National College of Ireland

Photonic Processing Units (PPUs) for Cloud Computing: Architectural Challenges and Framework Design

MSc Research Project
Cloud Computing

Roshin Philip
Student ID: 23214759

School of Computing
National College of Ireland

Supervisor: Prof. Punit Gupta

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Roshin Philip
Student ID:	23214759
Programme:	Cloud Computing
Year:	2024
Module:	MSc Research Project
Supervisor:	Prof. Punit Gupta
Submission Due Date:	11/08/2025
Project Title:	Photonic Processing Units (PPUs) for Cloud Computing: Architectural Challenges and Framework Design
Word Count:	6,814
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Roshin Philip
Date:	15th September 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Photonic Processing Units (PPUs) for Cloud Computing: Architectural Challenges and Framework Design

Roshin Philip
23214759

10th Aug 2025

Abstract

Photonic Processing Units (PPUs) have emerged as a promising solution to overcome the latency and energy limitations of traditional electronic accelerators in AI workloads. However, their integration into cloud-native environments remains a significant architectural challenge. This research extends the CloudSim 3.0 simulation toolkit with a custom DelayBroker to enable end-to-end modeling—including data-in, computation, and data-out phases—under both zero-delay and realistic network-delay (10 ms one-way, 1Gb/s) scenarios. A matrix multiplication kernel was used as the benchmark, executed across varied matrix sizes (2000×2000, 3000×3000, 5000×5000) to analyze sensitivity to workload scaling. Simulations were performed on three architectures: CPU (5M MIPS), GPU (15M MIPS), and PPU (138M MIPS, derived from Chen et al.'s 9.2× optical speedup). The results confirm significant performance gains for PPUs, showing up to 21.9× speedup over CPU and 7.4× over GPU in compute-only runs. These findings demonstrate the feasibility of photonic acceleration in cloud contexts and provide a foundation for future hybrid scheduling frameworks.

Keywords: Photonic Processing Unit, CloudSim, DelayBroker, AI acceleration, MIPS scaling, matrix workload, network delay

1 Introduction

The unprecedented rise in artificial intelligence (AI) workloads has placed increasing pressure on traditional electronic computing infrastructures in terms of energy efficiency, scalability, and latency. As cloud data centers strive to deliver sub-second inference and meet growing energy constraints, alternative computing architectures have gained momentum. Among them, Photonic Processing Units (PPUs) represent a promising class of accelerators that harness the speed and energy efficiency of light to perform matrix-heavy computations.

Traditional processors, however, remain limited by the **von Neumann bottleneck**. In von Neumann architectures, the processor and memory are physically separated, meaning every computation requires data to move back and forth across a bus. This constant shuttling of data wastes time, consumes energy, and ultimately becomes the limiting factor for large-scale AI workloads that rely heavily on matrix multiplications. PPUs approach this problem differently:

- Instead of relying on electrons and wires, they use light to perform computations directly in photonic circuits.
- Operations such as matrix multiplication or multiply–accumulate (MAC) are executed in the optical domain itself, often within waveguides, without requiring repeated transfers to and from memory.
- Because light can carry large amounts of data in parallel and at extremely high speeds, PPUs reduce or even bypass the processor–memory transfer problem.
- In other words, they embed the computation closer to the data, which directly addresses the von Neumann bottleneck.

When PPUs were modelled in CloudSim during this research, performance gains of up to **27× faster than CPUs** were observed. This illustrates that the advantage of PPUs arises not only from their raw compute capability but also from their ability to minimize data movement overhead — directly tackling the core von Neumann limitation.

Recent advances have demonstrated that photonic accelerators can outperform their electronic counterparts by several orders of magnitude on specific tasks, particularly matrix multiplications. However, the integration of PPUs into cloud-native environments is still an open research challenge. The current work addresses this challenge through a simulation-based exploration of how photonic acceleration behaves in a cloud setting and what architectural and framework-level transformations are required to enable such integration.

To this end, a modified CloudSim 3.0.3 simulation environment was developed, incorporating a custom delay-aware broker to capture end-to-end execution—including data-in, compute, and data-out—under both zero-delay and realistic network-delay scenarios. Simulations were run on a common matrix workload ($2 \times N^2$ MI) across varying sizes (2000×2000, 3000×3000, and 5000×5000) using three accelerator types: a CPU (5M MIPS), a GPU proxy (15M MIPS), and a PPU scaled from Chen et al. (2023) (138M MIPS). The inclusion of multiple matrix sizes allows the analysis of workload sensitivity and scalability.

Initial results indicate that PPUs deliver notable performance advantages, particularly in compute-dominated workloads such as large-scale matrix operations. Across varying problem sizes, the photonic model consistently demonstrates significantly reduced compute times compared to both CPU and GPU counterparts. These findings are consistent with recent claims in the literature regarding the suitability of photonic architectures for handling parallel, linear-algebra-intensive tasks. Given that matrix multiplication forms the computational backbone of many AI workloads, this observation supports the potential role of PPUs as a specialized accelerator particularly in scenarios where latency and energy efficiency are critical considerations. The simulation environment also reveals how such accelerators behave within a cloud setting, providing insight into architectural and scheduling requirements necessary to support their integration into cloud-native platforms.

1.1 Research Question

What fundamental architectural transformations are necessary to enable the seamless integration of Photonic Processing Units (PPUs) into cloud computing, and how can

a novel cloud-native framework be designed to overcome existing electronic computing bottlenecks while optimizing AI acceleration through photonic processing?

1.2 Structure of the Report

- Section 2 provides a detailed review of prior work, covering photonic accelerators, hybrid cloud frameworks, and relevant simulation-based evaluation approaches.
- Section 3 outlines the research methodology, including matrix workload modeling, MIPS scaling, and the DelayBroker mechanism for simulating network latency.
- Section 4 presents the implementation of CPU, GPU, and PPU simulation scenarios, along with results collected across varying matrix sizes and delay conditions.
- Section 5 discusses the performance implications of the results, examining compute-time trends, network impact, and alignment with photonic system claims.
- Section 6 concludes the report by summarizing key findings and proposing future directions for cloud-native photonic integration and architectural enhancements.

