Vrije Universiteit Amsterdam

Bachelor Thesis

Understanding the Energy Impact of Cloud Gaming Through Large-Scale Simulation

Author: Rom Savidor 2688305

1st supervisor:	Jesse Donkervliet
daily supervisor:	Jesse Donkervliet
2nd reader:	Alexandru Iosup

A thesis submitted in fulfillment of the requirements for the VU Bachelor of Science degree in Computer Science

August 27, 2023

Abstract

As video games continue to grow in popularity and complexity, cloud gaming has emerged as a transformative technology, reshaping the landscape of interactive entertainment by enabling users to stream resource-intensive video games remotely. This helps players with less financial means to play the newest games without buying the newest hardware. Furthermore, it could help reduce the electronic waste that is generated by replacing old hardware. However, this convenience comes at the cost of increased energy consumption in data centers, which is already on the rise. Researching the environmental cost of cloud gaming is critical if we want to be able to keep these services running, and could benefit governments, data center operators, video game companies, and players. This paper presents an evaluation of cloud gaming's energy consumption, addressing key research questions related to data center design and operational choices. Through a series of experiments, we analyze the impact of factors such as GPU power models, CPU and GPU utilization, and resolution on energy consumption and cost. Our evaluation leverages a simulation-based approach that bridges real-world data and computational models, offering insights into both the environmental and economic implications of cloud gaming. Our findings not only validate our simulation model and GPU implementation but also shed light on optimization strategies for energy-efficient cloud gaming platforms. This paper equips stakeholders with quantitative information to guide decisions in cloud gaming system architecture, resource allocation, and sustainability efforts, ultimately contributing to a more informed and responsible evolution of cloud gaming technology. This work is relevant to researchers, industry professionals, and policymakers interested in understanding and improving the energy footprint of cloud gaming. The project can be accessed at https://github.com/romSavid/opendc-RS-2023.

Contents

1

Introduction

The rise of smartphones and the widespread of computer networks have made video games more accessible than ever, and today, they play a big part in the everyday lives of many. In 2022, the number of active gamers was estimated at 3.09 billion (1) worldwide. A big part of these games takes place online, whether as games that run locally and are connected to a server that manages the game's states, thus enabling a shared virtual world, or via cloud gaming services, which run a game on a remote server and stream it as a video to the client's machine.

To support these technologies on a large scale, online games operate as distributed systems in cloud data centers. The problem is that data centers require large amounts of energy to operate (2), and as players, developers, and publishers continue to drive the growth of this industry, internet service providers and data centers are facing pressure to support this demand. Environmental organizations, government agencies, and communities near data centers are raising concerns about the energy consumption and emissions associated with online gaming. According to a 2020 European Commission study (3), the energy consumption of data centers within the EU is expected to increase by 28% by 2030 when compared to 2018. The study also states that there is no one solution to achieve the goal of sustainable data centers by 2030.

A significant factor for the growth in data centers' energy consumption is video streaming (4), and gaming is often treated as video streaming. While cloud gaming could reduce energy consumption on the user's side, it is one of the most energy-intensive forms of gaming (5), and the most harmful form of Internet gaming. This technology is new and evolving, and despite our general knowledge of the energy use of data centers, more research is needed to provide a comprehensive understanding of the impact in the specific context of cloud gaming. We feel that as computer scientists, we have a moral obligation

to make sure the rapid evolution of cloud technologies is also a sustainable one. While gaming hardware and software developers make constant progress in making more sustainable gaming consoles and video games (5), not enough is done on the data centers side to improve the energy use of the different types of online gaming. Currently, most gaming data servers share data centers with servers of varying uses, which makes it harder to improve energy efficiency (6).

Thus, this study aims to create tools that could help us investigate the environmental cost of playing online video games, and how we can improve the layout of data centers in order to decrease this cost. As data centers are spread all over the world, ensuring their energy efficiency is a global concern, and through this project, we hope to reach conclusions that will benefit the environment in the long run.

1.1 Problem Statement

In order to conduct experiments and gain meaningful insights regarding the aforementioned issues, it is essential that we simulate cloud gaming on discrete-event simulators. In its current state, OpenDC does not offer this kind of simulation and requires designing and implementing a model that accurately captures the intricacies of cloud gaming systems.

To address this issue, we plan to implement a model that is built to simulate cloud gaming and includes a tool for building work traces. This presents a significant challenge as cloud gaming service providers are very secretive about the infrastructure of their services, and therefore, we are required to make many assumptions regarding hardware, its capabilities, and the resulting energy consumption.

Moreover, OpenDC currently only simulates CPU utilization and does not include GPU utilization in its simulations. In order to achieve the desired simulation level, we aim to incorporate GPU utilization into OpenDC.

Lastly, to gain further insight into the environmental costs of cloud gaming, we will use our additions to OpenDC to test different data center topologies and their energy consumption effects.

1.2 Research Questions

RQ1 *How to model energy usage in cloud gaming services?*

To address the energy consumption of cloud gaming through simulation, we must first address the question of modeling cloud gaming. A model will help us conduct

experiments regarding the energy consumption of cloud gaming. The main challenge of creating the cloud gaming model stands from the fact that there is hardly any data available from cloud gaming service providers. Ideally, we want to model our experiments using actual, validated data. The lack of transparency, however, forces us to make several assumptions regarding user patterns, service infrastructures, and energy consumption.

RQ2 *How to implement the cloud gaming energy usage model into a discrete-event simulator?*

To gather the required energy consumption data, we will create an experiment template that can take different inputs, such as the hardware, the number of virtual machines (VMs), or the number of game instances on each VM, create appropriate workloads, and utilize them to run experiments on OpenDC. Translating the model into useful experiments can prove to be challenging. We will need to implement a workload builder that can create traces that fit the large number of users that fit the real-world numbers of cloud gaming users. Furthermore, most modern video games require GPUs to run properly (7), but currently, OpenDC does not offer GPU as a parameter for simulation. This will make the outcome of the experiments less valuable. To address it, we plan to implement GPU utilization, and its resulting energy consumption, as a parameter in the experiments in OpenDC. Implementing our model can contribute to research on the environmental effects of online gaming by providing a tool to simulate different cloud gaming instances.

RQ3 *How does data center design affect cloud gaming's energy consumption?*

Finally, we want to look into the effects different data center topologies have on energy consumption. Different designs could potentially help reduce the environmental price of cloud gaming. To do this, we will make use of our work by creating experiments that test different data center layouts and resource allocation and hope to reach conclusions regarding those variables in terms of their environmental footprint. We aim to test possible alternatives to our model and see if they help improve energy consumption.

1.3 Thesis Contributions

This thesis suggests an approach to modeling and analyzing cloud gaming energy consumption. The OpenDC simulator will be integrated with advanced models to provide a

framework for analyzing resource allocation and energy efficiency for GPU-intensive workloads in cloud gaming infrastructures.

Among the main contributions of this thesis are:

MC1 Cloud Gaming Modeling: In Chapter 3 we develop a methodology to model cloud gaming energy consumption, utilizing player count traces and informed assumptions to estimate concurrent usage accurately.

MC2 GPU Power Usage Modeling: In Chapter 3, we propose a model for estimating GPU power usage within data centers, considering GPU utilization.

MC3 Integration into OpenDC: In Chapter 4 we implement the cloud gaming and GPU power usage models into the OpenDC simulator, providing researchers with an easily extendable tool for future investigations into cloud gaming energy efficiency.

MC4 Energy Cost Experiments: In Chapter 5 we conduct experiments to measure the energy cost of running a cloud gaming service, providing valuable insights into the environmental impact and cost-effectiveness of such platforms.

1.4 Plagiarism Declaration

I confirm that this thesis work is my own work, is not copied from any other source (person, Internet, or machine), and has not been submitted elsewhere for assessment.

1.5 Thesis Structure

This thesis is structured as follows: Chapter 2 introduces the concepts and terminology necessary to understand this work. Chapter 3 introduces our cloud gaming energy consumption model. Here we give an overview of the model, explain the assumptions we have made, and elaborate on our GPU utilization model and the cloud gaming service providers we modeled. Chapter 4 presents our implementation of the cloud gaming energy consumption model in the OpenDC simulator. Here we give an overview of the components we have implemented and how they interact with the existing components of OpenDC. We then present our experiment generator tool. Chapter 5 evaluates our model and implementation and presents the experiments we have conducted looking at different data center topologies and their effects on energy consumption. We also discuss possible threats to the validity of our work. Chapter 6 presents the related literature in the fields of data centers, cloud

gaming, and the OpenDC simulator. Finally, Chapter 7 summarises the work, provides answers to our research questions, and offers our recommendations for future research.

2

Background

This research deals with energy consumption in cloud gaming. The following section presents the key concepts needed to understand this work, including cloud gaming, data centers, and OpenDC.

2.1 Cloud Gaming

Online gaming is a broad term that encompasses a wide range of digital gaming experiences, from massively multiplayer online games to mobile and social games. For the purpose of this research, emphasis will be placed on cloud gaming. The term cloud gaming refers to the delivery of gaming content and services over the internet rather than through traditional consoles and PCs.

Cloud gaming services allow users to play games on virtually any device with an internet connection, including a PC, laptop, tablet, or even smartphone. This eliminates the need to purchase a dedicated gaming console and upgrade hardware to keep up with the latest releases. Due to the fact that games can be streamed directly from the cloud, cloud gaming also requires less storage space. While allowing for increased scalability and accessibility, cloud gaming also increases the energy consumption and carbon emissions associated with online gaming (5).

The workflow of a cloud gaming session is as follows: The user accesses a cloud gaming platform, like Microsoft's Xbox Game Pass Cloud Gaming, through a device with internet access, such as a computer or mobile device. The platform streams a video of the chosen game to the user's device, which is displayed in real-time. The user then interacts with the game using the device's control scheme, and their inputs are sent back to the cloud gaming platform over the internet. Following that the platform processes the user's inputs,

updates the game state accordingly, and then streams the updated video back to the user's device. This cyclical process continues, allowing the user to play the game on their device as if it were running locally, but with the processing and graphics rendering taking place on servers in a data center.

By utilizing technologies such as compression, dynamic resolution scaling, and efficient video codecs, the platform tries to improve the user's experience even in less ideal network conditions. An overview of a cloud gaming service can be seen in Figure 2.1.

2.2 Data Centers

Data centers are the backbone of the internet and play a crucial role in supporting cloud gaming infrastructure. A data center is a large, centralized facility that houses servers and other computing hardware. A high-speed network connects these servers to each other and to the internet, allowing data and information to be transmitted quickly and efficiently. Data and information are stored, processed, and transmitted by servers for many applications, including websites, emails, cloud computing, and data analysis.

To ensure the reliability and availability of these services, data centers are designed with multiple layers of redundancy. This includes backup power supplies, network connections, and cooling systems (2). Data centers consume large amounts of energy to power and cool servers, storage, and networking equipment. They also produce significant carbon emissions, which contribute to climate change (3).

As the demand for cloud gaming, and other data center services, continues to grow (8, 9), the energy consumption and carbon emissions associated with data centers will also increase. Data center design can significantly affect energy consumption (10). The choice of hardware, cooling systems, and energy efficiency technologies all influence the amount of energy that is consumed in a data center (2). By optimizing these factors, data centers can reduce their energy consumption and associated carbon emissions.

2.3 OpenDC

To maximize resource utilization, reduce energy consumption, and improve performance, data centers must be managed effectively (11). However, due to associated risks, potential disruptions to ongoing activity, and the high financial cost of their use and maintenance, conducting live analysis in operational data centers is often impractical (12). This neces-

sitates a cost-effective and scalable platform for conducting controlled and reproducible experimentation in data centers.

Designed to overcome these challenges, OpenDC provides a flexible, open-source simulation framework. By simulating the behavior of data centers, OpenDC allows for rapid prototyping, evaluation of novel strategies, and performance analysis under different scenarios, eliminating the challenges of conducting physical experiments (13, 14).

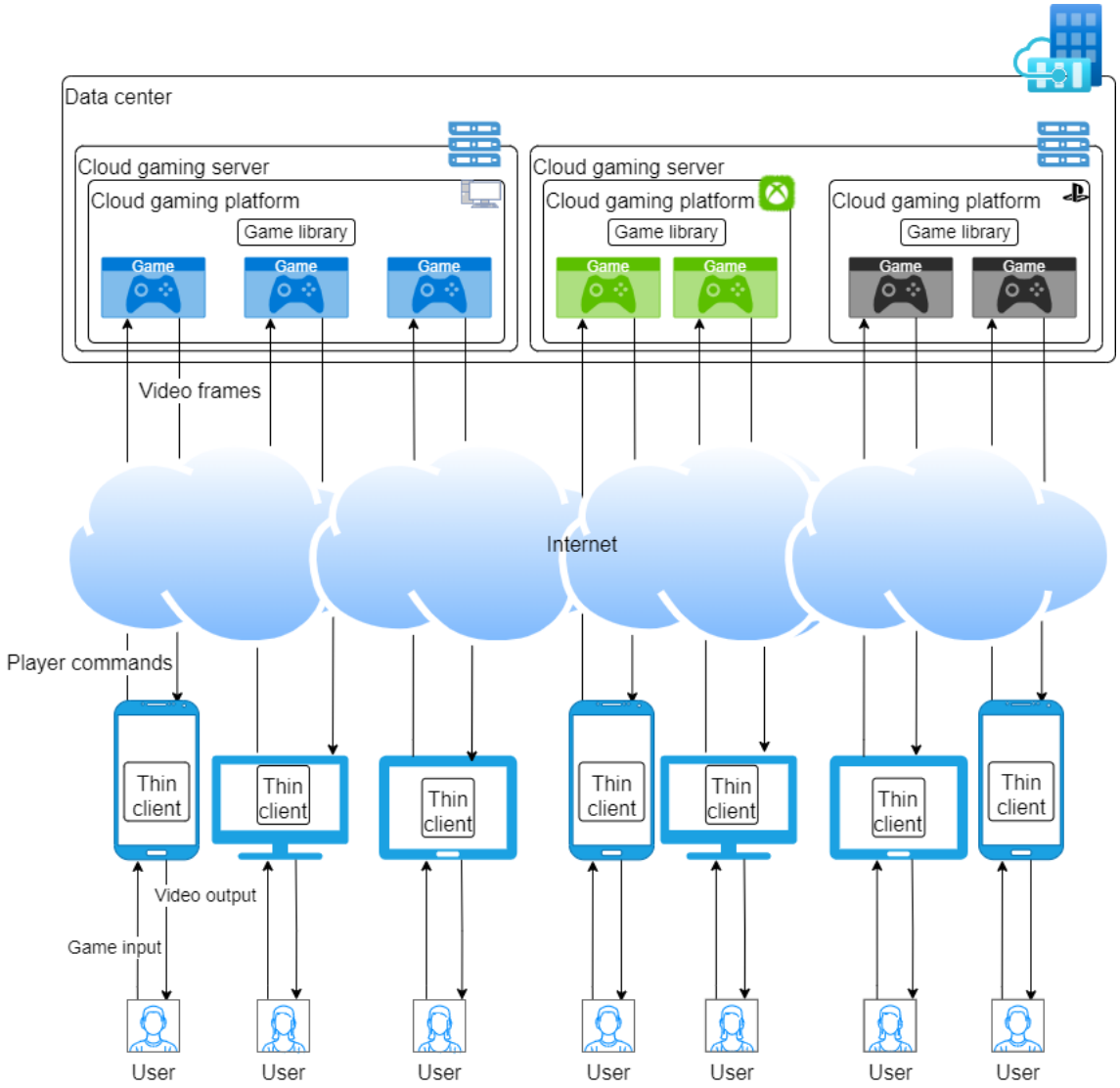

Figure 2.1: Cloud gaming

3

Modeling Cloud Gaming

This chapter addresses **RQ1**: *How to model energy usage in cloud gaming services?* To do this, we will outline the methodology and underlying assumptions that enable our simulation of the energy consumption of cloud gaming. Since cloud gaming services do not provide direct game traces and resource usage, our model relies on probable assumptions and indirect data. In particular, the model employs traces of the number of players in an online game during various timeframes. Thus, they can be used as a proxy for the number of concurrent users on a cloud gaming service. Additionally, we are able to make assumptions about resource allocation based on the technical specifications for cloud gaming hardware that are publicly available.

3.1 Model Requirements

In order to evaluate the environmental impact of cloud gaming, we need to first develop an energy consumption model that considers the various aspects of such a service, such as the number of users, the resources required for each game instance, CPU and GPU utilization, and power consumption. The inclusion of GPU utilization is paramount to offer a holistic simulation environment, recognizing the importance of both CPU and GPU components in cloud gaming workloads. Furthermore, the model should enable the modeling of authentic cloud gaming services, enabling rapid experiments that are readily comparable to other real-world data sources, such as console energy consumption. The model's primary goal should be to provide valuable environmental impact insights, offering a means to measure the ecological implications of cloud gaming services across diverse usage scenarios.

