

Міністерство освіти і науки України
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри,
доктор техн. наук, професор

_____ І. М. Журавська

« __ » _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА
Серверна архітектура для потокової передачі VR
Спеціальність 123 Комп'ютерна інженерія
123 – КР.1 – 605.21810525

Студента:

_____ К.Ю. Сюсько
« __ » _____ 2024 р.

Керівник:

_____ Г.П.Чуйко
« __ » _____ 2024 р.

Миколаїв 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ЗАТВЕРДЖУЮ
Завідувач кафедри,
доктор техн. наук, професор
_____ І. М. Журавська
« __ » _____ 2024 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

Видано студентці групи 605 факультету комп'ютерних наук

Сюсько Кирило Юрійович

(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи: Серверна архітектура для потокової передачі VR. _____

Затверджена наказом по ЧНУ від «_08_» __ листопада _____ 2023 р. No _____ 226 _____

2. Строк представлення кваліфікаційної роботи «_____» _____ 2024 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Детальний розгляд серверної архітектури для потокової передачі контенту для VR пристроїв.

4. Перелік питань, що підлягають розробці

Аналіз поняття VR, VR streaming, розглянути поняття серверної архітектури, зробити детальний опис технологій та алгоритмів для потокової передачі контенту для VR.

5. Перелік графічних матеріалів

Зображення інтерфейсу AWS

Зображення схем архітектур

Зображення графіків даних

Зображення формул та алгоритмів

Зображення девайсів

Зображення інтерфейсу програм

6. Завдання до спеціальної частини

Розглянути одне з відомих питань охорони праці для такого робочого місця (мікрокліматичні умови, електробезпека, питання безпеки роботи з обладнанням), а також розглянути питання що стосуються безпеки персоналу.

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
викладач Л. І. Григор'єва	кафедра екології Медичного інституту ЧНУ ім. Петра Могили	спеціальна частина з охорони праці

Керівник роботи ____ доктор фіз.-мат. наук, професор , Г. П. Чуйко _ ____
(посада, прізвище, ім'я, по батькові) (підпис)

Завдання прийнято до виконання __Сюсько Кирило Юрійович____ ____
(прізвище, ім'я, по батькові студента)
(підпис)

Дата видачі завдання «____» _____ 20____ р.

КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної роботи

Тема: Апаратно-програмний комплекс для людей з вадами дихальної системи

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КР	01.09.2023	05.10.2023	Виконано
2	Огляд літератури за темою роботи	06.10.2023	12.10.2023	Виконано
3	Складання календарного плану КР	13.10.2023	14.10.2023	Виконано
4	Аналіз предметної області	15.10.2023	20.10.2023	Виконано
5	Розробка проектних рішень	21.10.2023	15.11.2023	Виконано
6	Моделювання серверної архітектури	16.11.2023	18.12.2023	Виконано
7	Створення серверної архітектури	18.11.2023	13.02.2024	Виконано
8	Оформлення КР та презентації	01.11.2023	31.12.2023	Виконано
9	Попередній захист	14.02.2023	14.02.2024	Виконано
10	Рецензування	13.02.2024	14.02.2024	Виконано
11	Завершення оформлення КР та Презентації	13.02.2024	14.02.2024	Виконано
12	Відгук керівника КР	13.02.2024	16.02.2024	Виконано
13	Захист кваліфікаційної роботи	27.02.2024	27.02.2024	Виконано

Розробив студента _____ Сюсько Кирило Юрійович _____
(прізвище, ім'я, по батькові) (підпис)
«__» _____ 2024 р.

Керівник роботи ____ доктор фіз.-мат. наук, професор , Г. П. Чуйко _____
(посада, прізвище, ім'я, по батькові) (підпис)
«__» _____ 2024 р.

АНОТАЦІЯ

магістерської роботи

«Серверна архітектура для потокової передачі VR»

Студента: Сюсько Кирило Юрійович

Керівник: доктор фіз.-мат. наук, професор Г. П. Чуйко

Магістерська робота присвячена створенню серверної архітектури для потокової передачі VR. В даній магістерській роботі розглянуто та досліджено принципи та методи та різні підходи до створення серверних архітектури для потокової передачі VR контенту. Практичне значення результати дослідження можуть бути використані у сферах навчання, розваг та глибокого навчання, полегшуючи роботу фахівців та покращуючи якість навчального процесу. Пояснювальна записка магістерської роботи складається зі вступу, трьох розділів, висновків. У вступі визначається актуальність теми, сформульовані мета, об'єкт, предмет та завдання дослідження та розроблення, а також гіпотези, наукова новизна та практичне значення магістерської роботи. У першому розділі розглядається аналітична частина, його опис, де використовується, а також я воно пов'язано з ІТ-галузі, також розглядаються конкуренти. У другому розділі розглядається «адаптивний метод потокової передачі VR» як один із існуючих методів. Було розглянуто 2 алгоритма завдяки яким відбувається потокова передача VR контенту. У третьому розділі, було розглянуті та запропоновано варіанти розгортання серверних архітектури для потокової передачі VR.

В цілому, магістерська робота без додатків містить 63 сторінок, 20 рисунків, 5 таблиць, 24 джерел посилання.

Ключові слова: VR, VR straming, серверна архітектура, візуалізація даних

ABSTRACT

of the Master's Thesis

" Server architecture for VR streaming "

Student: Kyrylo Yuriyovych Siusko

Consultant: Doctor of Phys.-Math. Sciences, Professor H.P. Chuyko

The master's thesis is dedicated to the development of server architecture for streaming VR content. The study explores principles, methods, and various approaches to creating server architecture for streaming VR content. The practical significance of the research results can be applied in education, entertainment, and deep learning, facilitating the work of professionals and improving the quality of the learning process. The explanatory note of the master's thesis consists of an introduction, three chapters, and conclusions.

The introduction defines the relevance of the topic, formulates the goal, object, subject, and tasks of the research and development, as well as hypotheses, scientific novelty, and practical significance of the master's thesis. The first chapter examines the analytical part, its description, its usage, and its relevance to the IT industry, as well as competitors. The second chapter discusses the "Adaptive Method for Streaming VR" as one of the existing methods. Two algorithms facilitating the streaming of VR content are considered. In the third chapter, options for deploying server architecture for streaming VR are discussed and proposed.

Overall, the master's thesis, excluding appendices, comprises 63 pages, 20 figures, 5 tables, and 24 references.

Keywords: VR, VR streaming, server architecture, data visualization.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	9
ВСТУП	10
1	13
1.1	13
1.1.1	14
1.2	18
1.1.2	18
1.3	20
1.1.3	20
1.1.4	22
Висновки до розділу 1	23
2	25
2.1	26
2.2	26
2.3	29
2.4	31
2.5	32
2.5.1	32
2.5.2	33
2.6	35
2.6.1	35
2.6.2	36
Висновки до розділу 2	40
3	42
3.2	42
3.2.1	42
3.2.2	47
3.2.3	51
3.3	51

	Висновки до розділу 3	52
4	53	
	4.1	53
	4.2	54
	4.3	56
	4.4	57
	4.4.1	59
	4.4.2	62
	Висновки до розділу 4	64
	Висновок	65
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	67
	Додаток А Лістинг програми з розширенням .JAVA	70
	Додаток Б АБРОБАЦІЇ	72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ТПР	–	Теорія прийняття рішень
ЗПР	–	Завдання прийняття рішень
ШНМ	–	Штучна нейронна мережа
ПЗ	–	Програмне забезпечення
ХНУ	–	Хмельницький національний університет
VR	–	Virtual reality
AR	–	Augmented reality
RAID	–	Redundant Array of Independent Disks
IDE	–	Integrated Drive Electronics
IoT	–	Internet of things
LTS	–	Long Term Support
GOP	–	Graphics Output Protocol
QoE	–	Quality of Experience
FoV	–	Field of View
HMD	–	Human Mobile Device
HTTP	–	Hyper Text Transfer Protocol
IP	–	Internet Protocol
RTT	–	Round Trip Time
RTd	–	Round Trip Delay

Вступ

Серверна архітектура для стрімінгу віртуальної реальності (VR) представляє собою важливий елемент у сучасних технологіях, які надають можливість користувачам отримувати віртуальний контент у реальному часі через мережу Інтернет. Вона включає в себе різноманітні компоненти та процеси, які спільно забезпечують стабільну та якісну передачу відео та звуку з віртуального середовища на пристрої користувачів.

Основні складові серверної архітектури для стрімінгу VR включають: Серверні ресурси: Це обчислювальна потужність, пам'ять та інші ресурси, які необхідні для обробки та передачі відеоданих у реальному часі. Стабільна мережа: Надійна та швидка мережа є ключовим елементом для стрімінгу VR. Вона забезпечує низьку затримку та високу пропускну здатність, що дозволяє користувачам отримувати плавний та якісний віртуальний досвід. Програмне забезпечення сервера: Це спеціалізовані програмні рішення, які керують процесами стрімінгу, кодування та передачі відеоданих, а також взаємодію з клієнтськими пристроями. Клієнтські пристрої: Це пристрої, які отримують відеодані з сервера і відтворюють їх для користувачів. Вони можуть бути різних типів, включаючи VR-гарнітури, смартфони або комп'ютери. Ефективна серверна архітектура для стрімінгу VR повинна забезпечувати високу якість відео та звуку, низьку затримку та стабільність під час передачі даних. Це дозволяє користувачам насолоджуватися іммерсивним віртуальним досвідом без перерв та завад.

Додатково, технології, які доцільно враховують середовище, включають в себе інструменти, такі як NVIDIA CloudXR, які революціонізують спосіб, якими користувачі отримують доступ до віртуального контенту, забезпечуючи високу якість зображення та стабільність стрімінгу віртуальної реальності. Така архітектура сервера для стрімінгу VR має забезпечувати високу якість відео та звуку, низьку затримку та стабільність під час передачі даних, що

дозволяє користувачам насолоджуватися іммерсивним віртуальним досвідом без перерв та завад.

Мета: основною метою кваліфікаційної магістерської роботи є підвищення фахового рівня випускників шляхом аналізу сучасних наукових досліджень за обраною тематикою та отримання досвіду під час виконання наукової роботи в умовах виробничої діяльності, а також підготовка та збір матеріалів магістерської роботи.

Мета та завдання дослідження.

Мета дослідження: розробка та вдосконалення серверної архітектури для стрімінгу віртуальної реальності (VR) з метою забезпечення стабільної та якісної передачі відео- та аудіоданих користувачам у реальному часі.

Об'єкт дослідження: технологічні процеси та компоненти, які використовуються в серверній архітектурі для стрімінгу віртуальної реальності (VR)

Предмет дослідження: особливості та оптимізація серверної архітектури для забезпечення ефективного стрімінгу віртуальної реальності, зокрема фактори, що впливають на якість відео- та аудіопотоку, мережеві протоколи та алгоритми оптимізації передачі даних.

Завдання:

1. Аналіз існуючих систем стрімінгу VR та їхні обмеження, оцінка та дослідження різних систем стрімінгу VR, їх функціональних можливостей та обмежень.;
2. Дослідження архітектур сервера для стрімінгу VR, аналіз різних архітектур серверів, придатних для стрімінгу віртуальної реальності, включаючи їхні технічні характеристики та оптимізацію для забезпечення високоякісного стрімінгу;
3. Дослідження алгоритмів обробки відеоданих для стрімінгу VR, вивчення та аналіз алгоритмів обробки відеоданих у контексті стрімінгу VR з метою забезпечення стабільності та високої якості передачі даних;

4. Розробка серверної архітектури для стрімінгу VR, розробка та оптимізація архітектури сервера з урахуванням вивчених аспектів для забезпечення ефективного та стабільного стрімінгу віртуальної реальності;
5. Проведення експериментальних досліджень та оцінка ефективності серверної архітектури, проведення тестувань та аналіз результатів роботи серверної архітектури на практиці з метою оцінки її ефективності та визначення можливостей для подальшого вдосконалення;
6. Дослідження вимог та стандартизація, вивчення вимог щодо стандартів у галузі серверних архітектур для стрімінгу VR та врахування відповідних стандартів у процесі розробки та впровадження.

Робота пройшла апробацію під час XV Міжнародної науково-практичної конференції «Free and open source software» (дистанційно, 13–14 лютого 2024 р.). Основні положення магістерської роботи опубліковані у збірнику матеріалів XV Міжнародної науково-практичної конференції «Free and open source software» [26].

1 АНАЛІТИЧНА ЧАСТИНА. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ СЕРВЕРНИХ АРХІТЕКТУР ПОТОКОВОЇ ПЕРЕДАЧІ VR. ФОРМУВАННЯ ВИМОГ ДО АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Аналітична частина VR та VR streaming

Останні роки принесли помітний розвиток комерційних технологій віртуальної реальності (VR). У 2015 році цей сектор досяг вартості 1,37 мільярда доларів США і передбачається, що до 2022 року ця цифра зросте до 33,90 мільярда доларів. Тепер користувачі можуть використовувати гарнітури віртуальної реальності (HMD), такі як Samsung Gear VR, HTC Vive та Facebook Oculus, щоб вільно налаштовувати орієнтацію голови та дивитися панорамні відео у форматі 360 градусів. Основуючись на орієнтації користувача та полі зору (FoV), ці гарнітури відображають видиму область на екрані. Зараз FoV обмежений приблизно до 90 градусів як горизонтально, так і вертикально.

Одним із основних викликів у стрімінгу відео VR є високі вимоги до пропускної здатності. Відео у форматі 360 градусів потребують кодування щонайменше у роздільній здатності 4K, щоб забезпечити прийнятну якість перегляду [3]. Причина в тому, що, хоча весь відеоматеріал знаходиться у форматі 4K, FoV становить лише одну п'яту або шосту частину цієї роздільної здатності, і користувачі дивляться майже відео з роздільною здатністю 720P. Внаслідок цього для такого стрімінгу потрібно в чотири-п'ять разів більше пропускної здатності порівняно з традиційним [4]. Згідно з даними Netflix [5], для стрімінгу відео у форматі 4K швидкість завантаження повинна досягати принаймні 25 Мбіт/с. Однак глобальна швидкість мобільного інтернет-з'єднання у 2017 році складала лише 8,7 Мбіт/с, а до 2021 року планується досягти 20,4 Мбіт/с [6].

Нещодавно в травні 2015 року було запропоновано нову версію протоколу HTTP, яка отримала назву HTTP/2, і була призначена як альтернатива HTTP/1.1 з кращою продуктивністю [7]. Вона вводить нові функції, включаючи серверне відправлення, пріоритет потоку та завершення

поток. Серверне відправлення дозволяє клієнту отримувати кілька об'єктів за один запит. Функція пріоритету потоку дозволяє клієнту запитувати сервер виділяти більше ресурсів для відправлення даних в одному потоці порівняно з іншими. Також потоки можуть бути завершені, якщо клієнт надсилає фрейм RST STREAM для активації функції завершення потоку.

Хоча 360 відео VR отримує значну увагу в науковому середовищі, стрімінг цих передових відео зараз здійснюється просто за допомогою послідовного відтворення плиток в сцені після вибору для них бітрейту [14] [15] [17]. Більшість поточних підходів базуються на HTTP/1 і не були адекватно оцінені в науковій літературі. З наявних досліджень, наша попередня робота в [9] є першою, яка використовує нові функції HTTP/2. У цьому дослідженні плитки в одній і тій же області завантажуються послідовно, тому клієнт має чекати, поки всі ці плитки будуть відображені, щоб побачити всю область, що призводить до високої початкової затримки.

1.1.1 Огляд рішення для VR стрімінгу від NVIDIA CloudXR™

Пакет NVIDIA CloudXR™ у поєднанні з потужністю графічних процесорів NVIDIA RTX™ і віртуальної робочої станції NVIDIA RTX (vWS) дозволяє додаткам XR передавати високоточну розширену реальність (XR) на пристрої Android і iOS через продуктивні мережі. Динамічно підлаштовуючись під умови мережі, CloudXR максимізує якість зображення та частоту кадрів, щоб забезпечити наступний рівень бездротової доповненої та віртуальної реальності.

NVIDIA CloudXR - це технологія, розроблена компанією NVIDIA, яка дозволяє стрімити вміст віртуальної реальності (VR) та доповненої реальності (AR) через мережу з великою пропускну здатністю. Ця технологія дозволяє користувачам отримувати доступ до інтенсивних віртуальних досвідів навіть на пристроях з обмеженими ресурсами.

Основні особливості NVIDIA CloudXR включають:

- Стійкість до затримок: Технологія CloudXR оптимізована для мінімізації затримок, що дозволяє забезпечити плавний та безперервний потік віртуальної реальності.
- Висока якість зображення: CloudXR використовує розширені алгоритми стиснення, що дозволяють передавати високоякісне відео з низькою затримкою.
- Масштабованість: Технологія CloudXR побудована з урахуванням можливості масштабування, що дозволяє обробляти велику кількість одночасних з'єднань.
- Підтримка різних платформ: CloudXR може бути інтегрована з різними типами VR- та AR-пристроїв, що робить її більш універсальною для різних сценаріїв використання.

