual production and post-production tasks using the above tools to arrive at pathways for system-wide optimisation and energy estimation tools

The remainder of this paper is organised as follows. The related work section outlines existing tools and workflows for energy estimation and monitoring, from vendor flows to academic solutions. The methodology section describes the hardware and software architecture of the site-level monitoring framework and the invasive hardware monitoring system. The results section describes the test setup and observations from benchmarking virtual production systems and post-production tasks using industry-standard tools. The discussion section outlines the data collection and training flow for a non-invasive energy estimation flow that is enabled by the systems developed in this work. Finally, the conclusion section concludes the paper.

Related Work

Existing power measurement tools can be broadly categorised into schemes that directly measure system-wide power consumption, use hardware on-device and on-board sensors through software APIs, or indirect methods that use modelling techniques to estimate energy consumption on the CPU/GPU for different workloads [4–6]. System-wide power can also be measured through external AC power meters (wall power) and from power rail data on PSUs (where available) to capture a system-wide view of the run-time power consumption, voltage and current levels, and energy consumption [6]. Wall power measurements also factor in the inefficiencies of the power supply unit (PSU); however, these solutions are often unable to isolate a specific components' power consumption when executing a specific task, and hence offer limited potential for optimising an algorithm or a workflow during the development phase.

On-chip and on-device sensors are embedded in GPUs and specific motherboards, allowing for the monitoring of device-level / rail-level power consumption at runtime. Vendors provide software IPMIAPIs that can probe such sensors to compute and log the real-time power consumption, as seen in tools such as NVIDIA-SMI [7]. In high-end servers, such information can be extracted through out-of-band interfaces such as the intelligent platform management interface (IPMI), which probes voltage and current sensors on the motherboard to determine run-time power consumption [8]. However, multiple studies have identified the inconsistency of such measurements stemming from lower accuracy of sensing and limited data acquisition (sampling) rates [4,5,6,9,10]. Recent works have also shown that different CUDA versions only have a minor impact on the power measurements reported by NVIDIA-SMI, although variations exist between GPU generations [11]. Additionally, many vendors restrict access to IPMI through proprietary solutions, limiting their widespread adoption for high-precision energy monitoring.

Invasive measurement techniques have been shown to be the most accurate method to measure power directly off the internal power circuits. These approaches embed instrumentation circuits into the output of the PSU to monitor the CPU/GPU power lines directly. The main advantages of this approach are its higher sensitivity and acquisition capabilities compared to on-device and on-chip solutions. Multiple research papers have explored the use of invasive solutions to assess the power consumption of GPU and CPU while profiling specific tasks [12–17]. The invasive nature of this solution and the expensive sensing/acquisition systems limit its applicability for real-world profiling solutions in media servers for characterising hardware components and for benchmarking applications.

Energy estimation techniques are another widely used method to predict the energy consumed by applications on CPUs and GPUs. Popular methods, such as running average power limit (RAPL) and extensions pyRAPL [18], use low-level CPU metrics and performance counters to estimate the run-time power consumption to a high degree of accuracy. Internally, RAPL estimates are used by profiling tools such as Intel's VTune profiler and SoC Watch [19] energy estimation tools for Intel CPUs. Studies show that the RAPL estimates are within 10% to 22% of the ground truth energy measured by physical invasive sensors [20] when profiled on matrix multiplication tasks. Multiple tools have been built on RAPL and pyRAPL modules, such as CodeCarbon [21] and Experimental-Impact Tracker [22]. It should be noted that RAPL registers are limited to 32 bits, are not updated frequently or deterministically, and do not include timestamps for the data, making them less suited for long-term profiling. Additionally, RAPL and similar tools are unable to isolate a specific task from background activity, which poses significant challenges for systems where multiple tasks can be executed in parallel.

Predictive and simulation models are another way to estimate the power consumption of tasks on CPU/GPU platforms. Predictive models carefully model the system through extensive simulations or emulations, and by establishing correlation with low-level performance counters for estimating the energy costs [5,6]. Simulation models, however, should capture the architecture details of the CPU/GPU platform for accurate results. Wattch [23], McPAT [24] and GPUWatch [25] are examples of such frameworks, which model numerous internal blocks to accurately estimate energy costs and other performance parameters when running specific algorithms on them. Other tools use machine learning models to establish a correlation between low-level performance counters and reported power for a set of applications benchmarked from a training dataset [26,27,28,29,30]. However, their accuracy is driven by the diversity of the dataset and the accuracy of the ground truth environment.

In virtual production and extended reality (XR) environments, media servers and computing systems form one part of the chain, with LED displays and signal routing components forming the remaining energy-consuming components. It has been shown that display modules can consume up to 34% of the energy of the entire chain [31]. Other experiments [32, 33] have shown that the brightness of the colour and the content displayed have a significant impact on the power draw of LED walls. However, there is no tooling in existence that can pool the energy information from different resources in a VP/XR chain.

This work thus focuses on addressing the challenges of energy monitoring across the chain (for VP/XR components) for estimating end-to-end energy costs of a media pipeline and the low-sampling rate and accuracy of existing invasive measurement systems (on-device or invasive probes) to capture rapid power peaks caused by dynamic loads in post-production algorithms.

Methodology

The main contribution of this work is to develop a baseline for the energy consumption of existing video processing pipelines by building a set of tools and a standardised method to perform measurements at runtime.

One technique focuses on end-to-end monitoring through out-of-band services for a VR/XR platform, leveraging the management interface existing in XR servers and by developing tools to automate the measurements at runtime. The developed tools will characterise the end-to-end and machine-specific power consumption when running a curated set of test cases in a scaled model of a VP environment. The second technique aims to characterise the accuracy of on-device and on-chip power measurements through an invasive monitoring platform and subsequently characterise the energy costs of mapping existing VP tasks to specific resources (CPU or GPU) and use the framework to establish the correlation between low-level instructions being executed for these tasks and the energy costs they incur.

This dual approach provides both production-ready monitoring tools deployable on existing systems and precision measurement techniques for detailed optimisation during tool development and platform design, establishing a comprehensive framework for energy assessment of VP systems and energy-aware development of new VP solutions.

Out-of-band end-to-end energy profiling for XR

In XR pipelines, real-time rendering, compositing, and display operations demand substantial electrical power. For this work, we focus on hardware infrastructure, specifically power-intensive components such as Disguise media servers and LED display volumes. Power monitoring in these systems can be approached through either hardware or software techniques, leveraging existing sensors available through low-level monitoring interfaces or physically probing power pins. The approach discussed in this subsection leverages integrated sensors and non-invasive methods rather than developing custom external hardware. We used a combination of hardware and software approaches most suitable to the interfaces available in the standard Disguise system architecture shown in Figure 1.