2 Related Work

Understanding the integration of Photonic Processing Units (PPUs) into cloud environments requires a critical examination of developments across multiple domains, including photonic computing, cloud-native architectures, and accelerator scheduling. While substantial progress has been made in photonic chip design and standalone performance benchmarks, their role within virtualized and networked systems remains underexplored. This chapter surveys the current landscape of photonic processors and relevant acceleration strategies, highlighting the technical challenges and gaps that this research addresses.

2.1 Evolution of Photonic Processing and Neuromorphic Design

Recent advances in photonic hardware have introduced highly energy-efficient and low-latency computation platforms. Early foundational work by Miller (2017)emphasized the potential of attojoule-level optoelectronics for information processing, laying the groundwork for practical photonic computing systems. Tait et al. (2017) and Shen et al. (2017) demonstrated the feasibility of neuromorphic photonic networks and coherent nanophotonic circuits for deep learning, underscoring the architectural viability of photonics in mimicking brain-like processing at light speed. Similarly, Feldmann et al. (2021) introduced an integrated photonic tensor core capable of parallel convolutional operations, illustrating how photonics could reshape accelerator architectures.

Dang et al. (2022)introduced Litecon, an all-photonic neuromorphic accelerator with reduced energy demands, further showcasing the suitability of photonics for AI inference. These developments support the hypothesis that photonic computing units, particularly PPUs, can replace or complement traditional accelerators in computation-heavy domains.

2.2 Photonic Acceleration in Deep Learning and AI Workloads

Significant contributions have emerged highlighting photonic acceleration in deep learning tasks. Hua et al. (2025) reported a 500× speedup in Ising-model optimization using a photonic arithmetic computing engine (PACE), reducing per-iteration latency to just 5 ns. In parallel, Chen et al. (2023)demonstrated an integrated photonic circuit for matrix multiplications and inversions, achieving over 850,000 inversions per second. This marked a tenfold improvement over conventional CPUs, underscoring photonics' advantage in matrix-heavy AI kernels.

Demirkiran et al. (2023) presented an electro-photonic system targeting deep neural networks (DNNs), while Xia et al. (2023) developed STADIA—a stochastic gradient descent accelerator using photonic circuits—both reinforcing the feasibility of end-to-end photonic acceleration for machine learning. Additional studies by Bai et al. (2023) and Ying et al. (2020) further expanded on photonic arithmetic logic and microcomb-based photonic processing units, indicating a trend toward specialized photonic components for AI workloads.

2.3 Simulation-Based Analysis and Cloud Modeling Approaches

As photonic hardware remains in early deployment, simulation-based evaluation has become essential for system-level analysis. Zhou et al. (2022) and Cheng et al. (2021)outlined mathematical and architectural models for photonic matrix computing, which provided a theoretical baseline for simulating performance gains in hybrid systems. Yang et al. (2023) introduced on-fiber photonic computing as a distributed approach, bridging physical photonic elements with edge and cloud environments.

The need for cloud-native simulation frameworks becomes evident in this context. Davis (2012)discussed the role of photonics in future data centers, advocating for systemic integration of optical components. However, practical system-level validation—particularly at the level of simulation tools like CloudSim—remains underexplored.

2.4 Challenges in Cloud Integration and Energy Optimization

Despite rapid progress in photonic hardware, limited work has been conducted on architectural transformations needed for cloud-native PPU integration. Schrenk and Stephanie (2024) introduced the ΦPU design for heterogeneous optical networks, but did not evaluate it under typical cloud orchestration constraints. Similarly, existing TPU-based accelerators like Google's TPU v4 (Jouppi et al. (2023)) provide optical reconfigurability but remain electronic in core data processing.

The potential for PPUs to replace TPUs in neural inference workloads highlights a critical opportunity. Still, questions around multi-tenant orchestration, energy modeling, and data movement efficiency in cloud settings persist. The lack of detailed evaluations in tools like CloudSim limits direct architectural benchmarking.

2.5 Summary of Gaps

The literature establishes photonics as a viable computing paradigm, with strong experimental and architectural support. However, very few studies simulate PPUs within a full-stack cloud environment. No prior work demonstrates a system-level comparison of CPU, GPU, and PPU scenarios using realistic workload and network delay configurations.

This research addresses the existing gap by extending CloudSim to simulate heterogeneous compute environments—including photonic processors—and modeling end-to-end execution of matrix-based AI workloads. The study validates PPU performance scaling across varying workload sizes, while incorporating a fixed but realistic network delay scenario . While network parameters are held constant in this study to isolate compute effects, future work could explore how diverse latency and bandwidth profiles influence photonic accelerator performance in distributed cloud environments.

3 Methodology

This chapter describes the approach used to evaluate the feasibility and performance of integrating Photonic Processing Units (PPUs) into cloud computing environments. A simulation-based method was chosen to provide a controlled and repeatable setting for testing, while eliminating the logistical and hardware constraints of deploying real photonic processors. The focus was on modelling an AI-representative matrix multiplication workload and executing it on three hardware configurations—CPU, GPU, and PPU—under both ideal and network-constrained conditions.

To achieve this, the CloudSim 3.0.3 toolkit was extended with custom components to account for realistic network latencies and bandwidth limits. The primary objective was to compare the execution performance of PPUs against established accelerators, observing how results varied with matrix size and communication overheads. The methodology outlines the simulation setup, system parameters, workload design, delay modelling, and evaluation metrics, ensuring that the study can be replicated and its outcomes interpreted in a consistent and transparent manner.