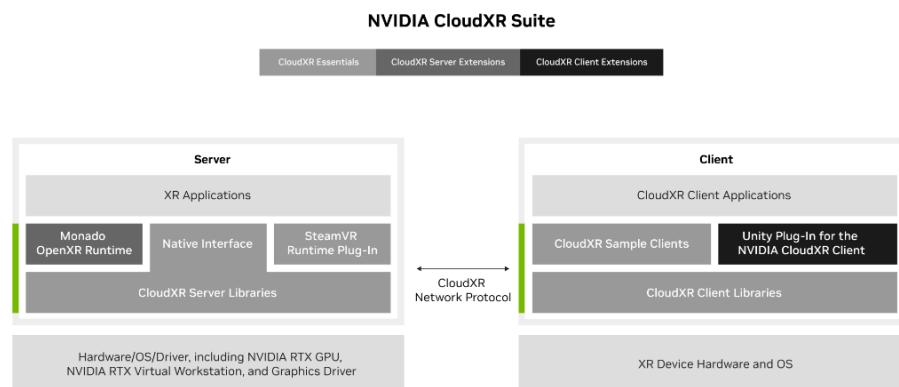


Рисунок 1.1 – Блочна архітектура взаємодії клієнт-серверу від пакету
NVIDIA CloudXR

NVIDIA CloudXR використовується в різних областях, таких як медицина, освіта, розваги та промисловість, де він дозволяє користувачам отримувати доступ до високоякісних віртуальних і доповнених реальностей, навіть якщо їхні пристрої не мають достатньої обчислювальної потужності для обробки цих даних локально.

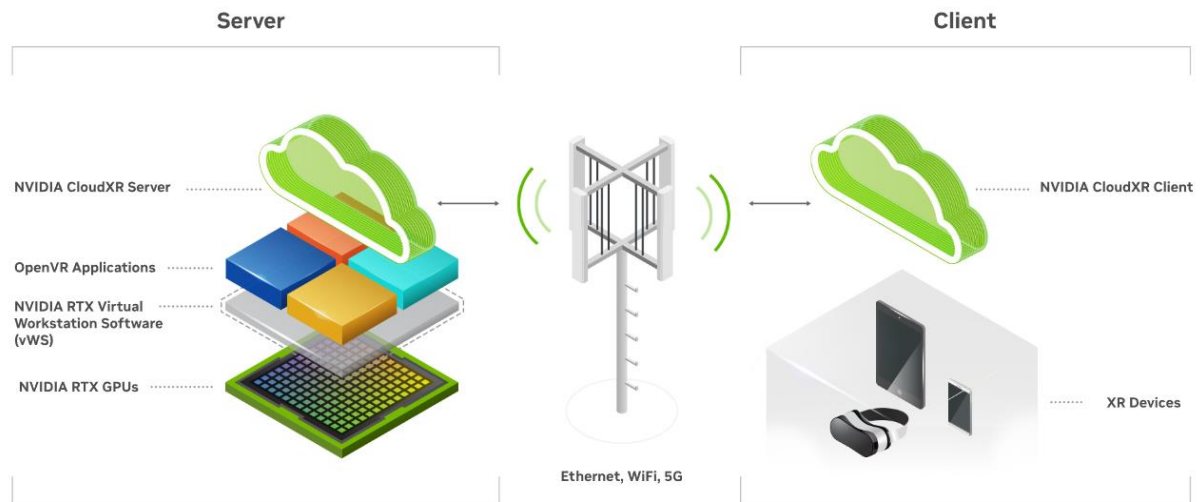


Рисунок 1.2 □ Інтерактивна схема архітектури взаємодії клієнт-серверу від пакету NVIDIA CloudXR

Передача складної графіки з VR та AR високошвидкісними мережами. CloudXR розширює межі VR та AR, забезпечуючи стрімінг XR-контенту на портативні пристрої без зниження якості порівняно з традиційними рішеннями. CloudXR дозволяє використовувати ПЗ для віртуалізації GPU та сервери на основі NVIDIA RTX для стрімінгу VR- та AR-контенту з будь-якої програми OpenVR. Це означає, що навіть найскладнішу графіку з VR та AR можна передавати з віддаленого сервера через мережу 5G та Wi-Fi на будь-який пристрій без обмежень фізичного обладнання.

Безкомпромісна продуктивність та мобільність. Сучасні рішення для VR/AR потребують фізичного високопродуктивного комп'ютера, який може забезпечити достатню потужність рендерингу для складних імерсивних

середовищ. CloudXR використовує графічні процесори NVIDIA RTX, які декодують та передають зображення на клієнтські пристрої з ОС Windows та Android. Динамічно підлаштовуючись під параметри мережі, CloudXR підвищує якість зображення та частоту зміни кадрів, мінімізуючи затримку та розриви зображення.

Стрімінг AR, VR та XR з периферійних пристроїв. Мобільні мережі нового покоління відрізняються наднизькою затримкою, що розширює можливості доповненої та віртуальної реальності. Сервери на базі RTX забезпечують чудову продуктивність для трансляції графіки з CloudXR. Висока пропускна здатність та низька затримка мережі 5G дозволяє транслювати ресурсомістку графіку з VR та AR для професійних користувачів.

Високі стандарти керованості, безпеки та багатоорендності. Поєднання продуктивності платформи EGX та ПЗ NVIDIA RTX Virtual Workstation (vWS) у CloudXR забезпечує надійну багатоорендність, що є стандартною вимогою операторів зв'язку та виробників серверів. Завдяки віртуалізації, кілька віртуальних машин можуть використовувати один GPU при зниженні навантаження. Це оптимізує щільність користувачів та експлуатаційні витрати. А підтримка систем віртуалізації VMware та Red Hat та додаткові інструменти управління покращують керованість.

Програмне та апаратне забезпечення в інноваційному наборі інструментів. Організація передачі XR-контенту з периферійних пристроїв потребує досвіду в галузі оркестрації хмарних систем, візуальних обчислень, П, віртуалізації, роботи мереж 5G та інших сфер. Оптимізувати візуалізацію на периферійних пристроях дозволить набір інструментів CloudXR 5G MEC для розробки рішень MEC для мереж 5G за допомогою CloudXR на платформі NVIDIA EGX. Це повний стек програмно-апаратного забезпечення NVIDIA. Набір інструментів підтримується сертифікованими NVIDIA системами та об'єднує GPU NVIDIA RTX, RTX vWS, SDK CloudXR та сторонні програми.

Рішення дозволяє розробникам розпочати роботу з VR, AR та віртуальним GPU по мережі 5G та Wi-Fi.

1.2 Огляд існуючих рішень системи, що розробляється

1.1.2 Browse Streaming apps VR Games on Meta Quest

«Browse Streaming apps VR Games on Meta Quest» — це платформа, яка дозволяє користувачам Meta Quest переглядати та підтримувати доступ до широкого спектру віртуальних розваг та додатків для віртуальної реальності (VR).

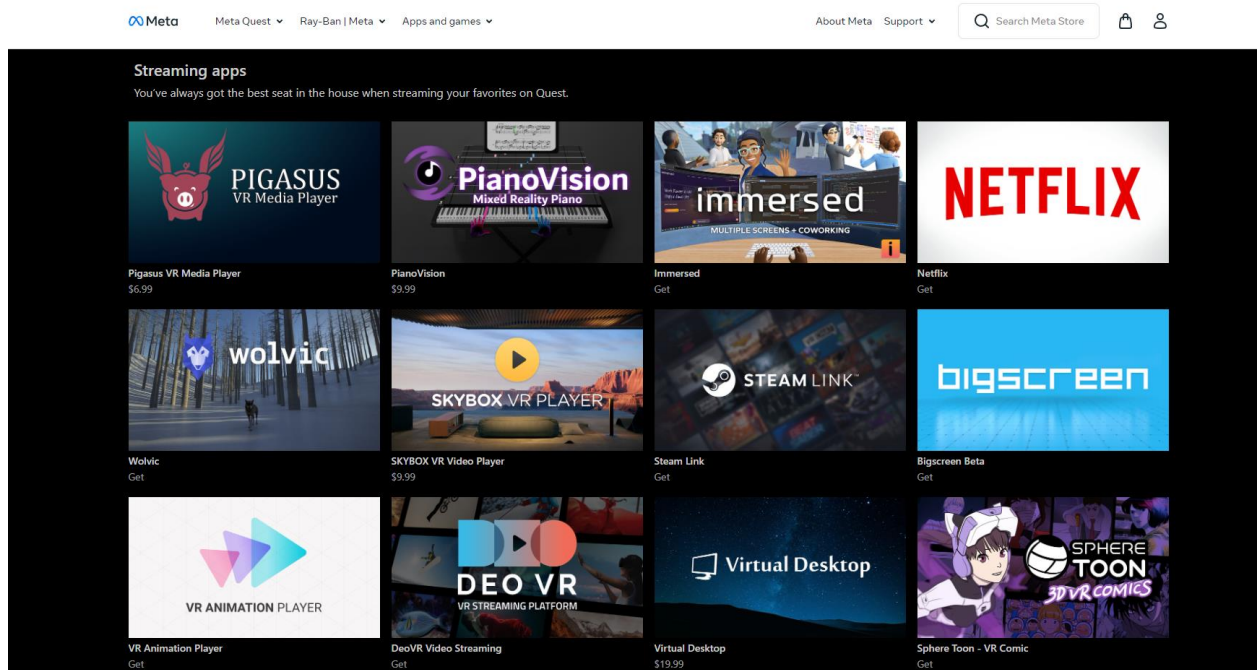


Рисунок 1.3 □ Перелік доступних функцій та програм Browse Streaming apps
VR Games on Meta Quest

Особливості:

Широкий вибір додатків та ігор: Платформа надає доступ до різноманітних VR-ігор, додатків та стрімінгових сервісів, які охоплюють різні жанри та категорії інтересів користувачів.

Навігація та пошук: Користувачі можуть швидко знайти ігри та додатки за допомогою зручного пошуку та фільтрів за категоріями, рейтингами та рекомендаціями.

Огляд та опис: Кожна гра або додаток має докладний огляд та опис, які користувачі роблять правильний вибір перед завантаженням.

Підтримка стрімінгової платформи: Платформа дозволяє користувачам отримувати доступ до стрімінгових сервісів, таких як Netflix, Hulu, YouTube VR та інших, де вони можуть переглядати відео в імерсивному VR-середовищі.

Спільнота та рекомендації: Користувачі можуть ділитися відгуками, рекомендаціями та враженнями від інших учасників спільноти, що дозволяє створити активне та дружнє середовище.

Оновлення та нові випуски: Платформа регулярно оновлюється з випуском нових ігор, додатків та функцій, щоб користувачі могли завжди насолоджуватися свіжим контентом.

Ключові переваги:

Різноманіття контенту. Користувачі можуть знайти щось на свій смак серед великого вибору ігор і додатків.

Зручність використання. Інтуїтивний і простий інтерфейс робить пошук та вибір контенту легким та приємним.

Спільнота користувачів. Можливість ділитися враженнями та отримувати рекомендації від інших користувачів створює взаємодопомогу та співпрацю.

Ключові недоліки:

Обмеженість вибору. Хоча вибір і досить великий, проте може бути обмежений порівняно з іншими платформами.

Не всі додатки безкоштовні. Деякі ігри та додатки можуть мати вартість, що може бути обтяжливим для деяких користувачів.

Підтримка тільки пристроїв компанії Meta, тобто тільки лінійка Meta Oculus VR.

Потреба в хорошому інтернет-з'єднанні. Відтворення стрімінгових відео та ігор може вимагати стабільного та швидкого Інтернет-з'єднання, що не завжди доступно.

1.3 Формування вимог до апаратно-програмного забезпечення

1.1.3 Вимоги до апаратного забезпечення

Сервер бази даних — найпотужніший з серверів системи з найбільшим обсягом оперативної пам'яті і високопродуктивною і відмовостійкою дисковою підсистемою. У ньому необхідно використовувати від 2 до 4 багатоядерних процесорів, максимальний обсяг ОЗУ, швидкі диски або масив зберігання даних.

На серверах додатків виконуються всі розрахунки. Сервер цього типу повинен мати обсяг оперативної пам'яті від 6 до 12 Гб і високопродуктивну дискову підсистему. Рекомендується використовувати RAID-масив або налагоджену систему резервного копіювання всього сервера додатків. Для ефективної роботи досить використовувати 2 серверних багатоядерних процесора сучасної архітектури з частотою ядра від 2,0 ГГц. Сервери системи повинні виконувати тільки програмний код системи і відповідних компонент операційної системи. Будь-яке додаткове ПЗ на сервері веде до зниження швидкості обслуговування клієнтів.

Рекомендується використовувати серверне обладнання, яке задовольняє наступним вимогам: багатоядерні процесори з підтримкою 64-розрядних команд, оперативна пам'ять з контролем парності і виправленням помилок; апаратний RAID-контролер; диски SAS 15k rpm; мережеві плати зі швидкістю роботи 1 Гб/с і вище.

Сервери додатків і БД повинні знаходитися в рамках однієї локальної мережі. Рекомендується зв'язок по каналу 1 Гб/с. Вимога до каналу між клієнтської робочої станцією і сервером додатків — будь-яке з'єднання, що забезпечує взаємодію по протоколу TCP.

З боку клієнта необхідно 128 Кб/с, рекомендовано з'єднання зі швидкістю 512 Кб/с і вище (мінімально — 56 кб/с). З боку сервера додатків для одночасної роботи до 25 клієнтів необхідний канал з пропускною спроможністю 1 Мб/с.

Клієнтськими робочими місцями можуть бути будь-які сучасні пристрої — персональний комп'ютер, ноутбук, нетбук з ОС, сумісної з Microsoft Windows, тонкі робочі станції, iPad, iPhone, Windows- і Android-планшети, смартфони і будь-які інші пристрої, які функціонують в якості клієнта до сервісу терміналів або віртуальних робочих столів. Але для зручного використання потрібен пристрої на кшталт окулярів віртуальної реальності для повного занурення у контент.

Гарним приклад є Oculus Quest 2: VR-гарнітура. Бренд Oculus спільно з розробниками з корпорації META (FaceBook) створили оптимальну модель для тих, хто все ще з побоюванням розглядає VR-сети. Стильний дизайн, просте інтуїтивне керування за допомогою джойстиків, широкий асортимент ігор та нові враження — все це ви отримаєте, якщо вирішите купити окуляри віртуальної реальності Oculus Quest 2.



Рисунок 1.4 □ Oculus Quest 2: VR-гарнітура

Окуляри віртуальної реальності Oculus Quest 2, мабуть, одна з найкращих моделей, яку здатний запропонувати сучасний ринок технологій. Серед основних переваг гаджета для віртуальної реальності варто відзначити: стильний, зручний дизайн, комфортна вага та оптимальні розміри, неймовірно захоплюючий досвід від гри, потужна вбудована аудіосистема з глибоким звуком, простота у використанні та інтуїтивне керування, вражаючий

асортимент ігор у фірмовій онлайн-бібліотеці, доступна ціна окулярів віртуальної реальності Oculus Quest 2.

Oculus Quest 2 має підтримку зворотної сумісності з оригінальною бібліотекою Oculus Quest. Додайте до цього всі сучасні VR-проекти, такі як Resident Evil 4 VR та отримайте сотні різних програм та ігор. І з кожним днем це число зростає. А оскільки такі заходи як Meta Quest Gaming Showcase стають щорічними, ми спостерігаємо здорову частоту появи великих ігор в системі, що вселяє впевненість у світлому майбутньому хедсетів віртуальної реальності.

Головною проблемою оригінального Oculus Quest був його процесор – Snapdragon 835. Насамперед він був розроблений як процесор для смартфонів, що помітно обмежувало його можливості як чіпсета для VR.

Oculus Quest 2, своєю чергою, отримав чіпсет Snapdragon XR2, що подарував гаджету відчутний приріст потужності. Новий Quest також має 6 Гб оперативної пам'яті, що дозволяє йому працювати набагато стабільніше, забезпечуючи плавну та швидку навігацію по меню, а також більш детальний ігровий досвід.

Oculus Quest 2 йде в комплекті з переробленою та покращеною версією Oculus Touch Controllers — контролерів першого Квесту. Це оновлення покликане в першу чергу для покращення трекінгу жестикуляції.

Зовні вони такі ж, як і раніше: кожен із контролерів виконаний у формі, що нагадує ківш. Є дві кнопки для стрільби та захоплення об'єктів, а також інтерактивні аналогові стіки та пара кнопок на лицьовій стороні кожного контролера.

1.1.4 Вимоги до програмного забезпечення

Програмне забезпечення повинно використовувати технологію Nvidia CloudXR для ефективного підключення, рендеру зображення та передачі з серверу до клієнта не витрачаючи час.

PyCharm

PyCharm – це кросплатформна IDE, яка використовується в комп'ютерному програмуванні спеціально для Python. Платформа, розроблена JetBrains, в основному використовується для аналізу коду, графічного налагоджувача тощо. Вона підтримує веб-розробку за допомогою Django, а також Data Science за допомогою Anaconda.