As such, we can define three model requirements (**MR**):

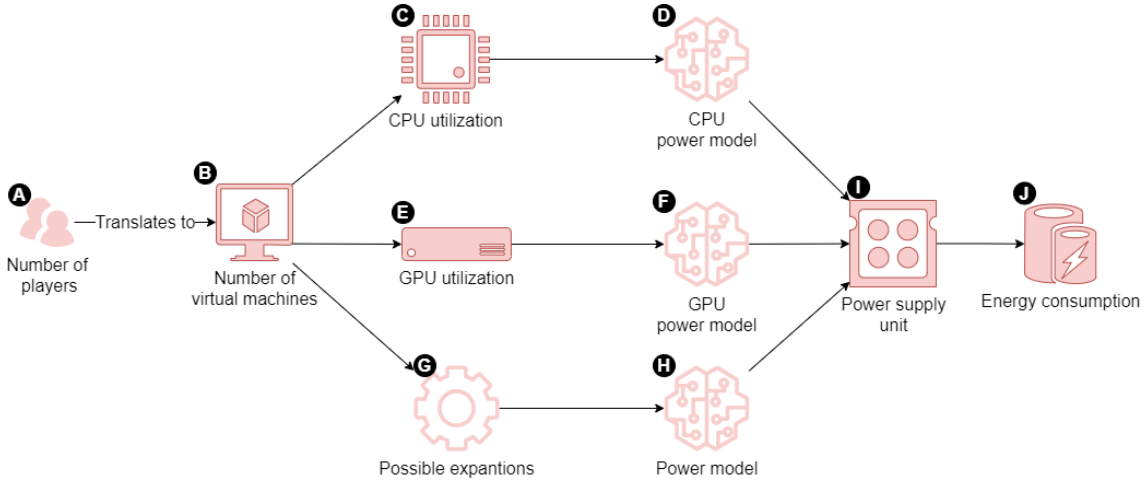

Figure 3.1: Our cloud gaming model

MR1: Design and implement a model replicating cloud gaming services' energy consumption patterns. This requirement is centered around capturing the intricacies of energy usage in cloud gaming, with particular emphasis on the interplay between hardware components, user behavior, and gaming workloads. The model must be capable of reflecting how various factors, such as CPU and GPU utilization, influence energy consumption in a cloud gaming context.

MR2: Enhance the model to encompass GPU utilization alongside CPU utilization. This extension is pivotal in capturing a comprehensive view of energy consumption within cloud gaming, as modern gaming workloads heavily rely on both CPU and GPU resources. The model should replicate how GPU utilization translates to energy consumption and consider the interdependence between CPU and GPU components.

MR3: Model real-world cloud gaming services, enabling researchers to conduct experiments that facilitate direct comparisons between cloud gaming energy consumption and other real-world data, such as energy usage in traditional console gaming. This requirement focuses on creating a platform that simplifies and accelerates experimentation by allowing researchers to easily assess the energy consumption of cloud gaming in relation to established benchmarks, thus providing a practical perspective on the environmental implications of cloud gaming services.

3.2 Cloud Gaming Energy Consumption Model Overview

The cloud gaming energy consumption model we developed is designed to emulate the dynamics of cloud gaming platforms (**MR1**). At its core, the model aims to provide accurate estimates of the energy consumption and costs associated with operating such platforms. The architecture comprises several key components, each contributing to the model's completeness.

In Figure 3.1, the number of players Ⓐ over the given timeframe is estimated by analyzing player count traces from multiplayer games' servers to help build a realistic workload for the model. These player counts are translated into virtual machines (VMs) Ⓑ on a one-to-one basis, aligning the generated VMs with user behavior and session lengths. The CPU and GPU utilization values Ⓒ Ⓔ from the trace are applied to their respective power models Ⓓ, Ⓕ. Then, the results are used in the relevant power supply unit (PSU) profile Ⓘ to calculate energy consumption. This is done for every entry in the work trace, with its relative user count, resource allocations, and GPU and CPU utilization values and power models. Ultimately, the result is the total energy consumption of the simulation.

It is also possible to further extend our model with more components Ⓖ, like network adapters or cooling, and their power models Ⓗ, enabling more intricate simulations.

The last requirement is modeling real-world cloud gaming services. In order to accomplish this, we make assumptions based on publicly available data about these services. The modeling of these services enables us to conduct experiments with results that can be compared with other real-life data, like the energy consumption of gaming consoles, and provide valuable insights.

3.3 Utilizing Player Count Traces

The first component of our model is the incorporation of player count traces from multiplayer games' servers (**MR1**). In this context, the term trace refers to a time-stamped log of the number of active players during different time intervals. Since there are no publicly available traces from real cloud gaming services, which would have allowed us to simulate the most accurate user interaction, and assuming that gaming habits are generally consistent across platforms, these traces serve as a useful indicator of cloud gaming user activity patterns.

As an alternative, we considered using traces from other cloud services not necessarily related to gaming. For instance, Microsoft published traces of Azure Functions (15), which

allowed us to estimate how many players there were. However, this approach was not adopted since Azure Functions invocation patterns differ greatly from those of cloud gaming services. Additionally, it is possible to use a fixed number (either arbitrary or somewhat substantiated) of users for all timeframes of the trace. This will result in a less realistic simulation but still provide a useful and fast way to experiment with different topologies and compare them with a fixed user count. A better alternative to our solution, is to use real-world traces, if they are available. This will lead to the most accurate simulation.

3.4 Resource Allocation Estimation

The second core element of our model involves estimating how resources are allocated on a per-player, per-virtual machine (VM), and per-host basis on a cloud gaming server (**MR1**). Given the scarce availability of explicit data related to cloud gaming server specifications and operation, our model makes informed assumptions about how resources might be distributed. Drawing on existing knowledge about the general operational principles of cloud servers and the few specific details available about cloud gaming servers, like the NVIDIA cloud gaming RTX blade server (16), we hypothesize that each active player corresponds to a distinct VM instance. Each VM, in turn, is allocated a portion of the host's resources – such as processing power, memory, and bandwidth – based on the needs of the game and the quality of service. This allocation takes into account the game's graphical fidelity, processing requirements, and network demands, among other factors. For the purpose of our model, we assume that the resource allocation per player remains relatively consistent, allowing us to predict the total resources needed based on the number of concurrent players.

When considering resource allocation, we again ran into the problem of the lack of actual data from real-world services that we could apply to our model. The only official resource we found was the aforementioned NVIDIA cloud gaming RTX blade server (16), where we could deduct plausible resource allocation for cloud gaming in certain scenarios. However, different cloud gaming services will have different resource allocation procedures, and ideally, we would have the data for each service we want to model and simulate. Another alternative was to look up the system requirements for different games and use them as the resources allocated to each VM. While this could simplify the model, it treats each VM as a separate physical machine and ignores the resource-sharing aspect of cloud gaming.

We made the decision to keep the resource allocation fixed throughout our generated experiments, meaning each VM will get the same resources. This was done to keep the experiment generator simple. We also considered the option to include a random element in resource allocation to get a more diverse topology but decided this would not benefit our model. Ideally, a user of the system will have a real-world topology to use to get the most accurate simulation.

3.5 Modeling Real-World Cloud Gaming Systems

We have modeled three platform presets based on whatever publicly available information we could find, and the assumption we could make using that information (**MR3**). Note that we make a lot of generalizations here for simplicity reasons, but these services are more complex than what we model here. Different hardware is used to run different games (17), and we cannot reach that level of granularity with the scope of our work and with the knowledge we possess. We base a lot of our assumptions on the NVIDIA Cloud Gaming Server (16), as it has a publicly available hardware specification that we can use. To have a uniform setup for the experiments, we make the assumption that one CPU core is required to run one game instance, based on the maximum number of game instances that NVIDIA claims the server can run (160) (16) and the overall number of CPU cores in a server (also 160). So, we define one cluster to have 160 CPU cores and calculate the rest of the parameters based on this number.

Defining the power models requires making assumptions regarding the idle power draw of the GPUs and CPUs of the systems, as this metric is not widely available. We treat the published TDP (thermal design power) as the maximal power draw and calculate the idle power draw based on TDP, and the publicly available overall power draw of the different systems.

The three presets are:

1. **GeForce Now**: GeForce Now is NVIDIA's cloud gaming service (18). While we do not know which hardware is used to run this service, we will make the assumption that NVIDIA's RTX Cloud Gaming Blade Servers (16) are used. Every server is made of 20 CPU nodes, each with eight cores that run at 3.5 GHz, 64 GB of memory, and two NVIDIA RTX GPUs. The base clock speed is not indicated, but we will make the assumption that the GPU that is being used is the RTX 3060 (19), which has a base clock speed of 1.32 GHz. The TDP for the GPU is 150 W (16). The CPU model is not provided, but according to the available information, we make an assumption

that the used model is the Intel® Core™ i9-11900K Processor (20), which has a TDP of 125 W. As for the idle power draw, we do not have any information, but we will use a calculation similar to that used for the Xbox and PlayStation presets, where we divide the TDPs by around 5.5 to make the calculation. To assume the idle power draw for both the CPU and GPU we divide the maximal power draw by 5.5 and get roughly 27 W for the GPU and 23 W for the CPU.

NVIDIA indicates that a single server can run up to 160 games concurrently. This translates into one CPU core per game instance. While the actual number of instances may vary according to the type of game that is run, for the scope of this project we will keep this one CPU per one instance ratio. In our implementation, a cluster is one of these servers. So that translates to 160 CPU cores that run at 3.5 GHz, 40 GPUs running at 1.32 GHz, and a memory of 1280 GB.

2. **xCloud**: The second preset is based on Microsoft's Project xCloud, also known as Xbox Cloud Gaming (21). While the infrastructure of the xCloud servers is not public, it is known that the server blades are powered by Xbox Series X hardware (22). Furthermore, while it is not known how many Xbox units are installed on each server, for previous iterations it has been published that each server blade contains eight Xbox units (23). As a general rule of thumb for this implementation, we are going to assume 160 CPU cores in a cluster, based on the information published about NVIDIA's RTX Cloud Gaming Blade Server (16). Based on the previous assumptions, one cluster of xCloud servers will include hardware that corresponds to 20 Xbox series X units (24). This translates to 160 CPU cores that run at 3.8 GHz, 20 GPUs, each running at 1.825 GHz (25), and a memory of 320 GB. The TDP for the GPU is 200 W (26) and while the TDP for the CPU is not available, the Series X's CPU has similar specifications to the AMD Ryzen 7 5700G (27), which has a TDP of 65 W. As for the idle power draw, we consider the Navigation Mode power consumption from the official Xbox energy consumption report (28), which is 48 W. The maximal power draw of the GPU and CPU combined is 265 W. By dividing it by 48 we got roughly 5.5. Now to assume the idle power draw for each we divide the maximal power draw by 5.5 and get roughly 36 W for the GPU and 12 W for the CPU.

   While it is unclear how many game instances each Xbox unit can run, we will make the same assumption we made for the GeForce Now infrastructure and say that each CPU core translates to one game instance. Note that this is not based on real

information, and the real number of instances is probably lower and depends on the type of game that is run.

3. **Playstation Plus**: PlayStation Plus (29), is a subscription service for Sony PlayStation users that includes, among other things, a cloud gaming service, formerly known as PlayStation Now. Out of the three, this service has the least public information available, so we will make assumptions regarding the overall structure of a cluster similar to the assumption we made for xCloud. Hardware that corresponds to 20 PlayStation 5 units (30). This translates to 160 CPU cores that run at 3.5 GHz, 20 GPUs, each running at 2.23 GHz (31), and a memory of 320 GB. The TDP for the GPU is 180 W. For the CPU we have a similar case to that of the Xbox, where we assume the CPU has similar specifications to the AMD Ryzen 7 5700G (27), which has a TDP of 65 W. As for the idle power draw, we consider the Home menu user interface power consumption from the official PlayStation energy consumption report (32), which is 45.6 W. The maximal power draw of the GPU and CPU combined is 245 W. By dividing it by 45.6 we got roughly 5.4. Now to assume the idle power draw for each we divide the maximal power draw by 5.4 and get roughly 33 W for the GPU and 12 W for the CPU. Like before, we are going to assume a one CPU core per one game instance ratio.

It is important to emphasize that these presets and their numbers are given as a starting point. But to get accurate results that correspond to real-world situations, users will be better off using real-world data. Our tool is extendable and modifiable and real-world data can be implemented into it in addition to our presets.

3.6 Modeling of the GPU

GPUs (Graphics Processing Units) play a significant role in rendering graphics and processing complex visual computations. Most modern games require the hosting machine to have a GPU with sufficient capabilities to run at the desired settings. GPUs can also be used to offload computational tasks from the CPU (Central Processing Unit) to improve performance (33) To complete our cloud gaming model, we introduce a model of GPU resource allocation and energy consumption (**MR2**).

In our original model, we did not treat the GPU resources as separate GPU nodes (or graphic cards) but as the number of cores available to the host and the clock speed that corresponds to the modeled GPU node. This means that for a single host, we could only

model a number of GPU nodes with the same clock speed. This decision was made to enable the model's implementation within our time limits. But as we started running the experiments, we ran into the problem of scaling them up. When the experiments got bigger, the amount of GPU cores objects created used large amounts of the system's memory and made it impossible to run on our local machine, decreasing our solution's usability even for smaller experiments. In addition, modeling the GPU cores did not add much to the accuracy of the system as they work differently than the CPU cores and do not operate on an individual level like them.

Instead, we have opted for a different model. In our current model, we make use of the GPU virtualization concept (34), a general term that describes the techniques and methods used to virtualize GPUs in order to enable their efficient utilization in cloud computing or large-scale distributed computing environments. So a GPU's computing power can be shared by multiple virtual machines (VMs). We assign a number of graphics cards to a cluster and then divide their computing capacity (MHz) between all of the different hosts equally.

A different approach to the individual graphic nodes, the equal division of compute power, and the inclusion of additional metrics like memory clock speed and architecture, can be explored further in future work.

3.6.1 GPU Power Model

To enable power consumption estimation, we have added a new power model, responsible for translating GPU utilization into corresponding power consumption values in watts. A factor is calculated by subtracting the given idle power draw from the given max power draw, and then multiplied by the utilization and added to the idle power draw to calculate the GPU power usage. This can be formulated in the following way:

$$E(u) = E_{idle} + (E_{max} - E_{idle}) \times u$$

where:

- E is the energy consumption
- u is the utilization

This is a linear power model which suggests a straightforward power usage calculation, but it can be also used as a framework to implement more complex power models later.

3.6.2 Power Supply Unit Gaming Profile

A dedicated power supply unit (PSU) profile for gaming power consumption was created. This profile considers both CPU and GPU usage, calculating the total power usage by aggregating the individual power requirements of the CPU and GPU components. This was added so the model could support pushing and retrieving CPU and GPU usage separately. We considered treating CPU and GPU usage as one value which is the sum of both, but we decided that the added flexibility of keeping them separate benefits the implementation more than the simplicity of having them as a single value.

4

GPU Model and Experiment Generator Implementation in OpenDC

In this chapter, we address **RQ2**: *How to implement the cloud gaming energy usage model into a discrete-event simulator?* To do this, we describe the implementation details of the cloud gaming model and the GPU power usage model within the OpenDC simulator.

4.1 Requirements Analysis for Cloud Gaming Model and GPU Implementation

When implementing our models, it is important to meet specific requirements to ensure the flexibility and usability of the system, to support future research. The GPU implementation and cloud gaming model should be designed with extensibility in mind, so further improvements could be implemented easily and support more accurate simulation. This means providing a modular architecture that allows for easy integration of new power consumption models. Researchers and developers should be able to add new components, power models, and cloud gamine service presets without significant modifications to the codebase.

The implementation should also be easily modifiable to accommodate different requirements and scenarios. This includes the ability to adjust configuration parameters such as GPU utilization, core counts, gaming workloads, and power models. The system should provide clear interfaces that allow researchers and developers to modify and experiment

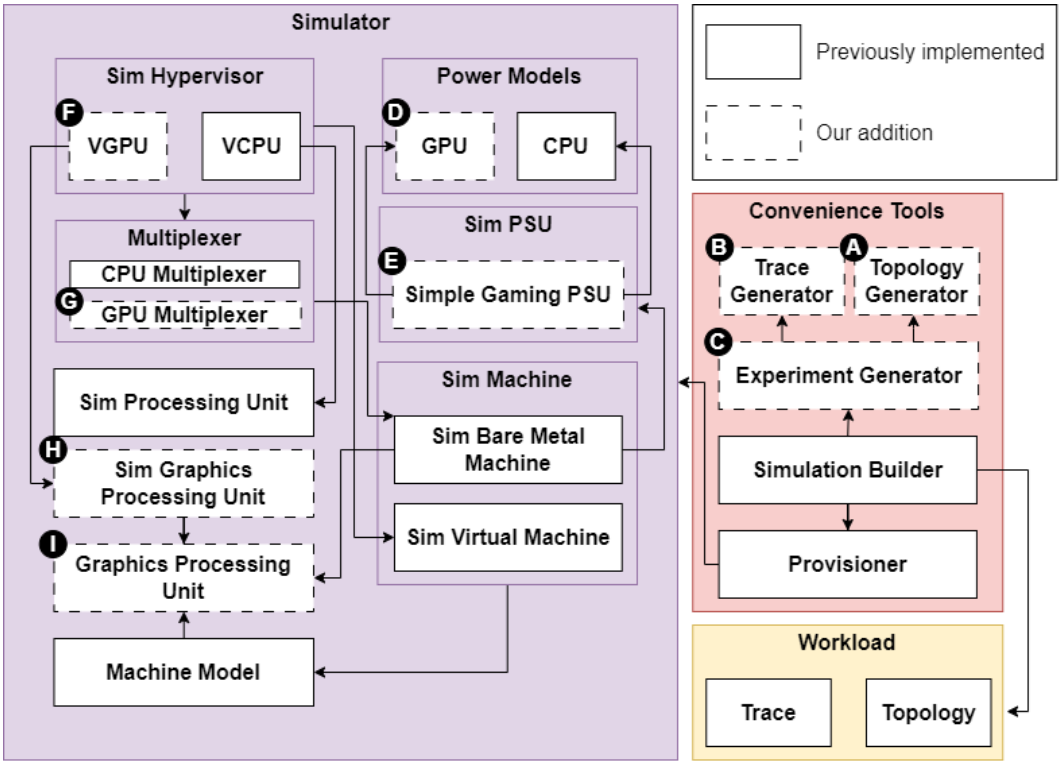

Figure 4.1: An overview of the components we implemented in OpenDC and their interactions with other components

with these aspects effectively. This flexibility enables customization and fine-tuning of the system to match specific cloud gaming environments and research needs.