Embedded Profiling Infrastructure

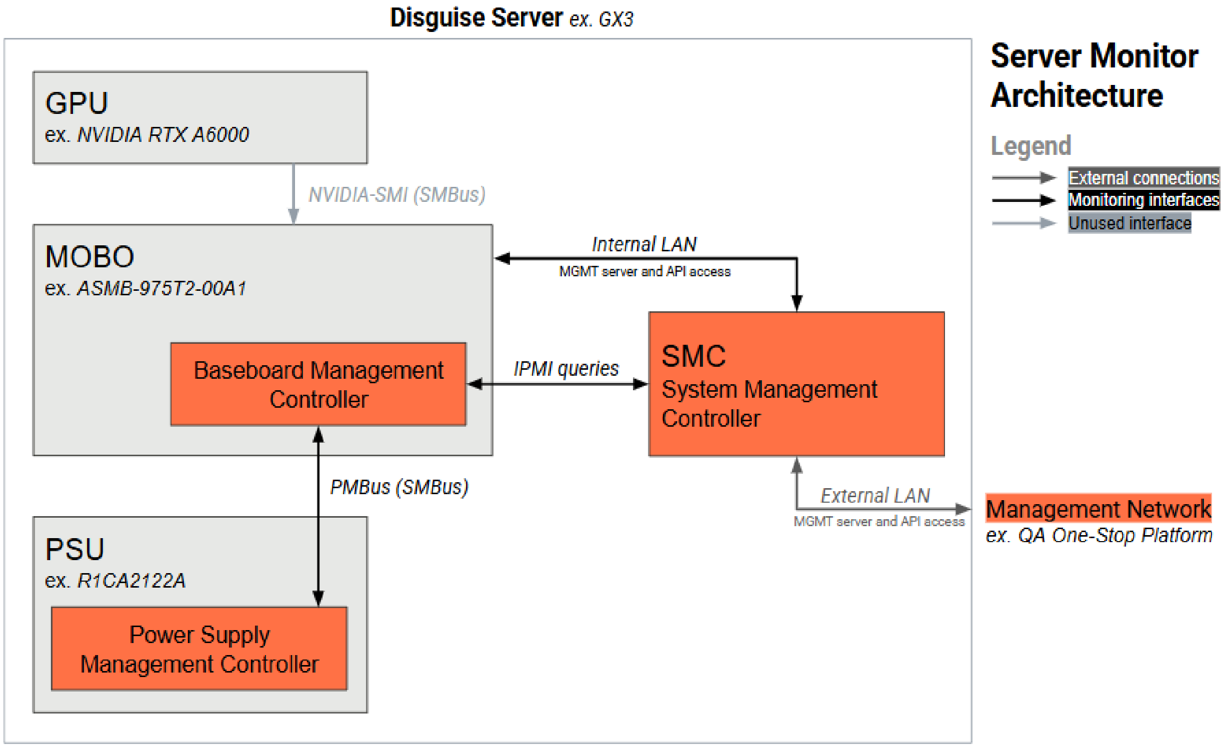

Figure 1. Internal media-server monitoring architecture (GX3) [36]

The principal monitoring agent is the System Management Controller (SMC) available on most Disguise systems. The SMC is a Disguise embedded platform that allows direct out-of-band interaction with server components via the Baseboard Management Controller (BMC) and host operating system [37] relying on open standards such as IPMI, SMBus, and PMBus.

As described in *Figure 2,* using the Intelligent Platform Management Interface (IPMI), we can query the BMC Sensor Data Repository, which collects readings from the PSU Power Supply Management Controller (PSMC) via the Power Management Bus interface (PMBus), a subset of the widely used System Management Bus interface. The PMBus protocol is an application-layer command set that operates over the transport layer of the System Management Bus (SMBus) [38,39].

Each SMC has tooling to support IPMI. Using this tooling, the contents of the Sensor Data Repository sourced from the PSU can be queried. An Infrastructure Assessment was conducted to evaluate which Disguise media-server architectures allowed this real-time monitoring. The primary advantage of this approach is that we avoid significant overhead on the measured system. The use of the NVIDIA System Management Interface (SMI) [7] was also investigated to be able to differentiate between overall system power draw and component-level draw.

PSU Characterization

The PSU unit in the servers under test was characterised for measurement accuracy and sampling rates, which are captured in Table 1.

Table 1: PSU parameters for the servers under test

Sampling Rate	Fixed at 1 Hz by the PSMC-interfaced sensor. ASPEED AST2600 BMC SoC has a significantly higher sampling rate. [40]
Resolution	Standard IPMI readings display 10W quantisation when integrated into the Sensor Data Repository. I2C raw commands are used to directly communicate with the I2C bus, bypassing the BMC's Sensor Data Repository (SDR) abstraction layer. [41]
Accuracy	Tests comparing PMBus with calibrated external meters showed a variance of ±1-2W. For characterisation, we used a steady-state testing sequence under controlled GPU load using Furmark at 11520×2160, 8× MSAA, 85-87% TDP.

For this development, SMC firmware was developed to query the BMC SDR at configurable intervals, extracting both input and output power metrics from the PSU. Excessive polling was avoided, which could otherwise result in duplicate readings as the PSU Management Controller pulls sensor data at a fixed 1 Hz interval regardless of IPMI polling frequency. [41]

Energy Profiling API

Power metrics are exposed through the SMC API endpoints, enabling integration with monitoring systems, such as the Disguise QA One-Stop Platform across management networks. This integration provides both real-time visualization and historical data collection. While PMBus does not provide individual power rail measurements (GPU-specific consumption), the total system power data correlates effectively with workload characteristics, as evaluated in initial calibrations. For component-level analysis, PMBus data can be supplemented with GPU-specific metrics from NVIDIA SMI for a more granular energy profile [7], though we leave this for future work. These monitoring capabilities are included in firmware release 2025.01 with full API documentation available to enable customers to develop custom power monitoring and optimisation solutions on developer.disguise.one. [42]

Site-Level System Monitoring

A similar framework was used to create a remote, automated power monitoring setup using Tapo P110 WiFi-enabled power monitors. These monitors exposed data over the network on scrapable endpoints, allowing automation of power data collection on devices without internal profiling capability. This technique was used to characterise the power consumption of LED volumes - an extension of the initially stated aims - providing readings of XR stage power usage and further allowing realistic and distributed analysis. The architecture of the L2 stage media server and LED volume monitoring is shown in Figure 2. The “L2 Stage” is described in the following XR Stage Evaluation section. An independent setup was defined to use this integrated monitoring system to conduct explicit tests on XR workloads. The broad system architecture is visible in Figure 2. In practice, this setup comprised a single output machine (in-house VX4+) in conjunction with a Brompton S8 processor and 36 DB2.6 V2 LED panels. [43,44].

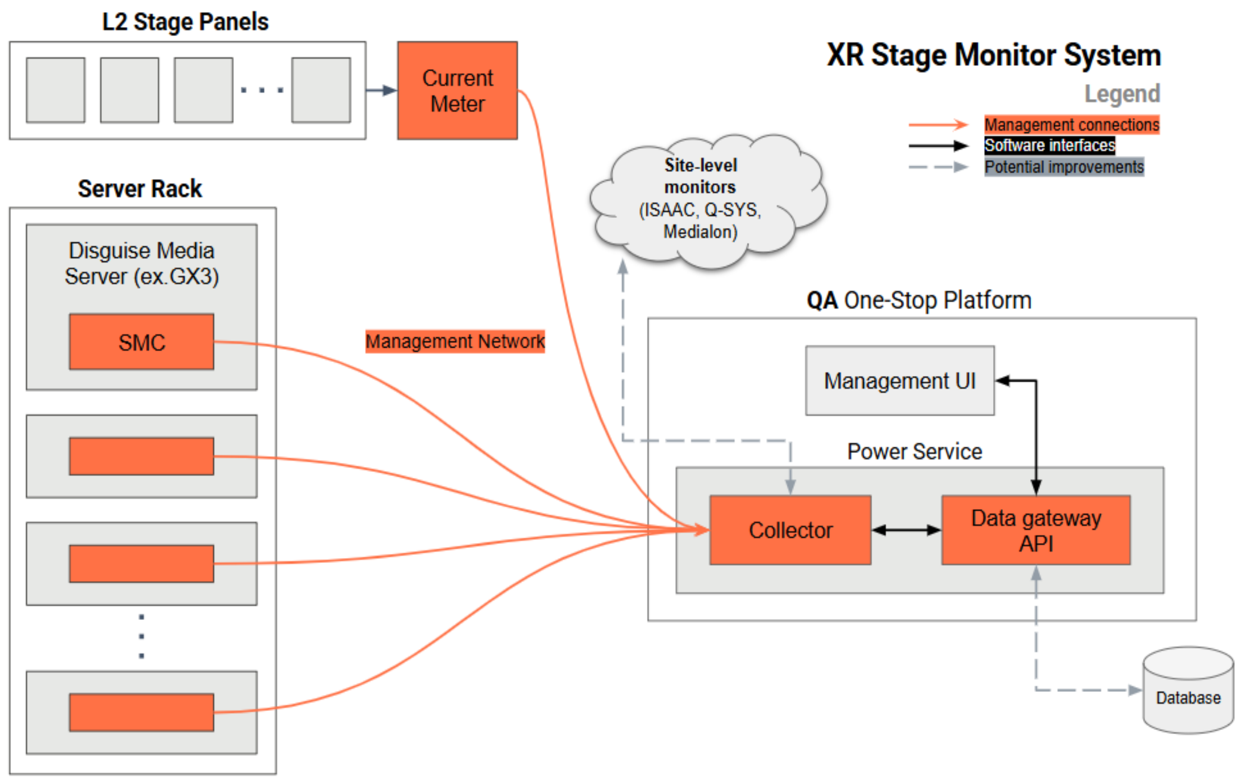

Figure 2: L2 stage media-server and LED volume monitoring architecture (representative)