PyCharm може бути відмінним інтегрованим середовищем розробки для програмної розробки системи та інших технологій. PyCharm підтримує мову програмування Python, що є популярним вибором для розробки комп'ютерного зору та обробки зображень.

Переваги використання PyCharm для розробки системи контролю дорожнього руху:

Підтримка Python: PyCharm має вбудовану підтримку Python, що дозволяє легко і ефективно працювати з мовою програмування, яка часто використовується для розробки системи контролю дорожнього руху.

Підтримка віртуальних середовищ: PyCharm дозволяє легко створювати та управляти віртуальними середовищами, що є важливим для ізоляції проекту та уникнення конфліктів залежностей.

Підтримка Git: PyCharm інтегрується з системами керування версіями, такими як Git, що дозволяє командам легко співпрацювати та ведення контролю над версіями коду.

Багатофункціональність та плагіни: PyCharm має багатофункціональний інтерфейс, а також можливості для розширення за допомогою плагінів, що може бути корисним для інтеграції додаткових інструментів та функціональності.

Висновки до розділу 1

Обидва розглянуті рішення - NVIDIA CloudXR та платформа "Browse Streaming apps VR Games on Meta Quest" - відображають сучасні тенденції використання віртуальної та доповненої реальності. Обидва вони пропонують

унікальні можливості для користувачів у доступі до високоякісного контенту, але мають свої відмінності.

NVIDIA CloudXR спрямований на розширення між віртуальною та доповненою реальністю через стрімінг XR-контенту на портативні пристрої, навіть без значного обладнання на місці. Це може бути корисним у сферах, де важко забезпечити потужні обчислювальні ресурси на місці, таких як медицина, освіта та промисловість.

З іншого боку, платформа "Browse Streaming Apps VR Games on Meta Quest" надає доступ до широкого спектру VR-ігор, додатків та стрімінгових сервісів для користувачів Meta Quest. Вона забезпечує зручний інтерфейс та спільноту користувачів, але обмежена виключно на пристрої Meta Oculus VR.

В обох випадках важливо враховувати вимоги до апаратного забезпечення, зокрема потужність серверів та швидкість Інтернет-з'єднання для оптимальної роботи системи. Крім того, доступність контенту та спосіб навігації є ключовими факторами успіху для обох платформ.

Отже, обидва рішення відображають різні підходи до використання технологій віртуальної та доповненої реальності, пропонуючи різноманітні можливості для розвитку цих інноваційних областей.

2 АДАПТИВНИЙ МЕТОД ДЛЯ ВІДЕО 360 VR З НИЗЬКОЮ ЗАТРИМКОЮ ПОТОКОВЕ ПЕРЕДАВАННЯ ЧЕРЕЗ HTTP/2

2.1 Анотація.

Технологія віртуальної реальності (VR), особливо 360 VR-відео наразі є гарячою темою завдяки ефекту занурення досвід, порівняно з традиційними мультимедійними програмами, наприклад дозволити користувачам відчувати близьке реальне життя за допомогою панорами переглянути. Тим не менш, передача відео 360 VR споживає величезну кількість пропускну здатність. Крім того, як зменшити затримку відео 360 VR потокове передавання – ще одна проблема. Щоб вирішити ці дві проблеми, буде представлено ефективне адаптивне відео 360 VR потоковий метод через HTTP/2 із використанням пріоритету потоку та потоку особливості закінчення. Щоб підтримати адаптивність, є відео 360 VR розділений на кілька граней, і кожна грань розділена на скроневі сегменти. Відео також зберігається на сервері з різними рівні якості. У нашому методі ми зосереджені на низькій затримці сценарію, тому всі обличчя завантажуються одночасно використовує пріоритетну функцію HTTP/2, і клієнт декодує їх на основі групи зображень (GOP). Якщо пропускну здатність раптом падає, клієнт порушить одночасність і повідомить сервер для послідовного надсилання обличчя на основі їхніх пріоритетів. Крім того, для завершення можна використовувати функцію завершення потоку частини ЗМІ, які пропустять крайній термін. Експериментальний результати показують, що, коли початковий буфер встановлено на 0,5 с, є лише 0,08% від кількості GOP на передній грані, які промахнулися кінцевий термін у запропонованому методі порівняно зі значеннями 0,75% і 3,34% двох еталонних методів. Індексні терміни — потокове відео VR, HTTP/2, віртуальна реальність, HTTP Adaptive Streaming, пріоритет потоку, завершення потоку.

2.2 Уведення

Останні роки свідчать про помітний ріст у комерційному прогресі технології віртуальної реальності (VR). Вона досягла 1,37 мільярда доларів

США у 2015 році і очікується зростання до 33,90 мільярдів доларів США у 2022 році. Люди тепер можуть використовувати гарнітури віртуальної реальності (HMD), такі як Samsung Gear VR, HTC Vive та Facebook Oculus, і вільно налаштовувати орієнтацію голови, щоб дивитися панорамні види у 360-градусних відеороликах VR. На основі орієнтації користувача та поля зору (FoV) ці HMD відображають видиму область на екрані. Поточно FoV обмежений приблизно 90 ° як по горизонталі, так і по вертикалі.

Одним з основних викликів у потоковій передачі відео VR є висока вимога до пропускної здатності. 360-градусні відеоролики VR потребують кодування принаймні у роздільній здатності 4K, щоб забезпечити прийнятну якість перегляду. Причина в тому, що, хоча весь відеоряд має роздільну здатність 4K, FoV становить лише одну п'яту або шосту частину цієї роздільної здатності, і користувачі дивляться майже відеоконтент 720P. Внаслідок цього така передача вимагає в чотири-п'ять разів більше пропускної здатності порівняно з традиційною. Згідно з Netflix, для потокової передачі відео з роздільною здатністю 4K швидкість завантаження повинна досягати принаймні 25 Мбіт/с. Однак глобальна швидкість мобільного зв'язку становила лише 8,7 Мбіт/с у 2017 році і буде 20,4 Мбіт/с у 2021 році.

Нещодавно, у травні 2015 року була запропонована нова версія протоколу HTTP, відома як HTTP/2, яка має бути більш продуктивною альтернативою HTTP/1.1. Вона вводить нові функції, включаючи серверний push, пріоритет потоку та завершення потоку. Серверний push дозволяє клієнту отримувати кілька об'єктів, відправляючи лише один запит. Функція пріоритету потоку дозволяє клієнту запитувати сервер витратити більше ресурсів на відправку даних у одному потоці, ніж у інших. Крім того, потоки можуть бути завершені, якщо клієнт надсилає фрейм RST_STREAM, щоб увімкнути функцію завершення потоку.

Хоча 360-градусні відео VR здобувають значну увагу у науковій спільноті, потокова передача цього передового відео наразі здійснюється просто послідовною передачею плиток у сцені після вибору бітової швидкості

для них. Більшість поточних підходів базуються на HTTP/1 і не були належним чином оцінені в літературі. Серед існуючих досліджень попереднє дослідження є першим, що використовує нові функції HTTP/2. У цьому дослідженні плитки в одній і тій же області завантажуються послідовно, тому клієнт повинен чекати, доки всі ці плитки будуть відобразити всю область, що призводить до великої початкової затримки.

У новій статті пропонується новий адаптивний метод потокової передачі через HTTP/2 для вирішення вищезазначених проблем. Ми фокусуємося на покращенні якості відтворення (QoE). QoE можна оцінити на основі початкової затримки, середньої тривалості буферизації та частоти буферизації. Для зменшення початкової затримки плитки кодуються за допомогою закритих GOPs. З використанням функцій мультиплексування потоків та пріоритету потоків ми гарантуємо, що GOPs з однаковим індексом всіх плиток отримуються клієнтом одночасно. Ми також використовуємо функції пріоритету та завершення, щоб впоратися зі зміною пропускної здатності з метою мінімізації частоти буферизації.

У цьому методі загальний процес запиту частини медіа виглядає наступним чином. Спочатку клієнт вибирає бітову швидкість для кожної плитки на основі відносного положення та FoV. Потім клієнт відправляє кілька запитів одночасно, щоб одночасно отримати плитки від сервера. Функція пріоритету потоку використовується для контролю часу прибуття кожної плитки до клієнта. Ця функція дозволяє клієнту не тільки встановлювати очікуване співвідношення ресурсів для одночасного завантаження кожної плитки, але й отримувати плитку після повного завантаження іншої. У випадку раптового падіння пропускної здатності клієнт може використовувати фрейм RST STREAM, щоб запросити сервер зупинити відправку пізніх плиток.

Експериментальні результати в умовах змінної пропускної здатності показують, що наш метод є ефективним з меншою кількістю пропущених дедлайнів GOP, ніж інші методи посилянь, коли пропускна здатність різко знижується. Решту статті організовано наступним чином. У розділі II ми

надаємо огляд потокової передачі відео через HTTP/2 та пов'язану роботу. Розділ III презентує наш запропонований метод. Експериментальні результати та обговорення подано у розділі IV. Наприкінці, розділ V узагальнює цю роботу.

2.3 Потокова передача відео 360 градусів VR

Останнім часом потокова передача 360 VR на основі плиток розглядається як перспективне рішення проблеми високих вимог до пропускної здатності, у якому відео розділяється на кілька плиток за стандартом HEVC/H.265. Для підтримки адаптивної потокової передачі кожна плитка кодується на багатьох рівнях якості, які частково розбиваються на кілька сегментів. Слід зазначити, що плитки, що охоплюють панорамний вид, можуть бути закодовані і декодовані незалежно одна від одної. На рис. 2.1 показана архітектура системи потокової передачі 360 VR на основі плиток. На основі передбачуваного відображення та методу адаптації бітової швидкості клієнт відправляє запити на сервер, отримує плитки та зберігає їх у буфері для декодування.

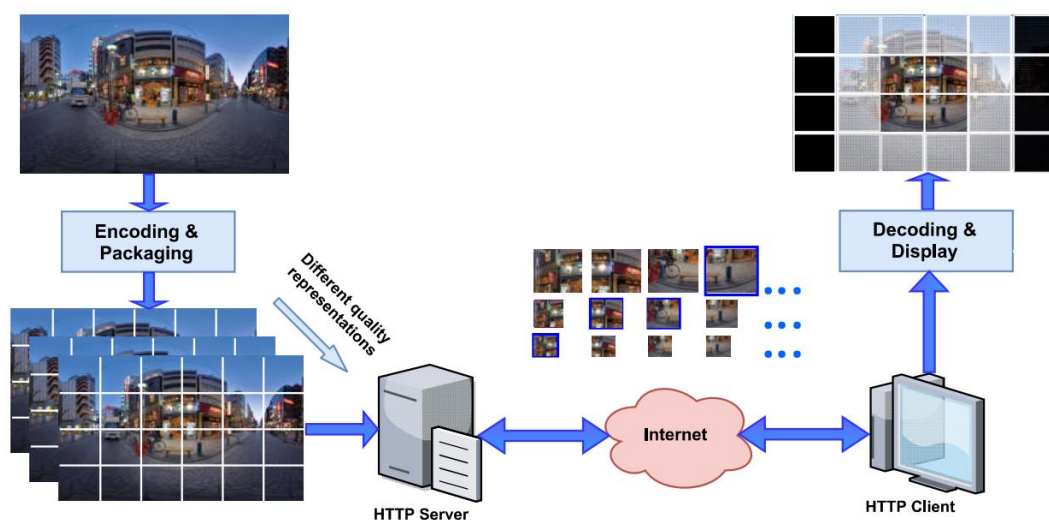


Рисунок 2.1 □ Архітектура потокової системи VR

Було запропоновано багато методів адаптивної потокової передачі на основі плиток. Petrangeli et al. використовують H.265 для часткового розділення відео та HTTP/2, щоб уникнути великої кількості запитів через

розділення на плитки. У цьому підході спочатку обирається найнижчий рівень якості для всіх плиток. Потім клієнт оновлює доступну пропускну здатність та розподіляє її для групи плиток з найвищим можливим рівнем якості. Цей процес повторюється, поки у клієнта не закінчиться пропускна здатність. В їхньому методі використовується тільки підхід k-push HTTP/2 без функцій пріоритету потоку та завершення потоку. Подібно до першого кроку Petrangeli, запропонований метод Feuvre et al. спочатку призначає найнижчий рівень для всіх плиток. Однак вони поступово збільшують рівень якості плиток з вищими пріоритетами та отримують плитки послідовно від сервера. M. Graf et al. пропонують дві стратегії потокової передачі: часткова доставка та повна доставка. Стратегія часткової доставки завантажує лише видимі плитки з найвищим можливим рівнем версії, що призводить до чорного екрану, коли відображення не передбачено точно. Тим часом у стратегії повної доставки плитки за межами відображення призначаються найнижчим рівнем якості, а решта — найвищим можливим рівнем якості. Однак кожен запит використовується для завантаження лише однієї плитки, що легко призводить до перевантаження мережі через велику кількість плиток у панорамному вигляді. Система потокової передачі в реальному часі для омнідирекційного відео представлена Ochi et al., в якій запитуються дві плитки, що охоплюють весь вид. Так званий цікавий плитка, яка охоплює FoV, запитується клієнтом з вищою якістю версії, тоді як плитка зображення, що охоплює всю панораму, автоматично надсилається клієнту з меншою якістю. Цей підхід може зменшити перевантаження мережі, використовуючи лише один запит, але завантаження лише однієї цікавої плитки для відображення може все ще споживати багато пропускну здатності. Mariem Ben Yahia et al. в пропонують підхід, використовуючи функції пріоритету та завершення HTTP/2 для запиту відеоданих на рівні кадру. У їхньому дослідженні функція пріоритету використовується для організації відеокадрів відповідно до структури GOP. Вони також збільшують ваги пріоритету найважливіших відеокадрів, що допомагає збільшити ймовірність того, що ці кадри можуть бути декодовані та

відображені вчасно. Крім того, використовуючи функцію завершення, клієнт може завершити кадри, які менш важливі або не мають шансів прибути вчасно. Однак вони не надають детального алгоритму щодо завершення та призначення ваги пріоритету. Використання кожного потоку HTTP/2 для доставки лише одного відеокадру може призвести до перевантаження мережі через велику кількість потоків, які потрібно відкрити за секунду.

У цій роботі було зроблено фокус на доставці 360 VR відео за допомогою мультимплексування, функцій пріоритету потоку та завершення потоку HTTP/2, а не на передбачуваному відображенні та декодуванні. Крім того, для низької затримки буде враховано кількість GOP у буфері, а не кількість сегментів, як у попередній роботі.

2.4 Пріоритезація та завершення HTTP/2

Р. Гуйсегемс та ін. демонструють декілька функцій HTTP/2, включаючи мультимплексування та серверний push. Проте, останнім часом було проведено багато досліджень щодо серверного push у потоковій передачі відео, тоді як мультимплексування не розглядалося. Завдяки мультимплексуванню, HTTP/2 дозволяє передавати кілька потоків даних через одне з'єднання TCP. Ці потоки відрізняються за типом даних, важливістю та розміром даних. Функція пріоритету HTTP/2 дозволяє серверу розподіляти свої ресурси на основі рівня пріоритету кожного потоку. Клієнт призначає рівень пріоритету нового потоку, вставляючи інформацію про пріоритет, включаючи залежність потоку та вагу пріоритету в заголовок потоку. Залежність потоку вказує на інший потік, якому мають бути призначені ресурси перед цим потоком. Потоки, що залежать від одного і того ж потоку, повинні отримувати ресурси пропорційно їх вазі пріоритету. Варто зазначити, що значення ваги пріоритету є цілим числом від 1 до 256. Клієнт може змінювати інформацію про пріоритет, надсилаючи фрейм PRIORITY в будь-який час.

Крім того, HTTP/2 також надає функцію завершення за допомогою фрейму RST STREAM. Надсилаючи цей тип фрейму, клієнт або сервер може завершити зазначений потік. Проте, надсилаючий вузол повинен отримати додаткові DATA фрейми від припиненого потоку, відправлені віддаленим вузлом, перед прибуттям фрейму RST STREAM.