The implementation should have a user-friendly tool that provides a quick way to start experimenting without requiring a deep understanding of the system or having real-world traces so that it could be used both for research and teaching (14).

As such, we can define three implementation requirements (**IR**):

IR1: The implementation should support the addition of new power models and cloud gamine service presets by modifying existing components without architectural changes.

IR2: The implementation should support the simulation of a large variety of state-of-the-art gaming hardware through configuration parameters.

IR3: The implementation should provide users with a tool to run cloud gaming simulations without needing to have real-world traces.

```
ClusterID;ClusterName;cpuCores;cpuSpeed;
    gpuCount;gpuSpeed;Memory;numberOfHosts
A01;A01;160;3.5;40;1.32;1280;160
A02;A02;160;3.5;40;1.32;1280;160
A03;A03;160;3.5;40;1.32;1280;160
A04;A04;160;3.5;40;1.32;1280;160
A05;A05;160;3.5;40;1.32;1280;160
A06;A06;160;3.5;40;1.32;1280;160
A07;A07;160;3.5;40;1.32;1280;160
A08;A08;160;3.5;40;1.32;1280;160
A09;A09;160;3.5;40;1.32;1280;160
A10;A10;160;3.5;40;1.32;1280;160
A11;A11;160;3.5;40;1.32;1280;160
A12;A12;160;3.5;40;1.32;1280;160
A13;A13;160;3.5;40;1.32;1280;160
A14;A14;160;3.5;40;1.32;1280;160
```

```
// Generate topology file
        CloudGamingTopologyGenerator.generateTopologyTxt(
            numClusters = numClusters,
            cpuCoresPerCluster = cpuCount,
            cpuSpeed = cpuCap,
            graphicsCardPerCluster = gpuCount,
            gpuSpeed = gpuCap,
            memory = memCap,
            numHosts = gameInstancesPerCluster,
            topologyName = "$platform-topology",
            cpuIdleDraw = cpuIdleDraw,
            cpuMaxDraw = cpuMaxDraw,
            gpuIdleDraw = gpuIdleDraw,
            gpuMaxDraw = gpuMaxDraw
        )
```

Figure 4.2: Creating a topology manually versus with our generator

4.2 Cloud Gaming Workloads

To include our cloud gaming model in OpenDC, we implemented three generator objects that are in charge of the different aspects of our model. While these tools help the user start experimenting quickly (**IR3**), it is also possible to manually create topologies, trace files, and meta files.

4.2.1 Topology Generator

As can be seen in Figure 4.1 Ⓐ we have implemented an object that receives from the user the desired topology characteristics, like the graphics card specifications, the number of CPU cores, the number of clusters, the power models, and the number of hosts, and generates a new topology text file that includes different clusters based on these characteristics (**IR2**). This topology will later be used by the simulator to generate a list of hosts for the experiments.

Previously, as can be seen in Figure 4.2, the user had to create the topology file manually, by creating the said text file. Now, the user can input the desired characteristics and the file will be generated automatically. This makes it easier to start experimenting without much preparation (**IR3**). On the other hand, to create topologies with varying characteristics per cluster, it is still required to create the topology manually.

4.2.2 Trace and Meta Generator

The next step was to implement an object that receives from the user the characteristics of the game traces Ⓑ. This includes the number of hours to include, the number of users per

Ⓐ

```
// Generate trace file
        CloudGamingTraceGenerator.generateTraceCsv(
            hours = hours,
            usersPerHour = usersPerHour,
            cpuCount = cpuCount / gameInstancesPerCluster,
            cpuUsage = (cpuUtilization * cpuCap * 1000) * (cpuCount / gameInstancesPerCluster),
            cpuCap = cpuCap,
            gpuCount = 1,
            gpuUsage = (gpuUtilization * partitionedGpuCap * 1000) * 1,
            gpuCap = partitionedGpuCap,
            memCap = memCap,
            outputDir = "$platform-trace"
        )
```

Ⓑ

```
id,timestamp,duration,cpuCores,cpuUsage,gpuCount,gpuUsage
1,1690116021222,3600000,1,2659.9999999999995,1,216.71875
2,1690116021222,3600000,1,2659.9999999999995,1,216.71875
3,1690116021222,3600000,1,2659.9999999999995,1,216.71875
4,1690116021222,3600000,1,2659.9999999999995,1,216.71875
5,1690116021222,3600000,1,2659.9999999999995,1,216.71875
6,1690116021222,3600000,1,2659.9999999999995,1,216.71875
7,1690116021222,3600000,1,2659.9999999999995,1,216.71875
8,1690116021222,3600000,1,2659.9999999999995,1,216.71875
9,1690116021222,3600000,1,2659.9999999999995,1,216.71875
10,1690116021222,3600000,1,2659.9999999999995,1,216.71875
11,1690116021222,3600000,1,2659.9999999999995,1,216.71875
12,1690116021222,3600000,1,2659.9999999999995,1,216.71875
13,1690116021222,3600000,1,2659.9999999999995,1,216.71875
14,1690116021222,3600000,1,2659.9999999999995,1,216.71875
15,1690116021222,3600000,1,2659.9999999999995,1,216.71875
```

Ⓒ

```
id,startTime,stopTime,cpuCores,cpuCapacity,gpuCount,
    gpuCapacity,memCapacity
1,1690116021222,1690130421224,1,3.8,1,0.228125,320
2,1690116021222,1690130421224,1,3.8,1,0.228125,320
3,1690116021222,1690130421224,1,3.8,1,0.228125,320
4,1690116021222,1690130421224,1,3.8,1,0.228125,320
5,1690116021222,1690130421224,1,3.8,1,0.228125,320
6,1690116021222,1690130421224,1,3.8,1,0.228125,320
7,1690116021222,1690130421224,1,3.8,1,0.228125,320
8,1690116021222,1690130421224,1,3.8,1,0.228125,320
9,1690116021222,1690130421224,1,3.8,1,0.228125,320
10,1690116021222,1690130421224,1,3.8,1,0.228125,320
11,1690116021222,1690130421224,1,3.8,1,0.228125,320
12,1690116021222,1690130421224,1,3.8,1,0.228125,320
13,1690116021222,1690130421224,1,3.8,1,0.228125,320
14,1690116021222,1690130421224,1,3.8,1,0.228125,320
15,1690116021222,1690130421224,1,3.8,1,0.228125,320
```

Figure 4.3: Creating trace and meta files manually versus with our generator

hour, and the CPU and GPU utilization. The object then generates two CSV files. The first file is the trace file, which includes a row per VM per hour, and includes its length in milliseconds, its GPU and CPU usage, the number of virtual GPUs, and its CPU number of cores (**IR2**). The scope of one hour for the game traces was chosen for simplicity reasons but can be easily changed for future work. The second file is the meta file, which includes one row per VM that includes its GPU and CPU capacity, its number of virtual GPUs, its CPU number of cores, memory capacity, and its start and stop times.

Figure 4.3 shows Ⓐ how the tool can be used to create new trace and meta files by providing the desired characteristics, the manual creation of trace files Ⓑ, and the manual creation of meta files Ⓒ. Like the topology generator, this tool makes it easier to start experimenting quickly, without requiring real-world traces (**IR3**). However, for more intricate simulations the user should either create these manually or provide their own traces.

```
@Test
    fun testBasicRun() = runSimulation {

        val tracesDir = "psplus-trace"
        val usersPerHour = listOf(1777, 1693, 1560, 1406, 1242, 1106, 1045, 1011, 973, 938,
            949, 1031, 1182, 1357, 1563, 1779, 1925, 2013, 2074, 2096, 2029, 1961, 1890, 1826)

        ExperimentGenerator.generateExperiment("psplus", 0.5, 0.95, 24, usersPerHour)

        val workload = getWorkload(tracesDir)
        val topology = createTopology("psplus-topology")
```

Figure 4.4: Creating an experiment using our experiment generator

4.2.3 Experiment Generator

The last object Ⓒ is in charge of combining the previous in a convenient, easy-to-start way. As can be seen in Figure 4.4, this tool can be used to run experiments with a relatively quick setup, by choosing a platform, the varying number of users, the length of the session in hours, and the CPU and GPU utilization (**IR2**). The object, in turn, uses the previous two generators and our presets to create a topology file, a trace file, and a meta file. We have implemented three cloud gaming service presets into the object so users are able to start experimenting quickly without having real-world traces (**IR3**). Adjusting these presets, or adding new ones is a simple procedure that only requires adding an option to the experiment generator (**IR1**), as can be seen in Figure 4.5.

4.3 GPU Simulation Implementation

OpenDC had not previously incorporated GPU modeling, presenting an opportunity to extend the tool's capabilities and improve the fidelity of our cloud gaming service model. This implementation encompasses various components, including power modeling Ⓓ, machine and virtual machine modifications, a new multiplexer for GPU core management Ⓖ, and dedicated power supply unit (PSU) profiles Ⓔ. Importantly, all these GPU-related changes were implemented in a backward-compatible manner and were based upon previously implemented concepts, so their integration into the system is done in a way that makes use of the system's capabilities. Existing experiments that do not involve GPU modeling can continue to function as intended.

Future work in this area includes refining power models to align them with real-world data and benchmarks, further enhancing power consumption estimation accuracy (**IR1**). Additionally, exploring different parameters that influence GPU performance and power

```
// Set preset attributes based on the chosen platform
    when (platform.lowercase()) {
        "xcloud" -> {
            cpuCount = 160
            gpuCount = 20
            cpuCap = 3.8
            gpuCap = 1.825
            memCap = 320L
            gameInstancesPerCluster = instancesPerCluster
            cpuIdleDraw = 12.0
            cpuMaxDraw = 65.0
            gpuIdleDraw = 36.0
            gpuMaxDraw = 200.0
        }
        "psplus" -> {
            cpuCount = 160
            gpuCount = 20
            cpuCap = 3.5
            gpuCap = 2.23
            memCap = 320L
            gameInstancesPerCluster = instancesPerCluster
            cpuIdleDraw = 12.0
            cpuMaxDraw = 65.0
            gpuIdleDraw = 33.0
            gpuMaxDraw = 180.0
        }
        "geforcenow" -> {
            cpuCount = 160
            gpuCount = 40
            cpuCap = 3.5
            gpuCap = 1.32
            memCap = 1280L
            gameInstancesPerCluster = instancesPerCluster
            cpuIdleDraw = 23.0
            cpuMaxDraw = 125.0
            gpuIdleDraw = 27.0
            gpuMaxDraw = 150.0
        }
        else -> throw IllegalArgumentException("Invalid platform: $platform")
    }
```

Figure 4.5: The different cloud gaming service presets available in our experiment generator

usage, such as memory bandwidth and architecture, can provide a more comprehensive understanding of GPU behavior.

4.3.1 GPU Power Model

The implementation of the GPU power model Ⓓ was largely based on the implementation of the already existing CPU power model. We added a new interface and class for the GPU power model, with a linear power model and a constant power model. Additional power models can be easily added later by adding classes that extend the power model interface (**IR1**). Furthermore, a dedicated power supply unit (PSU) profile for gaming power consumption was created. This profile considers both CPU and GPU usage, calculating the

total power usage by aggregating the individual power requirements of the CPU and GPU components.

4.3.2 Power Supply Unit Gaming Profile

Adding the gaming profile for the power supply unit (PSU) Ⓔ consisted of creating a new class for the profile, including the behavior for pushing the demand. The profile includes both GPU and CPU usage values and separate methods for getting and setting them.

4.3.3 GPU in Machine Models

The GPU model was integrated into the machine and virtual machine models, allowing for precise allocation and management of GPU resources. By extending the existing infrastructure, OpenDC enables users to define the number of GPUs and their frequency for the simulation.

As indicated in the design section, a GPU core was originally implemented as a single processing unit with an id, a frequency, and a connection to a processing node, which in this case was the graphics card. But due to the difficulties that were introduced with creating a large number of GPU core objects, we have decided to implement at the graphics card level, having a graphics processing unit Ⓗ simulating both a graphics card and a virtual GPU, depending on the situation. Because of the similarities between the CPU implementation and the GPU implementation, many classes, like the power model and the graphics processing unit Ⓘ, were based on the CPU versions of these classes.

4.3.4 Graphics Processing Unit

The graphics processing unit Ⓘ was implemented as an abstraction for both a graphics card and a virtual GPU. This way it could be used to simulate either a graphics card or only a part of its computing power. Similar to the processing node class, it stores data such as vendor, model name, and arch. These could be later used when extending the implementation to have prefabs, for example. The main piece of information stored is the frequency, which will be used to calculate energy usage. When the class is used as a virtual GPU, the frequency will represent the partial computing power that is available for the host.

While different GPUs have different capabilities in terms of virtualization, and the number of virtual GPUs they can have (35), we have opted to not enforce it in our implementation, to have more flexibility for the users. In future work, these restrictions could

be implemented to provide a more realistic simulation that relies on real-world hardware limitations.

4.3.5 Hypervisor

To keep a separation between CPU and GPU usage metrics, we added a second multiplexer Ⓖ to the hypervisor that is only in charge of the GPU. When starting the hypervisor, all of its GPU inputs are connected to multiplexer outputs. The multiplexer is used to get the various metrics of GPU usage. Currently, we do not consider scenarios where a certain host has more than one virtual GPU/GPU, so it will always be only one connection. But this can be easily changed later when a more complex version of the GPU simulation is implemented (**IR2**). A VGPU Ⓕ was added, similar to the VCPU, to keep track of the GPUs on a virtual machine.

5

Experimentation

In this chapter, we address **RQ3**: *How does data centers' design affect cloud gaming's energy consumption?*. The following experiments focus on the following aspects: verifying our model, verifying our implementation, a general energy consumption analysis of cloud gaming services, and testing how different topologies affect the energy consumption of cloud gaming. These experiments can help us understand the price and environmental effects of running a cloud gaming platform, and help stakeholders make informed decisions when designing such platforms.

We run seven experiments, which are set up based on the specifications we have presented in the model section, and on Minetrack (36), a public domain Minecraft (37) multiplayer server statistics dataset. Because we have no access to real-world user counts of cloud gaming services, we used other sources to assume the number of users that log in to the service and play a game during different time frames. The numbers that we used, are at a smaller scale than what is realistically the case for a whole cloud gaming service (38). This approach allows us to focus on outcomes that are more akin to simulating the energy consumption of running a specific game title on the service or a localized cluster of server rooms, rather than attempting to encompass the entirety of the service's operations. We hypothesize that the smaller scope can still provide us with results that are meaningful and that we can learn from. The main metric that we measure is energy use in joules, which is converted to kWh, Wh, or W, depending on the experiment, for better readability. The main findings from these experiments are summarized below:

1. **What CPU and GPU utilization values we could use for our simulators?:** We evaluate this question in Section 5.1. To test that, we run four games with varying graphics settings on a local machine and log the CPU and GPU utilization values we get. For the CPU we see utilization values that vary from 7% for the least demanding

game, through 30%-60% for the more demanding games for most scenarios, up to 100% when loading a new level. For the GPU we see utilization values from as low as 1% for the least demanding game, through 30%-70% for more demanding games, and up to 90%-100% for the most demanding game. For the most demanding game, increasing the graphics settings when the GPU utilization was already at 100% only results in decreasing FPS (frames per second).

2. **Does our implementation produce realistic results?:** In Section 5.2, we run an experiment where we use a simple trace of 160 users over four hours, so we can perform a simple comparison. We divide the results we get to see the individual power consumption of each game instance and compare it to the power consumption of running a console at home. We ran this experiment once for the xCloud preset and once for the PlayStation Plus preset. We want to see how far off our results are from the energy usage of these platforms so we can either confirm or reject the validity of our GPU implementation into OpenDC. For xCloud, we get a simulated average of 137 W, which is very close to the official reported value of 153 W. For PlayStation Plus, we get a simulated average of 110.8 W, which is quite different than the official reported value of 210 W. Regardless, we believe that this is more due to the lack of official data available regarding the PlayStation Plus hardware rather than our implementation, and deem it successful.