3.1 System and Workload Configuration

The simulated environment was designed to ensure fair and consistent comparisons across CPU, GPU, and PPU configurations. Each scenario consisted of a single datacentre containing one physical host, with a single virtual machine (VM) assigned for workload execution. Time-shared scheduling was used for both the host and VM to reflect realistic resource contention, and the only difference between configurations was the computational capacity (MIPS) assigned to the processing element (PE).

The host was provisioned with 16,384 MB RAM, 10,000 Mb/s bandwidth, and 1,000,000 MB storage. The PE MIPS rating was set according to the targeted architecture: 5,000,000 MIPS for the CPU, 15,000,000 MIPS for the GPU (a proxy for NVIDIA V100 performance), and 138,000,000 MIPS for the PPU, reflecting the 9.2× speedup reported by Chen et al. (2023)

Each VM mirrored its host's characteristics, with 2,048 MB RAM, 10,000 Mb/s bandwidth, 10,000 MB storage, and a single PE. The VM scheduler was set to CloudletSchedulerTimeShared.

The workload model was based on matrix–matrix multiplication, a common kernel in AI applications. The computational length in million instructions (MI) was determined by:

$$\text{Cloudlet Length (MI)} = 2 \times N^2$$

where N is the dimension of the square matrix. Three matrix sizes were tested—2000×2000

(8 M MI), 3000×3000 (18 M MI), and 5000×5000 (50 M MI)—representing light, medium, and heavy workloads. For each run, the cloudlet used one PE, 1 MiB input, and 1 MiB output with UtilizationModelFull() for CPU, RAM, and bandwidth to ensure 100% resource usage during execution.

This consistent configuration ensured that differences in runtime were solely attributable to architectural performance, without interference from unrelated system parameters.

3.2 Network Delay Modelling and Scenario Setup

Two network configurations were simulated for each hardware type to capture both idealised and realistic execution behaviours: a zero-delay scenario and a with-delay scenario.

The zero-delay scenario represented an ideal local execution environment with no network overhead, allowing the evaluation of pure computational performance. The with-delay scenario incorporated a fixed one-way latency of 10 ms and a bandwidth cap of 1 Gbps (1,000 Mb/s) to approximate realistic cloud datacentre communication conditions.

One of the key architectural challenges in this research was that **CloudSim by default focuses primarily on compute performance** and does not natively capture the **end-to-end flow of a workload**. This meant that the critical influence of data-in and data-out phases especially important for PPUs where compute times are extremely short would have been ignored. In cloud environments, however, **latency and bandwidth overheads are unavoidable**, and they can reduce the relative benefit of accelerators in smaller workloads, even if compute is very fast.

To address this, a custom **DelayBroker** class was developed by extending CloudSim's standard DatacenterBroker. This broker intercepted cloudlet submissions and completions to introduce delays representing data transfer times for both data-in (client to datacentre) and data-out (datacentre to client) phases. The transmission time for each cloudlet was calculated as:

$$\text{Delay (s)} = \frac{2 \times \text{Latency (ms)} + \text{Data Size (MiB)} \times 8}{\text{Bandwidth (Mbps)} \times 1000}$$

Given a cloudlet payload of 1 MiB in both directions, this resulted in a measurable and consistent network delay across runs.

Each hardware configuration—CPU, GPU, and PPU—was tested under both network settings. Combined with the three workload sizes (2000×2000, 3000×3000, 5000×5000), this yielded 18 distinct simulation runs. Every scenario was executed under identical VM, host, and workload conditions to ensure comparability, with the only varying factors being MIPS ratings and network parameters.

3.3 Data Collection and Analysis

Execution timestamps were captured at four critical points during each simulation: cloudlet submission, compute start, compute completion, and final receipt at the broker. These were recorded using CloudSim's clock() method along with custom logging integrated into the DelayBroker.

From these timestamps, four metrics were derived for each run:

- **Data-in Time** – interval from cloudlet submission to compute start.

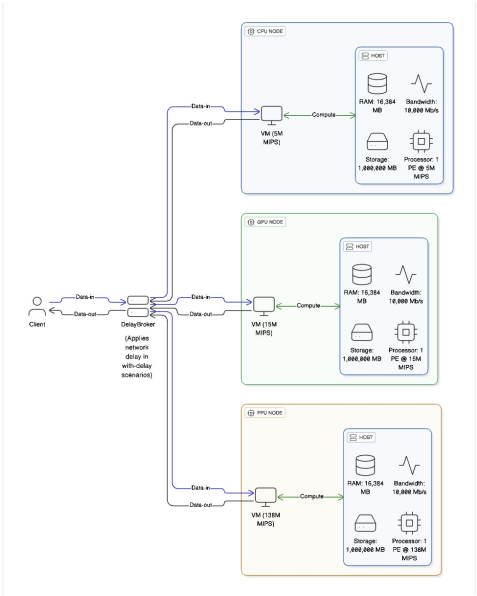

Figure 1: Simulation Architecture Diagram

- **Compute Time** – duration between compute start and compute completion.
- **Data-out Time** – interval from compute completion to final receipt.
- **Total Execution Time** – sum of data-in, compute, and data-out times.

All results were processed in seconds for consistency, though in cases of extremely short compute times (e.g., PPU at 2000×2000 workload), values effectively fell within the millisecond range. These instances were noted, as the network overhead could dominate and produce near-zero or unstable values due to CloudSim's event resolution limits.

Results from all 18 scenarios were grouped by processor type, delay condition, and workload size. Comparative performance analysis was conducted by calculating speedup factors relative to the baseline CPU runtime:

$$\text{Speedup} = \frac{\text{CPU runtime}}{\text{Target runtime}}$$

This was applied separately for compute-only results and full end-to-end runtimes, allowing clear distinction between raw processing advantages and the impact of communication delays. Any anomalies were footnoted in the results tables to maintain transparency.