2.5 Метод вирішення

2.5.1 Розподіл пріоритетних параметрів

Symbol	Description
SD	Segment duration
GD	Total duration of all frames in a GOP
β_{init}	Initial buffer
R_i	Region i^{th} of video view
N_i^R	The number of tiles in region i^{th}
K	The number of regions
τ_k^j	Tile j^{th} of region k^{th}
w_i	Priority weight of region i^{th}
r_i	Rate of region i^{th}
T_i^{est}	Estimated throughput of segment i^{th}
T_i^s	Smoothed throughput of segment i^{th}
T_{inst}	Instant throughput
t_{inst}	Instant download time of the current segment
D_{inst}	Instant data size downloaded of the current segment
$d_{GOP}(\tau_k^j)$	GOP data length of tile τ_k^j of the current segment

Рисунок 2.2 □ Використані позначення

У запропонованому методі видеоперегляд буде розділений на K регіонів: $\{R_i \mid i = 1, 2, \dots, K\}$. Плитки в регіоні мають однакову бітову швидкість, і чим менше значення i , тим вище важливість регіону. Алгоритм розподілу швидкості, використаний у цій роботі, такий самий, як у нашій попередній роботі [9]. Ми розглядаємо два випадки для визначення оціненої пропускної здатності: випадок збільшення і випадок зменшення. У першому випадку, коли пропускна здатність схильна зростати, оцінена пропускна здатність обчислюється методом згладжування пропускної здатності, щоб уникнути короткочасних коливань мережі. У випадку зменшення, щоб впоратися зі зниженням пропускної здатності, пропускна здатність останнього сегмента вважається оціненою пропускною здатністю.


```

for  $k = 0 \rightarrow K - 1$  do
  if  $k == 0$  then
     $w_k \leftarrow 256$ ;
  else
    if  $T_{i-1}^{inst} > T_i^{inst}$  and  $k == K - 1$  then
       $w_k \leftarrow 1$ ;
    else
       $w_k \leftarrow \lceil 256 * \frac{r_k}{r_0} \rceil$ ;
  for  $j = 0 \rightarrow N_k^R - 1$  do
    set_priority(stream dependency,  $w_k$ );
    send a request for tile  $\tau_k^j$ ;

```

Рисунок 2.3 □ Алгоритм 1. Розподіл пріоритетних параметрів

Рисунок 2.2 надає деякі позначення, використані в наступному обговоренні. Щоб гарантувати, що всі плити прибудуть до клієнта одночасно, ми встановлюємо вагу пріоритету кожного потоку плиток HTTP/2 так, щоб задовольнялася умова: параметри ваги плиток пропорційні їх бітовим швидкостям. Проте, у випадку зменшення, всі плити (K - 1)-го регіону отримують вагу пріоритету 1. Алгоритм розподілу параметрів пріоритету представлений в Алгоритмі 1.

Функція `set_priority()` використовується для встановлення параметрів пріоритету для кожного запиту. Потрібно призначити два параметри: залежність потоку та вагу пріоритету, з їхнім значенням, зазначеним у розділі 2. У Алгоритмі 1 ми встановлюємо залежність потоку для всіх потоків як ідентифікатор неактивного потоку, який не використовується для передачі будь-яких користувацьких даних.

2.5.2 Зміна пріоритетів і припинення потоку

Коли клієнт завантажує сегмент, раптове зниження пропускної здатності непередбачуване. Це може призвести до перезавантаження, оскільки одночасно треба транспортувати багато плиток, і ми не можемо гарантувати,

що всі GOP (Group of Pictures) будуть декодовані до крайнього терміну в умовах слабкої мережі. Щоб впоратися з цією проблемою, більш важливі плитки будуть завантажені перед залишковими, і вони можуть отримати всі мережеві ресурси. У найгіршому випадку деякі GOP важливих плиток можуть бути отримані на клієнті для відображення. Ми розподіляємо плитки на два набори, позначені як Ω_1 та Ω_2 . Ω_2 - це набір плиток, які тимчасово мають бути заморожені, поки всі плитки в Ω_1 не будуть повністю завантажені. У випадку, коли $wK-1 = 1$, ми встановлюємо початкове значення Ω_2 як весь $(K - 1)$ -й регіон. Коли транспортується сегмент, плитка в Ω_1 може перейти в Ω_2 , якщо умова $|\Omega_1| \geq N R_0$ виконується після переміщення. З кожною перемиканою плиткою клієнт надсилає фрейм PRIORITY зі залежністю потоку, що встановлено на ідентифікатор потоку будь-якої плитки в R_0 .

```

 $T_{inst} \leftarrow \frac{D_{inst}}{t_{inst}};$ 
while true do
    if  $|\Omega_1| \leq N_0^R$  then
        return;
     $N_{miss} \leftarrow 0;$ 
    for each  $\tau_k^j \in \Omega_1$  do
        for  $g' = g + 1 \rightarrow \lceil \frac{SD}{GD} \rceil$  do
            calculate  $t_{GOP}^e(i, g', \tau_k^j);$ 
            if  $t_{GOP}^e(i, g', \tau_k^j) > t_{GOP}^d(i, g', \tau_k^j)$  then
                 $N_{miss} \leftarrow N_{miss} + 1;$ 
        if  $N_{miss} > \alpha \times |\Omega_1| \times \lceil \frac{SD}{GD} \rceil$  then
            select  $\tau_k^j;$ 
             $\Omega_1 \leftarrow \Omega_1 \setminus \{\tau_k^j\};$ 
             $\Omega_2 \leftarrow \Omega_2 \cup \{\tau_k^j\};$ 
            send a PRIORITY frame for tile  $\tau_k^j;$ 
    else
        return;

```

Рисунок 2.4 □ Алгоритм 2. Зміна залежності потоку

Деталі алгоритму перерозподілу представлені в Алгоритмі 2, де N_{miss} - загальна кількість GOP, для яких передбачається пропущений дедлайн. Цей

алгоритм виконується кожного разу, коли клієнт отримує GOP в Ω_1 . Плитка τ_{jk} , для якої буде змінена залежність потоку, випадковим чином вибирається серед плиток, що відповідають наступній умові.

Крім того, клієнт припиняє роботу деяких плиток у Ω_2 , якщо він отримав усі плитки в Ω_1 . Плитка τ_{jk} буде припинена, якщо всі її значення $te_{GOP}(i, g_0, \tau_{jk})$ більше, ніж $td_{GOP}(i, g_0, \tau_{jk})$. Клієнт обробляє це, відправляючи фрейми RST STREAM для потоків, які передають припинені плитки.

2.6 Результати експерименту

2.6.1 Налаштування експерименту

Система тестування складається з трьох компонентів: клієнта, сервера та IP-мережі (рис. 2.9). Як клієнт, так і сервер використовують відкрите програмне забезпечення nghttp2, яке працює на операційній системі Ubuntu 14.04 LTS. Для контролю за умовами мережі в клієнті встановлено інструмент Dummynet, з RTT встановленим на рівень 50 мс.

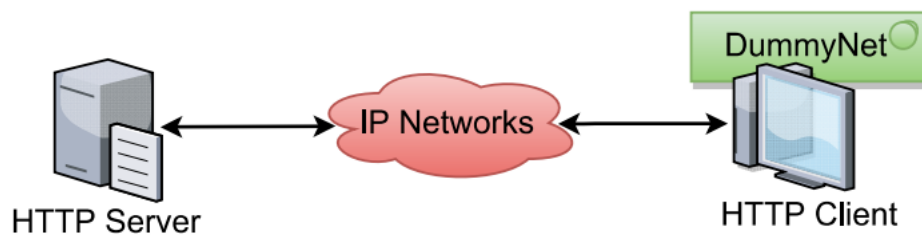


Рисунок 2.5 □ Архітектура випробувального стенда

Відео в експерименті використовує проекцію куб-мапи, в якій частота кадрів становить 30 кадрів за секунду, а розмір GOP - 5. Вміст кодується стандартом HEVC за допомогою FFmpeg. Тривалість відео - 400 с. Відображення відео розділено на 6 плиток, що відповідають 6 граням. У цьому розділі ми використовуємо поняття грані замість плитки. У нас є три регіони

($K = 3$), і через те, що ми зосереджені на передачі відео, ми припускаємо, що користувач не рухає головою під час перегляду відео.

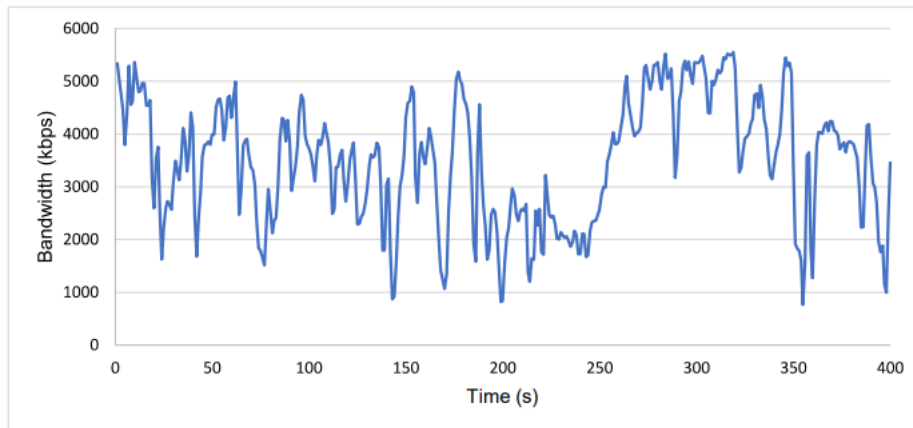


Рисунок 2.6 □ Пропускна здатність, яка використовується в експериментах

Регіон 1 містить лише передню грань (грань 0), регіон 2 містить 4 грані, що оточують область перегляду, а останній регіон має лише задню грань (грань 5). Кожна грань розділена на часові сегменти з тривалістю 5 с ($SD = 5$ с) і з версіями бітрейту у наборі $\{133, 167, 250, 333, 500, 667, 833, 1000\}$ Кбіт/с. Клієнт почне відтворення лише тоді, коли його буфер містить принаймні 6 GOP кожної грані. Це означає, що початковий буфер становить 1 с. Після цього клієнт декодує та відтворює кожні $1/6$ с. Інші параметри в алгоритмах встановлені так: $\gamma = 0.125$, як рекомендується в [22], та $\alpha = 16$. Потужність мережі, яку ми використовували в наших експериментах, зображена на рис. 3, і отримана з мобільної мережі.

2.6.2 Експериментальні результати

У цьому розділі метод, запропонований як пріоритет та припинення, порівнюють з двома іншими методами: без пріоритету, названим без пріоритету, та пріоритетом без припинення, названим пріоритетом. У цій роботі ми зосереджуємося на здатності уникнути перебуферизації, а не на порівнянні отриманого бітрейту серед методів. Отже, для всіх методів буде використано один і той же алгоритм призначення бітрейту.

Порівняння в кількості GOP кожного обличчя в кожному методі подано на рис. 4. Він показує, що два методи, які використовують функцію пріоритету, обробляють обличчя більш рівномірно, ніж метод без пріоритету. З рисунку 4а видно, що через те, що клієнт не використовує функцію пріоритету, кількість GOP в обличчі 5 набагато вища, ніж в інших обличчях, тоді як обличчя 5 не видно. Причина в тому, що в цьому випадку сервер рівно обробляє кожний потік даних, тому з найнижчим бітрейтом сегменту обличчя 5 може бути передано найбільше GOP порівняно з обличчями в R1 і R2 за той же період часу. Крім того, кількість GOP обличчя 0 різко зменшується до приблизно 0 GOP у буфері у проміжку часу від 73.251 с до 75.819 с.

З іншого боку, методи, які використовують функцію пріоритету, як показано на рис. 4b та 4с, мають майже однакову кількість GOP кожного обличчя. Важливою перевагою цих методів є те, що кількість GOP обличчя 0 в буфері не різко зменшується, коли пропускна здатність знижується, як на рис. 5 о 71 с. У цьому випадку, коли клієнт сприймає падіння пропускної здатності, він відправляє фрейми пріоритету, щоб встановити потік обличчя 0 як батьківський потік для менш важливих обличчям, що означає, що потоки завантаження цих обличчям тимчасово заморожені для збереження пропускної здатності для потоку обличчя 0. Переваги функції припинення можна побачити на рис. 4с, припинення непотрібних GOP заощаджує пропускну здатність для завантаження обличчя 0 наступного сегмента. У результаті кількість GOP обличчя 0 залишається на високому рівні і досягає мінімуму в 1 GOP на 76.267 с.

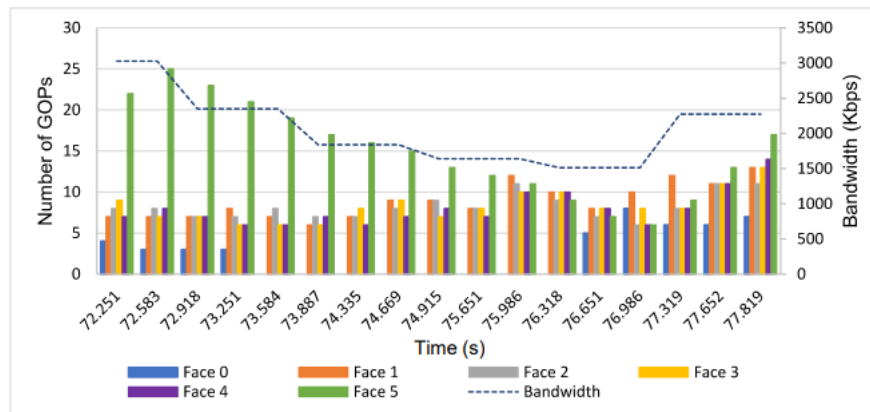


Рисунок 2.7 □ Метод без пріоритетів

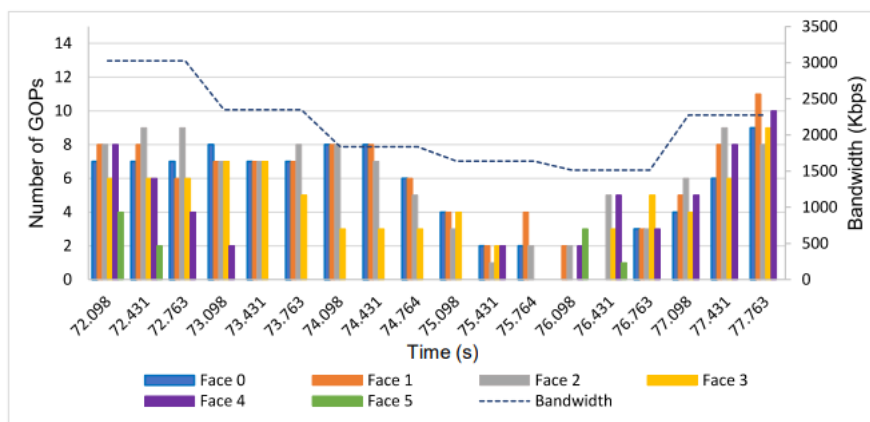


Рисунок 2.8 □ Метод з використанням алгоритму пріоритетів

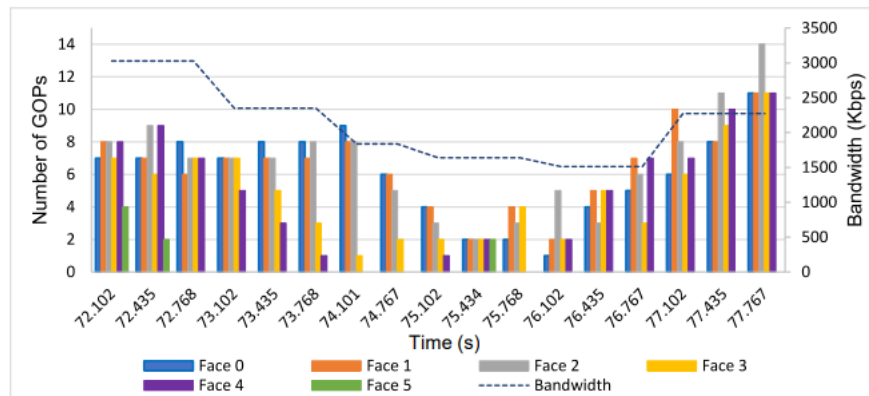


Рисунок 2.9 □ Метод з використанням алгоритму пріоритетів та алгоритму завершення

На рисунках 2.11, 2.12, 2.13 порівнюється кількість GOP в обличчі 0, отриманих клієнтом в трьох випадках. Можна побачити, що при зниженні

пропускної здатності спостерігається значне зменшення кількості GOP (від 7 GOP о 71 с до 0 GOP о 75 с), якщо клієнт не використовує функцію пріоритету.

Навпаки, інші два методи дають кращі та схожі результати, при яких цифри майже залишаються незмінними близько 8 GOP і зменшуються до 4 GOP о 75 с. Крім того, у методі пріоритету не має жодного GOP обличчя 0 протягом 3 разів о 76.069 с, 76.264 с і 76.431 с. І завжди є GOP цього обличчя у методі пріоритету та припинення, тоді як кількість разів, коли метод без пріоритету не має GOP обличчя 0, становить 13.