3. **How do different GPU power models affect energy consumption?** In Section 5.3 we test the effects of the GPU power model on energy consumption. We use a workload similar to that of the previous experiment with varying CPU and GPU utilization levels and GPU power models. We observe that the choice of power model can affect the end energy consumption greatly and should be thought out thoroughly. Furthermore, we can verify that adding new power models to our implementation is simple and does not require architectural changes.

4. **What is the cost of running a small cloud gaming service for a day?:** In Section 5.4 we conduct experiments using a 1-day trace for the three modeled services. The trace is based on data from the Minetrack dataset. 27 different runs are performed, each with distinct topology and GPU and CPU utilization levels, to cover a reasonable range of results. We then calculate the total energy consumption and the cost of running a small cloud gaming platform for a whole day. The calculated price for the xCloud run is 3,343 euros, the PlayStation Plus run is 3,087 euros, and

the GeForce NOW run is 3,308 euros. For xCloud and PlayStation Plus, the prices are close to those of running the same amount of game instances on Xbox Series X and PlayStation 5, respectively. For GeForce NOW, the price seems to be significantly lower than that of running the same amount of game instances natively on a personal gaming computer.

5. **How much energy does a cloud gaming service consume compared to running a game locally?:** Also, in Section 5.4, we compare the total energy consumption and the overall price of running a cloud gaming service with the total energy consumption and cost of running the same amount of game instances natively. For xCloud, PlayStation Plus, and their respective consoles Xbox Series X and PlayStation 5 the results are rather similar. The prices for running each service or console for a full day are 3,342.7 euros for the xCloud, 2,646 euros for the Xbox Series X, 3,087.2 euros for PlayStation Plus, and 3,631.7 euros for PlayStation 5. The results for GeForce NOW and personal gaming computers differ widely with 3,308.1 euros for GeForce NOW and 9,511.6 euros for personal gaming computers.

6. **Does running the same amount of game instances with more GPUs lowers consumption?** In Section 5.5, we look at the effects of decreasing GPU utilization by increasing the number of GPUs on the server on energy consumption. The results show that keeping GPU utilization low by having more GPUs could help reduce energy consumption significantly, as we can see a decrease of 29.63% in energy consumption when employing this strategy. However, our GPU implementation needs to be improved for us to have more confidence in these results.

7. **How does running multiple game instances on a VM affect energy consumption?** In Section 5.6 we examine how different amounts of game instances that run on one VM affect the overall energy consumption. We run four simulations, with one instance per VM, two instances per VM, four instances per VM, and eight instances per VM. All with the same user count traces, for 24 hours. The results show an increasing improvement in energy consumption the more game instances we run on each VM. For one game instance per VM, the energy consumption is 8774.4 kWh, for two game instances per VM, the energy consumption is 5667.7 kWh, for four game instances per VM, the energy consumption is 2898.4 kWh, and for eight game instances per VM, the energy consumption is 1449.2 kWh. These results suggest that cloud gaming service providers should aim to run more game instances on

fewer VMs. However, this experiment does not take into account the overhead that is gained by running more games per VM, and further work is required to verify these results.

8. **How much more expensive it is to stream games in a cloud gaming service in 4K as compared to 1080p** To answer this question, in Section 5.7 we add a new cloud gaming service that is better fitted to stream games in 4K, in order to investigate the difference in energy consumption and price between a 1080p run and a 4K run. We modify the number of CPUs and GPUs required to run a single game instance. The main difference between the two runs is the GPU utilization level, which is 75% for the 1080p run and 99% for the 4K run. The results suggest that the increase in energy consumption and cost that incurs from streaming in 4K as opposed to 1080p is relatively moderate. For the 1080p run, we observe an energy consumption of 9262.9 kWh for 24 hours, and a price of 4,706 euros. For the 4K run, we observe an energy consumption of 10746 kWh for 24 hours, and a price of 5,459 euros. According to this experiment, service providers could benefit from offering 4K streaming to their users. In addition, we can verify that adding new cloud gaming service presets to our implementation is simple and does not require architectural changes.

5.1 CPU and GPU Utilization Values of Different Games

Our main findings from this experiment are:

1. **MF1:** CPU and GPU utilization values vary widely between different games, and even in the same game in different scenarios.

2. **MF2:** The graphical settings do not affect the CPU utilization of the game.

3. **MF3:** Good CPU utilization values for our experiments are 15%, 30%, and 70%. Good GPU utilization values for our experiments are 15%, 60%, and 100%.

In the first experiment, we investigate the CPU and GPU utilization values of different games running locally on our machine with different graphics settings, to obtain real values to use for our simulations in the following experiments.

5.1.1 Experimental Setup

The experiment is run on an ASUS ROG GL552VW laptop, launched in 2016. The device is powered by an Intel Core i7-6700HQ CPU running at 2.6GHZ, and a NVIDIA GEFORCE GTX 960M GPU. Despite being relatively dated, this hardware setup can still support newer games with lower graphical settings.

We chose four diverse games to provide a spectrum of graphical demands:

1. **Marvel's Spider-Man:** A modern AAA title with complex graphics and physics.
2. **A Hat in Time:** A less demanding indie title.
3. **Tetris.com:** An online version of the popular game. A simple game with modest graphics requirements.
4. **Tetris.NET:** A .NET implementation of Tetris, representing basic graphical demands.

For Marvel's Spider-Man and A Hat in Time, we tested different graphical settings to better understand the variability in utilization values within a single game. In both cases, we employed the game's provided preset sliders to assess low, medium, and high graphics settings, rather than manually adjusting individual settings. The two Tetris games lack the option to modify graphics settings.

5.1.2 Experimental Results

1. **A Hat in Time:** The CPU utilization value we get for the three graphics settings we tested is uniform, peaking at 50% during intensive gameplay and averaging at 40%. The GPU varies more, averaging 30% on the low settings with spikes up to 50% when there are more particles, 60% on the medium settings with spikes up to 80%, and 75% on the high settings with spikes up to 100%. The average values can be seen in Figure 5.1.
2. **Marvel's Spider-Man:** The CPU utilization remains fairly stable at around 70%, regardless of the graphics setting. The GPU utilization consistently stays between 99-100%. Changing the graphics settings mainly affects the frame rate, increasingly dropping as we test higher graphics settings, and has no effect on resource utilization. The average values can be seen in Figure 5.2.

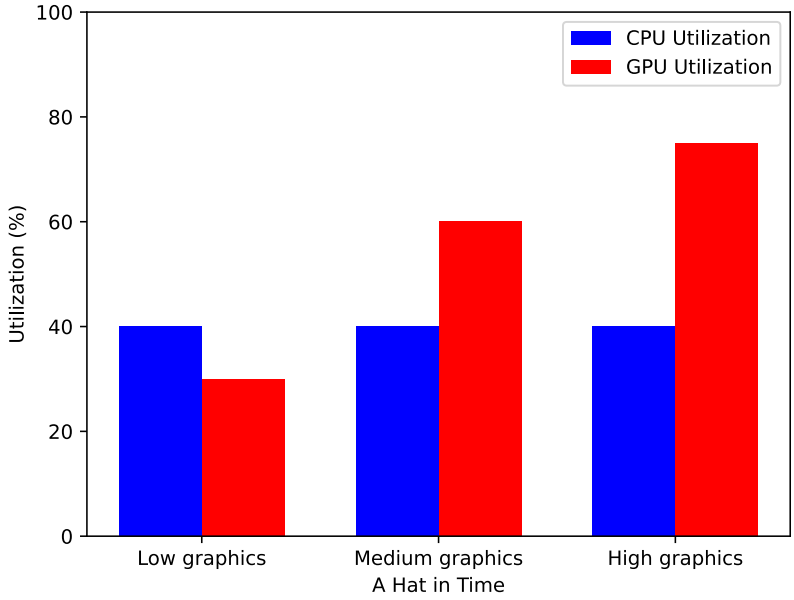

Figure 5.1: CPU and GPU utilization values for A Hat in Time for different graphics settings

3. **Tetris.com:** A steady 16% for both CPU and GPU is observed during the whole play through.

4. **Tetris.NET:** This game demonstrates the lowest requirements, with the CPU only demanding 7% and the GPU only demanding 1-2%.

5.1.3 Experiment Results Discussion

According to the results, CPU and GPU demands vary across different games and graphical settings. It is evident that there's no "one-size-fits-all" benchmark for simulations, necessitating a spectrum of utilization values. For our simulations, we exclude Tetris.NET, owing to its minimalist graphical fidelity, making it an unlikely candidate for cloud gaming services. Conversely, Tetris.com is kept as a benchmark for simple games. Thus, for our following simulations, we decided on GPU utilization values of 15%, 60%, and 100%, and 15%, 30%, and 70% for CPU utilization values.

5.1.4 Limitations and Threats to Validity

The results presented in this experiment, and all those to follow, are valuable, but their limitations should also be taken into account:

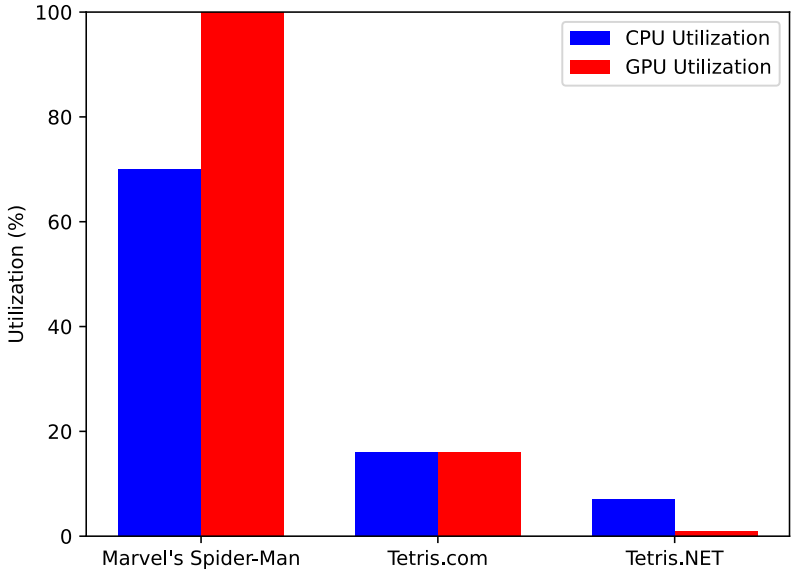

Figure 5.2: CPU and GPU utilization values for different games

1. **Game Diversity:** The selected games represent only a fraction of the diverse game genres and graphical intensities available.

2. **Hardware Variability:** While sufficient for our needs, our chosen device is just one of numerous possible configurations. Most importantly, machines used to run cloud gaming services are built to run games and are optimized to be used in these services. This, presumably, allows them to manage resources better than any personal gaming laptop. GPU utilization results may also be influenced by a machine's strength and age. Running Marvel's Spider-Man on the latest version of the gaming laptop in this experiment might result in much lower GPU utilization than, for example, that of a weaker one that was not designed for gaming.

5.2 A Comparison Between the Simulated Energy Consumption of Cloud Gaming Platforms and Their Respective Consoles' Reported Energy Consumption

Our main findings from this experiment are:

1. **MF4:** Our implementation of GPU simulation in OpenDC produces realistic results

for cloud gaming simulations.

2. **MF5:** Further refinement and data collection are necessary to improve the accuracy of the power models for our cloud gaming services presets.

In this experiment, we aim to verify the validity of our implementation of GPU simulation into OpenDC.

5.2.1 Experimental Setup

For this experiment, we use a trace of four hours with 160 users. Four hours (39) is a reasonable time for a gaming session. The choice of 160 users is based on the assumptions made in Chapter 3. We use two of our present, XCloud and PlayStation Plus, because we have official data on the energy consumption of the Xbox Series X (28) and PlayStation 5 (32) that we can base our comparison on.

Traces are created using our experiment generator. User counts and time frames are identical for every run, except for three parameters that vary:

1. **Topology:** We use two of the presets that we introduced earlier. This means varying CPU and GPU amounts and capacities.

2. **CPU utilization:** CPU utilization varies greatly between different video games, different settings, different CPUs, and different machines (40). For the scope of this experiment, we run each experiment with 15% CPU utilization, 30%, and 70%, based on our findings in Section 5.1.

3. **GPU utilization:** GPU utilization also varies depending on different variables such as the game played, the settings, the GPU itself, and so on. There seems to be a higher variance than in the CPU utilization, and a 95% utilization is perfectly reasonable for a game (41). For the scope of this experiment, we run each experiment with 15% GPU utilization, with 60%, and with 100% utilization, again, based on our findings in Section 5.1.

The presets for each service provider are run nine times, once for each variation in CPU and GPU utilization. After converting our results to Watts per game instance, we plot them, along with their respective averages, against publicly available data regarding each platform's relevant console's energy consumption.

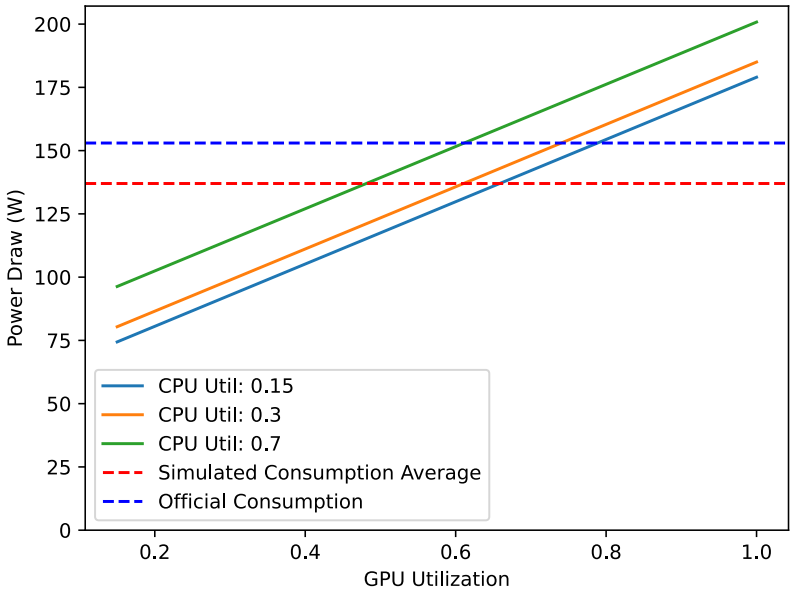

Figure 5.3: xCloud: power draw per game instance

5.2.2 Experimental Results

The results of the experiment are presented in two plots. The first plot, Figure 5.3, represents the xCloud run, and the second plot, Figure 5.4, represents the PlayStation Plus run. Each plot has three graphs, each representing a different CPU utilization. The X-axis represents different GPU utilizations. The Y-axis represents the power draw in W. In addition, we have a graph representing the average power draw for all experiment runs and a graph representing the official reported power draw for the relevant console.

For the xCloud run, the minimum value we observe is 74.4 W for a CPU utilization value of 15% and a GPU utilization value of 15%, the highest value we observe is 200.8 W for a CPU utilization value of 70% and a GPU utilization value of 100%, the simulated average is 137 W, and the official reported value for Xbox Series X is 153 W (28). For the PlayStation Plus run, the minimum value we observe is 69.4 W for a CPU utilization value of 15% and a GPU utilization value of 15%, the highest value we observe is 185 W for a CPU utilization value of 70% and a GPU utilization value of 100%, the simulated average is 110.8, and the official reported value for PlayStation 5 is between 209.8 and 210.9 W, depending on the definition (32).

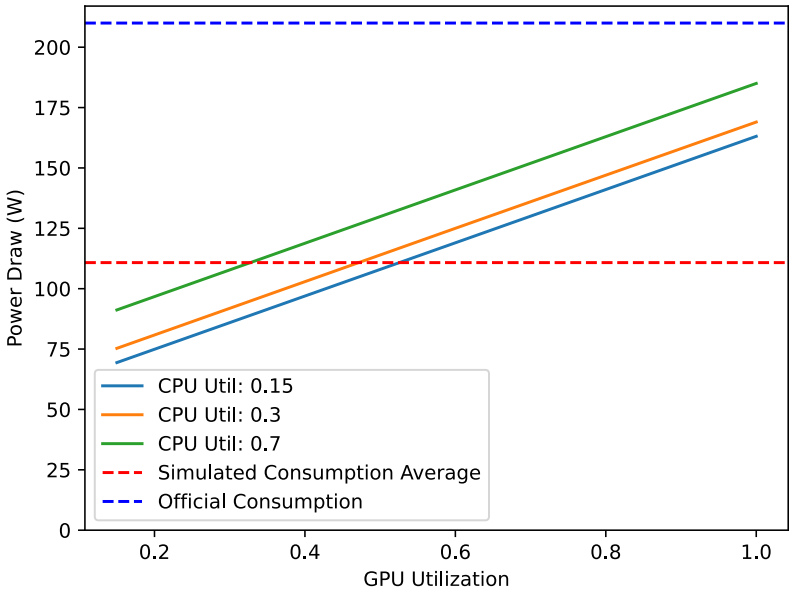

Figure 5.4: PlayStation Plus: power draw per game instance

5.2.3 Experiment Results Discussion

This experiment evaluated the validity of our implementation by running each preset with various GPU and CPU utilization combinations. Our goal is to compare the power draw results obtained from our simulation with the officially reported values for the respective hardware (Xbox Series X and PlayStation 5).