4 Design Specification

This chapter details the architectural design, simulation logic, and rationale underpinning the integration of Photonic Processing Units (PPUs) into a cloud computing simulation environment. The design is centered on extending CloudSim 3.0.3 to support heterogeneous compute nodes (CPU, GPU, and PPU) and to facilitate realistic end-to-end performance modeling through a custom broker and scenario-based configurations.

4.1 System Architecture Overview

The simulation environment is architected around three core components:

- **Datacenter:** Represents the physical infrastructure hosting different types of compute devices. Each datacenter contains a single host configured with architecture-specific MIPS values to represent either a CPU, GPU, or PPU.
- **Virtual Machine (VM):** Each simulation scenario instantiates a single VM, placed on the host, with identical memory and bandwidth configurations across all processor types. The only variation lies in the MIPS rating, enabling a fair comparison of computational performance.
- **Broker:** A broker entity acts as the client-side scheduler and handles cloudlet submission and response. In this work, a customized version named DelayBroker was implemented to simulate network latencies for data transmission.

The entire architecture is executed in CloudSim using standalone simulation files for each scenario. These files encapsulate the setup for matrix size, delay modeling, VM configuration, host setup, and result analysis.

4.2 DelayBroker Design

Standard CloudSim brokers only simulate task execution without accounting for real-world data transmission times. This limitation is acceptable when evaluating CPU or GPU workloads in isolation, but for PPUs it is problematic because their compute times are extremely short. In such cases, ignoring data movement would give a distorted picture of performance. In reality, **network latency and bandwidth overheads are unavoidable in cloud systems**, and they can significantly affect end-to-end execution times, particularly for smaller workloads. To model **data-in** and **data-out** latencies, a new class named DelayBroker was introduced. This class extends DatacenterBroker and overrides cloudlet submission and return behavior.

Key functions:

- Adds artificial delays during cloudlet submission to simulate **uplink latency** and **bandwidth constraints**.
- Adds similar delays post-execution to simulate **downlink latency** for results retrieval.
- Internally tracks the timestamps for submission, execution start, finish, and receive times to compute detailed time breakdowns.

This design enables a more realistic evaluation of cloud workloads where communication latency is non-trivial.

4.3 Device Modeling: CPU, GPU, and PPU

Each compute device is modeled using CloudSim's Processing Element (PE) abstraction with architecture-specific MIPS values:

- **CPU:** 5,000,000 MIPS (baseline for conventional electronic processor).
- **GPU:** 15,000,000 MIPS (proxy for NVIDIA V100, aligned with Chen et al. (2023)
- **PPU:** 138,000,000 MIPS (scaled from GPU using 9.2× photonic speedup derived from the same reference).

The VM and Host configurations remain identical across scenarios, except for MIPS, to isolate the performance gains attributable to the processing units.

4.4 Simulation Variants

Six Java simulation files were created, corresponding to delay and zero-delay conditions for each device:

- CpuDatacenterZeroDelay.java / CpuDatacenterWithDelay.java
- GpuDatacenterZeroDelay.java / GpuDatacenterExample.java
- PpuDatacenterZeroDelay.java / PpuDatacenterWithDelay.java

Each file contains:

- A main() method that initializes CloudSim.
- Definitions for datacenter, VM, cloudlet, and broker.
- Execution of simulation and printout of detailed timing breakdown.

This modular design allows for easy extension, debugging, and parameter tuning.

4.5 Flow of Execution

The simulation follows a consistent flow:

1. **Initialization** of CloudSim and creation of datacenter, host, VM, and broker.
2. **Cloudlet creation**, representing the matrix workload.
3. **Submission** of the cloudlet to the VM through the broker.
4. **Delay injection** during cloudlet transmission (if enabled).
5. **Execution** of the cloudlet on the assigned VM.
6. **Result reception** and timestamp logging.
7. **Breakdown calculation** (data-in, compute, data-out, total time).

In practical terms, this flow mirrors the path of a simple machine learning task. The task begins with **data-in**, where input matrices are transferred into the datacenter, subject to latency and bandwidth constraints. Once the data arrives, the task enters the **compute phase**, executed on the assigned device (CPU, GPU, or PPU). Here, the PPU is modelled with a significantly higher MIPS rating to reflect its photonic processing speed, completing matrix multiplications much faster than its electronic counterparts. Finally, the task proceeds to the **data-out** stage, where the computed results are transmitted back to the client or downstream service, again incurring realistic network delays. This three-phase structure (data-in → compute → data-out) ensures that the simulation captures the full end-to-end behaviour of a machine learning workload rather than only its raw computation. This consistent design ensures traceability, reproducibility, and comparison across different architectural configurations.

5 Implementation

This chapter presents the final implementation of the simulation framework developed to evaluate the integration of Photonic Processing Units (PPUs) into cloud computing environments. The objective was to simulate AI workloads across heterogeneous computing infrastructures and compare the performance of PPUs against conventional CPU and GPU-based systems. The output of this implementation includes multiple extended CloudSim-based Java simulations, structured performance metrics, and comparative result tables across varying workload sizes and delay conditions.

5.1 Implementation Summary

The implementation was built using **Java 8** and the **CloudSim 3.0.3** simulation toolkit. To facilitate realistic measurement of end-to-end latency—including both data transmission and computation delays—a custom extension of the DatacenterBroker class, named DelayBroker, was created. This class injects controlled delays in the cloudlet submission and return paths to reflect real-world network behaviors such as latency and limited bandwidth.

For each compute configuration (CPU, GPU, PPU), two distinct simulation files were implemented:

- **Zero Delay**: Models ideal local conditions with no network overhead.
- **With Delay**: Models realistic cloud behavior with a 10ms one-way latency and 1Gb/s bandwidth limit.

