Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____*Комп'ютерних наук*_____
(повна назва)
Кафедра_____*Системотехніки*_____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____другий (магістерський)_____

_____Розробка компонентів інформаційної системи_____
_____відновлення зображення_____
(тема)

Виконав:
студент _2__ курсу, групи _*СПРм-22-1*_____
_____*Гаджиєв Е. Р.*_____
(прізвище, ініціали)

Спеціальність ___*122 Комп'ютерні науки*____
__
(код і повна назва спеціальності)
Тип програми ____освітньо-наукова_____
Освітня програма __*Системне проектування*_
_____
(повна назва освітньої програми)

Керівник ___*проф. Саваневич В. Є.*_____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ __*Гребеннік І.В.*__
(підпис) (прізвище, ініціали)

2024 р.

Я як студент(ка) ХНУРЕ розумію і підтримую політику закладу із академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

12.06.2024 

Кваліфікаційна робота не містить відомостей, заборонених до відкритого опублікування.

Кваліфікаційна робота виконана у відповідності до стандартів, що діють в Україні.

Попередній захист проведено 12.06.2024

Керівник кваліфікаційної роботи  проф. Саваневич В. Є.

Харківський національний університет радіоелектроніки

Факультет *Комп'ютерних наук*

Кафедра *Системотехніки*

Рівень вищої освіти другий (магістерський)

Спеціальність *122 Комп'ютерні науки*
(код і повна назва

Тип програми освітньо-наукова

Освітня програма *Системне проектування*
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри СТ
Проф. Гребенік І.В. _____
(підпис)
«_01_»____04___2024_ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові *Гаджиєву Емілю Рафіковичу*
(прізвище, ім'я, по батькові)

1. Тема роботи: *Розробка компонентів інформаційної системи відновлення зображення*
затверджена наказом по університету від 01.04.2024. № 259Ст
2. Термін подання студентом роботи до екзаменаційної комісії: 15.06.2024 р
3.Вихідні дані до роботи: *Розробити компоненти інформаційної системи відновлення зображення, що забезпечують автоматизацію процесу відновлення астрономічних зображень спотворених зовнішніми впливами. Операційна система: Ubuntu 24.07 LTS. Програмне забезпечення: IDE PyCharm, застосунок Docker-Desktop.*
4. Перелік питань, що потрібно опрацювати в роботі: *Вступ. Аналіз предметної області. Розробка методу відновлення астрономічних зображень. Проєктування та розробка архітектури компонентів інформаційної системи відновлення зображень. Розгортка та тестування компонентів інформаційної системи. Висновки.*
5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій: *Use Case діаграма інформаційної системи відновлення зображень, діаграма послідовностей для процесу «Створення нового завдання відновлення зображень в системі», діаграма послідовностей для процесу «Розміщення завдань відновлення зображень у черзі виконання завдань», діаграма послідовностей для процесу «Виконання завдання відновлення зображень», діаграма станів завдання обробки зображення, діаграма класів пов'язаних із створенням завдань обробки зображень, діаграма класів пов'язаних із виконанням завдань з обробки зображень, логічна модель даних інформаційної системи, діаграма архітектури інформаційної системи відновлення зображення, фізична модель даних, графік середнього часу обробки завдань для різних конфігурацій системи.*

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім’я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	*Отримання завдання на виконання роботи*	*01.04.2024*	
2	*Визначення ключових потреб у відновленні астрономічних зображень*	*02-04.04.2024*	
3	*Огляд і порівняльний аналіз існуючих інструментів для роботи з астрономічними зображеннями*	*05-15.04.2024*	
4	*Аналіз ролі хмарних обчислень у процесах відновлення астрономічних зображень*	*16-20.04.2024*	
5	*Визначення вимог до компонентів інформаційної системи відновлення зображень*	*21-23.04.2024*	
6	*Розробка комбінованого підходу до відновлення астрономічних зображень*	*24-30.04.2024*	
7	*Проєктування компонентів інформаційної системи відновлення зображень*	*01-07.05.2024*	
8	*Розробка масштабованої архітектури інформаційної системи відновлення зображень*	*08-15.05.2024*	
9	*Розгортка компонентів інформаційної системи засобами Cloud-технологій*	*15-16.05.2024*	
10	*Перевірка продуктивності розроблених компонентів інформаційної системи*	*17-18.05.2024*	
11	*Оформлення пояснювальної записки*	*18-25.05.2024*	
12	*Представлення на рецензування*	*13.06.2024*	

Дата видачі завдання 01.04.2024 р.