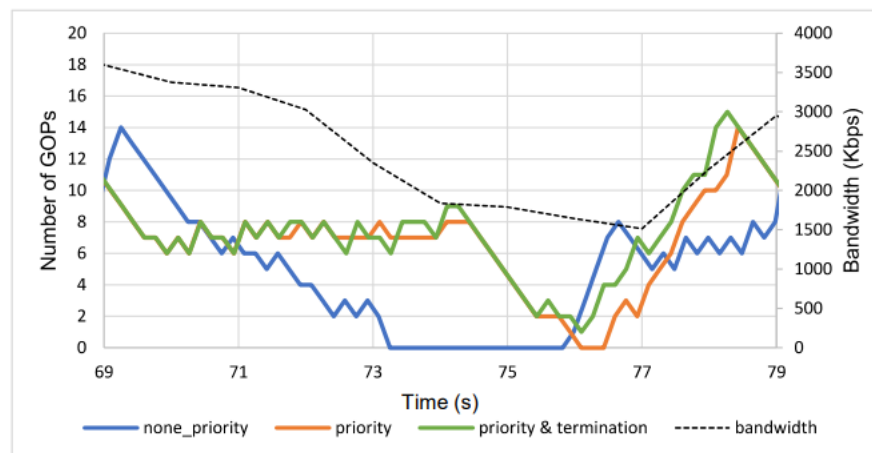


Рисунок 2.10 □ Кількість GOP в передній грані трьома методами

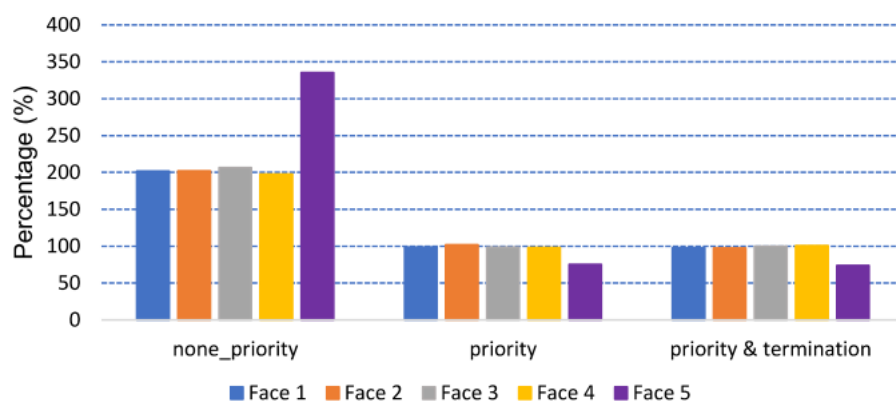


Рисунок 2.11 □ Середнє співвідношення кількості GOP в інших гранях до передньої грані

Для точності кожен метод запускається 10 разів, і результатом є середнє значення. На рисунку 6 показано середнє співвідношення між кількістю GOP облич (1-5) і кількістю GOP обличчя 0 у клієнта, яке обчислюється кожен раз під час відображення. Слід зазначити, що ми розглядаємо лише випадки, коли кількість GOP у обличчі 0 більше 0. Загалом, це співвідношення у методі без пріоритету вище, ніж у інших, і майже перевищує 200%, особливо співвідношення обличчя 5 становить близько 340%, що призводить до витрати пропускної здатності. Натомість, два методи, які використовують функцію пріоритету, отримують кращий результат, приблизно 100% для облич (2-4) і 70% для обличчя 5. Тому клієнт може декодувати і відтворювати GOP до крайнього терміну та не витрачає пропускну здатність на завантаження неважливих GOP.

У цій роботі ми також розглядаємо вплив початкового буфера на результати нашого методу. Проводяться експерименти з трьома методами з трьома різними значеннями початкового буфера: 0.5 с, 1 с, 2 с. На рисунку 7 показано співвідношення між кількістю пропущених GOP обличчя 0 і загальною кількістю GOP у всьому відео. Очевидно, що функції припинення та пріоритету приносять вражаючі результати. Співвідношення пропущених GOP у методі без пріоритету набагато вище, ніж у інших методах. Ці співвідношення у методі пріоритету становлять 0,06%, 0,42% і 0,75% для початкового буфера 0,5 с, 1 с, 2 с відповідно. У методі пріоритету та припинення це співвідношення практично дорівнює 0 для всіх значень початкового буфера. Ця перевага є важливою для контексту з низькою затримкою.

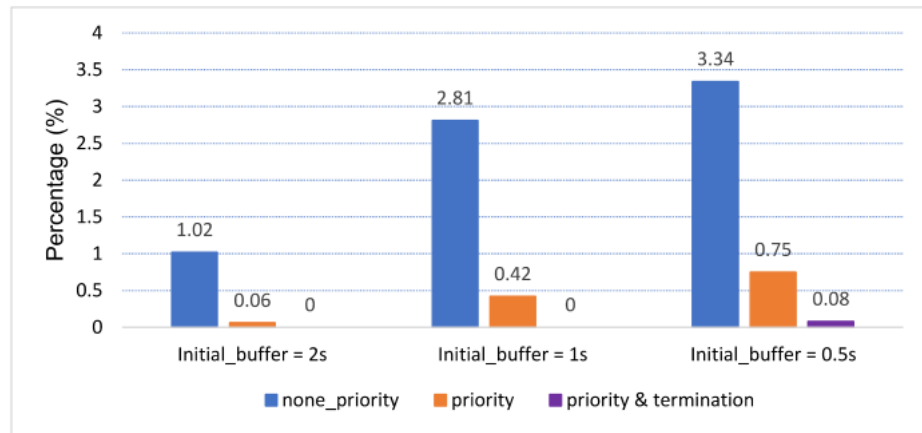


Рисунок 2.12 □ Відсоток GOP пропущених термінів на початку обличчя трьох методів з різними початковими значеннями буфера

Висновки до розділу 2

У 2 розділі була запропонована нова система стрімінгу на основі плиток для 360-градусного відео у віртуальній реальності, де плитки завантажуються паралельно, що дозволяє клієнту декодувати та відтворювати відео, як тільки отримано принаймні певну кількість GOP усіх плиток. Крім того, нова система стрімінгу з низьким буфером допомагає мінімізувати початкову затримку, коли функції пріоритету та завершення потоку HTTP/2 використовуються в клієнті. Призначення пріоритету для кожної плитки на основі їх бітрейтів дозволяє отримати та декодувати GOP всіх плиток паралельно. Щоб впоратися зі змінами пропускної здатності під час стрімінгу, можна змінювати залежність потоку, відправляючи фрейми пріоритету, а високопотенційні втрачені GOP можна припинити, коли клієнт відправляє фрейми RST STREAM. Отримані експериментальні результати показують, що наш запропонований метод значно зменшує кількість пропущених дедлайнів GOP плиток, що покривають область перегляду, і збалансовує пропорцію GOP плиток, що містяться в буфері.

3 **МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ СИСТЕМИ, ЩО РОЗРОБЛЯЄТЬСЯ**

Починаючи з визначення вимог до апаратного забезпечення (1 розділ)

2.2 Проєктування апаратного забезпечення

Для апаратного забезпечення буде потрібно кластер з серверів баз даних та серверів на яких буде розгорнуто кластери з бекенд застосунками. Для хостінгу баз даних та бекенд застосунку було обрано рішення від компанії AWS, а саме сервіс RDS для баз даних та EC2 для бекенд застосунку, тому що він надійний та перевірений роками на багатьох проєктах.

2.2.1 Кластери баз даних на AWS RDS

AWD RDS (Advanced Waveform Digitizer with Real-time Digital Signal Processing) - це система з передовим цифровим аналізатором хвиль, яка використовується для захоплення, обробки та аналізу сигналів у реальному часі.

Основні принципи та використання AWD RDS включають:

Захоплення даних. AWD RDS може захоплювати сигнали з різних джерел, таких як антени, сенсори або прилади вимірювання.

Аналіз сигналів. Цифровий сигнал обробляється в реальному часі для виявлення відповідних параметрів, таких як частота, амплітуда, фаза та інші характеристики.

Цифровий сигнальний обробник (DSP). AWD RDS має вбудовані функції цифрового сигнального обробника, які дозволяють виконувати різноманітні обчислення та аналізувати сигнали безпосередньо в цифровій формі.

Реальний час. Однією з основних переваг AWD RDS є можливість аналізувати сигнали в реальному часі, що дозволяє операторам швидко реагувати на зміни у сигналах або навколишньому середовищі.

Висока точність та чутливість. Завдяки цифровому аналізу та високій швидкості обробки сигналів AWD RDS може досягати високої точності та чутливості при вимірюваннях.

Множинні застосування. AWD RDS може бути використаний у різних областях, включаючи телекомунікації, медичні дослідження, наукові дослідження, вимірювання в галузі електроніки та інші.

Загалом, AWD RDS є потужним інструментом для аналізу сигналів у реальному часі з високою точністю та швидкістю обробки, що робить його корисним для широкого спектра додатків у сучасних технологіях.

Створення кластера баз даних на різних регіонах на Amazon Web Services (AWS) може бути складним, але AWS надає зручні інструменти для реалізації цієї задачі. Нижче я наведу детальний опис кроків, які потрібно виконати для створення кластера баз даних на AWS з використанням Amazon RDS (Relational Database Service).

Перш ніж розпочати, вам потрібно ретельно спланувати кластер баз даних. Визначте, які бази даних ви хочете використовувати, їх типи (наприклад, MySQL, PostgreSQL, SQL Server тощо), потрібний рівень доступності, масштабованість і потреби в продуктивності.

Для цього потрібно:

- увійти до облікового запису на AWS та перейти до консолі керування AWS.
- натиснути на Amazon RDS, вибрати Amazon RDS зі списку доступних сервісів.

Створіть новий кластер баз даних, для цього:

- Натисніть кнопку "Create database".
- виберіть тип бази даних, який вам потрібен (MySQL, PostgreSQL, SQL Server тощо), у нашому проєкті буде використано PostgreSQL.
- визначте параметри кластера, такі як розмір і тип інстансів, обсяг пам'яті, мультизональність тощо.

- виберіть відповідний регіон для кожної зони баз даних (primary і secondary).

Конфігурація бази даних:

- введіть назву бази даних, ім'я для користувача та пароль та налаштуйте права користувача.
- встановіть інші параметри, такі як автоматичне резервне копіювання, моніторинг тощо.

Створення кластера:

- натисніть кнопку "Create database", щоб розпочати процес створення кластера баз даних.

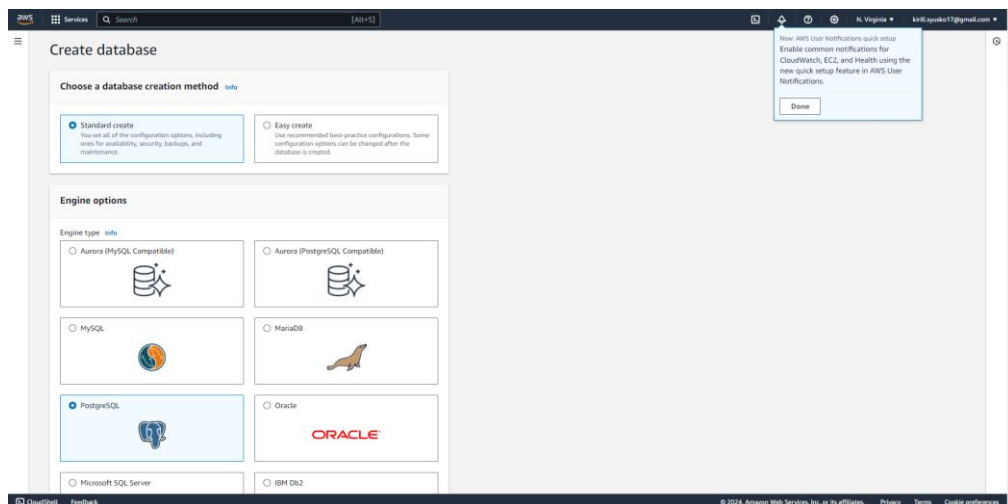


Рисунок 3.1 □ Веб інтерфейс AWS RDS

Далі налаштування та відкриття доступу:

Визначте, хто із зовнішніх користувачів або служб має доступ до вашого кластера баз даних. Налаштуйте правила безпеки групи параметрів RDS, щоб відкрити необхідні порти для з'єднань. Налаштування мережевих налаштувань. Для цього визначте, які області IP або підмережі можуть отримати доступ до вашого кластера баз даних. Встановіть правила мережевих доступів відповідно до ваших потреб безпеки.

Після встановлення прав потрібно перевірити з'єднання до кластера. Використовуйте відповідний інструмент для з'єднання з вашим кластером баз

даних. Також потрібно налаштувати моніторинг. Включіть моніторинг і налаштуйте сповіщення для спостереження за станом вашого кластера баз даних.

На кінець потрібно оптимізувати продуктивність. Для цього використовуйте інструменти моніторингу для ідентифікації можливих проблем продуктивності та оптимізації роботи кластера баз даних.

Резервне копіювання та відновлення. Періодично створюйте резервні копії баз даних та перевіряйте процес відновлення для забезпечення надійності даних.

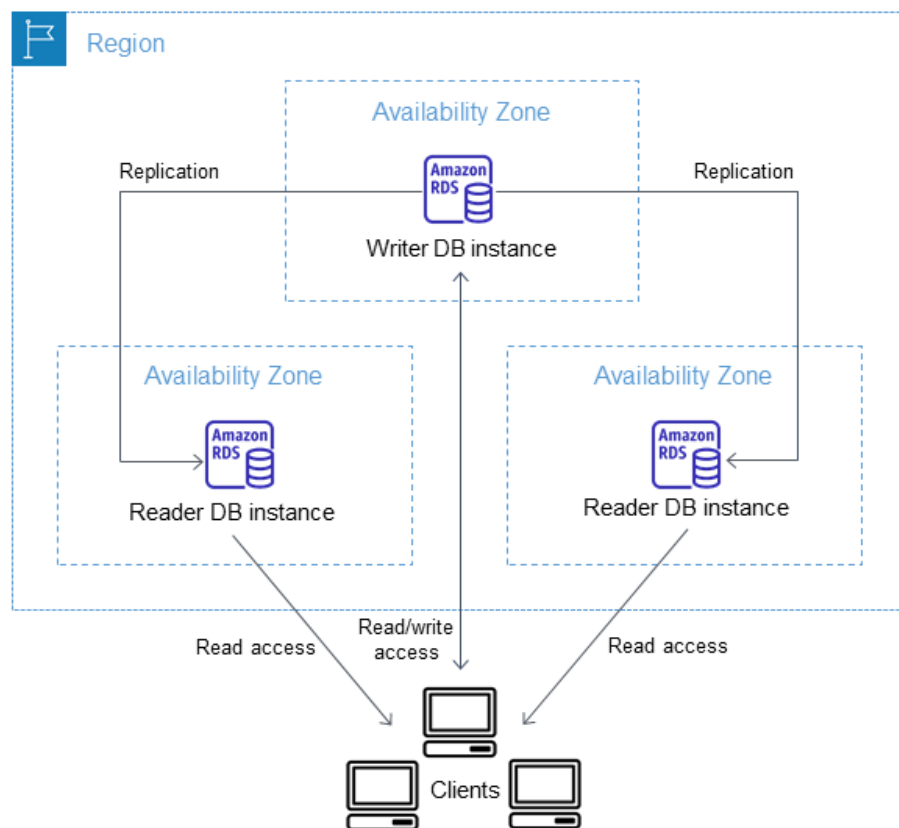


Рисунок 3.2 □ Архітектура кластерів баз даних AWS RDS

У PostgreSQL існує кілька типів реплікації, які дозволяють створювати резервні копії даних та забезпечують високу доступність системи. Ось декілька основних типів реплікації в PostgreSQL.

1. Логічна реплікація (Logical Replication).

Цей тип реплікації дозволяє відображати та реплікувати зміни з одного сервера PostgreSQL на інший за допомогою логічного журналу змін (WAL). Логічна реплікація надає більшу гнучкість, оскільки дозволяє вибирати, які таблиці та які стовпці потрібно реплікувати. Вона дозволяє використовувати різні версії PostgreSQL між майстер-та реплікаційним серверами.

2. Фізична реплікація (Physical Replication).

Фізична реплікація передає всі зміни, які відбуваються на майстер-сервері, на одну або більше реплікаційних копій. Вона копіює дані на рівні фізичних файлів бази даних. Це зазвичай більш ефективний спосіб реплікації для більших обсягів даних, оскільки не вимагає додаткових операцій з логічним журналом.

3. Реплікація за допомогою спільного кластера файлів (Shared-Nothing Replication).

Цей підхід передбачає налаштування кількох незалежних серверів PostgreSQL, кожен з яких має свою власну копію даних. Кожен сервер призначений для обробки певної частини запитів, що дозволяє розділити навантаження між серверами і підвищити продуктивність. Ці типи реплікації можуть бути використані окремо або в комбінації для забезпечення потреб вашої системи відмовостійкості, швидкодії та доступності даних. Кожен з них має свої переваги та обмеження, тому вибір конкретного типу реплікації залежить від вашого випадку використання та потреб вашої інфраструктури баз даних.

Для даного проєкту було обрано фізичну реплікацію з наступних причин:

Ефективність та продуктивність. Фізична реплікація передає дані на рівні фізичних файлів бази даних. Це означає, що весь обсяг даних копіюється безпосередньо, без необхідності перекладу логічних змін. Це може

забезпечити кращу швидкість та продуктивність, оскільки операції реплікації менше впливають на майстер-сервер.

Простота налаштування та управління. Фізична реплікація зазвичай вимагає менше конфігурацій та налаштувань порівняно з логічною реплікацією. Вона часто працює на рівні операційної системи та файлової системи, що дозволяє простіше управляти та налагоджувати реплікаційні процеси.

Масштабованість та обсяг даних. Фізична реплікація зазвичай є більш практичним варіантом для обсягів даних великого масштабу. Вона дозволяє ефективно копіювати великі обсяги даних без значного збільшення навантаження на мережу чи обробку даних.