XR Stage Evaluation

A structured test was designed to get readings from the LED panel and output VX4+ in several states. The test sequence is illustrated in Table 2. An initial baseline test with no content being displayed (black screen) was conducted, followed by single colour tests (R, G, B, W) at various brightness levels (0-100%) with 5-minute durations for each level. This test sequence was repeated twice, while varying only the light intensity as set on the Brompton Tessera S8 LED processor between 500 nits and 1000 nits maximum. 1000 nits is the practical maximum for most XR stages, as described by the Disguise Support team, and the additional 500 nits test allows us to cover a greater range of intensities on the panels (from 1W25% at 125 nits to 2W100% at 1000).

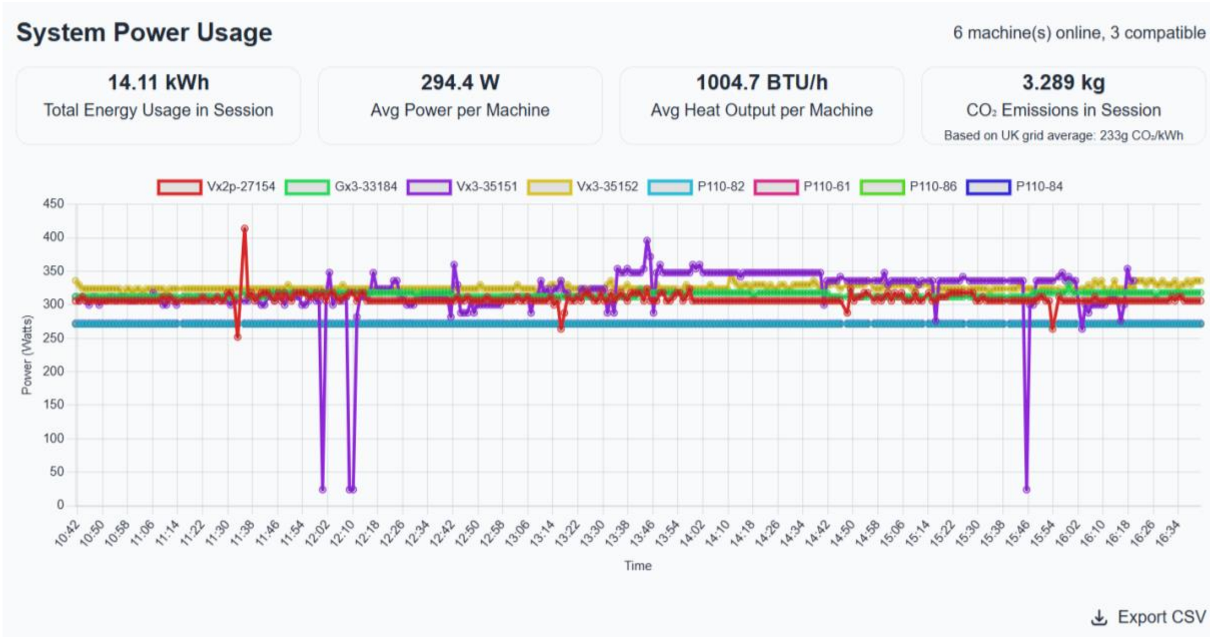

Figure 3: Site-level monitoring on QA platform, metrics collected from network SMCs via new API

Table 2: Full test sequence

Color	**Test Time** *(mins for each brightness setting)*	**Brightness Setting** *(% set in designer software)*	**Global Maximum** *(nits set on LED processor)*
Black (none)	5	25, 50, 75, 100	NA
White	〃	〃	500
Green	〃	〃	500
Red	〃	〃	500
Blue	〃	〃	500
White	〃	〃	1000
Green	〃	〃	1000

Red	〃	〃	1000
Blue	〃	〃	1000

The input power of the content output VX4+ machine was measured during this test; however, we observed an error with the PSU sensor and determined that the results were spurious, and hence did not report them.

Invasive Energy Monitoring and Granular Energy Estimation

While dedicated media solutions developed in the previous Section for XR workflows incorporate SMC to monitor power consumption using on-board sensors, multiple research publications have identified inaccuracies with on-board sensors. Power monitoring using SMC is also infeasible for bespoke elements within production pipelines. We investigate the accuracy of on-board and on-device sensors in practical systems to characterise their accuracy, develop methods to capture energy consumption in a granular fashion and establish the groundwork for granular energy consumption modelling in post-production pipelines.

To characterise on-device power sensors, we develop an invasive power measurement system and instrument a model server architecture with different computing elements (CPU, GPU, FPGA). Subsequently, the power consumption from each element is characterised using different workflows in a professional compositing environment, using tasks that are mapped to these specific resources. Combining this power profiling with low-level instruction profiling (using tools such as Intel VTune), we develop a granular energy estimation tool that uses the correlation between low-level performance counters signatures for specific tasks and the power draw signatures to predict the energy consumption of the specific task independently.

Hardware-based power measurement

A DC monitoring system is developed that can monitor power consumption on systems that do not feature an SMC (most general workstations). Two sensor options are provided to measure current consumption directly off the power lines that connect to the CPU and GPU: contactless split-core hall-effect sensors and a miniature coreless magnetic current sensor (TLI4971). Contactless clamp-on sensors allow current measurement without interrupting or modifying the main circuit. Unlike clamp-on sensors, the TLI4971 needs the main circuit to be broken to pass current through its internal rail.

For acquiring the data, we evaluated two options: the integrated 10-bit ADC on the Arduino, which supports a maximum of 10K samples per second, or a dedicated high-speed 10-bit MCP3008 ADC that supports a peak rate of 200K samples per second. The latter was interfaced over the available SPI interface on the Arduino device.

The measurement chain is calibrated for the entire range of operation of the hall-effect sensors using calibrated loads, power supplies and measurement probes in TCD, using TTi EL-R Series

Digital Bench Power Supply readings, BSIDE ACM 92 DC Current Clamp Meter and programmable loads. The sensors are then instrumented into a model workstation that uses a high-end Intel processor and NVIDIA GPUs. The specification of the measurement system is listed in Table 3.

Table 3: Specifications of the invasive measurement system based on hall effect current transformers.

Rated Input	25A, 30 A, 200 A
Input measurement range	45A, 200A
Rated output	2.5 ± 0.625 V
Sampling frequency	10 K samples/second, 200 K samples/second
Sensitivity of the sensor	20.8333 mv/A, 3.125 mv/A,48 mV/A

Granular energy estimation model for CPUs

As mentioned, a key challenge of existing energy estimation/prediction models is their inability to isolate the task from background activity while estimating the energy cost. To this end, we developed an energy profiling model that can extract the low-level parameters specific to the task under consideration and use a lightweight learning model to estimate the energy overhead. To achieve this, we use Intel's VTune profiling framework to invoke the task under consideration and extract the low-level metrics associated with this task. The extracted VTune profiling data for each application is aggregated and averaged over multiple runs to create an instruction-level signature for each application. Simultaneously, the invasive hardware measurement system records the power consumption at a high sampling rate to capture power peaks and troughs, generating a unique power signature for the task. We profile a set of media applications through Foundry's NUKE tool, using internal plugins, custom graph operations and open source applications from Cattery, Foundry's community framework, simultaneously generating the power consumption profile for each of these tasks. Together, they form an energy-instruction dataset, which we used to train a random forest model to establish the correlation between task patterns and power consumption signatures. A random forest regressor with 100 decision trees was chosen as this provided the best results for our training and test datasets generated from the profiling tasks. For validating the model, we use an unseen set of tasks within NUKE, invoked through the VTune profiler to generate the low-level signature data, which is then fed as input to the estimator. The results are compared against baseline measurements from the invasive hardware monitor to evaluate the estimation accuracy across different tasks.