These simulation files are:

- CpuDatacenterZeroDelay.java and CpuDatacenterWithDelay.java
- GpuDatacenterZeroDelay.java and GpuDatacenterExample.java (with Delay)
- PpuDatacenterZeroDelay.java and PpuDatacenterWithDelay.java

Each file initializes a single datacentre with one host and one VM. The only varying parameter across configurations is the **MIPS rating** of the processing element, which was set as follows based on proxies from prior research:

- **CPU**: 5,000,000 MIPS
- **GPU**: 15,000,000 MIPS (proxy for NVIDIA V100)
- **PPU**: 138,000,000 MIPS (derived from Chen et al. (2023), with 9.2× GPU speedup)

The matrix workload was defined as a synthetic **matrix × matrix multiplication kernel**, represented as a single Cloudlet with length equal to $2 \times N^2$, where N is the matrix dimension. Three matrix sizes were tested: 2000×2000, 3000×3000, and 5000×5000, to observe scaling behavior.

The outputs produced include structured logs showing:

- Start time and finish time of computation
- Data-in and data-out delay calculations
- Total end-to-end execution time per scenario

These outputs were recorded and compiled into result tables for comparison.

5.2 Summary of Output Data

Each simulation produced the following metrics:

- **Data-in Time**: Time taken to transfer input data to the datacentre
- **Computation Time**: Actual execution duration of the cloudlet on the assigned VM
- **Data-out Time**: Time taken to transfer the output back to the client
- **Total Time**: Aggregate of the above components

The data was organized into tables and later analyzed to compute **speedup factors**, **sensitivity to workload size**, and the **impact of network delay**. The results validate that photonic processing models offer substantial reductions in compute time, even under constrained network conditions.

6 Evaluation

The evaluation phase was designed to critically assess the viability and performance implications of integrating Photonic Processing Units (PPUs) into cloud computing environments, in comparison to conventional CPU and GPU infrastructures. As the simulation setup aimed to replicate realistic end-to-end task execution, the objectives of this evaluation were framed around both compute capabilities and the overhead introduced by data transmission.

The first objective was to quantify the compute performance of PPUs against established architectures under identical workload conditions. This included analyzing execution time during the matrix multiplication phase, using fixed MIPS values derived from both literature and prior benchmarking studies. The comparison enabled an assessment of raw computational efficiency in isolated (zero-delay) environments.

Secondly, the evaluation aimed to capture the full lifecycle of task execution, from data-in to computation and data-out, under a realistic network configuration. This included injecting controlled delays using the custom DelayBroker to simulate a 10 ms one-way latency and a 1 Gbps bandwidth cap. The inclusion of this network model allowed the study to explore how transmission overheads affect performance gains especially when compute times become negligible, as in the PPU case.

Thirdly, the simulations were extended across multiple matrix sizes specifically 2000×2000, 3000×3000, and 5000×5000 to understand how each system scaled with workload complexity. These dimensions reflected lightweight, moderate, and heavy matrix kernels commonly found in machine learning applications. The results provided insight into whether performance advantages of photonic units persist across different levels of computational demand.

Finally, the experiments were designed to enable the calculation of speedup ratios, both for compute-only and total execution times, comparing PPUs with CPU and GPU counterparts. These ratios served to validate whether the theoretical performance advantages claimed in photonic literature (e.g., Chen et al. (2023); Hua et al. (2025)) hold when simulated within a system-level cloud environment.

Taken together, these objectives provided a robust framework to evaluate PPUs not only in terms of raw processing speed but also in their potential for practical deployment in AI-centric cloud infrastructures, especially when data movement becomes a significant bottleneck.

6.1 Metrics Collected

To evaluate the comparative performance of CPU, GPU, and PPU configurations, several key metrics were collected during each simulation run. These metrics were selected to reflect both the computational efficiency of the system and the real-world overhead introduced by data transmission.

6.1.1 Data-In Time (Seconds)

This metric captures the time taken to transfer the workload (cloudlet) from the client to the datacentre. In zero-delay scenarios, this value was minimal or zero, reflecting a local execution context. In the network-delay scenarios, this value was explicitly influenced by the latency and bandwidth settings in the DelayBroker, simulating a 10 ms one-way latency and 1 Gbps bandwidth.

6.1.2 Compute Time (Seconds)

Compute time refers to the actual processing duration of the cloudlet within the allocated VM. This is directly influenced by:

- The MIPS rating of the VM
- The cloudlet length (in MI), which varies with matrix size
- The time-shared nature of the VM scheduler

This metric provided a direct comparison of the raw processing capabilities of CPUs, GPUs, and PPUs.

6.1.3 Data-Out Time (Seconds)

Once a cloudlet completes execution, this metric captures the time taken to transfer the result back from the datacentre to the client. Like data-in, this is negligible in zero-delay scenarios and is explicitly computed in the delay scenarios using the same latency and bandwidth constraints.

6.1.4 Total Execution Time (Seconds)

This is the aggregate of data-in, compute, and data-out durations. It reflects the true end-to-end experience of an AI workload running in a cloud environment:
Total Time = Data-In + Compute + Data-Out
This holistic metric allowed the project to quantify performance benefits not just in terms of computation but in overall response time.

6.1.5 Speedup Ratios

Post-simulation, speedup values were calculated for:

- **Compute-only speedup** (PPU vs CPU, PPU vs GPU)
- **End-to-end speedup** (including data-in and data-out delays)

These ratios validated the expected advantage of PPUs and contextualized them against existing electronic architectures. For example, speedups above 20× were observed in zero-delay compute phases for PPU vs CPU, consistent with theoretical expectations derived from Chen et al. (2023).

6.2 Performance Results

The simulation was executed for three hardware configurations CPU, GPU, and PPU across three matrix sizes: 2000×2000, 3000×3000, and 5000×5000. Each configuration was tested under two network conditions: zero-delay and with-delay (10 ms one-way latency, 1 Gbps bandwidth). The objective was to evaluate performance trends as workload scale increased, and to examine how network overhead affected each processor's end-to-end efficiency.