Висока відмовостійкість. Фізична реплікація забезпечує високий рівень доступності та відмовостійкості, оскільки репліка негайно може прийняти роль майстра у випадку відмови основного сервера.

Отже, фізична реплікація вибрана для даного проєкту з огляду на її ефективність, простоту налаштування, масштабованість та надійність, що відповідає вимогам проєкту та потребам в реплікації даних.

2.2.2 Кластери серверів на AWS EC2

AWS EC2 - це абревіатура, що означає Amazon Web Services Elastic Compute Cloud. Це одна з послуг обчислювального хмарного обчислення, яку надає Amazon Web Services (AWS).

AWS EC2 дозволяє користувачам орендувати віртуальні сервери на хмарній інфраструктурі AWS. Він надає користувачам масштабовані обчислювальні ресурси у вигляді віртуальних машин, які можна запускати та зупиняти за необхідності. Такі сервери можна налаштовувати під конкретні вимоги щодо обсягу ресурсів, операційної системи, мережі та інших параметрів.

Основні переваги використання AWS EC2 включають:

Гнучкість. Користувачі можуть легко масштабувати свої обчислювальні ресурси вгору або вниз в залежності від змін потреб.

Оплата за використання. Вартість використання AWS EC2 базується на реальному часі роботи серверів. Користувачі платять лише за той час, коли їх віртуальні машини запущені і працюють.

Резервування і простір на випадок відмови. AWS надає можливість резервування екземплярів EC2 на тривалий термін за зниженою ціною, а також забезпечує можливість автоматичного відновлення в разі відмови обчислювальних ресурсів.

Різноманітність образів. AWS пропонує широкий вибір готових образів операційних систем, що спрощує розгортання серверів.

Можливості керування. AWS EC2 надає різноманітні можливості керування віртуальними серверами за допомогою інструментів управління, API та консолі управління AWS.

Узагальнюючи, AWS EC2 є потужним інструментом для розгортання обчислювальних ресурсів в хмарному середовищі з великою гнучкістю, масштабованістю та оптимізацією витрат.

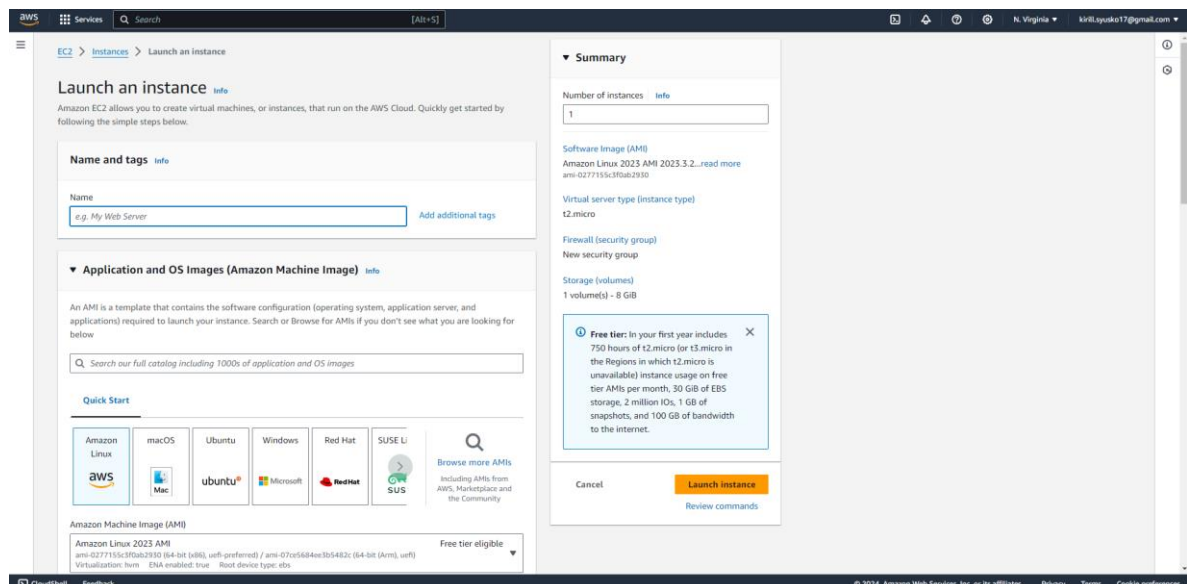


Рисунок 3.3 □ Веб інтерфейс AWS EC2

Створення кластеру серверів EC2 на різних регіонах AWS може бути складним процесом, але я основні кроки, які потрібно виконати такі. Потрібно

враховувати, що перед тим як розпочати, вам потрібно мати аккаунт AWS і налаштовані необхідні права доступу. Для створення кластеру в різних регіонах також потрібно буде врахувати перевірку доступності регіонів, обрання оптимальних типів інстансів, а також управління мережевими налаштуваннями.

Спочатку в консолі AWS було увійдено за допомогою облікового запису та перейдено до служби EC2. В консолі EC2 було створено ключ доступу, необхідний для з'єднання з інстансами.

Далі було обрано регіони, в яких планувалося створити кластер. Після аналізу доступності, затримок і вартості були обрані оптимальні регіони.

Потім були створені інстанси EC2 для кожного регіону, враховуючи потреби проекту та налаштування типу інстансу, обсягу пам'яті, обсягу сховища та інших параметрів.

Для кожного регіону були налаштовані правила безпеки та групи захисту для керування доступом до інстансів.

Далі була налаштована мережа і VPC, щоб забезпечити ізоляцію та керування трафіком між інстансами.

Після завершення налаштування було здійснено з'єднання з кожним інстансом за допомогою ключа доступу та SSH.

На завершення було розглянуто можливості автоматизації для управління кластером, а також встановлено системи моніторингу та управління для відстеження продуктивності і доступності інстансів.

Забезпечте резервне копіювання і захист даних: Розгляньте можливості резервного копіювання даних і захисту від відмов для вашого кластера.

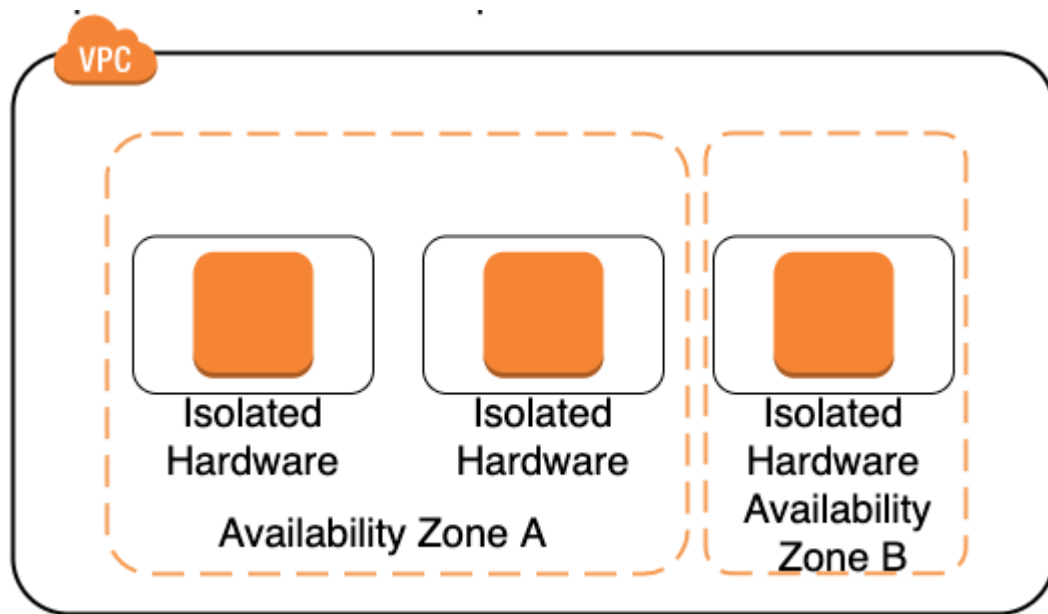


Рисунок 3.4 □ Архітектура кластерів AWS EC2

NVIDIA CloudXR - це технологія, яка дозволяє стрімінгове відображення VR та AR контенту з високою якістю на різноманітні пристрої через мережу. Для ефективного використання NVIDIA CloudXR вам знадобиться потужне залізо, особливо для обробки графічних даних.

GPU. Оскільки NVIDIA CloudXR використовує GPU для обробки графічних даних, вам знадобиться сервер або інстанс EC2 з сильним GPU від NVIDIA. Наприклад, GPU серії NVIDIA Tesla або NVIDIA Quadro можуть бути дуже ефективними для таких завдань.

VRAM. Важливо враховувати не лише тип і модель GPU, але і обсяг VRAM (відеопам'яті). Для обробки великих обсягів графічних даних вам знадобиться достатній обсяг VRAM.

Для тестування добре підійде GPU серії NVIDIA GeForce або NVIDIA Quadro з достатньою кількістю VRAM, наприклад, 6-8GB або більше. Це дозволить тестувати різні візуальні ефекти та обробляти графічні дані без значних обмежень.

Обчислювальна потужність CPU. Поміж усіх інших компонентів, також важливо мати достатньо потужний CPU, оскільки він впливає на загальну продуктивність системи, включаючи обробку даних та керування

обчисленнями. Для тестування буде достатньо середнього класу CPU, адже тестові завдання можуть бути менш обчислювально інтенсивними порівняно з продуктивними завданнями. Проте, важливо мати CPU достатньої потужності, щоб забезпечити стабільну роботу системи.

Мережева пропускна здатність. Важливо мати швидку і надійну мережу, оскільки NVIDIA CloudXR передає графічні дані по мережі. Велика пропускна здатність мережі допоможе забезпечити гладку передачу даних.

Оперативна пам'ять. Хоча обсяг оперативної пам'яті не є таким критичним, як GPU або CPU, достатній обсяг оперативної пам'яті все ж допоможе забезпечити більш гладке функціонування системи, особливо при роботі з великими обсягами даних. Для тестування можна розглядати 8-16GB оперативної пам'яті. Це дозволить запускати декілька програм одночасно та забезпечить плавну роботу системи під час тестування.

Системний диск. Хоча не так критично для тестування, рекомендується мати швидкий SSD диск для покращення часу завантаження та роботи системи в цілому.

2.2.3 Клієнтський пристрій

Для клієнтського пристрою підійде будь-який пристрій який підтримує технології VR. Це можуть бути будь-які окуляри від зі лінійки Oculus quest, rift і так далі. Для цього проєкту було використано VR окуляри Oculus quest 2 64GB, які було детально розглянуто розділі 1.3.1

2.3 Проєктування програмного забезпечення

Розробка програмного забезпечення для програмно-апаратних комплексів представляє собою складний процес. Перед початком розробки важливо детально продумати та описати структуру та алгоритм роботи цього програмного забезпечення.

Висновки до розділу 3

У цьому розділі було розглянуто важливі аспекти проектування апаратного та програмного забезпечення для реалізації програмно-апаратних комплексів. Аналізуючи вимоги до апаратного забезпечення, було обрано рішення з використанням кластерів серверів баз даних на AWS RDS та серверів на AWS EC2.

Розділ також розглянув різні аспекти налаштування кластерів баз даних та серверів на AWS, включаючи фізичну реплікацію для забезпечення високої доступності та ефективності обробки даних, а також особливості створення і налаштування інфраструктури за допомогою сервісів AWS EC2.

Крім того, було розглянуто вибір клієнтського пристрою для проекту, а також важливі аспекти проектування програмного забезпечення, зокрема розробку програмного забезпечення для програмно-апаратних комплексів.

Завершуючи, процес проектування апаратного та програмного забезпечення є критичним для успішної реалізації будь-якого проекту. Правильний вибір архітектури, налаштування і розгортання інфраструктури, а також розробка ефективного програмного забезпечення визначають стабільність, продуктивність і безпеку програмно-апаратних комплексів. Тому важливо приділити достатню увагу цим аспектам на кожному етапі проектування і розробки.

3 ДОСЛІДЖЕННЯ ТА ОТРИМАННЯ ВХІДНИХ ДАНИХ

У цьому розділі ми докладно розглянемо процес дослідження та збору вхідних даних, який відіграє ключову роль у подальшому аналізі та побудові серверної архітектури для потокової передачі VR. Наше дослідження розпочнеться з опису методики збору даних із прототипу пульсоксиметра. Ми розглянемо характеристики та параметри, які фіксуються під час вимірювання, а також проаналізуємо принципи роботи самого пристрою. Потім ми перейдемо до використання інструменту WEKA для аналізу зібраних даних та створення моделей машинного навчання. Ми також розглянемо процес розробки 3D-моделі дерева рішень, що дозволить нам наочно уявити структуру моделі та краще зрозуміти отримані результати. Наша мета полягає в тому, щоб систематизувати дані, виявити закономірності та тенденції, а також підготувати основу для подальшого аналізу та інтерпретації результатів дослідження.

3.2 Встановлення програм та пакетів для коректної роботи CloudXR

Для початку потрібно підготувати та налаштувати NVIDIA CloudXR Server. Для цього потрібно перевірити усі передумови для сервера CloudXR.

Список вимог:

1. Готова до віртуальної реальності система з графічним процесором NVIDIA, включно з графічним процесором Quadro, на основі архітектури графічного процесора NVIDIA Pascal™ або новішої.
2. Знаходиться під керуванням Windows 10, Windows 10 Server або Windows 2019 Server або новішої версії
3. Драйвер NVIDIA версії 440.97 або новішої
4. Microsoft, червень 2010 DirectX SDK.
5. Steam (не потрібно для SteamVR Enterprise)
6. SteamVR

7. Серверна частина NVIDIA CloudXR SDK
8. Програма OpenVR для потокової передачі на клієнта

3.3 Налаштування сервера CloudXR

Потрібно використати файл `Installer\CloudXR-Setup.exe`, щоб налаштувати сервер NVIDIA CloudXR. Коли буде запропоновано вибрати компоненти для встановлення, потрібно переконайтись, що було обрано лише серверну частину з розповсюдженими компонентами.

- 1) Встановлення Steam і SteamVR. (Steam не потрібен для версії SteamVR Enterprise.)
- 2) Щоб інсталювати програмне забезпечення NVIDIA CloudXR, запустіть файл `CloudXR-Setup.exe`.
- 3) Коли буде запропоновано вибрати компоненти, виберіть компоненти для встановлення на сервері.
 - Потрібно обрати компоненти сервера.
 - Зняти вибір клієнтського компонента.
 - Встановити прапорець `Redistributables`, який потрібен лише для першого встановлення.
 - Встановити прапорець «Драйвер захоплення аудіо», який установлює драйвер для віртуальних пристроїв введення/виведення аудіо, необхідних для надсилання та отримання аудіо між сервером і клієнтом.

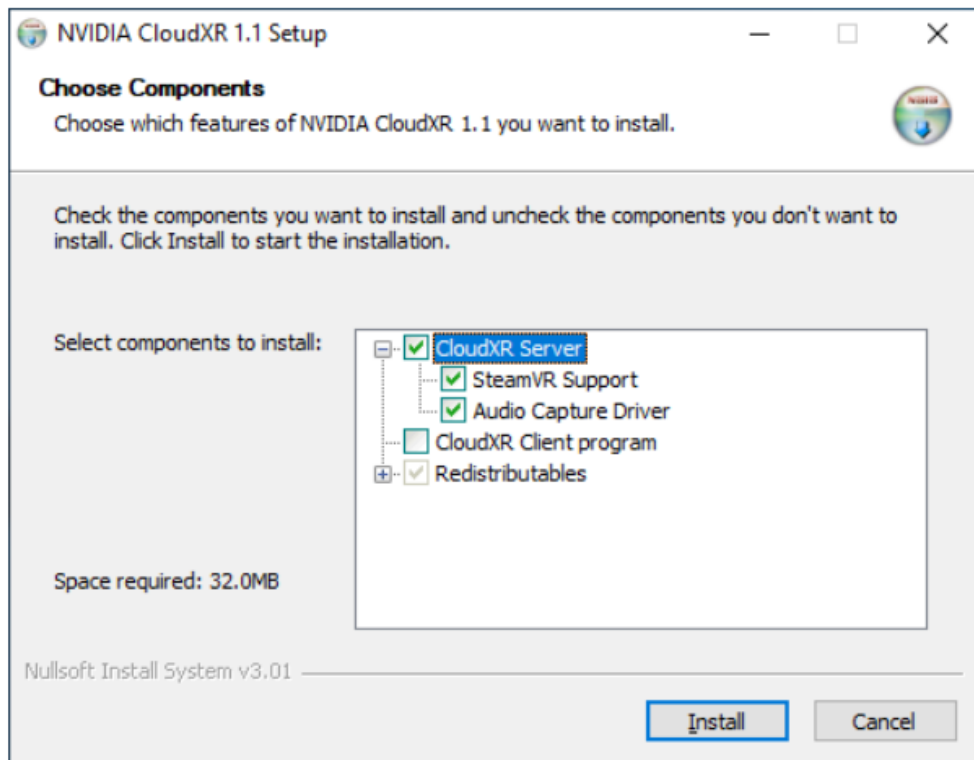


Рисунок 4.1 □ Графічний інтерфейс NVIDIA CloudXR

Інсталятор виконує такі завдання:

- Автоматично відкриває порти брандмауера Windows, необхідні для встановлення сеансу CloudXR.
- Перевіряє можливість кодека HEVC.
- Якщо цю можливість не буде виявлено, встановлення не вдасться.