Results and Discussions

In this section, we first discuss the observations from the non-invasive end-to-end monitoring framework that can be deployed in production-ready environments, focused on the PMBus interface and external modelling of LED volume power. Figure 3 shows the results of the site-level monitoring on the QA platform, displaying the metrics collected from network SMCs for the system described in Figure 2. Initially, RS PRO Energy Meters were used to validate internal PSU measurements and moved toward TAPO P110 Wifi enabled meters for stage measurement. Figure 3 shows the results from in-house servers as well as power supply monitors (TPLink P110 devices) that track the 9 DB2.6 Infiled LED panels.

LED Panel Characteristics

Monitoring the power consumption of an entire LED volume presents significant challenges. LED walls are a primary energy consumer in virtual production, and their power draw is content-dependent. The lack of standardised energy reporting protocols across the heterogeneous hardware from different manufacturers complicates efforts to build accurate energy models. This section details a characterisation of a specific LED volume to establish a baseline methodology without access to detailed diode characteristics or the functional impact of LED processors such as the Brompton S8 [45].

Initial power consumption analysis of the DB2.6 S8 V2 LED panels based on the architecture (shown in Figure 2) is illustrated in Figure 4. The analysis revealed patterns across different colours and brightness levels. Unexpected discrepancies were found in power consumption between theoretically equivalent brightness levels from the analysis of white content between the 500-nit and 1000-nit maximum tests. This discrepancy can be observed in Figure 4, where at 500 nits maximum, white input power does not correspond to 50% brightness (white value) at 1000 nits, which was the assumption.

This discrepancy arises from the sRGB colour space [46] employed by display systems, which incorporates a non-linear transfer function to match human visual perception. The sRGB standard applies a gamma correction of approximately 2.2 to map between linear light intensity and the encoded digital values. To accurately correlate power consumption with actual light output, the following methodology was implemented:

(1) Convert the fraction f from sRGB colour back into linear intensity by applying the inverse gamma correction ($f^{2.2}$), which transforms the perceptually-uniform software setting back to actual light intensity.
(2) Multiply that linear intensity by the max nits setting configured on the LED processor (500 or 1000 nits in our case) to determine the luminance.
(3) Use this "Corrected Nits" value, which better reflects actual photometric light output for power consumption analysis, as LED power draw correlates with actual luminance rather than software-encoded values.

This analysis revealed that a software intensity setting of 50% produces approximately 22% of maximum brightness (since 0.5 raised to the power of 2.2 equals approximately 0.22), and achieving 50% actual brightness requires a software intensity setting of approximately 73.5%. Power consumption correlates more closely with actual light output than with software intensity settings, which can be seen in the greater linearity of the transformed brightness values in Figure 4. From our tests, we observe that setting a panel brightness to 50% in software does not reduce the power consumption by 50% of the peak consumption, but rather to approximately 22%, indicating a non-linear relationship where power scales with actual luminance rather than encoded values. This also points to our prior discussion that the frame mapping to the panel is effectively a black box, and the practical difficulties in characterising the exact behaviour of the LED processing pipeline in relation to the diode outputs, as this information is withheld by the manufacturers.

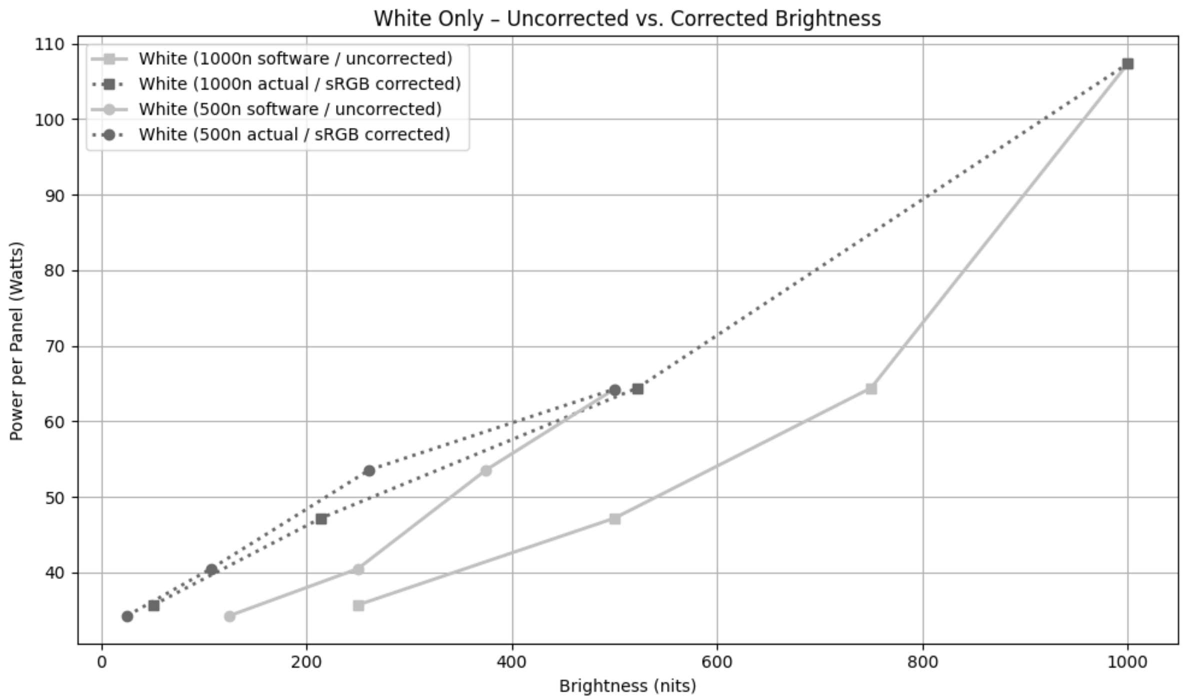

Figure 4: LED volume power vs. brightness, showing sRGB correction for white values

The corrected curves for all colours (along with the greyed-out original curves) based on the Test pattern (defined in Table 2) can be observed in Figure 5. There is clearly some non-linearity in the 500 nit test in the behaviour of the Red, Green, and Blue colours, though White seems to more consistently conform to the sRGB colour space.

As access to the problem-space is limited - considering limited resources in modelling for XR pipeline hardware outside of Disguise products - the scope of this study is limited in its further analysis, and only rough coefficients to approximate colour behaviour at differing LED processor brightness levels can be provided [45].

An extension of this work that would be enabled by a wider variety of maximum brightness level tests, allowing the full characterisation of the setup to provide predictions of DB2.6 power usage given a matrix of pixel values, or an input frame. This could serve as the basis for a framework for characterisations of additional LED panels, and thus the prediction of power consumption for a larger variety of XR setups.