When comparing the power draw data obtained from our simulation with the officially reported values for the respective hardware (Xbox Series X and PlayStation 5), we found inconsistent results. For the xCloud preset, the simulated average power draw (137 W) corresponded well to the official value of an Xbox Series X (153 W) (28), indicating that our simulation model for xCloud's power consumption is relatively accurate and can be relied upon for further analysis. However, for the PlayStation Plus preset, we found a significant difference between our calculated average power draw (110.8 W) and the official reported value (210 W) (32). Still, we believe this difference is reasonable enough and does not invalidate our implementation.

5.2.4 Limitations and Threats to Validity

The decision to compare an average power draw over all of our simulated results using different utilization values with the reported power draw values of the respective consoles during active gaming was made to see if our results were reasonable values for the power draw of a gaming instance. While we aim to capture the essence of power consumption accurately, our results indicate further refinement and data collection are necessary to improve the accuracy of the power models for PlayStation Plus.

Aside from inherent contextual differences between our simulated cloud gaming platform and a single console, several factors can contribute to the discrepancy between our results. First, there are the underlying power models, which are based on both real data and assumptions. Secondly, the method of calculating the average power draw should be considered. Games may run on varying utilization levels for varying durations, so averaging power draw over various utilization levels might not be the most suitable approach.

Further, it is essential to understand that gaming workloads are dynamic in nature. Games can run on GPUs with high utilization and CPUs with moderate utilization, leading to power draw patterns that differ from the average. The officially reported values may reflect specific gaming scenarios, which adds to the variance in power consumption observed between our simulation and the official values. For example, the reported value for a PlayStation 5 game is 210 W, while a PS4 game running on a PlayStation 5 reports a power draw of 97.2 W, illustrating the significant variance possible in gaming scenarios.

Overall, while fine-tuning the calculation method or considering different utilization profiles might yield results closer to the officially reported values, we believe the results are sufficiently close to validate the reliability of our implementation of GPU simulation into the OpenDC simulator

5.3 The Effect of Different GPU Power Models on Energy Consumption

Our main findings from this experiment are:

1. **MF6:** The GPU power model we employ can significantly affect overall energy usage and needs to be carefully considered.

2. **MF7:** Adding new power models to our implementation is simple and does not require any architectural changes.

This experiment explores the impact of using different GPU power models for the same simulation and the steps required to take to add them to our implementation.

5.3.1 Experimental Setup

In this experiment, we use a trace of four hours with 160 users and our xCloud preset. We use four different utilization values for each power model we test 15%, 30%, 70%, and 100%, always identical for CPU and GPU, so we can see how the effect of the power model changes. We run the simulation twelve times, once for every combination of power model and utilization we test. The power models are the existing GPU linear power model, the GPU square root power model, and the GPU cubic power model. To show the effects of the different GPU power models, the CPU power model stays linear during all of the simulations. The square root and cubic power models were added for this experiment based on the CPU version of these power models. The square root power model formula is:

$$E(u) = E_{idle} + (E_{max} - E_{idle}) \times \sqrt{u}$$

and the cubic power model formula is:

$$E(u) = E_{idle} + (E_{max} - E_{idle}) \times u^3$$

where:

- E is the energy consumption
- u is the utilization

An example of the required code addition for a new power model can be seen in Figure 5.5.

5.3.2 Experimental Results

The results are presented in Figure 5.6. Each bar represents the energy consumption for a combination of CPU and GPU utilization and GPU power model. For CPU and GPU utilization of 15%, the results are 36.1 kWh for the cubic power model, 47.6 kWh for the linear power model, and 66.3 kWh for the square root power model. For CPU and GPU utilization of 30%, the results are 41.7 kWh for the cubic power model, 63.2 kWh for the linear power model, and 82.7 kWh for the square root power model. For CPU and GPU utilization of 70%, the results are 76.8 kWh for the cubic power model, 104.9 kWh for the linear power model, and 115.7 kWh for the square root power model. For CPU and GPU utilization of 100%, the results are 136.1 kWh for all power models, varying only

```java
/**
 * Construct a square root {@link GpuPowerModel} that is adapted from CloudSim.
 *
 * @param maxPower The maximum power draw of the server in W.
 * @param idlePower The power draw of the server at its lowest utilization level in W.
 */
public static GpuPowerModel sqrt(double maxPower, double idlePower) {
    return new GpuPowerModels.SqrtPowerModel(maxPower, idlePower);
}

private static final class SqrtPowerModel extends GpuPowerModels.MaxIdlePowerModel {
    private final double factor;

    SqrtPowerModel(double maxPower, double idlePower) {
        super(maxPower, idlePower);
        this.factor = (maxPower - idlePower) / Math.sqrt(100);
    }

    @Override
    public double computePower(double utilization) {
        return idlePower + factor * Math.sqrt(utilization * 100);
    }
}
```

Figure 5.5: The code required for a new GPU power model

about 0.000001 kWh between them all. Figure 5.6 shows the results for 100% utilization as 136 instead of 136.1 for better readability. The energy consumption values are all for a workload of 4 hours and 160 users.

5.3.3 Experiment Results Discussion

Several conclusions can be drawn from the results. First, the GPU power model we employ Significantly affects overall energy usage. For 15% GPU utilization, we see an increase of 39.29% in overall energy consumption when using the square root power model instead of the linear power model and a decrease of 24.2% when using the cubic power model instead of the linear one. This means that the choice of power model for the simulation should be carefully considered.

Second, the effect of the power model varies between different GPU utilization levels. If we had an increase of 39.29% between the linear power model and the square root power model for 15% utilization, for 30% we get an increase of 30.85%, and for 70% utilization, we only get an increase of 10.25%. At 100%, the increase is almost unnoticeable as it is less than 0.01%. The increase we see between the cubic power model and the linear power model does not grow but acts differently. For 15% we get an increase of 31.93%, for 30% an increase of 51.65%, and for 70% the increase percentage goes down to 36.59%. This is due to the nature of the performed calculations in the power models we tested. All three

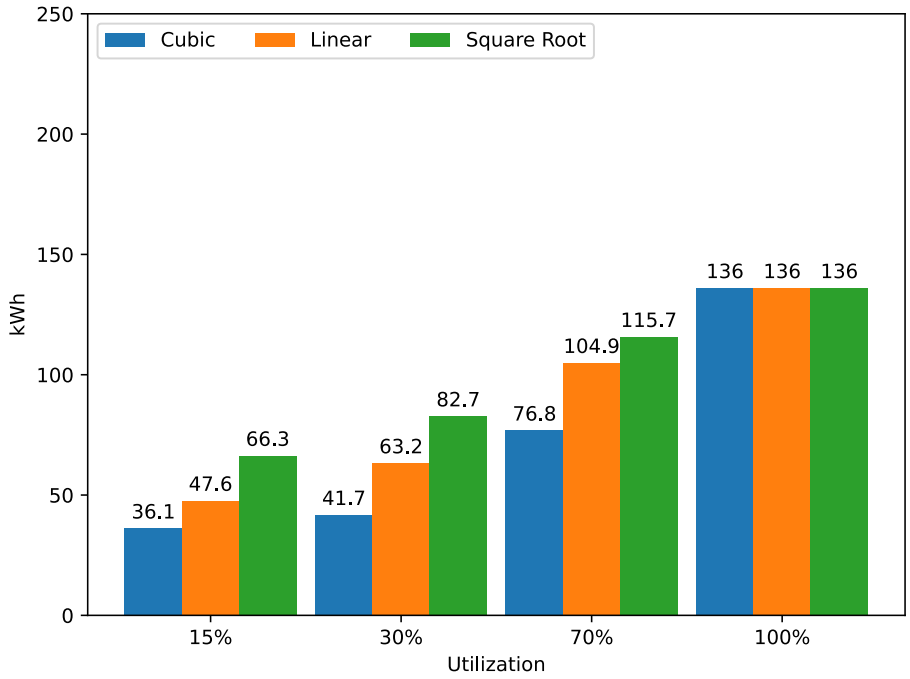

Figure 5.6: Energy consumption by CPU and GPU utilization and GPU power model

power models affect the utilization value in the calculation. Different power models could emphasize different components of the energy consumption calculation more.

Last, adding new power models to our implementation is simple and does not require any architectural changes (**IR1**).

5.3.4 Limitations and Threats to Validity

We use power models that do not differ greatly, all based on the way utilization is used for the calculation. They are based on CPU power models from (42) and might fit less for GPU simulation. There might be more complex GPU power models that will affect the overall energy consumption differently. Future work could further explore GPU power models.

5.4 How Much Energy is Consumed While Running a Cloud Gaming Service for One Day?

Our main findings from this experiment are:

1. **MF8:** CPU and GPU utilization values have a significant effect on energy consumption.

2. **MF9:** For our xCloud and PlayStation 5 preset simulations, we found that the price of operation is close to that of running Xbox Series X or PlayStation 5.

3. **MF10:** For our GeForce NOW preset simulation we found that it is significantly cheaper than running the same amount of game instances natively on a personal gaming computer.

In this experiment, we calculate the energy cost of running a cloud gaming service for one day. Through this experiment, we aim to reach two goals:

1. Calculate the energy consumption and costs of running a cloud gaming platform for a full day.

2. Compare the energy consumption and costs of running a cloud gaming platform with running games on a console or personal computer.

5.4.1 Experimental Setup

For this experiment, the user counts that we use, are based on Minetrack (36): We take the user count statistics of the Minecraft servers as an indication of how many players use a cloud gaming platform to play a certain game, and how the workload changes during a 24-hour time frame. While these statistics do not represent the number of players that use one of the bigger cloud gaming services, we argue that there is still a similarity between the number of users that play Minecraft online on their machine, to the number of users that play Minecraft on a cloud gaming service during a certain time frame. Furthermore, these user numbers can correspond to a smaller cloud gaming service.

The Minetrack dataset includes logs of reported player count in several different Minecraft online servers every roughly three seconds, over a time span of 24 hours. Since 3 seconds is far too precise of a time frame for the scope of our experiment the first step was to get the average player count per hour, for 24 hours. The traces are spread over the different tracked servers, which have varying player counts. To not generalize the counts too much, we didn't take the average count for all of the servers together but took the hourly average per server, over the whole 4 months of the dataset. For this experiment, we took the average user count per hour for the play.inpvp.net host.

Using our experiment generator, we created traces using these user counts for a period of 24 hours, and we employed identical user counts and timeframes for each run. The variations in topology, CPU utilization, and GPU utilization are identical to the previous

Table 5.1: Average player count by hour and IP

Hour	IP	Average Player Count
00:00:00	play.inpvp.net	1777.1
01:00:00	play.inpvp.net	1693.6
02:00:00	play.inpvp.net	1560.7
03:00:00	play.inpvp.net	1406.4
04:00:00	play.inpvp.net	1242.9
05:00:00	play.inpvp.net	1106.2
06:00:00	play.inpvp.net	1045.5
07:00:00	play.inpvp.net	1011.5
08:00:00	play.inpvp.net	973.1
09:00:00	play.inpvp.net	938.5
10:00:00	play.inpvp.net	949.2
11:00:00	play.inpvp.net	1031.8
12:00:00	play.inpvp.net	1182.5
13:00:00	play.inpvp.net	1357.6
14:00:00	play.inpvp.net	1563.5
15:00:00	play.inpvp.net	1779
16:00:00	play.inpvp.net	1925.5
17:00:00	play.inpvp.net	2013.5
18:00:00	play.inpvp.net	2074.1
19:00:00	play.inpvp.net	2096.2
20:00:00	play.inpvp.net	2029.8
21:00:00	play.inpvp.net	1961.8
22:00:00	play.inpvp.net	1890.3
23:00:00	play.inpvp.net	1826.8

experiment. Each service provider's preset is run 9 times, once with each variation in CPU and GPU usage

To determine the estimated cost in euros, we calculate the average kWh usage per platform for a 24-hour timeframe and multiply it by the price per kWh for a company in the Netherlands (43). To estimate the operating price of a similar number of game instances when running natively on consoles or personal computers in the users' homes, we calculate the average hourly user count, which is 1517, and multiply it by the consumption of the said machine in W and divide by 1000 to get kWh. For the consoles, we have the published data(32) (28), but for personal computers, a single number cannot be obtained because there are many different components in personal computers that can greatly influence the energy consumption. For this case, we use an estimation of 550 W for a mid-range gaming computer (44). Then, we multiply by 24 to get the kWh value for a whole day and multiply

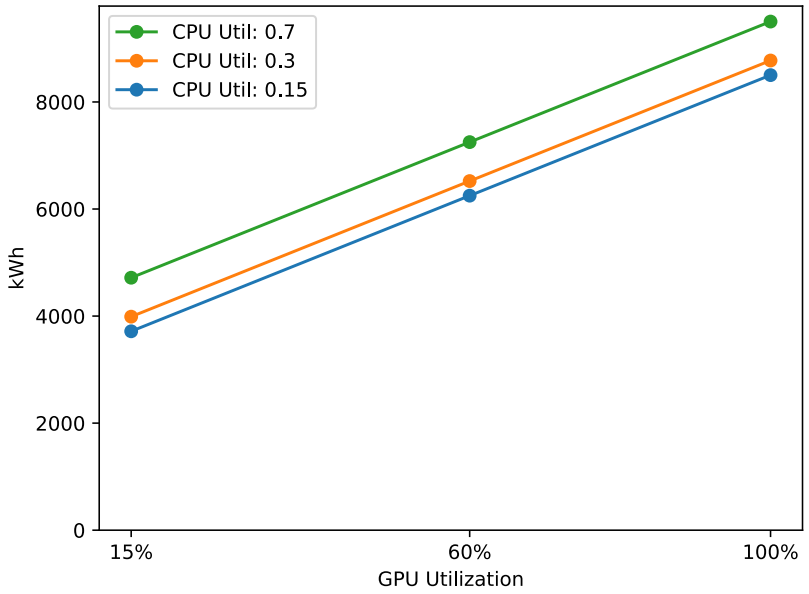

Figure 5.7: xCloud: Energy consumption over 24 hours for different CPU and GPU utilization levels

that by the price per kWh for a private user in the Netherlands.

5.4.2 Experimental Results

The results of the experiment are presented in five plots. Figure 5.7 presents the energy consumption results in kWh for the xCloud preset, which are 3715.8, 3988.7, and 4716.5 for 15% GPU utilization, 6249.4, 6522.3, and 7250.1 for 60% GPU utilization, and 8501.4, 8774.4, and 9502.1 for 100% GPU utilization. Figure 5.8 presents the energy consumption results in kWh for the PlayStation Plus preset, which are 3465.9, 3738.8, and 4466.6 for 15% GPU utilization, 5736.8, 6009.7, and 6737.5 for 60% GPU utilization, and 7755.4, 8028.4, and 8756.2 for 100% GPU utilization. Figure 5.9 presents the energy consumption results in kWh for the GeForce NOW preset, which are 3865.3, 4390.6, and 5791.2 for 15% GPU utilization, 5765.5, 6290.7, and 7691.3 for 60% GPU utilization, and 7454.5, 7979.8, and 8756.2 for 100% GPU utilization. Each plot has three graphs, each representing a different CPU utilization. The X-axis represents different GPU utilizations. The Y-axis represents the energy consumption in kWh.

The fourth plot, Figure 5.10, shows the estimated price in euros of running each of the cloud gaming platforms for 24 hours, compared to the estimated price of running the same

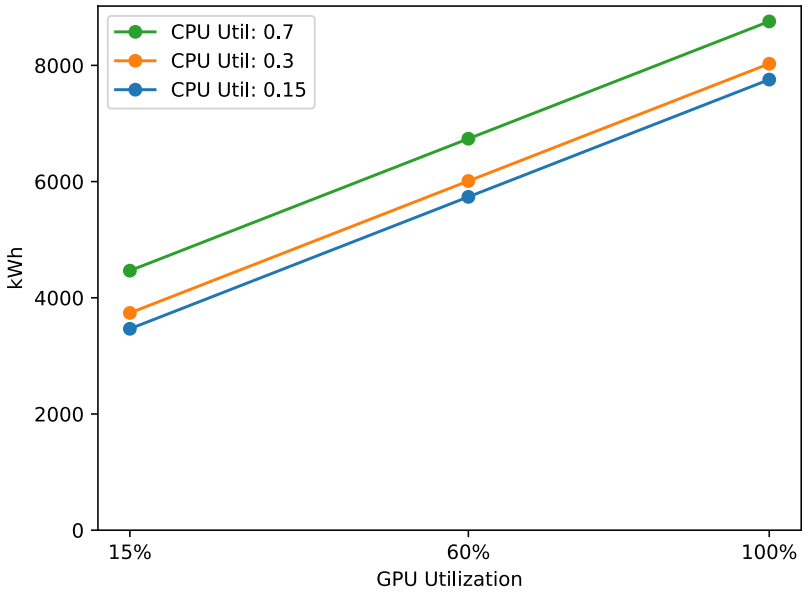

Figure 5.8: PlayStation Plus: Energy consumption over 24 hours for different CPU and GPU utilization levels

amount of game instances for the same amount of time on a console or a personal computer. The simulated price for xCloud is 3,342 euros, for PlayStation Plus it is 3,087 euros, and for GeForce Now it is 3,308 euros. The price we calculated for Xbox Series X is 2,646 euros, 3,631 euros for PlayStation 5, and 9380 euros for an average gaming personal computer.