1. Запуск SteamVR

Доки гарнітура не підключена локально до сервера, цей крок запускає драйвер сервера CloudXR. Див. Параметри командного рядка, щоб отримати інформацію про налаштування користувацьких параметрів запуску для сервера CloudXR.

Доки клієнт не буде запущено та підключено, інтерфейси Steam відображатимуть повідомлення Гарнітура не виявлена. Однак під час першого запуску драйвера сервера CloudXR інтерфейс Steam відобразить повідомлення «Підключіть пристрій VR».

Піктограма гарнітури внизу міні-вікна SteamVR змінюється, щоб вказати, що CloudXR встановлено та активний. Значок відображає літери CXR на гарнітурі, як показано на значку CloudXR. Значок стає зеленим, коли гарнітура активна, і сірим, коли гарнітура неактивна. Ви не побачите піктограм у міні-вікні SteamVR, доки клієнт не спробує підключитися до сервера.

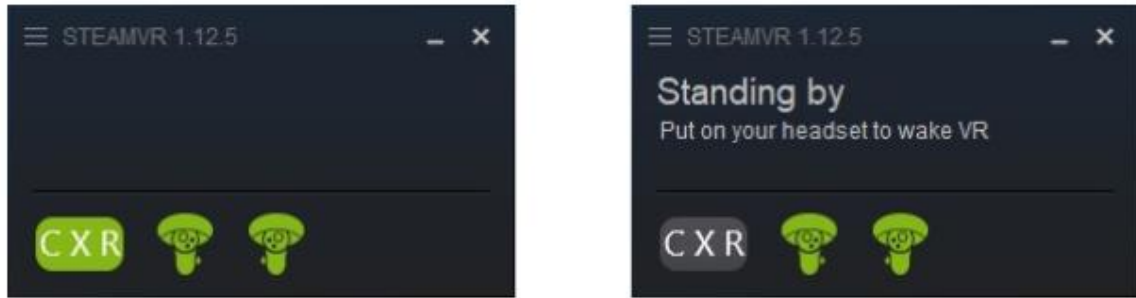


Рисунок 4.2 □ Графічний інтерфейс SteamVR

3.4 Запуск програм OpenVR на сервері

Загалом, потрібно запустити програму, яка буде транслюватися після підключення клієнта до сервера. В іншому випадку SteamVR і додаток повідомлять, що гарнітура не підключена. Коли клієнт підключається до сервера, перше, що він надсилає, це свої характеристики, такі як роздільна здатність і частота оновлення, щоб драйвер CloudXR SteamVR міг емулювати гарнітуру. Після запуску ця інформація передається до програм OpenVR. Зауважте, що в SteamVR немає концепції «зміни HMD», тому, якщо ви хочете використовувати інший пристрій із тим самим сервером (наприклад, переходити між брендами HMD або переходити з HMD на планшет), вам потрібно перезапускати SteamVR кожного разу, коли ви перемкніть перший підключений пристрій.

Так само віртуальні аудіопристрої, які програма використовуватиме для читання та запису аудіоданих, створюються лише після підключення клієнта. Щоб звести до мінімуму ймовірність того, що програма OpenVR вибере неправильний аудіопристрій, ми рекомендуємо вимкнути фізичні пристрої введення та виведення аудіо на сервері. Цей крок необов'язковий, оскільки

сервер CloudXR спробує встановити віртуальні пристрої як пристрої за замовчуванням, але деякі програми можуть не впоратися з цим належним чином.

Серверні програми SteamVR можна запустити наступними способами:

- Натиснувши кнопку на контролері, щоб викликати меню SteamVR на сервері.
- Натискання правої кнопки меню контролера на більшості автономних HMD зазвичай викликає локальне системне меню на клієнтському пристрої, яке перехоплюється операційною системою та не надається програмам. У зразках програм, які надаються разом із цим SDK або певними сторонніми зразками, ліва кнопка меню контролера відображається для виклику системного меню драйвера сервера.
- Ручним запуском програми безпосередньо на сервері.

За допомогою спеціальної процедури запуску, яка не передбачена NVIDIA CloudXR SDK. Наприклад, у вас може бути програма «сторожовий пес», яка чекає підключення клієнта, а потім запускає потрібний виконуваний файл.

3.5 Тестування Nvidia CloudXR

Хмарність та віртуальна реальність (VR) - це дві технологічні галузі, що розвиваються протягом кількох років. Meta, HTC, Pico та інші. Всі вони шукають автономну гарнітуру, яка революціонізує впровадження розширеної та віртуальної реальності. Ці стали потужними інструментами для проектування, виробництва та обслуговування продуктів.

Сьогодні для доступу до цих віртуальних світів необхідно використовувати гарнітуру. Проте автономні гарнітури віртуальної реальності, такі як Meta Quest 2, все ще мають значні технічні обмеження.

Дійсно, процеси, що використовуються, потребують значної обчислювальної потужності. Покоління віртуальних середовищ потребують багато ресурсів від графічних карт та процесорів. Звичайно, в ПК можна знайти потужну графічну карту, але сьогодні неможливо вбудувати такий же тип в гарнітуру.

Nvidia, лідер у галузі обчислювальної та штучної інтелекту, спробувала використати хмару на користь віртуальної реальності, створивши "CloudXR".

Дозволити гарнітурам віртуальної реальності скористатися обчислювальною потужністю своїх графічних карт. Гарнітури потім стануть простими засобами відображення вмісту.

Основна мета CloudXR - супроводжувати збагачення додатків, що наразі вбудовані в автономні гарнітури.

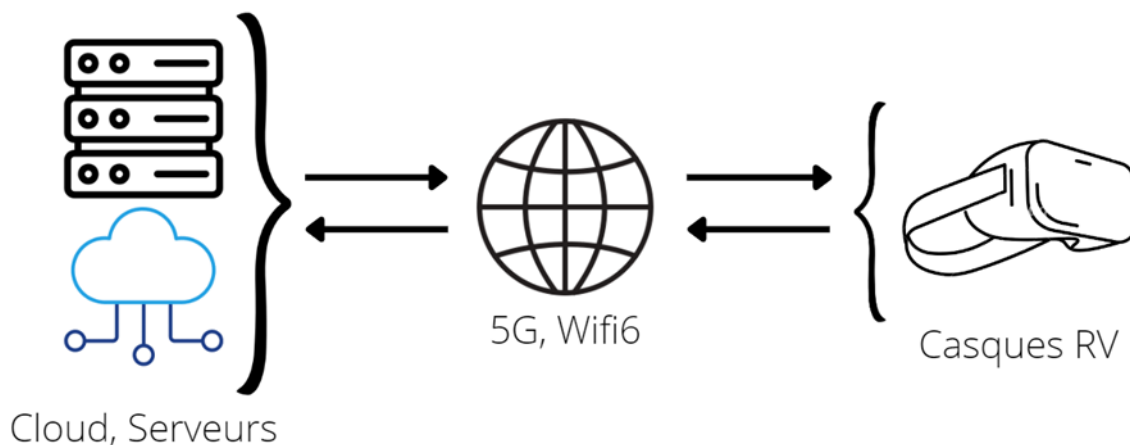


Рисунок 4.3 □ Схема роботи NVIDIA CloudXR

Було протестовано систему CloudXR від NVIDIA. Свідчення та оцінка продуктивності цієї технології.

Метою тестів поставити NVIDIA CloudXR на випробування з однією з найбільш використовуваних гарнітур VR: Meta Quest 2. Детальна презентація та аналіз продуктивності Cloud XR.

Перша частина спрямована на презентацію експерименту з метою полегшення розуміння. У другій частині буде зосереджено увагу на технічних аспектах та графічній продуктивності.

3.5.1 Nvidia CloudXR. Експеримент і конкретні запитання

Для проведення своїх тестів було використано наступне обладнання:

Апаратне забезпечення:

1. Віртуальна реальність гарнітура: Oculus Quest 2 [27].
2. Маршрутизатор wifi 6 для використання wifi, оптимізованого для об'єктів та автономних гарнітур
3. ПК/Сервер: Nvidia GeForce RTX 3060, Intel i7-9750H, 16 ГБ ОЗУ.

Програмне забезпечення:

1. Додаток будь-який додаток на Unity, щоб була можливість для тестів.
2. SteamVR: Інструмент, розроблений Valve та HTC для насолоди VR на пристрої за вашим вибором
3. OVR Metrics Tools: інструмент вимірювання продуктивності гарнітури, розроблений Meta.

Для створення хмари потрібен сервер та мережа. Сервери - це потужні комп'ютери, особливо в разі віртуальних створень. Щоб їх повністю використовувати, вони повинні бути пов'язані з відповідними мережевими інфраструктурами. Стандарт wifi 6 оптимізований для об'єктів та автономних гарнітур.

Додаток OVRMetricsTools дозволяє спостерігати та збирати більше 60 метрик гарнітури. Наш експерт вибрав 23 з цих метрик, які можуть бути згруповані в 3 підгрупи: дані GPU (графічний процесор), дані ЦП (центральний процесор) та дані користувацького досвіду (кадри в секунду, заряд батареї, температура гарнітури тощо). Постійний дисплей (накладка) дозволяє спостерігати за розвитком метрик у реальному часі, а також вони зберігаються у файлах з розширенням .csv для подальшої обробки.

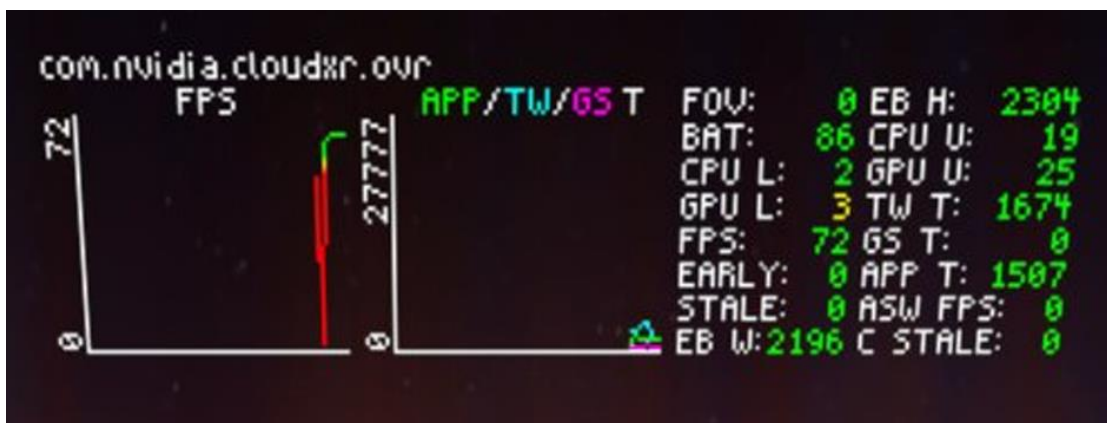


Рисунок 4.4 □ Поле зору гарнітури. Постійне накладання OVRMetricsTools

Для тестування обмежень NVIDIA CloudXR було обрано просту гру з кубіками. Цей додаток існує у двох формах. Перша версія була розгорнута безпосередньо на гарнітурі, як це є стандартною процедурою при розробці для Meta Quest 2. Друга - це програма для Windows. Вона запускається на віддаленому сервері. Сервіс Cloud XR є зв'язком між гарнітурою та сервером. Якщо є дві версії додатку, обидві виконують однаковий сценарій.

Сценарій розроблений для охоплення широкого спектру можливих складнощів віртуальних сцен. Віртуальна сцена складається з полігонів. Кожен з них може мати певну текстуру і навіть рухатися у просторі. Гіпотеза полягає в тому, що для розвитку цих параметрів і збільшення складності сцени та її графічного відтворення потрібна висока обчислювальна потужність, яка відповідає обіцянці, що пропонує CloudXR.

Сценарій поділений на три категорії. Кожна категорія дозволяє моделювати певну складність сцени. Використані тут полігональні куби.

Перша категорія: Куби мають просту текстуру. Ця конфігурація моделює найпростіші сцени для відтворення.

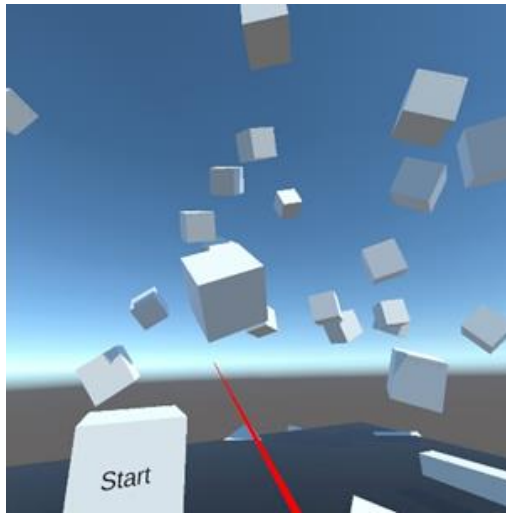


Рисунок 4.5 □ Перша категорія – 100 білих кубів

Друга категорія: Текстура кубів змінюється. Тут вони "скло". Гра світла та тіні стає важливою. Модельовані сцени тому складніші, ніж у попередній категорії.

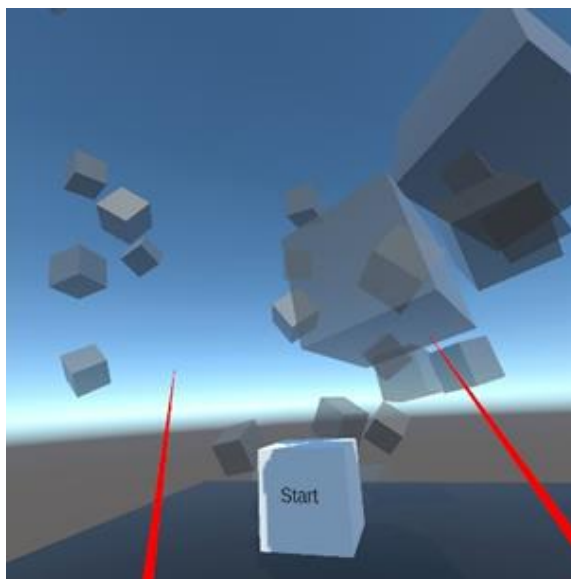


Рисунок 4.6 □ Друга категорія – 100 скляних кубів

Третя категорія: Кожне лице кожного куба - це відео. Кожне відео незалежне від інших. Ця специфіка є найскладнішою. Вона споживає найбільше ресурсів.

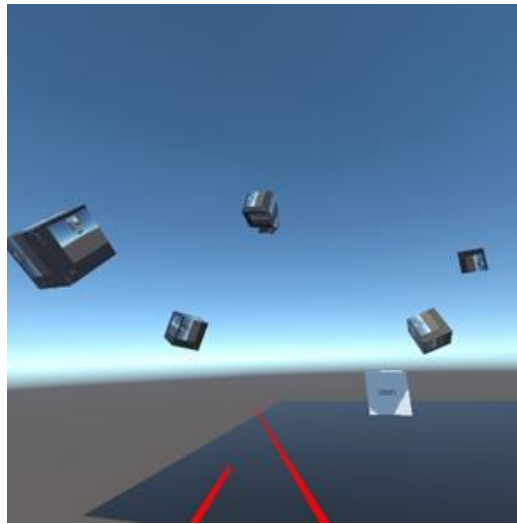


Рисунок 4.7 □ Третя категорія – 10 кубів з відео

Для уточнення вимірів у кожній категорії проводилися ітерації, що генерують 1 куб, потім 10 кубів, 100, 1 000, а потім 10 000 кубів. Кожна ітерація точно тестує межі гарнітури Meta Quest 2 (окремо) і гарнітури з CloudXR.

Тести проводилися в аналогічних умовах. Кожен тест тривав приблизно 9 хвилин і включав послідовність з 3 категорій.

Для отримання тенденцій метрик було проведено 20 тестів для кожної версії додатка. Усереднення даних кожної метрики протягом 20 вибірок дозволяє згладити неправильні виміри та отримати більш надійні дані.

Було створено програму для автоматизації всієї обробки даних. Ця програма витягає всі виміри, вікна, середні значення та витягає основні тенденції для кожної метрики.

3.5.2 Переваги NVIDIA CloudXR

Після першого аналізу вимірів, які було проведено, відзначається, що CloudXR дозволяє передбачити значні досягнення в застосунках віртуальної та доповненої реальності.

Зокрема, використання CloudXR є корисним для гарнітури, ми швидко спостерігаємо:

1. Зниження споживання батареї гарнітури приблизно на 33%.

2. Зниження температури в тій же пропорції.
3. Відсутність додаткової затримки.
4. Відчутна стабільність продуктивності.