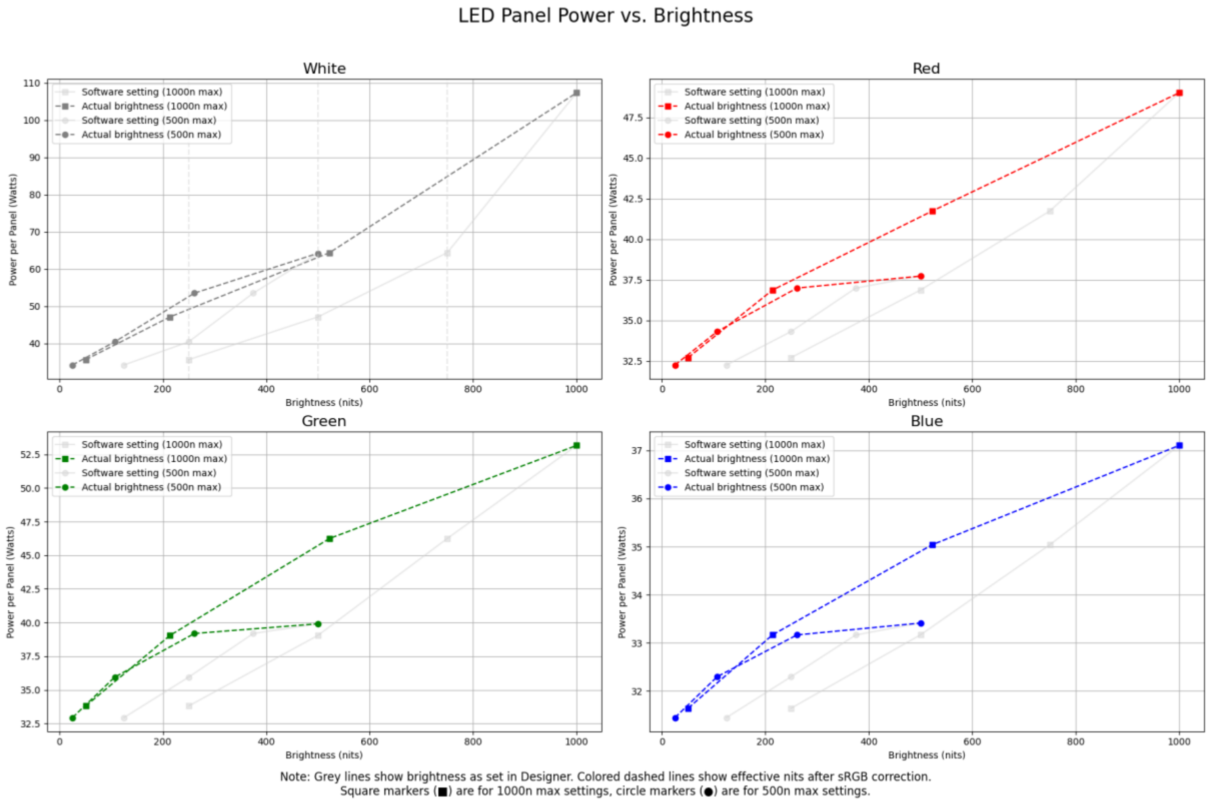

Figure 5. LED volume power vs. brightness, showing sRGB correction for all colours

Implications

Monitoring and understanding sources of power consumption can provide environmental and cost-saving benefits in the short term. Based on this study's methodology for system monitoring and subsequent cost optimisation, the overall XR system draw can be predicted using panel characteristics, allowing better capacity planning in the long term. The continued expansion of the XR pipeline scope and complexity will show increasing power demands and highlight the need for precise, real-time monitoring to inform more efficient in-panel and pipeline hardware choices [47].

As a result of the instrumentation of the Disguise in-house XR stage, it was found that the stage was left on during weekends and that a screensaver was playing during every weekday. This amounted to ~6.51 kWh draw per 9 panels each weekday, dropping to ~2.8 kWh weekend-day usage. Based on these measurements, yearly power consumption was around ~7,936 kWh for the whole L2 stage.

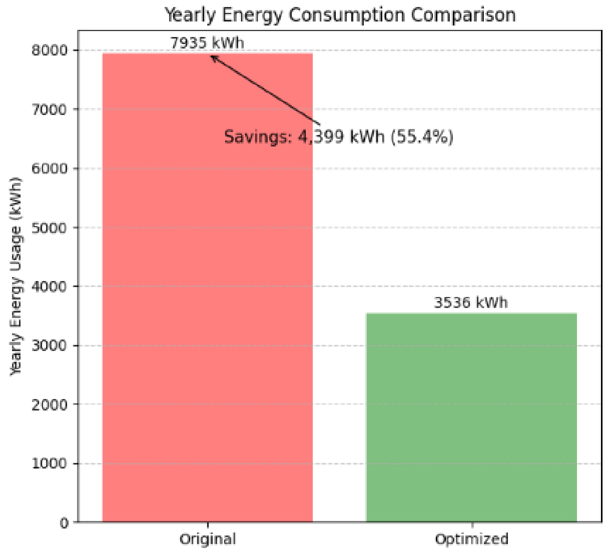

Figure 6: Comparison of previous L2 stage power consumption and optimised consumption

Considering that the stage was not permanently in use, a smart power-monitoring app was used to schedule time off over the weekend and regulate it during the week. By transitioning to this setup, yearly power consumption was reduced from ~7,936 kWh to ~3,224kWh, or roughly 59.4% as illustrated in Figure 6. Beyond operational benefits, the original annual carbon footprint of around 1,436 kgCO_2e for the L2 stage dropped by ~583 to around 853 kgCO_2e.

Our observations on the test platform can be adopted by users to examine their practices, specifically with respect to LED panels, where turning off the panels when not required saves significantly more energy than switching to a dark image. As such, while the implemented optimisations reduce overall power draw significantly, the usage of the P110s and Tapo's monitoring app will have to be disseminated for the effects to scale. In summary, collecting real-time metrics through the P110s and our new SMC Power API allowed Disguise to achieve cost and environmental savings without impacting operational capabilities. Table 4 captures the overall savings, which were estimated by adopting this flow in a Disguise test setup.

Table 4: Overall savings achieved through the energy-aware optimisation

Total Annual Consumption (original setup)	~7,936 kWh
Total Annual Consumption (optimised setup)	~3,224 kWh
Annual Energy Savings	~4,712 kWh (≈59.4%)
Overall Carbon Reduction	from ~1,436 kg CO_2e down to ~853 kg CO_2e (saving ~583 kg CO_2e)

Evaluating the accuracy of on-chip and on-device sensors

To understand the limitations of on-device sensors and vendor-provided APIs, we ran a set of benchmarking applications using Foundry's NUKE, using in-built example designs, custom node graphs and user-provided graphs through Foundry's open-source community network, Cattery. The tasks were mapped to CPU or GPU, with energy profiled through vendor tools such as NVIDIA-SMI. Simultaneously, the invasive energy monitoring module was attached to the GPU power lines from the PSU and the 12V board supply to capture the power delivered over the PCIe interface. Figure 7 shows the GPU utilisation and power consumption for a selection of NUKE tasks, measured through NVIDIA-NVML and NVIDIA-SMI interfaces when executed on an RTX 3080-Ti device on a standard Dell Workstation. We also benchmarked different resolutions for each task to push the resource utilisation on the GPU. Each task is run over a thousand times to ensure there are limited OS-level or interface (PCIe)-level biases to the measurements, and the 'reported' energy consumption and resource utilisation were averaged over these runs. The out-of-band energy monitoring tool is used simultaneously to log energy consumed by monitoring the GPU power lines. The key observation is that for compute-intensive tasks, the compute and memory utilisation, and thus the energy consumption, are impacted by the image resolution. A key observation in Figure 7(b) is the overshoots observed at different resolutions, hinting at sampling inaccuracies and/or the sensitivity of the measurements.