Figure 5.11 shows the average energy consumption over 24 hours in kWh for each of the simulated cloud gaming services. The results are 6,580 for xCloud, 6,077 for PlayStation Plus, and 6,512 for GeForce NOW.

5.4.3 Experiment Results Discussion

In this experiment, we pursue two primary objectives: to calculate the energy consumption and costs of operating a cloud gaming platform for a full day and to compare the costs of running a cloud gaming platform with running games natively on consoles or personal computers.

Figures 5.7, 5.8, and 5.9 depict the energy consumption for our three presets. As expected, the graphs demonstrate a linear relationship, owing to the use of a simple linear power model. Although the differences between the three presets are not highly significant, it is evident that the CPU utilizations have diverse effects on each platform. For example,

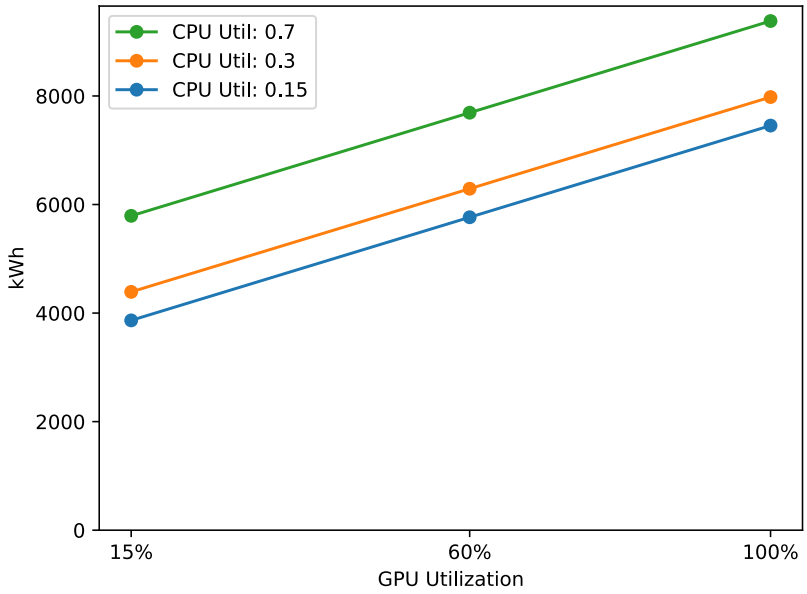

Figure 5.9: GeForce NOW: Energy consumption over 24 hours for different CPU and GPU utilization levels

in the case of GeForce Now, the rise in consumption between CPU utilization levels is more pronounced than in the other two presets. Conversely, the increments in consumption between GPU utilization levels for GeForce Now are smaller than those observed for the other two presets. These patterns align with our power model assumptions and are logically consistent. As we have seen in previous experiments, the utilization levels of both CPU and GPU affect the overall energy consumption greatly. We learn that games that require more resources or hardware less capable can cost significantly more to run.

Next, we address the estimated price of operating a cloud gaming service for 24 hours compared to running the same game instances natively on consoles or personal computers. It is important to consider that our simulation utilizes a user count of a small cloud gaming setup and results in operational prices that fit this scope rather than that of a whole commercial cloud gaming service. To obtain more accurate estimations of a bigger scope, future work with more real data can be conducted.

We then analyze the specific results for each platform. In the case of xCloud, our estimated cost surpasses the calculation for running Xbox consoles natively. This can be attributed to the fact that the reported power draw of an Xbox Series X is relatively low compared to the CPU and GPU TDPs. Conversely, the estimated cost for PlayStation

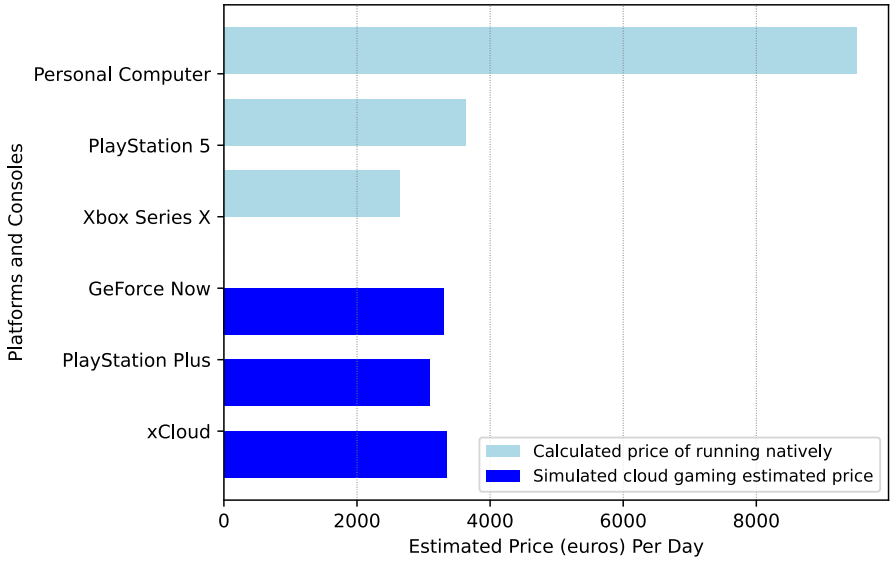

Figure 5.10: The estimated price (euros) of running a cloud gaming platform for 24 hours compared to the estimated price of running game instances natively

Plus is lower than the calculation for running PlayStation 5 natively. This is likely due to the reported power usage of a PlayStation 5 being relatively high compared to the CPU and GPU TDPs. Because our xCloud and PlayStation Plus presets are based on the Xbox Series X and PlayStation 5 consoles, the results are mostly as we expected. For a deeper understanding of the differences between cloud gaming services and their respective consoles, we will need more data on how these services are built. However, our results indicate little difference between running xCloud or PlayStation Plus and their respective consoles in terms of energy consumption.

The most interesting observation arises from the results obtained for GeForce Now. Despite the various assumptions made regarding TDP, power usage, and resource allocation, it is intriguing to witness the substantial cost difference in this case. Gaming computers are known for their significant power consumption, and here, the margin between estimated costs and native gaming costs is the most pronounced. According to our results, cloud gaming could save a lot of money and energy as compared to using a personal gaming computer

5.4.4 Limitations and Threats to Validity

We still lack a lot of data regarding the topology of cloud gaming services, and our results are based on the assumptions we made in Chapter 3. Furthermore, the data we use for

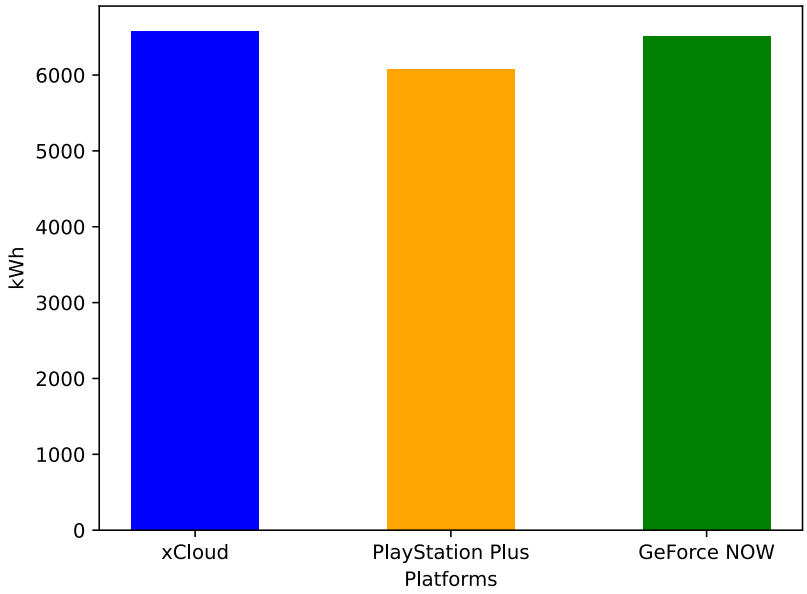

Figure 5.11: The average energy consumption of different cloud gaming services

console and personal computer power consumption calculations can be quite volatile. For consoles, the data could change significantly between different games, and personal computers vary even more as they are more flexible and can have multiple different hardware setups, leading to highly different power consumptions. The price of running a cloud gaming service has more to it than what we consider in our simulation, like cooling, hardware replacement, salaries, and so on.

5.5 Does Running the Same Amount of Game Instances With More GPUs Lower Consumption?

Our main findings from this experiment are:

1. **MF11:** Reducing the GPU utilization by increasing the amount of computing power per VM might help reduce the energy consumption of the cloud gaming service.

2. **MF12:** Further work is required on our GPU implementation to give a more definitive answer to this question.

In this experiment, we test how different GPU allocations could affect the overall energy consumption of a cloud gaming service. We want to see if service providers should try

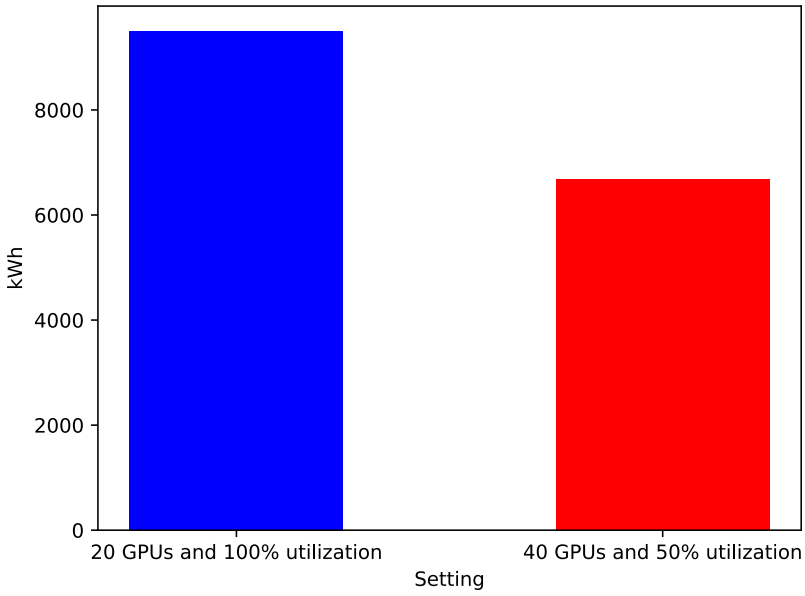

Figure 5.12: Energy consumption for two different settings

to minimize GPU utilization to keep the costs low or if maximizing utilization is better for energy consumption. The idea behind the experiment is that lower GPU utilization consumes less energy but requires more processing power to achieve, so more hardware is required to run the same amount of game instances.

5.5.1 Experimental Setup

We use the same user counts trace as in Section 5.4. We use the xCloud preset, with 70% CPU utilization. The variance between the two runs is the combination of GPU utilization and the overall number of GPU cards in the cluster. One run has 50% GPU utilization with 40 GPUs per cluster, like in the original xCloud model. The other run has 100% GPU utilization with 20 GPUs per cluster.

5.5.2 Experimental Results

Figure 5.12 shows the energy consumption for the two different simulated settings. The energy consumption for the run with 40 GPUs and 50% GPU utilization is 6687.1 kWh. The energy consumption for the run with 20 GPUs and 100% utilization is 9502.2.

5.5.3 Experiment Results Discussion

From the results, we can deduce that keeping GPU utilization low should be prioritized over getting the most computing power out of every GPU available on the server. The results suggest that there can be a significant reduction in energy consumption if data centers employ such a strategy. We did not expect this result, as we hypothesized that adding more hardware would introduce a baseline energy consumption for every added component that would be more significant than the saving achieved by the reduced GPU utilization.

5.5.4 Limitations and Threats to Validity

Our GPU implementation is utilization-based, so adding more or fewer GPUs when we set utilization does not change the final energy consumption. Further work on the GPU implementation is required to get more meaningful results. Also, the choice of power model affects the significance of the utilization in a major way so different power models might suggest different results.

5.6 How Does Running Multiple Game Instances On a VM Affect Energy Consumption?

Our main findings from this experiment are:

1. **MF13:** Increasing the amount of game instances per VM might improve the overall energy consumption of the cloud gaming service.

This experiment examines the effects of running more than one game instance on each VM. Doing that will require more CPU and GPU power for every VM but will require fewer VMs to start overall. This could prove economical for games that require fewer resources and can be run like that.

5.6.1 Experimental Setup

We again use the user counts trace as in Section 5.4. In this experiment, we do not use our experiment generator as it defaults to using one game instance per VM. Instead, we will use the topology and trace generators separately. The setup is based on our xCloud preset. We ran four simulations, all with 14 clusters, 160 CPU cores, and 20 GPUs.

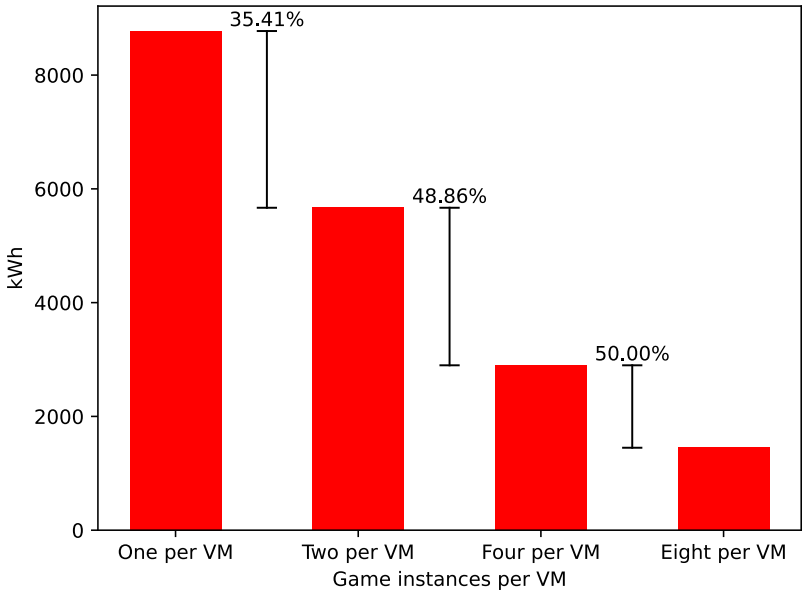

Figure 5.13: Energy consumption for four different settings

1. **One game instance per VM:** 160 hosts per cluster. Each host has one CPU core and a fraction of the GPU computing power which equals 228.125 MHz.

2. **Two game instances per VM:** 80 hosts per cluster. Each host has two CPU cores and a fraction of the GPU computing power which equals 456.25 MHz.

3. **Three game instances per VM:** 40 hosts per cluster. Each host has four CPU cores and a fraction of the GPU computing power which equals 912.5 MHz.

4. **Four game instances per VM:** 20 hosts per cluster. Each host has eight CPU cores and a fraction of the GPU computing power which equals 1,825 MHz.

Because the CPU utilization is calculated per CPU core, we left it at 30% for all cases. The GPU utilization stayed at 100% for all cases because while the capacity doubles, so does the usage. Because each CPU in our model of xCloud has eight cores, we are not testing more than eight game instances per VM.

5.6.2 Experimental Results

Figure 5.13 shows the energy consumption for 24 hours for the different simulated settings. The results are:

1. **One game instance per VM:** 8774.4 kWh.
2. **Two game instances per VM:** 5667.7 kWh.
3. **Two game instances per VM:** 2898.4 kWh.
4. **Two game instances per VM:** 1449.2 kWh.

Between the bars, we can see the percentage of energy consumption decrease between the results. The percentages are 35.41% between one instance per VM and two instances per VM, 48.86% between two instances per VM and four instances per VM, and 50% between four instances per VM and eight instances per VM.

5.6.3 Experiment Results Discussion

The results show a significant decrease in energy consumption the more games we run on one VM. This could suggest that cloud gaming service providers should strive to use fewer VMs for the same amount of game instances to save on energy costs. We did expect to see some improvement when running more game instances per VM as it cuts down on the overhead of running each VM. However, for four and eight instances per VM, we see bigger savings in energy consumption than what we expected based on recent work that suggests possible savings of 20%-40% by IT equipment optamization (8). Furthermore, we expected to see diminishing returns for every game instance we add, as the complexion of running more games could be costly, but it is the other way around, with the improvement in energy consumption increasing for every iteration of the experiment.

5.6.4 Limitations and Threats to Validity

The simulation does not take into consideration the extra overhead that is introduced when running more than one game instance on a VM, so the energy consumption of running multiple game instances per VM could be higher. Also, if one game instance per VM uses x of the GPU resources, it does not necessarily mean that running two games per VM will use 2x of the GPU resources. There might be extra resources needed for every added game instance, which also could increase energy consumption. Furthermore, as with the previous experiment, some improvements might be required to our GPU simulation implementation to provide more concrete results.

5.7 How Much Does Energy Consumption Increase When Running Games in 4K On The Cloud?

Our main findings from this experiment are:

1. **MF14:** The increase in energy consumption and price that incurs from streaming 4K gaming over streaming 1080p is moderate and service providers might benefit from offering this to users.

2. **MF15:** Adding new cloud gaming service presets to our implementation is simple and does not require any architectural changes.