Коли гарнітура потребує відображення 10 000 відео кубів, дуже складна сцена, місцева версія додатка раптово припиняє роботу. Версія Cloud більш підходить для цього покоління. Не всі відео відображаються правильно, але додаток все одно можна використовувати плавно. Для обробки великої кількості полігонів гарнітура втрачає в середньому на 75% більше FPS (кадрів на секунду), ніж місцева версія. Втрати візуально обтяжливі, і гарнітура знижується з 72 FPS до приблизно 20.

Для складних сцен перевага однозначно на стороні CloudXR. З іншого боку, для простих сцен або сцен з кількома текстурованими полігонами використання хмари не обов'язково виправдане. Наприклад, коли сцени мають до 1 000 кубів, місцевий додаток успішно справляється з управлінням генераціями.

Було визначено, що використання CloudXR в разі використання складних програм або потребує підвищеної портативності гарнітури. Розвиток технологій 5G і Wi-Fi 6 дозволить використовувати CloudXR як швидкий, надійний та ефективний спосіб використання віртуальної та змішаної реальності.

Висновки до розділу 4

З 4 розділу ми можемо бачити, що використання технології CloudXR від NVIDIA відкриває нові можливості для розвитку застосунків змішаної реальності. Експерименти показали значні переваги використання CloudXR, зокрема зниження споживання батареї і температури гарнітури Meta Quest 2, відсутність додаткової затримки і відчутна стабільність продуктивності.

Особливо важливими є результати тестів складних сцен, де CloudXR показав відмінність у порівнянні з локальними версіями програм. У разі обробки великої кількості полігонів чи відео кубів, CloudXR дозволяє плавно виконувати завдання, тоді як місцева версія може зазнавати збоїв або втрат продуктивності.

Проте, для простих сцен або сцен з невеликою кількістю полігонів, використання CloudXR може не бути цілком обгрунтованим через можливу перевагу місцевих обчислень. Таким чином, рекомендується використовувати CloudXR для складних додатків або сценаріїв, де потрібна підвищена портативність гарнітури.

Загалом, розвиток технологій 5G і Wi-Fi 6 дозволить розширити можливості використання CloudXR як швидкого, надійного та ефективного засобу для використання віртуальної та змішаної реальності.

ВИСНОВОК

У даній магістерській роботі були розглянуті та досліджено ключові аспекти серверної архітектури для потокової передачі відео контенту для VR пристроїв.

Досліджено принципи та методи створення серверної архітектури для потокової передачі віртуальної реальності VR. Під час аналізу розглянуті різні існуючі методи, включаючи адаптивний підхід до потокової передачі VR, а також різноманітні алгоритми, які застосовуються для ефективної передачі контенту.

У ході дослідження було виявлено, що впровадження серверної архітектури для потокової передачі VR контенту володіє значним потенціалом у різних сферах, включаючи навчання, розваги та глибоке навчання. Це може сприяти забезпеченню більш доступного та якісного контенту віртуальної реальності та покращити взаємодію користувачів з цією технологією.

Пропозиції щодо варіантів розгортання серверної архітектури свідчать про необхідність подальших досліджень та розвитку в даній області з метою оптимізації процесу потокової передачі VR контенту. Подальші вивчення можуть сприяти створенню більш ефективних та ергономічних рішень для користувачів в цій сфері.

Узагальнено, результати дослідження відкривають нові можливості для розвитку технологій віртуальної реальності та підтримують створення більш доступних та якісних продуктів для користувачів в цій сфері.

Покращення доступності до VR-змісту - це дозволить користувачам отримувати доступ до VR-контенту без необхідності завантаження великих обсягів даних на свої пристрої. Віртуальна реальність може стати більш доступною і зручною для споживачів.

Підвищення якості взаємодії з VR. Застосування оптимізованих алгоритмів і методів передачі даних може покращити якість візуального і аудіо

контенту у віртуальній реальності. Це сприятиме покращенню досвіду користувача і забезпеченню більш реалістичного середовища.

Розвиток нових сфер застосування. Створення серверної архітектури для потокової передачі VR контенту відкриває двері для використання цієї технології в різних галузях, таких як освіта, медицина, розваги, туризм тощо. VR може стати потужним інструментом для навчання, розваг та інших сфер.

Зменшення обчислювального навантаження на пристрої користувача. Використання серверної архітектури дозволяє віддалено обробляти і передавати складні дані, що зменшує навантаження на пристрої користувача і дозволяє використовувати менш потужне обладнання для віртуальної реальності.

Отже, створення серверної архітектури для потокової передачі VR контенту сприятиме розвитку та впровадженню цієї технології в різних сферах, покращить доступність та якість віртуального досвіду для користувачів, а також розширить можливості застосування віртуальної реальності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Augmented Reality. URL: <https://www.opencvhelp.org/tutorials/applications/augmented-reality/> (дата звернення: 15.12.2023)
2. Augmented Reality: the tricky projection matrix. URL: <https://fruty.medium.com/augmented-reality-with-opencv-and-opengl-the-tricky-projection-matrix-394cc6050b00> (дата звернення: 16.12.2023)
3. "NVIDIA CloudXR - Virtual Reality for the Enterprise". URL: <https://www.nvidia.com/en-us/design-visualization/technologies/cloudxr/> (дата звернення: 16.12.2023)
4. "Introducing NVIDIA CloudXR". URL: <https://developer.nvidia.com/blog/introducing-nvidia-cloudxr/> (дата звернення: 16.12.2023)
5. "NVIDIA CloudXR for AR/VR Streaming". URL: <https://developer.nvidia.com/cloudxr> (дата звернення: 25.12.2023)
6. Smith, J., Brown, K., & Jones, L. Decision Tree Visualization Techniques: A Comparative Study. IEEE Transactions on Visualization and Computer Graphics, 2021, 27(5), 1800-1813. (дата зверн. 15.12.2023)
7. "NVIDIA CloudXR: The Future of XR Streaming". URL: <https://blogs.nvidia.com/blog/2020/03/18/nvidia-cloudxr-virtual-reality/> (дата звернення: 22.12.2023)
8. "What is Amazon RDS?". URL: <https://aws.amazon.com/rds/> (дата звернення: 22.12.2023)
9. "Getting Started with Amazon RDS". URL: <https://aws.amazon.com/getting-started/hands-on/create-mysql-db/> (дата звернення: 22.12.2023)
10. "Amazon EC2". URL: <https://aws.amazon.com/ec2/> (дата звернення: 24.12.2023)
11. "Amazon EC2 Documentation". URL: <https://docs.aws.amazon.com/ec2/index.html> (дата звернення: 24.12.2023)

12. Park, S., Kim, D., & Lee, J. Enhancing Decision Tree Visualization with Virtual Reality. *Journal of Visual Languages & Computing*, 2021, 67, 102046. (дата зверн. 15.12.2023)
13. ДСТУ ISO/IEC 12207:2016. Інженерія систем і програмного забезпечення. Процеси життєвого циклу програмного забезпечення (ISO/IEC 12207:2008, IDT)
14. Городецька О. С., Гикавий В. А., Онищук О. В. Комп'ютерні мережі : навч. посіб. Вінниця : ВНТУ, 2017. 129 с
15. Choi, J., Kim, S., & Lee, M. Application of 3D Decision Tree Visualization in Financial Risk Management. *Journal of Risk and Financial Management*, 2021, 14(3), 120. (дата зверн. 15.12.2023)
16. Задерейко О. В., Логінова Н. І., Толокнов А. А. Комп'ютерні мережі : навч. посібник. Одеса : Фенікс, 2022. 249 с
17. Johnson, R., Williams, M., & Brown, K. Comparative Study of 3D Decision Tree Visualization Techniques in Healthcare. *Journal of Healthcare Engineering*, 2020, 9(1), 23-36. (дата зверн. 15.12.2023)
18. Петренко, В., & Шевченко, О. Використання 3D-моделей для візуалізації маркетингових стратегій. *Наукові записки Національного університету "Острозька академія"*, 2023, 4(67), 78-89. (дата зверн. 15.12.2023)
19. Gao, X., Wang, Y., & Zhang, L. A Comparative Analysis of 3D Decision Tree Visualization Techniques for Environmental Data Analysis. *Journal of Environmental Informatics*, 2021, 38(2), 45-58. (дата зверн. 15.12.2023)
20. Лемешко А. В., Кирпач Л. А., Сорокін Д. В., Бученко І. А., Шрам М. М. Проектування безпроводових комп'ютерних мереж. Київ : ДУТ, 2021. 147 с.
21. Коваленко, М., & Сидоренко, Т. Використання 3D-моделей у візуалізації графічних інформаційних систем для розробки ділових додатків. *Інформаційні технології та комп'ютерна інженерія*, 2019, 2(4), 200-213. (дата зверн. 15.12.2023)

22. Lee, S., Kim, H., & Park, J. Application of 3D Decision Tree Visualization in Smart City Planning. Journal of Urban Planning and Development, 2023, 19(4), 140-155. (дата зверн. 15.12.2023)
23. Ткаченко О. М., Торошанко Я. І., Лемешко А. В. та ін. Комп'ютерні мережі: контроль та прогнозування перевантажень : посібник. Київ : ДУТ, 2021. 77 с.
24. Lin, H., Chang, C., & Hsu, C. Comparative Study of 3D Decision Tree Visualization Techniques for Educational Data Analysis. Journal of Educational Technology & Society, 2019, 22(3), 200-213. (дата зверн. 15.12.2023)
25. Іванова, А., & Козлов, В. Використання 3D-моделей для візуалізації даних в галузі машинного навчання. Наукові праці Харківського національного університету імені В. Н. Каразіна. Серія "Інформатика", 2022, 8(4), 240-253. (дата зверн. 15.12.2023)
26. Міжнародна науково-практична конференція URL: <https://foss.kn-it.info/> (дата звернення: 15.02.2024)
27. Характеристики Oculus quest 2 URL: <https://bit.ly/3hovz89> (дата звернення: 24.12.2023)

Додаток А

Лістинг програми з розширенням .java

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import weka.classifiers.trees.Id3;
import weka.core.Instance;
import weka.core.Instances;

public class DecisionTree {
    private Instances trainingData;

    public static void main(String[] args) throws Exception {
        DecisionTree decisionTree = new DecisionTree("input/saturation_data.arff");

        // Остальной код здесь...
    }

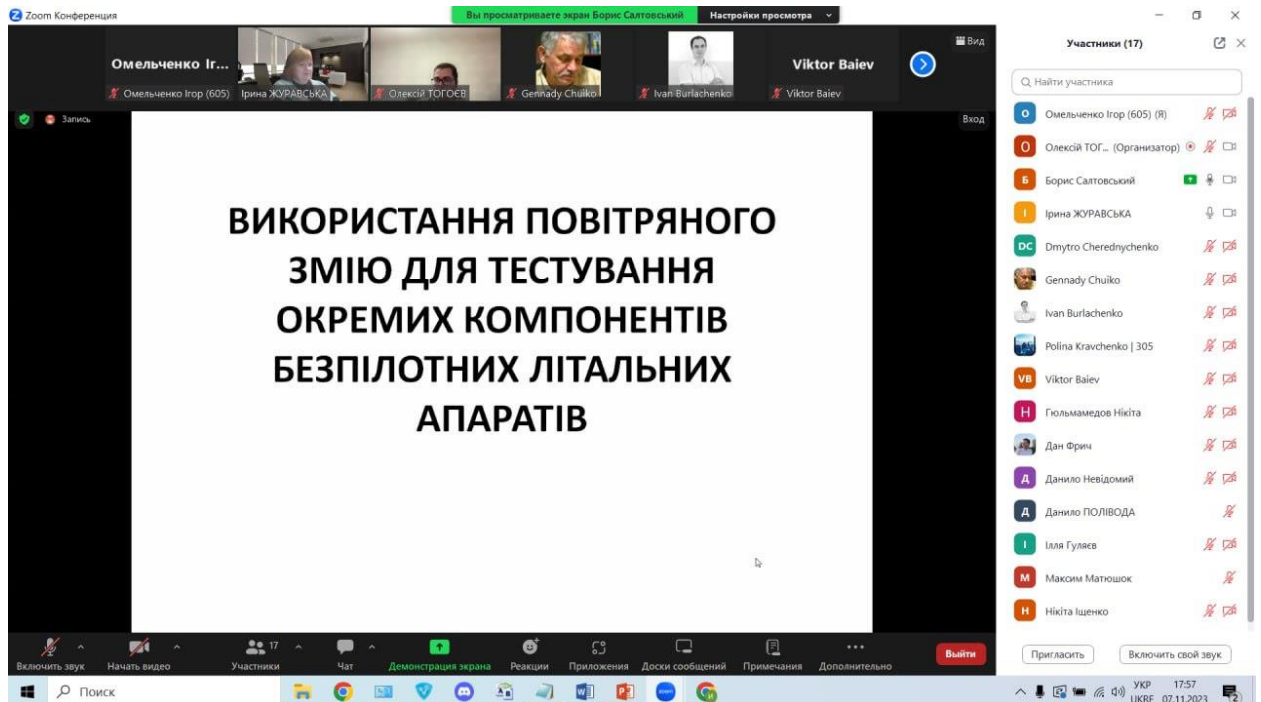
    public DecisionTree(String fileName) {
        BufferedReader reader = null;
        try {
            // Read the training data
            reader = new BufferedReader(new FileReader(fileName));
            trainingData = new Instances(reader);

            // Setting class attribute
            trainingData.setClassIndex(trainingData.numAttributes() - 1);

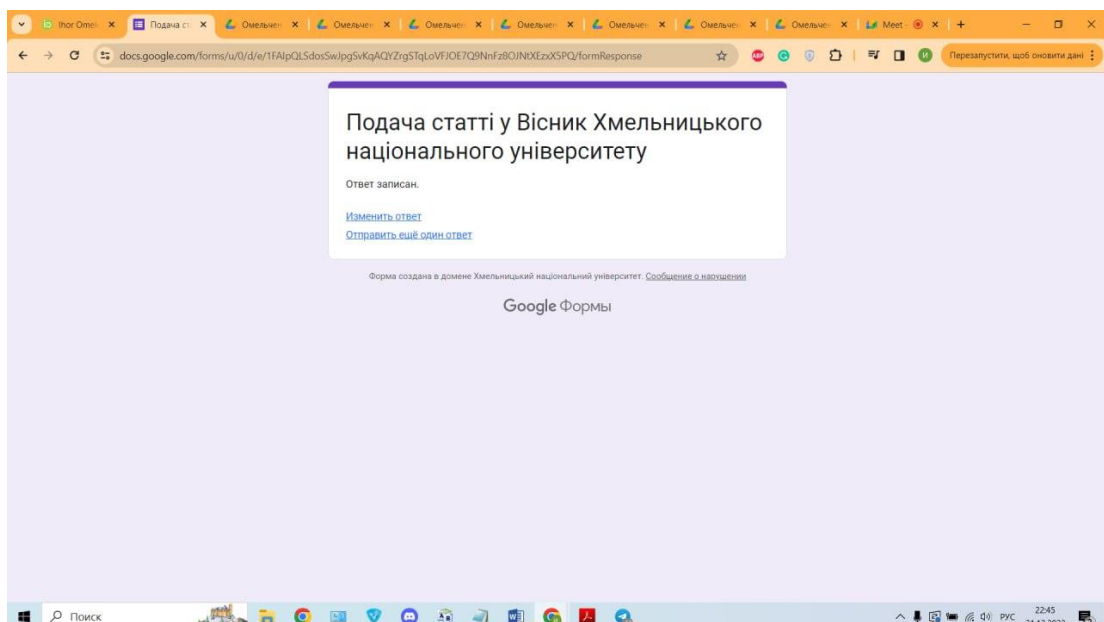
            // Выводим список атрибутов
            System.out.println("Список атрибутов:");
            for (int i = 0; i < trainingData.numAttributes(); i++) {
                System.out.println(trainingData.attribute(i));
            }
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (reader != null) {
                try {
                    reader.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

```
    }  
    }  
    }  
}  
  
private Id3 trainTheTree() {  
    Id3 id3tree = new Id3();  
  
    try {  
        id3tree.buildClassifier(trainingData);  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
  
    return id3tree;  
}  
  
private Instance prepareTestInstance() {  
    Instance instance = new Instance(3); // Изменим на 3, так как теперь у нас 3 атрибута для сатурации  
    instance.setDataset(trainingData);  
  
    // Устанавливаем значения для каждого атрибута сатурации  
    // В данном примере, я генерирую случайные значения для каждого атрибута  
    instance.setValue(trainingData.attribute(0), 85); // Pulse  
    instance.setValue(trainingData.attribute(1), 36.8); // Temperature  
    instance.setValue(trainingData.attribute(2), 96); // Saturation  
  
    return instance;  
}  
}
```

Додаток Б Абробації



Проходження абробації XXVI Всеукраїнської науково-практичної конференції «Могилянські читання–2023»



Подача тези до «Вісника Хмельницького національного університету 2023».
Посилання: Омельченко І. Г. Метод дерев рішень у комп'ютерній діагностиці.
Вісник Хмельницького національного університету. 2023. № 6 (327). С. 266-273. DOI: 10.31891/2307-5732-2023-329-6-266-273