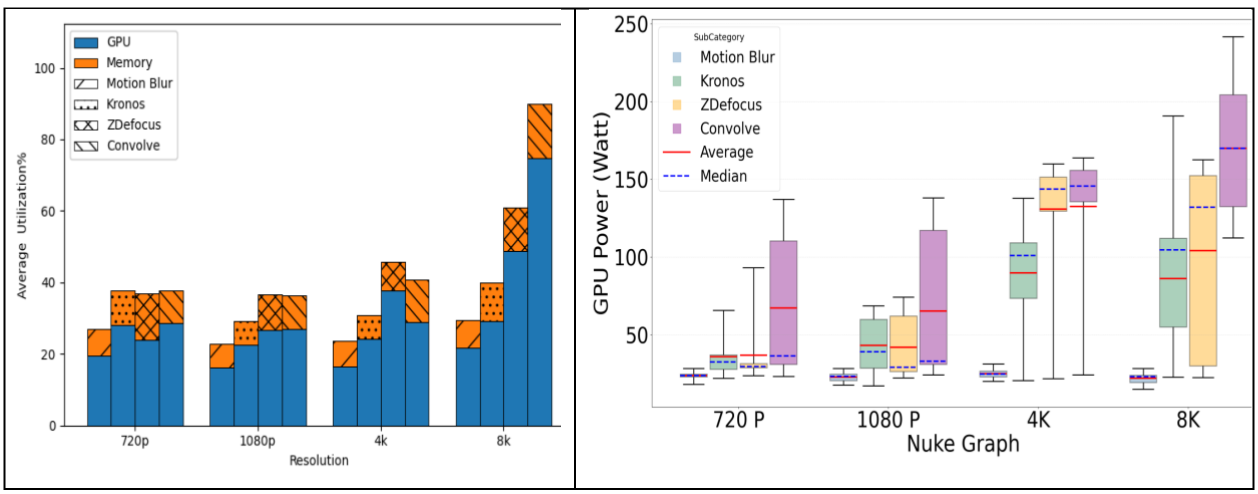

Figure 7: Analysis of GPU utilisation and GPU power (a) Plot of GPU and memory utilisation measured using NVIDIA tools across a set of representative NUKE plugins at different resolutions (b) Reported energy consumption across tasks for different resolutions

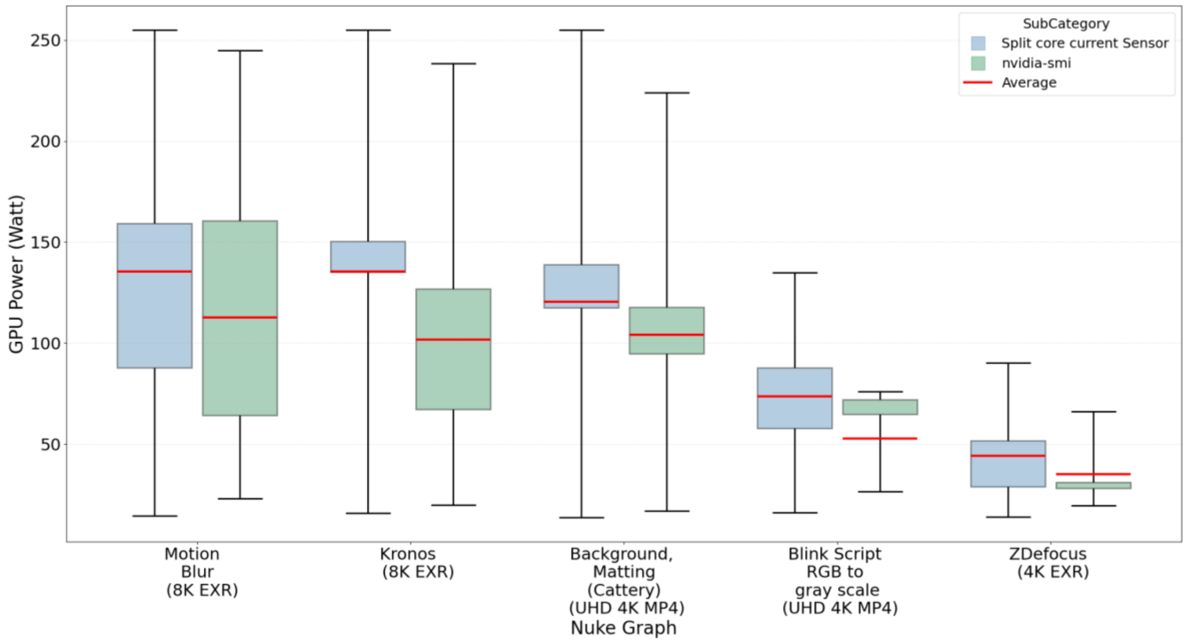

Figure 8: Comparison of the 'reported' power consumption against the measured power consumption for a set of representative NUKE plugins.

Figure 8 shows the comparison of the 'reported' power consumption from on-chip sensors and the measured power from the invasive sensors across different ranges of power consumption on the GPU. It can be seen that the on-chip sensors are deviating from the measured power consumption in all tasks, consistent with observations from previous research.

CPU energy estimator benchmarking

To characterise the energy estimator, we implemented multiple graph sequences on NUKE to provide unseen visual processing workflows. The estimation tool predicts the average power consumption as well as the maximum and minimum power consumption for the runtime of the task. Figure 9 shows the comparison of the predicted power consumption values (average, minimum and maximum) from the initial version of the energy estimator tool compared to the average, minimum and maximum power consumption measured by the invasive sensors. The NUKE tasks were set to execute on an Intel i9-10900KF CPU in a standard Dell Workstation. It can be observed that the minimum and average predictions have high correlation with the measured values, while the peak estimates show some deviation from the measured values in some cases. We are currently investigating the outlier cases for the peak power prediction to bring it in line with the minimum and average predictions. An interesting observation is that the power consumption estimated by the tool is not affected by the presence of other computing loads on the system (e.g., another task in the background), which is a unique ability of this tool compared to other solutions such as RAPL. Figure 10 shows the impact of running a NUKE graph firstly in isolation and subsequently in the presence of a background task (CPU stress) on our test setup. It can be observed that the additional load on the CPU results in our benchmark task consuming slightly lower power (with a slightly longer runtime) and is in line with the predicted power consumption when the task was executed in isolation. In the case of RAPL, the tool estimates the total CPU package power consumption, limiting its applicability for energy-driven task optimisations.

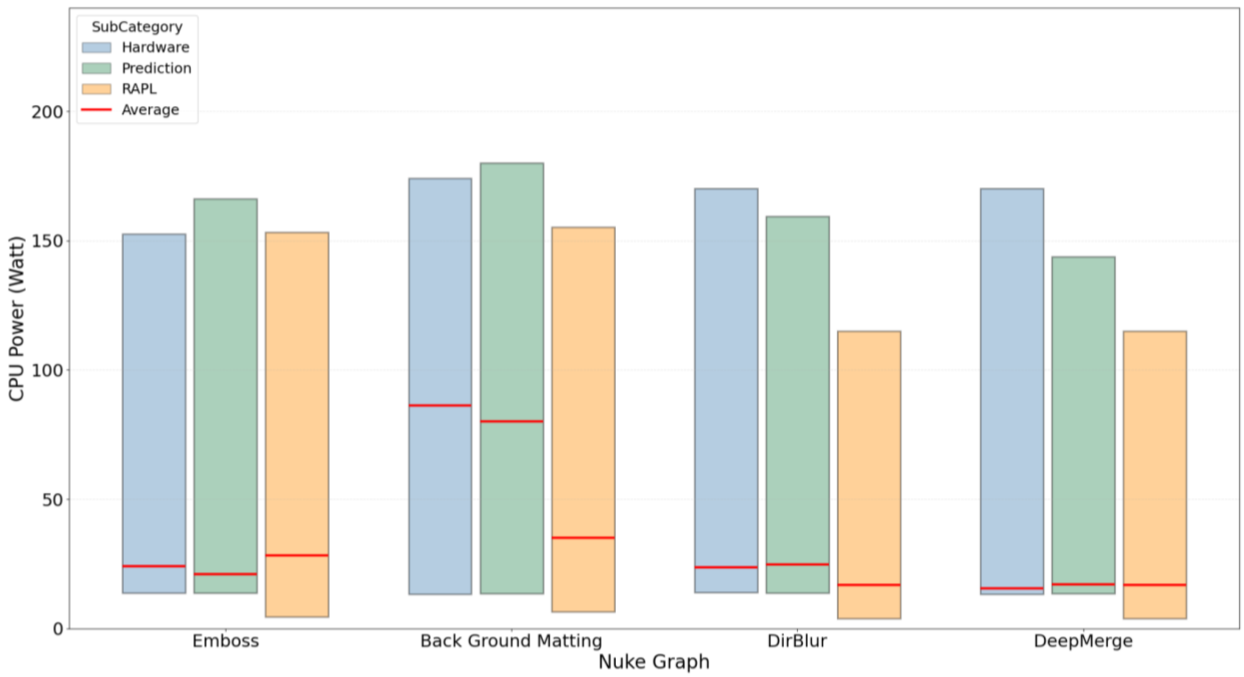

Figure 9: Comparison of the 'estimated' power consumption by the CPU energy estimator tool against the measured power consumption and predictions from RAPL for a set of representative NUKE plugins.