In this experiment, we test the difference in energy consumption and price between running a small cloud gaming service that streams games in 1080p and one that streams games in 4K. Cloud gaming service providers might want to offer their users the option to stream games in 4K to improve the value of their service and potentially charge a higher subscription rate, but 4K gaming can also incur a significant increase in operation costs. We want to test how much more expensive streaming in such resolutions might be.

5.7.1 Experimental Setup

We use the same player count trace as in Section 5.4 to simulate a small cloud gaming service. For this experiment, we use an adjusted version of the GeForce NOW preset that better fits 4K gaming. Our local hardware is not sufficient to test GPU and CPU utilization levels for 4K gaming, and we were unable to find any online reports, so we used a YouTube video by zWORMz (45) that tested the different utilization levels when running Marvel's Spider-Man Remastered at a variety of resolutions with NVIDIA GeForce RTX 3070 (46). The preset has 160 CPU cores of the same type per server, the same amount of GPUs, an RTX 3070 instead of an RTX 3060, and 40 game instances instead of 160 to provide more processing power per game instance.

The CPU utilization levels observed in the video are the same for both 1080p and 4K and sit at an average of 40%. The GPU utilization level for 1080p ranges from 60% to 90% but averages at around 75%. For 4K, the GPU utilization level stays constant at 99%. We ran the experiment twice, both times with 40% CPU utilization. Once with 75% GPU utilization, and once with 99% GPU utilization.

We calculate the overall energy consumption in kWh and the price in euros over 24 hours. The price is calculated the same way it is calculated in Section 5.4, by multiplying the energy consumption by the price of kWh for a company in the Netherlands (43).

```
"geforcenow4k" → {
                cpuCount = 160
                gpuCount = 40
                cpuCap = 3.5
                gpuCap = 1.5
                memCap = 1280L
                gameInstancesPerCluster = 40
                cpuIdleDraw = 23.0
                cpuMaxDraw = 125.0
                gpuIdleDraw = 40.0
                gpuMaxDraw = 220.0
            }
```

Figure 5.14: The code required for a new cloud gaming service preset

5.7.2 Experimental Results

In Figure 5.14 we can see the code that was needed in order to add the new preset. In Figure 5.15 we can see the energy consumption and price differences of running games in 4K over 1080p in a cloud gaming service for a whole day. The energy consumption for the 1080p run is 9262.9 kWh. The energy consumption for the 4K run is 10746 kWh. We can see an increase of 16.01% in energy consumption. The price for running the service for a whole day with a 1080p resolution is 4,706 euros, while for 4K, the price is 5,459 euros. An increase of roughly 50 cents per player for 24 hours.

5.7.3 Experiment Results Discussion

The results suggest a 16.01% increase in energy consumption, which might be a reasonable tradeoff for the gained improvement in service quality. The added price of around 50 cents per player per day is not very steep, and cloud gaming service providers could offer that to users who pay for a higher subscription tier, for example. This could be a way to lure more users into the service and ideally decrease the number of personal gaming computers used, which, as we have seen in Section 5.4, can consume considerably more energy and have a more significant negative effect on the environment.

On the other hand, the increase in energy consumption might prove too big for service providers, with the added cost making the addition of 4K streaming non-beneficial to them.

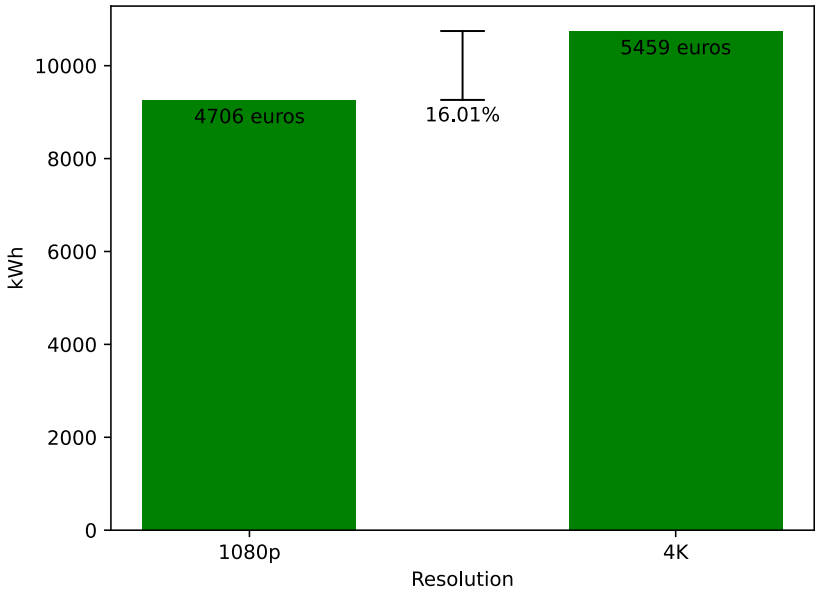

Figure 5.15: Energy consumption and price different resolutions

Furthermore, as established in Chapter 1, any increase in energy consumption per game instance has a direct negative effect on the environment.

5.7.4 Limitations and Threats to Validity

The experiment does not consider the extra power that is needed to stream 4K over 1080p. We only tested the data center side of the equation, but streaming 4K images requires more bandwidth and more power (47). This could increase the cost significantly. The utilization values we use are based on one video and not on experimentation. Furthermore, the spike in GPU utilization when streaming 4K was not the only effect. The FPS rate dropped from an average of around 150 when streaming 1080p to an average of around 70 when streaming 4K. While an FPS rate of above 30 is considered serviceable and an FPS of above 60 is largely considered very good (48), the decrease in frame rate could be even bigger than what was observed in this video, and service providers should consider the quality of service when opting to offer 4K streaming to users.

5.8 Summary

All seven experiments presented in this section were designed to verify our model and implementation. They also addressed **RQ3**: *How does data centers' design affect cloud gaming's energy consumption?*

The first experiment (Section 5.1) served as a guide for determining the appropriate CPU and GPU levels for subsequent experiments. This was accomplished by running four different games on our machine and keeping track of the CPU and GPU utilization levels that result from different graphical settings. The results showed that utilization levels can vary greatly between games and graphical settings. Based on our observations, we chose to use CPU utilization levels of 15%, 30%, and 70%, and GPU utilization levels of 15%, 60%, and 100%.

In Section 5.2, we ran an experiment to validate our implementation of GPU in OpenDC. We utilized a simple workload, divided the results per user, and compared the power consumption per player with the power consumption of the relevant console. The average power consumption we simulated for xCloud is 137 W, with the official reported value for Xbox Series X being 153 W. The average power consumption we simulated for PlayStation Plus is 110.8 W, with the official reported value for PlayStation 5 being between 209.8 and 210.9 W, depending on the definition. Our results for the xCloud run show that our implementation is valid, and while there is a bigger difference for the PlayStation Plus run, we still believe this is a reasonable difference, considering the fact that the reported PlayStation 5 power consumption values are very volatile, depending on the game, and go as low as 97.2 W for active gaming for a PS4 game.

In Section 5.3, we tested the effects of different GPU power models on energy consumption. We ran tests on three different GPU power models. Linear, square root, and cubic, each with four different utilization levels, 15%, 30%, 70%, and 100%, always with identical CPU and GPU, so we can see how the effect of the power model changes. The linear power model was already implemented. The two others were implemented for this experiment, showcasing the simplicity of adding new power models. For CPU and GPU utilization of 15%, the results are 36.1 kWh for the cubic power model, 47.6 kWh for the linear power model, and 66.3 kWh for the square root power model. For utilization of 30%, the results are 41.7 kWh for the cubic power model, 63.2 kWh for the linear power model, and 82.7 kWh for the square root power model. For utilization of 70%, the results are 76.8 kWh for the cubic power model, 104.9 kWh for the linear power model, and 115.7 kWh for the square root power model. For utilization of 100%, the results are 136.1 kWh for all power

models, varying only about 0.000001 kWh between them all. These results demonstrate the significant effect the GPU power model we employ can have on overall energy usage.

In Section 5.4, we examined the cost and energy consumption of running a cloud gaming service for a full day and compared it with the price of running the same amount of games on the relative console. Our energy consumption for a full day of running xCloud varies from 3715 kWh to 9502 kWh with an average of 6580 kWh and an estimated price of 3,343 euros. For PlayStation Plus it varies from 3466 kWh to 8756 kWh with an average of 6077 kWh and an estimated price of 3,087.2 euros. For GeForce NOW it varies from 3865 kWh to 9380 kWh with an average of 6512 kWh and an estimated price of 3,308.1 euros. The calculated prices for running the same amount of game instances on an Xbox Series X, a PlayStation 5, or a personal gaming computer are 2,646, 3,631.7, and 9,511.6 euros respectively. We can see that CPU and GPU utilization values significantly affect energy consumption. For our xCloud and PlayStation 5 preset simulations, the price of operation is close to that of running Xbox Series X or PlayStation 5. However, for our GeForce NOW preset simulation, we observe that it is significantly cheaper than running the same amount of game instances natively on a personal gaming computer. This is largely because generally speaking, personal gaming computers require a lot of energy to power. Also, we based our power consumption for a personal gaming computer on a mid-range gaming computer (44), but personal gaming computers vary significantly in their hardware and power consumption.

In Section 5.5, we tested whether running the same amount of game instances with more GPUs and a lower utilization per game decreases or increases overall energy consumption. The energy consumption for the run with 40 GPUs and 50% GPU utilization is 6687.1 kWh. The energy consumption for the run with 20 GPUs and 100% utilization is 9502.2. These results show a significant decrease in energy consumption achieved by lowering the GPU utilization by increasing the number of GPUs and could suggest that cloud gaming service providers follow a similar strategy. However, our GPU implementation in OpenDC mainly uses utilization to determine energy consumption, and no weight is given to the number of components. This means that we will see improvements as long as we decrease GPU utilization, regardless of the number of GPUs we add. To properly run this experiment, some improvements to our GPU implementation in OpenDC are required, like modeling the effect of having more GPUs on energy consumption.

In Section 5.6, we considered a different potential change to our cloud gaming service model. Instead of running one game instance per VM, we test running multiple game instances per VM. We ran tests with one, two, four, and eight game instances per VM.

The results show an increasing improvement in energy consumption for every extra game instance we add. For one game instance per VM, the energy consumption is 8774.4 kWh, for two, the energy consumption is 5667.7 kWh, for four, the energy consumption is 2898.4 kWh, and for eight, the energy consumption is 1449.2 kWh. While beneficial, the experiment does not consider the overhead that is gained by running more games per VM. Furthermore, the energy consumption value that is observed for running eight game instances on a VM seems to be too low for the number of game instances we simulate. Again, further improvements to the GPU implementation could provide more accurate results.

In Section 5.7, we looked at the increase in energy consumption and price when streaming games in 4K instead of 1080p. Cloud gaming service providers might want to offer 4K streaming to their users to increase the value of the service and attract more customers. This, however, could mean an increase in costs for the service provider. We implemented a new cloud gaming service provider preset that fits 4K gaming better, and we ran two simulations with varying GPU utilization to test how much the energy consumption and price will increase. For the 1080p run, we observed an energy consumption of 9262.9 kWh for 24 hours and a price of 4,706 euros, while for the 4K run, we observed an energy consumption of 10746 kWh for 24 hours and a price of 5,459 euros. An increase of roughly 50 cents per player for 24 hours. This is not such a substantial increase, and it could prove beneficial for the service providers to offer such a service to the users. However, this simulation does not consider the extra power that is needed to stream 4K over 1080p, as it is not modeled yet. Streaming 4K might prove to be more expensive than what we observe in our simulation. Furthermore, even a 16.01% increase in energy consumption, as we observe, still means a higher price for the environment.

In conclusion, the experiments we conducted helped us verify our model and implementation, gain insights regarding CPU and GPU utilization in video games, better understand the price of running cloud gaming services, and explore various data center topologies for cloud gaming services and their effects on energy consumption.

6

Related Work

The presented research is based on a body of recent research on the environmental impact of data centers. In this section, we provide a brief overview of the literature.

In 2009, Banerjee et al. (2) discussed the energy consumption of servers in the US, and suggested possible improvements in data center design. Also in 2009, Liu et al. (49) presented the GreenCloud architecture, which aims to reduce data center power consumption. Beloglazov et al. (11) defined an architectural framework and principles for energy-efficient Cloud computing and proposed energy-efficient resource allocation policies and scheduling algorithms in 2012. In 2023 Zhu et al. (8) analyzed the energy conservation and emission-reduction technologies and potential decarbonization paths for data centers. Regardless of it being a point of interest for a while, the energy consumption of data centers is only on the rise. Abu Bakar et al. (50) have raised concerns regarding this issue in 2021. Data centers have been discussed as a tool to improve the sustainability of the gaming industry through cloud gaming (51). However, the question of its actual sustainability has been raised (52). In 2019 the California Energy Commission looked into improving computer gaming energy efficiency and presented an analysis of the annual electricity use and annual costs of cloud gaming in California. Bhuyan et al. (47) looked into the energy consumption of cloud gaming in an end-to-end scope, going from cloud gaming services to mobile devices. However, they put their emphasis on mobile platforms, and test the energy consumption of the data center by setting up a local server with a desktop processor.

The work conducted by He (42) on data center power modeling was referenced when designing and implementing our model. Additionally, this research makes use of the OpenDC data center simulator (13, 14).

7

Conclusion

Video games are on a constant rise, and cloud gaming tries to cement itself as the new way to play, benefiting players who do not have to buy the newest hardware, video game companies that can reach a wider install base, and potentially, the environment, by reducing electronic waste. This thesis explores the environmental and economic aspects of cloud gaming, by designing a cloud gaming energy consumption model, implementing this model into the OpenDC data center simulator, and running experiments that explore different data center topologies and their effect on energy consumption.

7.1 Answering Research Questions

RQ1 *How to model energy usage in cloud gaming services?*

In Chapter 3 we introduce our design of a cloud gaming energy consumption model. Our approach includes player count traces, assumptions regarding resource allocation, and an addition of GPU utilization. The model considers both CPU and GPU utilization levels when calculating the energy consumption of a workload and includes a GPU power model that dictates how the utilization affects the final energy consumption calculation. We have also modeled three real-world cloud gaming service providers that fuse all of the different aspects of the model. This helps showcase the completeness of our model.

Our model supports simulating cloud gaming services at a realistic scale, with varying topologies. While our assumption is that every VM in our model is in charge of one game instance, our model is flexible and this can be changed to explore different solutions. As we have done in the Chapter 5.

RQ2 *How to implement the cloud gaming energy usage model into a discrete-event simulator?*

In Chapter 4 we present the implementation of our cloud gaming energy consumption model in OpenDC. This implementation is made up of two parts, the cloud gaming workload generator and the GPU simulation implementation. The workload generator is a set of tools that help the user quickly generate topologies, traces, and meta files by providing details like the number of CPUs, the number of GPUs, the number of users, the desired length of the workload, and so on. It includes our modeling of real-world cloud gaming services as presets to use and our resource allocation model. The tool is flexible and more cloud gaming presets could be easily added to it.

The implementation of GPU simulations required the addition of GPU power models, and a PSU (Power Supply Unit) gaming profile. Furthermore, we had to make structural changes to the machine models to accept a graphics processing unit. An additional multiplexer was added to the hypervisor to enable differentiation between CPU and GPU in the simulation. GPU simulation can now be used in OpenDC wether it is for cloud gaming workloads or different ones.

RQ3 *How does data center design affect cloud gaming's energy consumption?*

In Chapter 5 we look at different cloud gaming topologies and their effects on energy consumption. We investigate how different GPU power models affect the simulation results, and learn that the GPU power model we employ can have a significant effect on overall energy usage and needs to be carefully considered. We run a baseline test to learn how much money it costs to run a cloud gaming service for a day, and if it is more wasteful than running games at home, and observe that for a cloud gaming service in the scope we tested, the price for 24 hours ranges between 3,087.2 and 3,343 euros, and that while xCloud and PlayStation Plus do not present significant savings over their respective consoles, GeForce NOW presents a substantial saving over using personal gaming computers.

We experiment with different topologies, like reducing the GPU utilization by using more GPUs, or running more game instances per VM, and learn that potentially it can help reduce energy consumption, but our implementation requires further improvements to determine so.

Finally, we looked into the increase in energy consumption of streaming games in 4K over streaming in 1080p and determined that the increase might be reasonable enough

for cloud gaming service providers to offer 4K streaming for their users. However, we acknowledge that a more refined model and implementation may be needed to support this finding.

7.2 Limitation and Future Work

Our current model relies on many assumptions we made during the modeling phase, as a result of not having any real-world data from cloud gaming providers. In the future, achieving such traces will help create a more accurate model that could be used further to research the environmental costs of cloud gaming. These traces could include data like the types of CPUs and GPUs used, CPU core allocation per VM, the number of game instances that run on a VM, and so on.

Our GPU implementation gives a lot of emphasis to utilization, but not much to the amount of GPUs used and the GPU specifications, like the number of cores or the memory size. Implementing a more complete GPU model could help give more accurate results to questions like the ones we asked in Chapter 5 regarding running different amounts of game instances per VM and using different numbers of GPUs per server. Improving the GPU implementation could also enable further research into GPU virtualization, and GPU offloading

The model can also be enriched by the implementation of more hardware components, like different network components or cooling systems. By taking into consideration these components, we can achieve a more realistic data center simulation, and in turn, a more realistic cloud gaming energy consumption simulation.