Студент _____
(підпис)

Керівник роботи _____ *проф. Саваневич В. Є.*
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до магістерської кваліфікаційної роботи: 125 с., 1 табл., 40 рис., 4 додатки, 25 джерел інформації.

ІНФОРМАЦІЙНА СИСТЕМА, АСТРОНОМІЯ, ХМАРНІ ТЕХНОЛОГІЇ, МАСШТАБУВАННЯ, АЛГОРИТМ ЛЮСІ-РІЧАРСДОНА, МЕДІАННИЙ ФІЛЬТР, ВІДНОВЛЕННЯ ЗОБРАЖЕНЬ

Об'єктом дослідження є процес відновлення космічних зображень, що включає в себе поліпшення якості зображень, отриманих з телескопів.

Предметом дослідження є алгоритми відновлення зображень та їх інтеграція та масштабування в інформаційних системах для обробки зображень.

Метою дослідження роботи є проектування та розробка серверної та клієнтської частини компонентів інформаційної системи для відновлення космічних зображень з акцентом на можливість легкої інтеграції нових модулів та масштабованість системи.

Методи дослідження включають системний підхід, методи структурного аналізу та моделювання, а також застосування алгоритмів для обробки та покращення зображень.

У роботі проведено аналіз існуючих систем для обробки та відновлення космічних зображень, виділено основні недоліки та запропоновано нові підходи та методи для їх усунення, включаючи розробку гнучкої та масштабованої архітектури системи.

Сфера застосування результатів роботи включає не тільки космічну галузь, але й області, де важливе відновлення та покращення якості зображень, такі як дистанційне зондування Землі, метеорологія та екологічний моніторинг.

ABSTRACT

Master’s Thesis: 125 pages, 2 tables, 39 figures, 4 appendices, 25 title.

INFORMATION SYSTEM, ASTRONOMY, CLOUD TECHNOLOGIES, SCALING, LUCY-RICHARDSON ALGORITHM, MEDIAN FILTER, IMAGE RECOVERY

The object of the research is the process of recovering space images, which involves improving the quality of images obtained from telescopes.

The subject of the study is image recovery algorithms and their integration and scaling in information systems for image processing.

The aim of the research is to design and develop the server and client components of an information system for the recovery of space images, focusing on the ease of integrating new modules and system scalability.

The research methods include a systems approach, methods of structural and object analysis and modeling, and the application of algorithms for processing and improving images.

The work analyzes existing systems for processing and recovering space images, identifies major flaws, and proposes new approaches and methods to address them, including developing a flexible and scalable system architecture.

The scope of application of the results includes not only the space sector but also areas where image recovery and quality improvement are important, such as Earth remote sensing, meteorology, and environmental monitoring.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

GUI	– Graphical User Interface
IRAF	– Image Reduction and Analysis Facility
CL	– Command Language
БД	– База даних
AWS	– Amazon Web Services
EC2	– Elastic Compute Cloud
RDS	– Amazon Relational Database Service
S3	– Amazon Simple Storage Service
EBS	– Elastic Block Store
UML	– Unified Modeling Language
LTS	– Long Term Support

ВСТУП

У сучасному світі, де активно розвиваються наука та технології, особливе значення набуває дослідження космічного простору. Космічні зображення, отримані за допомогою телескопів, є важливим джерелом інформації для астрономів і вчених, які вивчають Всесвіт. Проте ці зображення часто піддаються впливу різноманітних спотворень, таких як атмосферне розмиття та обмеження, пов’язані з самим процесом отримання зображень, що ускладнює їх аналіз та трактування. У зв'язку з цими труднощами обробка та відновлення космічних зображень за допомогою комп'ютерних технологій стає одним із найважливіших напрямків у галузі астрономічних досліджень.