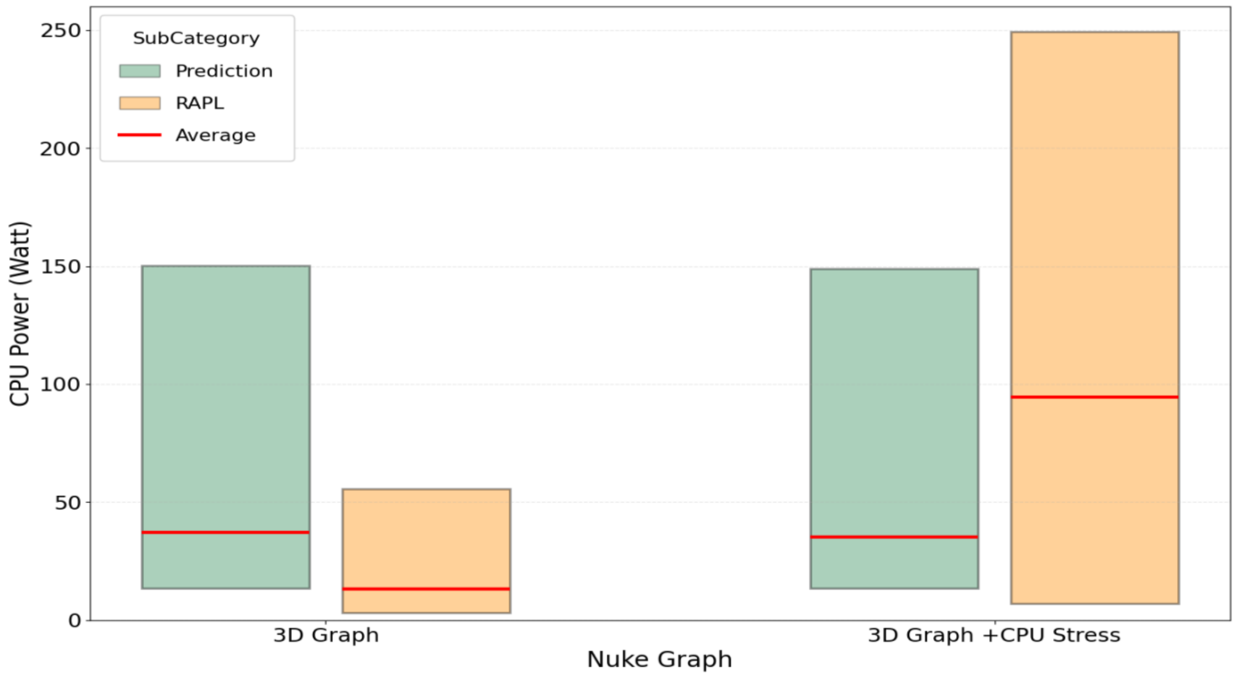

Figure 10: Comparison of the estimated power consumption by the CPU energy estimator tool against predictions from RAPL when the NUKE task is run in isolation versus the case where background tasks are active.

CONCLUSION

This paper presents two complementary methods for tracking the energy consumption of post-production and XR workflows, taking into account both computational and display components. An end-to-end framework is developed to characterise run-time energy consumption of individual servers and display components in an XR workflow, in addition to a granular invasive measurement system and an energy profiling tool that can provide insights into task, algorithm or device-level energy consumption. The key aim is to arm users and developers with these tools to aid in developing approaches for energy optimisation, environmental saving and cost reduction for media production pipelines. Our results show that the energy-awareness in the case of end-to-end virtual production pipelines can be used to develop automations that can reduce the overall energy consumption and carbon footprint of XR setups. Similarly, granular energy profiling of tools, invasive or predictive, can enable pathways for design-time optimisations to be explored for enhancing the processing algorithms, and/or platform configurations for improving the energy overheads of the algorithms.

In the future, we aim to expand the end-to-end workflow and automations further by improving the monitoring capabilities, developing a better understanding of display systems through extensive energy consumption-based characterisation and expanding the granular profiling flow for other computing resources.

ACKNOWLEDGEMENT

This work was funded by the Horizon CL4 2022 - EU Project Emerald – 101119800.

REFERENCES

[1] Albert, British Film Institute and Arup, "A screen new deal: A route map to sustainable film production," Tech Report, wearealbert.org, Mar. 2021. [Online]. Available: https://wearealbert.org/wp-content/uploads/2021/03/Screen-New-Deal-Report-1.pdf. [Accessed 25 Aug. 2025].

[2] V. Helzle, S. S. and T. J., "Green Screens, Green Pixels and Green Shooting," in ACM SIGGRAPH Talks, Feb. 2022. [Online]. Available: https://animationsinstitut.de/files/public/images/04-forschung/Publications/FABW_VirtualProductionSustainabilityReport_2_22_EN.pdf.

[3] D. Keeney, "Virtual Production's Role in Carbon Reduction and Net Zero Production in the Screen Industries," Future Observatory Research Report, 2023. [Online]. Available: https://futureobservatory.org/files/dcmsreports/futureobservatory_culturalpolicyreport_studioulster.pdf. [Accessed 25 Aug. 2025].

[4] M. Jay, V. Ostapenco, L. Lefèvre, D. Trystram, A.-C. Orgerie and B. Fichel, "An experimental comparison of software-based power meters: focus on CPU and GPU," in 2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid), pp. 106-118, 2023.

[5] S. Song, C. Su, B. Rountree and K. W. Cameron, "A simplified and accurate model of power-performance efficiency on emergent GPU architectures," in IEEE 27th International Symposium on Parallel and Distributed Processing, pp. 673-686, 2013.

[6] R. A. Bridges, N. Imam and T. M. Mintz, "Understanding GPU power: A survey of profiling, modeling, and simulation methods," ACM Comput. Surv., vol. 49, no. 3, pp. 1-27, 2016.

[7] NVIDIA, "System Management Interface" System Management Interface program [Online] Available: https://docs.nvidia.com/deploy/nvidia-smi/index.html

[8] Kavanagh, Richard, Django Armstrong, and Karim Djemame. "Accuracy of energy model calibration with IPMI." 2016 IEEE 9th International Conference on Cloud Computing (CLOUD). IEEE, 2016.

[9] Z. Yang, K. Adamek, and W. Armour, "Part-time power measurements: NVIDIA-smi's lack of attention," arXiv preprint arXiv:2312.02741, 2023.

[10] Z. Yang, K. Adamek, and W. Armour, "Accurate and convenient energy measurements for GPUs: A detailed study of NVIDIA GPU's built-in power sensor," in Proc. SC24: Int. Conf. High Performance Computing, Networking, Storage and Analysis, 2024, pp. 1–17.

[11] Yoshida, K., Shinobu, M., Yamaki, H., & Honda, H. (2024). Analyzing the impact of CUDA versions on GPU applications. Parallel Computing, 120 .