Another aspect that could be further researched is the quality of service (QoS). In our current implementation, we do not consider aspects like FPS (frames per second) rate and latency. Both are critical when considering cloud gaming. While we can look at energy savings through different topologies and techniques, if as a result of our suggestions the FPS drops too much, or the latency is too high, no one will want to use the service, and our suggestions will be rejected. In our 4K experiment, for example, the increase in costs for streaming 4K was not too high, but we did not consider how the FPS was affected by increasing GPU utilization. Furthermore, 4K streaming can be intensive on the network, and affect negatively the power consumption of the machine the player is using. Modeling these aspects could benefit the simulation greatly.

References

[1] Josh Howarth. **How Many Gamers Are There?**, October 2022. 1

[2] Prith Banerjee, Chandrakant D. Patel, Cullen Bash, and Parthasarathy Ranganathan. **Sustainable Data Centers: Enabled by Supply and Demand Side Management**. In *Proceedings of the 46th Annual Design Automation Conference*, DAC '09, page 884–887, New York, NY, USA, 2009. Association for Computing Machinery. 1, 8, 61

[3] European Commission. **Energy-efficient Cloud Computing Technologies and Policies for an Eco-friendly Cloud Market**, November 2020. 1, 8

[4] Janine Morley, Kelly Widdicks, and Mike Hazas. **Digitalisation, energy and data demand: The impact of Internet traffic on overall and peak electricity consumption**. *Energy Research & Social Science*, **38**:128–137, 2018. 1

[5] Evan Mills, Norman Bourassa, Leo Rainer, Jimmy Mai, Arman Shehabi, and Nathaniel Mills. **Toward Greener Gaming: Estimating National Energy Use and Energy Efficiency Potential**, 2019. 1, 2, 7

[6] Evan Mills, Norman Bourassa, Leo Rainer, Jimmy Mai, Ian Vaino, Claire Curtin, Louis-Benoit Desroches, and Nathaniel Mills. **A Plug-Loads Game Changer: Computer Gaming Energy Efficiency without Performance Compromise**, 04 2019. 2

[7] Jay Harris. **What is a GPU? (and Why You Need One for Gaming!)**. `https://www.geekawhat.com/what-is-a-gpu-and-do-you-need-one-for-gaming/`. Accessed: August 23rd, 2023. 3

[8] Hongyu Zhu, Dongdong Zhang, Hui Hwang Goh, Shuyao Wang, Tanveer Ahmad, Daijiafan Mao, Tianhao Liu, Haisen Zhao, and Thomas Wu. **Future data center energy-conservation and emission-reduction technologies**

in the context of smart and low-carbon city construction. *Sustainable Cities and Society*, **89**:104322, 2023. 8, 53, 61

[9] Flexera. **State of the Cloud**. `https://info.flexera.com/CM-REPORT-State-of-the-Cloud-2023-Thanks`, 2023. 8

[10] Georgios Andreadis, Laurens Versluis, Fabian Mastenbroek, and Alexandru Iosup. **A reference architecture for datacenter scheduling: design, validation, and experiments**. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis, SC 2018, Dallas, TX, USA, November 11-16, 2018*, pages 37:1–37:15. IEEE / ACM, 2018. 8

[11] Anton Beloglazov, Jemal Abawajy, and Rajkumar Buyya. **Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing**. *Future Generation Computer Systems*, **28**(5):755–768, 2012. Special Section: Energy efficiency in large-scale distributed systems. 8, 61

[12] Katarzyna Mazur and Bogdan Ksiezopolski. **On Data Flow Management: the Multilevel Analysis of Data Center Total Cost**, 2017. 8

[13] Fabian Mastenbroek, Georgios Andreadis, Soufiane Jounaid, Wenchen Lai, Jacob Burley, Jaro Bosch, Erwin van Eyk, Laurens Versluis, Vincent van Beek, and Alexandru Iosup. **OpenDC 2.0: Convenient Modeling and Simulation of Emerging Technologies in Cloud Datacenters**. In *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 455–464, 2021. 9, 61

[14] Alexandru Iosup, Georgios Andreadis, Vincent Van Beek, Matthijs Bijman, Erwin Van Eyk, Mihai Neacsu, Leon Overweel, Sacheendra Talluri, Laurens Versluis, and Maaike Visser. **The OpenDC Vision: Towards Collaborative Datacenter Simulation and Exploration for Everybody**. In *2017 16th International Symposium on Parallel and Distributed Computing (ISPDC)*, pages 85–94, 2017. 9, 22, 61

[15] Microsoft Research. **Azure Public Dataset**. `https://github.com/Azure/AzurePublicDataset`. Accessed: August 16th, 2023. 13

[16] NVIDIA. **RTX Blade Server Cloud Gaming - Nvidia**. https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/cloud-gaming-server/geforce-now-rtx-server-gaming-datasheet.pdf. Accessed: July 2nd 2023. 14, 15, 16

[17] Saad Muzaffar. **How Many Cores for Gaming?** https://beanstalk.io/how-many-cores-for-gaming/, July 2023. Accessed: July 4th 2023. 15

[18] NVIDIA. **NVIDIA GeForce Now**. https://www.nvidia.com/en-us/geforce-now/. Accessed: July 3rd 2023. 15

[19] NVIDIA. **NVIDIA GeForce RTX 3060/3060 Ti**. https://www.nvidia.com/nl-nl/geforce/graphics-cards/30-series/rtx-3060-3060ti/. Accessed: July 3rd 2023. 15

[20] Intel Corporation. **Intel Core i9-11900K Processor**. https://ark.intel.com/content/www/us/en/ark/products/212325/intel-core-i911900k-processor-16m-cache-up-to-5-30-ghz.html. Accessed: July 22nd, 2023. 16

[21] Microsoft. **Xbox Cloud Gaming**. https://www.xbox.com/en-US/cloud-gaming. Accessed: July 2nd 2023. 16

[22] I. Chyzmar and M. Hoblyk. **Xbox Cloud Gaming: Custom Xbox Series X hardware upgrade**. https://www.theverge.com/2021/10/7/22714067/xbox-cloud-gaming-custom-xbox-series-x-hardware-upgrade, Oct 2021. Accessed: July 2nd 2023. 16

[23] Mike Wild. **Project xCloud Server Blade Analysis**. https://fragwire.com/project-xcloud-server-blade-analysis/, 2020. Accessed: July 2nd 2023. 16

[24] Microsoft. **Xbox Series X**. https://www.xbox.com/en-US/consoles/xbox-series-x. Accessed: July 2nd 2023. 16

[25] Jarred Walton. **AMD Big Navi and RDNA 2 GPUs: Everything We Know**. https://www.tomshardware.com/news/amd-big_navi-rdna2-all-we-know, 2022. Accessed: July 3rd 2023. 16

[26] Techpowerup. **AMD Xbox Series X GPU**. https://www.techpowerup.com/gpu-specs/xbox-series-x-gpu.c3482. Accessed: July 20th 2023. 16

[27] AMD. **AMD Ryzen 7 5700G Processor**. https://www.amd.com/en/products/apu/amd-ryzen-7-5700g. Accessed: July 20th, 2023. 16, 17

[28] Xbox Support. **About power options on Xbox One and Xbox Series X|S**. https://support.xbox.com/en-US/help/hardware-network/power/learn-about-power-modes. Accessed: July 20th, 2023. 16, 36, 37, 38, 44

[29] Sony. **PlayStation Plus**. https://www.playstation.com/en-us/ps-plus/. Accessed: July 3rd 2023. 17

[30] Sony. **PS5 Specs**. https://ps-5.nl/ps5-specs. Accessed: July 3rd 2023. 17

[31] Techpowerup. **PlayStation 5 GPU**. https://www.techpowerup.com/gpu-specs/playstation-5-gpu.c3480. Accessed: July 3rd 2023. 17

[32] PlayStation. **PlayStation - Active Power Consumption**. https://www.playstation.com/en-dk/legal/ecodesign/. Accessed: July 22nd, 2023. 17, 36, 37, 38, 44

[33] Abu Asaduzzaman and Hin Y. Lee. **GPU Computing to Improve Game Engine Performance**. *Journal of Engineering and Technological Sciences*, 2014. 17

[34] Cheol-Ho Hong, Ivor Spence, and Dimitrios S. Nikolopoulos. **GPU Virtualization and Scheduling Methods: A Comprehensive Survey**. *ACM Comput. Surv.*, **50**(3), jun 2017. 18

[35] NVIDIA Corporation. *NVIDIA vGPU User Guide*. NVIDIA Corporation, 2021. Accessed: July 16th, 2023. 27

[36] Minetrack. **Minetrack Data: Public domain Minecraft multiplayer server statistics**. 29, 43

[37] Minecraft. **Minecraft**. https://www.minecraft.net/en-us. Accessed: July 8th, 2023. 29

[38] Gitnux. **Cloud Gaming Services Statistics**. https://blog.gitnux.com/cloud-gaming-services-statistics/, 2023. Accessed: July 8th, 2023. 29

[39] Gamerbolt. **How Much Time Does the Average Gamer Spend Gaming?** https://www.gamerbolt.com/how-much-time-does-the-average-gamer-spend-gaming/, Feb 2023. Accessed: July 23rd, 2023. 36

[40] Till Fischer, Axel Böttcher, Aaron Coday, and Helena Liebelt. **Defining and Measuring Performance Characteristics of Current Video Games**. In Bruno Müller-Clostermann, Klaus Echtle, and Erwin P. Rathgeb, editors, *Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance*, pages 120–135, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 36

[41] Christopher Harper. **What Should Your GPU Utilization Be?** `https://www.cgdirector.com/what-should-your-gpu-utilization-be/`, Nov 2022. Accessed: July 9th, 2023. 36

[42] Hongyu He. **How Can Datacenters Join the Smart Grid to Address the Climate Crisis? Using simulation to explore power and cost effects of direct participation in the energy market**. *CoRR*, **abs/2108.01776**, 2021. 42, 61

[43] Global Petrol Prices. **Global Petrol Prices - Netherlands Electricity Prices**. `https://nl.globalpetrolprices.com/Netherlands/electricity_prices/`. Accessed: August 22nd, 2023. 44, 54

[44] Adam Smith. **Eco Energy Geek - Gaming PC Power Consumption**. `https://www.ecoenergygeek.com/gaming-pc-power-consumption`, May 2023. Accessed: July 24th, 2023. 44, 58

[45] zWORMz Gaming. **RTX 3070 | Spider-Man Remastered - Very High - 1080p, 1440p, 4K - RTX / DLSS**. `https://www.youtube.com/watch?v=cxGcpdXR32M&ab_channel=zWORMzGaming`, Oct 2022. 54

[46] Techpowerup. **NVIDIA GeForce RTX 3070**. `https://www.techpowerup.com/gpu-specs/geforce-rtx-3070.c3674`. Accessed: August 23rd 2023. 54

[47] Sandeepa Bhuyan, Shulin Zhao, Ziyu Ying, Mahmut T. Kandemir, and Chita R. Das. **End-to-End Characterization of Game Streaming Applications on Mobile Platforms**. *Proc. ACM Meas. Anal. Comput. Syst.*, **6**(1), feb 2022. 56, 61

[48] Branko Gapo. **What Is A Good FPS For Gaming?** `https://www.gpumag.com/good-fps-for-gaming/`, July 2023. Accessed: August 23rd, 2023. 56

[49] Liang Liu, Hao Wang, Xue Liu, Xing Jin, Wen Bo He, Qing Bo Wang, and Ying Chen. **GreenCloud: A New Architecture for Green Data Center**. In *Proceedings of the 6th International Conference Industry Session on Autonomic Computing and Communications Industry Session*, ICAC-INDST '09, page 29–38, New York, NY, USA, 2009. Association for Computing Machinery. 61

[50] Landon Marston Md Abu Bakar Siddik, Arman Shehabi. **The environmental footprint of data centers in the United States**. *Environmental Research Letters*, 5 2021. 61

[51] Seong-Ping Chuah, Chau Yuen, and Ngai-Man Cheung. **Cloud gaming: a green solution to massive multiplayer online games**. *IEEE Wireless Communications*, **21**(4):78–87, 2014. 61

[52] Florian Kadiyan, Hayko;Wündsch. **Green Gaming: How sustainable is cloud gaming?** *IEEE Wireless Communications*, 1 2021. 61

Appendix A

Reproducibility

A.1 Abstract

This appendix discusses the methods used to conduct the experiments presented in Chapter 5, and the necessary steps to reproduce them.

A.2 Artifact check-list (meta-information)

- **Program:** OpenDC - A discrete-event data center simulator.
- **Compilation:** Code compiled using Gradle, JDK version 17.0.6.
- **Run-time environment:** Java(TM) SE Runtime Environment (build 17.0.6+9-LTS-190)
- **Hardware:** Local machine
- **Metrics:** Energy usage in Joules, CPU utilization, GPU utiliztation
- **Output:** .txt files for the results and topologies, .csv for the trace and meta files
- **How much disk space required (approximately)?:** 1.5G for OpenDC repository
- **How much time is needed to prepare workflow (approximately)?:** Approximately 20 minutes
- **How much time is needed to complete experiments (approximately)?:** A single run could take a minute including changing the inputs. Reproducing results with multiple runs around 15 minutes.
- **Publicly available?:** Yes

A.3 Description

The implementations of GPU and of the experiment generator are in a fork of the OpenDC repository.

A.3.1 How to access

Clone the master branch from the opendc-RS-2023 repository at https://github.com/romSavid/opendc-RS-2023

A.3.2 Software dependencies

The project was developed using the IntelliJ IDEA. We recommend using this IDE for running experiments, but it is not necessary.

A.4 Installation

To set up the development environment, follow these steps:

1. Install IntelliJ IDEA from the official website.
2. Clone the master branch of the project from https://github.com/romSavid/opendc-RS-2023
3. Open the cloned project in IntelliJ IDEA.
4. IntelliJ IDEA should automatically recognize the Gradle build file and attempt to download the necessary dependencies. If not, you can manually trigger a Gradle sync from the `View > Tool Windows > Gradle` menu.
5. After the dependencies have been downloaded and the project is successfully built, you can run the experiments from within IntelliJ IDEA. The relevant files are placed in `opendc-experiments-cloudGaming`. Go to `CloudGamingIntegrationTest.kt` and run the test by clicking on the small 'play' button to the left of the main class. The test indicates that everything works fine.

A.5 Experiment workflow

Go to `CloudGamingExperiments.kt` in the same folder. There you can see the `testBasicRun()` function. It can be run the same way the integration test was run. According to the chosen inputs, the test will run and create the following files in the respective folders in `opendc-experiments-cloudGaming/src/test/resources`:

1. `trace.csv` and `meta.csv` in the `trace/x-trace` folder, with x being the chosen platform.

2. `x-topology.txt` in the `env` folder, with x being the chosen platform.

3. `x_date.txt` in the `results` folder, with x being the chosen platform and date the current date and time.

In the result file, the most important parameter is `energyUsage` which is the overall energy usage in Joules.

A.6 Evaluation and expected results

For most of the OpenDC experiments, it is enough to use the experiment generator like it is used in `CloudGamingExperiments.kt`. The setup for every experiment is documented in Chapter 5. To reproduce the experiments follow these steps:

1. Choose the right `platform`, for example `xcloud`, `psplus`, or `geforcenow`.

2. Verify that there is a folder for the chosen platform in the `trace` folder.

3. For most experiments we use the `usersPerHour` value that is already provided, but it can be changed if needed.

4. In `ExperimentGenerator.generateExperiment()` choose the proper `cpuUtilization` and `gpuUtilization` values, between 0 and 1, the number of hours for the trace, and the users list.

5. Run the experiment

6. according to the specific experiment, take the `energyUsage` from the result file and treat it accordingly. For example, for Section 5.4, you can use the provided `usersPerHour`, choose the `xcloud` preset, and run it 9 times with the varying CPU and GPU utilization, 0.15, 0.3, and 0.7 for CPU and 0.15, 0.6, and 1.0 for the GPU. Now take all of the results files, add up the `energyUsage` from each one (convert them to kWh. This can be easily done here `https://www.rapidtables.com/convert/energy/Joule_to_kWh.html`), and divide by the number of runs (9). For `xcloud` you should get 6,580.09 kWh, and when multiplying by the price of kWh (0.508) you should get 3,342.68572 Euro. This can be done for all of the other presets as well. Other experiments might require more or fewer steps, but all of the OpenDC experiments revolve around choosing the basic parameters, running an experiment, and making calculations using the `energyConsumption` from the result.

7. The results are also printed back to the user in the terminal if you want to see a result for a specific run.

A.7 Experiment customization

Using the experiment generator it is easy to experiment with different topologies. The topology and trace generator can also be used separately from the experiment generator for more flexibility. And of course, if you have real-world traces in the right format they could also be loaded and used.

A.8 Notes

Sometimes the first run of a new preset returns different values than the expected ones, and then the rest of the runs return the proper results. I think this is due to the multithreading nature of the simulator and that sometimes the new run will use an old topology. However, this does not happen often.