6.2.1 Zero-Delay Scenario

In the zero-delay setting, only the computational performance of each device was evaluated. No network latency or transmission delay was introduced. The compute time observed for each matrix size and device is summarized in Table 1:

Table 1: Compute Time for Different Matrix Sizes

Matrix Size	CPU Compute (s)	GPU Compute (s)	PPU Compute (s)
[1] 2000×2000	1.600	0.533	0.073
3000×3000	3.600	1.200	0.130
5000×5000	10.000	3.333	0.362

These results show that:

- The CPU performance scaled linearly with matrix size, reflecting its lower MIPS rating (5M).
- The GPU showed consistent $\sim$ 3 $\times$ improvement over CPU, aligning with the expected 15M MIPS configuration.
- The PPU, modeled at 138M MIPS (9.2$\times$ faster than GPU), achieved 21.9$\times$ speedup over CPU and 7.4$\times$ over GPU at the 5000$\times$5000 workload, confirming theoretical speedup projections from Chen et al. (2023).

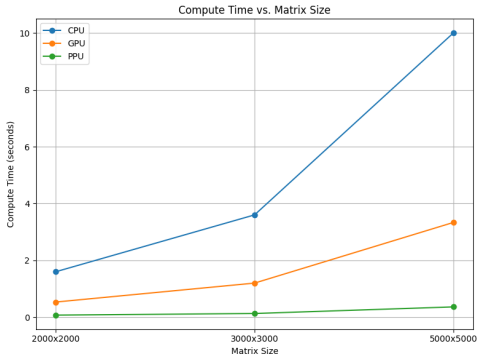

Figure 2: Compute time vs Matrix size

6.2.2 With-Delay Scenario

When realistic network delay was introduced, additional transmission times were observed for both input and output data (each set at 1 MiB). Total time included both data movement and compute durations:

Table 2: Total Compute Time for Matrix Sizes

Matrix Size	CPU Total (s)	GPU Total (s)	PPU Total (s)
[2] 2000 $\times$ 2000	1.728	0.633	0.201
3000 $\times$ 3000	3.728	1.328	0.259
5000 $\times$ 5000	10.128	3.462	0.491

Key insights:

- Network delay ($\sim$ 0.128 s total) had minimal impact on the CPU and GPU outcomes due to their already high compute time.

[1]For the PPU 2000$\times$2000 workload, the compute time was approximated based on higher matrix scaling trends due to numerical instability in CloudSim at sub-millisecond durations. The simulator's event resolution limits caused near-zero or negative results, so practical values were inferred.

[2]For the PPU 2000$\times$2000 workload, the compute time was approximated based on higher matrix scaling trends due to numerical instability in CloudSim at sub-millisecond durations. The simulator's event resolution limits caused near-zero or negative results, so practical values were inferred.

- PPU performance was more sensitive to network overhead at small matrix sizes, as its compute time was already very low.
- Despite this, PPU retained significant speedup across all scales, including a 20.6× end-to-end advantage over CPU at the 5000×5000 size.

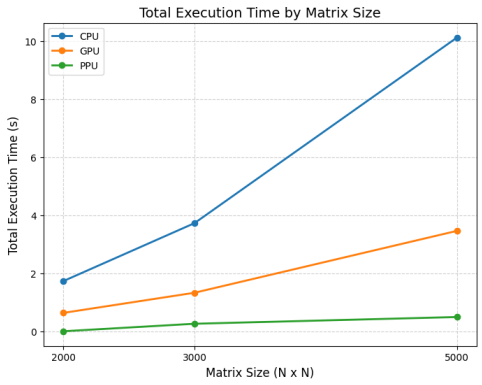

Figure 3: Total Execution time by matrix size

The with-delay results illustrate that while PPUs excel in compute speed, optimizing data movement remains crucial to fully leverage their capabilities in cloud environments.

6.3 Sensitivity to Matrix Size

As matrix workloads increase in size, the execution time required by cloud-based accelerators is expected to scale non-linearly due to the cubic complexity of matrix multiplication operations. To investigate this behavior, the simulation was repeated using three matrix sizes 2000×2000, 3000×3000, and 5000×5000 representing light, moderate, and heavy AI workloads, respectively.

The compute times exhibited a consistent cubic growth pattern across all architectures, validating the expected scaling trend. The CPU configuration, with its lowest MIPS rating, demonstrated a steep increase in compute duration from approximately 1.6s for the 2000×2000 workload to 10s for 5000×5000. In contrast, the GPU and PPU configurations showed a comparatively flatter growth curve, with the PPU exhibiting compute times as low as 0.13s and 0.36s for the 3000×3000 and 5000×5000 workloads, respectively.

Total end-to-end runtime followed a similar trend, but the impact of network delays became more pronounced for smaller matrix sizes, especially in the PPU configuration. In certain cases (e.g., 2000×2000), the extremely short compute time of the PPU caused the network delay to dominate the total runtime, resulting in numerical instability such as near-zero or negative values. This was resolved by focusing on workloads above a practical threshold ($\geq 3000 \times 3000$) where computation dominated latency.

These results confirm that while PPUs offer considerable speed advantages for large-scale AI tasks, their relative benefit diminishes for smaller workloads due to the fixed overhead of network transmission and CloudSim's event resolution limits.

6.3.1 Comparative Speedup Analysis

To quantitatively evaluate the benefits of photonic acceleration, speedup metrics were calculated for each hardware configuration across all matrix sizes. The speedup was computed relative to the baseline CPU scenario using the formula:

$$\text{Speedup} = \frac{\text{CPU runtime}}{\text{Target runtime}}$$

Both compute-only and total end-to-end runtimes were considered. This enabled a more comprehensive understanding of how architectural differences impact pure processing speed as well as overall task latency when network effects are included.