Наразі існують численні засоби обробки та відновлення астрономічних зображень, які можуть допомогти в аналізі космічних даних і виявленні прихованих фізичних об’єктів. Однак у більшості цих засобів є недоліки. Наприклад, деякі з них базуються на закритому програмному забезпеченні, що ускладнює доступ до їх внутрішніх механізмів та можливостей модифікації для конкретних потреб дослідника. Крім того, багато з цих інструментів мають складні інтерфейси та потребують значних знань у галузі астрономії та обробки зображень для їх ефективного використання, а значна кількість таких інструментів є дорогою у використанні.

У зв’язку зі швидким розвитком технологій деякі інструменти відновлення зображень можуть виявитися застарілими, а такі методи мають нижчий рівень ефективності у порівнянні з новітніми розробками.

Крім того, деякі підходи можуть вимагати великої обчислювальної потужності або спеціального обладнання, що робить їх менш доступними для широкого кола дослідників і астрономів. Тому розвиток нових інструментів, які були б відкритими, мали простий інтерфейс, а також були ефективними та актуальними у своїй діяльності, є важливою задачею для астрономічної спільноти.

Відповідно, основна мета поточної роботи полягає в розробці компонентів інформаційної системи, спрямованої на відновлення космічних зображень, з акцентом на забезпечення легкості використання, масштабування та доступності цих інструментів. Ця система має стати вагомим внеском у поліпшення процесів аналізу космічних зображень, дозволяючи дослідникам з різним рівнем технічної підготовки ефективно використовувати сучасні алгоритми відновлення зображень без потреби в глибоких знаннях в області інформаційних технологій та особливостей алгоритмів обробки та відновлення зображення.

Основу увагу приділено можливості масштабування інформаційної системи з метою прискорення наукових експериментів, в том числі з використанням хмарних сервісів та технологій. Це відкриває нові перспективи для підвищення ефективності та результативності наукових досліджень у галузі астрономії та космічних досліджень. Така інформаційна система не лише дозволить користувачам з легкістю зберігати та обробляти космічні зображення, а й забезпечить можливість розширення її функціональності шляхом використання хмарних ресурсів. Це сприятиме значному збільшенню швидкості аналізу даних та виконання обчислень, що в свою чергу прискорить наукові відкриття в галузі астрономії. Інформаційна система, що розробляється, не лише зробить наукові дослідження більш доступними та ефективними, а й сприятиме швидкій адаптації до зростаючих потреб у вивченні космосу.

Дана робота пройшла апробацію на 28-му міжнародному форумі «Радіоелектроніка та молодь XXI столітті» у конференції «Інформаційні інтелектуальні системи» та на міжнародній конференції «International Conference on Computational Linguistics and Intelligent Systems, CoLInS 2024» з подальшою публікацією у збірнику «CEUR Workshop Proceedings», який входить у наукометричну базу Scopus [1, 2].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Визначення ключових потреб у відновленні астрономічних зображень

Астрономічні дослідження значною мірою залежать від якості даних спостережень, серед яких важливу роль відіграють зображення космічних об’єктів. Ці зображення є ключовим джерелом інформації для аналізу та вивчення космічних явищ, від руху планет до вивчення галактик і далеких квазарів. Однак процес отримання високоякісних астрономічних зображень є складним через ряд природних і технічних проблем, які можуть спотворити вихідні дані. Виявлення та подолання цих проблем має вирішальне значення для розвитку астрономії та дослідження космосу.

Астрономічні зображення можуть бути спотворені різними факторами, які впливають на їх якість і точність. Вивчення цих факторів є важливим для розробки ефективних методів відновлення зображень, які забезпечують високу точність астрономічних спостережень.