[12] X. Ma, M. Dong, L. Zhong, and Z. Deng, “Statistical Power Consumption Analysis and Modeling for GPU-based Computing,” in *Proceedings of the 2nd Workshop on Power Aware Computing and Systems (HotPower ’09)*, 2009.
[13] Q. Zhao, H. Yang, Z. Luan, and D. Qian, “POIGEM: A programming-oriented instruction level GPU energy model for CUDA program,” in *Proc. 13th Int. Conf. Algorithms and Architectures for Parallel Processing (ICA3PP)*, Springer, 2013, pp. 129–142.
[14] W. Choi, D. Bedard, R. Fowler, and R. Vuduc, “A roofline model of energy,” in Proc. 27th IEEE Int. Symp. Parallel and Distributed Processing (IPDPS), Boston, MA, USA, May 2013, pp. 661–672.
[15] D. Bedard, M. Y. Lim, R. Fowler, and A. Porterfield, “Powermon: Fine-grained and integrated power monitoring for commodity computer systems,” in Proc. IEEE SoutheastCon, Concord, NC, USA, Mar. 2010, pp. 479–484.
[16] S. Sen, N. Imam, and C.-H. Hsu, “Quality assessment of GPU power profiling mechanisms,” in Proc. IEEE Int. Parallel and Distributed Processing Symp. Workshops (IPDPSW), Vancouver, BC, Canada, May 2018, pp. 702–711.
[17] J. H. Laros, P. Pokorny, and D. DeBonis, “PowerInsight: A commodity power measurement capability,” in Proc. Int. Green Computing Conf. (IGCC), Arlington, VA, USA, Jun. 2013, pp. 1–6.
[18] K. N. Han, M. Hirki, T. Niemi, J. K. Nurminen, and Z. Ou, “RAPL in action: Experiences in using RAPL for power measurements,” ACM Transactions on Modeling and Performance Evaluation of Computing Systems, vol. 3, no. 2, pp. 1–26, 2018.
[19] Intel Corporation, “Get Started with Energy Analysis,” Intel® SoC Watch Get Started Guide (Version 2023.1), Mar. 31, 2023. [Online].
[20] J. M. Paniego, S. Gallo, M. Pi Puig, F. Chichizola, L. De Giusti, and J. Balladini, “Analysis of RAPL Energy Prediction Accuracy in a Matrix Multiplication Application on Shared Memory,” in Communications in Computer and Information Science: Computer Science – CACIC 2017, vol. 790, Springer, Cham, 2018, pp. 37–46, doi: 10.1007/978-3-319-75214-3_4.
[21] CodeCarbon, “Track and reduce CO_2 emissions from your computing,” CodeCarbon – Home Page, [Online]. Available: https://codecarbon.io/.
[22] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, “Towards the systematic reporting of the energy and carbon footprints of machine learning,” Journal of Machine Learning Research, vol. 21, no. 248, pp. 1–43, 2020.
[23] D. Brooks, V. Tiwari, and M. Martonosi, “Wattch: A framework for architectural-level power analysis and optimisations,” in Proceedings of the 27th International Symposium on Computer Architecture (ISCA ’00), 2000, p. 83.
[24] S. Li, J.-H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, “McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures,” in Proc. 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), New York, NY, USA, Dec. 2009, pp. 469–480.

[25] J. Leng, T. Hetherington, A. ElTantawy, S. Gilani, N. S. Kim, T. M. Aamodt, and V. J. Reddi, “GPUWattch: Enabling energy optimisations in GPGPUs,” in Proc. 40th Annual Int. Symp. Computer Architecture (ISCA ’13), 2013, pp. 487–498, doi: 10.1145/2485922.2485964.
[26] P. Pathania, R. Mehra, V. S. Sharma, V. Kaulgud, S. Podder, and A. P. Burden, “Esave: Estimating server and virtual machine energy,” in Proc. 37th IEEE/ACM International Conference on Automated Software Engineering (ASE), Luxembourg, Sept. 2023, pp. 1–12. Association for Computing Machinery.
[27] X. Fan, W.-D. Weber, and L. A. Barroso, “Power provisioning for a warehouse-sized computer,” in Proc. 34th Annual International Symposium on Computer Architecture (ISCA), San Diego, CA, USA, Jun. 2007, pp. 13–23. Association for Computing Machinery.
[28] L. D. Gray, A. Kumar, and H. H. Li, “Workload characterization of the SPECpower_ssj2008 benchmark,” in Performance Evaluation: Metrics, Models and Benchmarks, S. Kounev, I. Gorton, and K. Sachs, Eds. Berlin, Heidelberg: Springer, 2008, pp. 262–282.
[29] H. Nagasaka, N. Maruyama, A. Nukada, T. Endo, and S. Matsuoka, “Statistical power modeling of GPU kernels using performance counters,” in Proc. Int. Green Computing Conf. (IGCC), Chicago, IL, USA, Aug. 2010, pp. 1–8,
[30] G. K. Alavani, J. Desai, S. Saha, and S. Sarkar, “Program Analysis and Machine Learning-based Approach to Predict Power Consumption of CUDA Kernel,” ACM Transactions on Modeling and Performance Evaluation of Computing Systems, vol. 8, no. 4, pp. 1–24, Dec. 2023, doi: 10.1145/3603533.
[31] D. Li, H. You, W. Jiang, X. Chen, C. Zeng, and X. Sun, “Enhanced power saving schemes for extended reality,” in Proceedings of the IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Sep. 2021, pp. 1–6.
[32] Barco, “Energy efficiency of different display technologies in control rooms,” Barco, 2023. Available:https://www.barco.com/en/inspiration/news-insights/2023-05-24-energy-efficiency -of-different-display-technologies-in-control-rooms
[33] P. Yuliantoro, S. Romadhona, and D. O. Ayuninda, “Monitoring Power Consumption on LED Using LED Matrix P10 Panel,” Journal of Applied Information and Communication Technologies (JAICT), vol. 8, no. 1, p. 146, Mar. 2023, doi: 10.32497/jaict.v8i1.4072.
[34] R. Ganber, *NUKE 101: Professional Compositing and Visual Effects*, New York, USA: Peachpit Press, 2014.
[35] J. Reinders, *VTune Performance Analyzer Essentials*, vol. 9, Santa Clara, CA, USA: Intel Press, 2005.
[36] Disguise Systems Ltd., “GX3 Specification Sheet,” 2023. Available: https://www.disguise.one/media/8371/spec_sheet_gx3.pdf
[37] H. Lyons, “System Management Controller,” Disguise User Guide, 2025. https://help.disguise.one/hardware/smc/smc-overview (accessed May 01, 2025).
[38] Power Management Bus (PMBus®) Specification Rev. 1.4. System Management Interface Forum, Inc., 2024. Accessed: May 01, 2025. [Online]. Available: pmbus.org

[39] System Management Bus (SMBus) Specification, Version 3.3.1. System Management Interface Forum, Inc., 2024. Accessed: May 01, 2025. [Online]. Available: https://smbus.org/specs/SMBus_3_3_1_20241020.pdf
[40] ASPEED AST2600 BMC Users Guide X12/H12. 980 Rock Ave. San Jose, CA 95131 U.S.A.: Super Micro Computer, Inc., 2022. Accessed: May 01, 2025. [Online]. Available: https://www.supermicro.com/manuals/other/BMC_Users_Guide_X12_H12.pdf
[41] D. Laurie, ipmitool(1) 1.8.11-5. 2011. Available: https://manpages.debian.org/wheezy/ipmitool/ipmitool.1.en.html
[42] H. Lyons, “System Management Controller (SMC) Application Programming Interface,” disguise.one, 2025. https://developer.disguise.one/smc/introduction/ (accessed May 01, 2025).
[43] Disguise Systems Ltd., “VX4+ Product Specification,” 2022. Available: https://www.disguise.one/en/products/vx-4plus
[44] Infiled, “DB Series V2,” Jan. 2023. Accessed: May 01, 2025. [Online]. Available: https://www.infiled.com/wp-content/uploads/2024/01/db-series.pdf
[45] Brompton Technology Ltd., “TESSERA S8 LED Processor,” Sep. 2024. Accessed: May 01, 2025. [Online]. Available: https://www.bromptontech.com/wp-content/uploads/2024/09/Brompton-S8-Data-Sheet-Sep2024-EN.pdf
[46] International Electrotechnical Commission, Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space - sRGB. 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland: IEC, 1999. Available:https://cdn.standards.iteh.ai/samples/10795/ae461684569b40bbbb2d9a22b1047f05/IEC-61966-2-1-1999-AMD1-2003.pdf
[47] Future Observatory, “Virtual Production and the Carbon Footprint of the Screen Industries,” 2023. Available: https://futureobservatory.org/files/dcmsreports/futureobservatory_culturalpolicyreport_studioulster.pdf