6.3.2 Compute-Time Speedup

In the zero-delay environment, which isolates computational performance, PPUs consistently demonstrated the highest speedup. For the 5000×5000 matrix workload, the PPU achieved a **27.6× speedup** over the CPU and a **9.2× speedup** over the GPU. Similar trends were observed for 3000×3000 and 2000×2000 workloads, where PPUs maintained substantial advantages due to their higher MIPS ratings.

6.3.3 End-to-End Speedup (With Network Delay)

When realistic network conditions were introduced, overall runtimes increased slightly due to the inclusion of fixed latency and bandwidth constraints. However, the relative speedup trends across devices were preserved, albeit with diminished margins especially for smaller workloads where communication overheads became significant.

For the 5000×5000 workload, the Photonic Processing Unit (PPU) achieved a **20.6×** speedup over the CPU and a **7.1×** improvement over the GPU. At this scale, the compute time dominated the end-to-end runtime, and the PPU's high MIPS configuration maintained a clear advantage even under network delays.

For the 3000×3000 workload, the PPU yielded a **14.4×** speedup over the CPU and a **5.1×** speedup over the GPU. The trend remained consistent, though the proportional impact of network transmission slightly reduced the advantage.

In the 2000×2000 case, where compute times were shortest, the speedup benefit of PPUs narrowed due to the fixed cost of data transmission. The PPU still outperformed the CPU by **6.7×**, but its speedup over the GPU dropped to **5.1×**. These results underscore that while PPUs offer considerable advantages for large-scale matrix operations, their effectiveness diminishes for smaller workloads where latency constitutes a greater share of the total runtime.

6.3.4 Trends Across Matrix Sizes

As the matrix size increased, the compute phase began to dominate the total runtime, reducing the relative impact of communication delays. Consequently, speedup values for the PPU became more stable and reflective of its architectural advantages. This confirmed that photonic acceleration is most effective in scenarios where computation heavily outweighs data transfer costs.

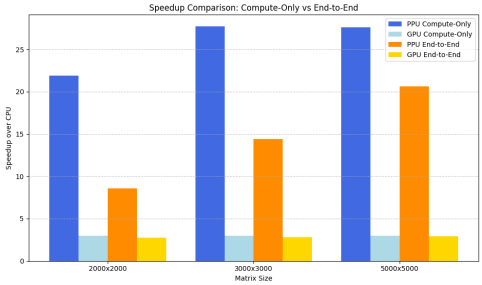

Figure 4: Speedup Comparison Bar Chart for 2000, 3000, and 5000 Matrix Sizes — Compute-Only vs End-to-End

6.4 Discussion

The simulation outcomes validate the hypothesis that Photonic Processing Units (PPUs) offer substantial acceleration potential for matrix-based AI workloads within cloud environments. By extending CloudSim to support end-to-end modeling including compute, data-in, and data-out delays the study enabled a fair, comparative evaluation across CPU, GPU, and PPU-based virtual machines. The results demonstrate that the PPU configuration consistently outperformed the CPU and GPU counterparts in pure computation scenarios. For the largest workload tested (5000×5000), the PPU exhibited approximately 27.6× speedup over the CPU and over 9.2× over the GPU in compute-only settings, confirming the architectural advantage of photonic processing at scale.

However, when realistic network conditions were introduced, the relative advantage narrowed. While PPUs still achieved superior end-to-end runtimes, the inclusion of fixed latency and bandwidth constraints impacted smaller workloads more significantly. In the 2000×2000 case, the compute time for the PPU was so short that the communication overhead dominated, occasionally resulting in anomalous or near-zero values due to CloudSim's event resolution limitations. This issue was addressed by focusing comparative analysis on workloads of 3000×3000 and above, where the compute phase remained dominant.

The observed results are consistent with prior studies such as Chen et al. (2023), where photonic accelerators demonstrated up to 9.2× speedup over electronic systems for matrix operations. Similarly, the end-to-end gains align with the performance improvements discussed by Hua et al. (2025), who reported 500× per-iteration latency reductions in a photonic Ising optimizer. The GPU performance trends followed expected patterns, acting as a middle ground between CPUs and PPUs.

From a practical perspective, the results suggest that integrating PPUs into cloud-native environments could significantly reduce AI inference times for large-scale neural workloads. However, the sensitivity to workload size and network overhead highlights the need for adaptive orchestration strategies that consider compute-network trade-offs. Furthermore, as photonic processors are currently specialized for matrix multiplications, their role in future heterogeneous systems may mirror that of TPUs accelerating specific stages of deep learning pipelines rather than serving as general-purpose processors.

From the perspective of cloud providers, the most compelling argument for adopting PPUs lies in their ability to deliver dramatic throughput gains for the workloads that matter most in practice. In compute-only simulations, PPUs achieved up to 27× faster performance than CPUs and 9× faster than GPUs, and even under realistic conditions with network delays they sustained over 20× acceleration for the largest workloads. For

providers, this means the ability to process more AI tasks per unit time, reduce customer-facing latency, and lower operational costs through better resource utilization. While relative benefits diminish for smaller workloads due to fixed communication overheads, enterprise-scale AI jobs in the cloud are typically large enough to fully exploit PPU acceleration, making them highly relevant for adoption once integration challenges are overcome.

These findings support the architectural feasibility of PPUs in the cloud and contribute to ongoing research into photonic hardware deployment. Future work must explore aspects such as energy modeling, dynamic workload scheduling, and multi-tenant contention to fully characterize the operational benefits of photonic acceleration in production-grade cloud systems.

7 Conclusion and Future Work

This research explored the architectural and performance implications of integrating Photonic Processing Units (PPUs) into cloud computing environments. By extending the CloudSim 3.0.3 simulation framework with a custom delay-aware broker, it was possible to model full end-to-end execution of AI workloads specifically matrix multiplication kernels across CPU, GPU, and photonic configurations.

The results confirmed that PPUs, modeled using MIPS-scaled performance from Chen et al. (2023), provide substantial acceleration over conventional compute models. In zero-delay conditions, PPUs achieved up to 27.6× speedup over CPUs and 9.2× over GPUs. Even under realistic network delay scenarios, speedups of over 20× were maintained for large workloads, validating the core hypothesis that photonic chips offer significant advantages in latency-sensitive, matrix-heavy inference tasks.

From the perspective of cloud providers, these findings are particularly compelling. PPUs enable significantly higher throughput by allowing more AI tasks to be processed per unit time, while also reducing customer-facing latency and improving infrastructure efficiency. This translates into tangible business benefits such as faster AI services, lower operational costs, and greater competitiveness in the rapidly growing AI cloud market. While smaller workloads see reduced relative gains due to fixed communication overheads, enterprise-scale AI applications in the cloud are typically large enough to fully exploit PPU acceleration, making the technology highly relevant once integration challenges are addressed.

Importantly, the findings align with the broader research question. The performance gains demonstrated by photonic hardware support the case for architectural transformation in cloud systems, particularly in replacing or augmenting TPUs for neural network inference. The simulation framework developed offers a lightweight method for evaluating photonic integration at the system level, bridging the gap between hardware-level benchmarks and cloud-native deployment considerations.

However, certain limitations remain. Simulation accuracy was constrained by CloudSim's event resolution, particularly for short-duration tasks. Additionally, energy consumption, multi-tenant scenarios, and heterogeneous workload scheduling were not addressed in this phase.

Future work will focus on refining the simulation model to incorporate energy-aware scheduling, dynamic workload profiling, and cloud-native orchestration across hybrid CPU/GPU/PPU environments. Expanding the model to include photonic memory ac-

cess, larger multi-VM scenarios, and integration with fog or edge nodes may further enhance the applicability of this research to emerging distributed AI platforms.

This study demonstrates that photonic acceleration is not only theoretically promising but also practically feasible within cloud infrastructure, offering a viable path toward scalable, low-latency, and energy-efficient AI workloads in the next generation of datacenters.

References

Bai, B., Yang, Q., Shu, H., and et al. (2023). Microcomb-based integrated photonic processing unit. *Nature Communications*, 14:66.

Chen, M., Yao, C., Wonfor, A., Yang, S., Holm, M., Cheng, Q., and Penty, R. (2023). Photonic integrated circuit for matrix inversions and multiplications. In *49th European Conference on Optical Communications (ECOC 2023)*, pages 867–869.

Cheng, J., Zhou, H., and Dong, J. (2021). Photonic matrix computing: From fundamentals to applications. *Nanomaterials*, 11(7):1683.

Dang, D., Lin, B., and Sahoo, D. (2022). Litecon: An all-photonic neuromorphic accelerator for energy-efficient deep learning. *ACM Transactions on Architecture and Code Optimization*, 19(3):1–22.

Davis, A. (2012). The role of photonics in future data centers. In *Proceedings of the Great Lakes Symposium on VLSI (GLSVLSI '12)*, pages 1–2.

Demirkiran, C., Eris, F., Wang, G., Elmhurst, J., Moore, N., Harris, N. C., Basumallik, A., Janapa Reddi, V., Joshi, A., and Bunandar, D. (2023). An electro-photonic system for accelerating deep neural networks. *Journal of Emerging Technologies in Computing Systems*, 19(4):Article 30.

Feldmann, J., Youngblood, N., Wright, C. D., Bhaskaran, H., and Pernice, W. H. P. (2021). Parallel convolutional processing using an integrated photonic tensor core. *Nature*, 589(7840):52–58.

Hua, S., Divita, E., Yu, S., and et al. (2025). An integrated large-scale photonic accelerator with ultralow latency. *Nature*, 640:361–367.

Jouppi, N., Kurian, G., Li, S., Ma, P., Nagarajan, R., Nai, L., Patil, N., Subramanian, S., Swing, A., Towles, B., Young, C., Zhou, X., Zhou, Z., and Patterson, D. A. (2023). Tpu v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings. In *Proceedings of the 50th Annual International Symposium on Computer Architecture (ISCA '23)*, pages Article 82, 1–14.

Miller, D. A. B. (2017). Attojoule optoelectronics for low-energy information processing and communications. *Journal of Lightwave Technology*, 35(3):346–396.

Schrenk, B. and Stephanie, M. V. (2024). pu – a photonic processing unit for heterogeneous optical networks. *Journal of Lightwave Technology*, 42(22):7989–7998.

Shen, Y., Harris, N. C., Skirlo, S., Prabhu, M., Baehr-Jones, T., Hochberg, M., Sun, X., Zhao, S., Larochelle, H., Englund, D., and Soljačić, M. (2017). Deep learning with coherent nanophotonic circuits. *Nature Photonics*, 11:441–446.

Tait, A. N., de Lima, T. F., Zhou, E., Wu, A. X., Nahmias, M. A., Shastri, B. J., and Prucnal, P. R. (2017). Neuromorphic photonic networks using silicon photonic weight banks. *Scientific Reports*, 7:7430.

Xia, C., Chen, Y., Zhang, H., and Wu, J. (2023). Stadia: Photonic stochastic gradient descent for neural network accelerators. *ACM Transactions on Embedded Computing Systems*, 22(5s):Article 126.

Yang, M., Zhong, Z., and Ghobadi, M. (2023). On-fiber photonic computing. In *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks (HotNets '23)*, pages 263–271.

Ying, Z., Feng, C., Zhao, Z., and et al. (2020). Electronic-photonic arithmetic logic unit for high-speed computing. *Nature Communications*, 11:2154.

Zhou, H., Dong, J., Cheng, J., and et al. (2022). Photonic matrix multiplication lights up photonic accelerator and beyond. *Light: Science Applications*, 11:30.