Однією з найбільших проблем при отриманні астрономічних зображень є вплив земної атмосфери. Турбулентність в атмосфері, спричинена коливаннями температури та тиску, призводить до того, що світло від зірок та інших космічних об’єктів поширюється нелінійним чином, спотворюючи (розмиваючи) зображення. Ефект розмиття особливо важливий в наземних обсерваторіях, де атмосфера стає своєрідним «фільтром», що знижує роздільну здатність телескопа.

На малюнку 1.1 показано приклад розмиття небесного тіла на астрономічному зображенні. В результаті атмосферної турбулентності зображення об'єктів на поверхні Землі, що сприймаються як точкові джерела ззовні атмосфери, втрачають різкість і набувають округлої форми.

Існують технології, що надають можливість мінімізувати або виправити ці спотворення. Одним з таких підходів є використання адаптивної оптики, яка

в режимі реального часу коригує спотворення, викликані атмосферною турбулентністю, за допомогою гнучких дзеркал, що змінюють свою форму, проте не зважаючи на високу ефективність, адаптивна оптика є дорогою у використанні [3].

Іншим методом є обробка зображень за допомогою спеціалізованого програмного забезпечення, що дозволяє виправити розмитості та покращити деталізацію зображень. Цей підхід частіше використовується дослідниками-початківцями з причини своєї доступності.

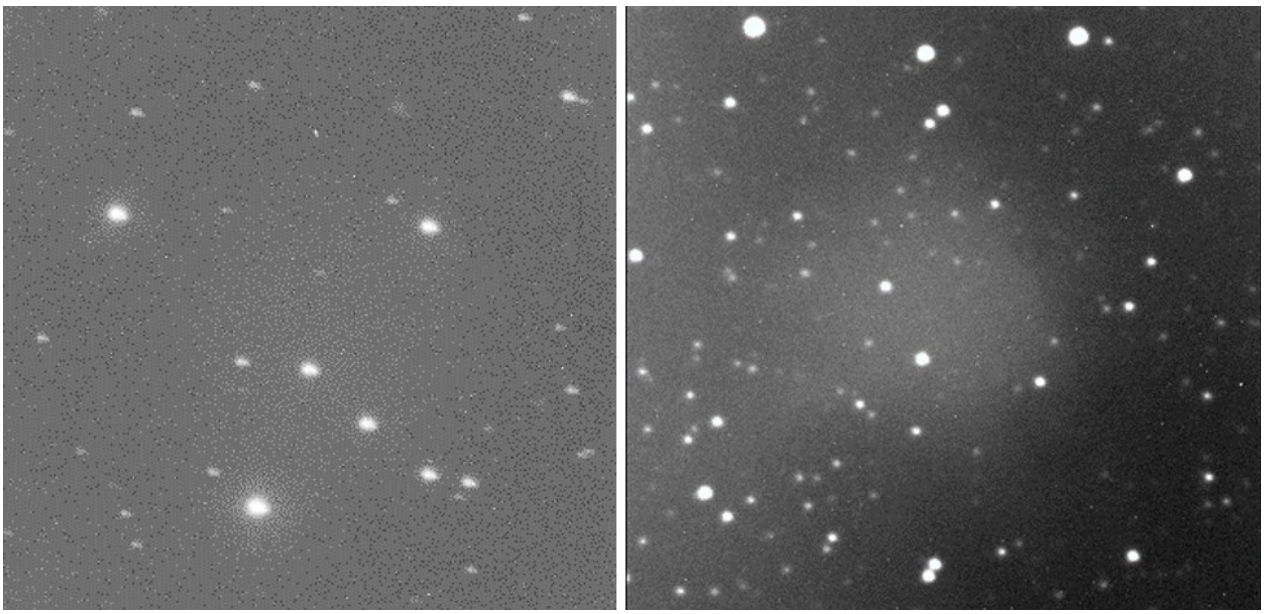


Рисунок 1.1 – Приклад розмиття зображення космічних об’єктів

Прояв негативного впливу атмосфери Землі на якість отриманих зображень можна побачити при використанні наземних обсерваторій або камер, розташованих на високих платформах. Пориви вітру, що виникають в таких випадках, можуть викликати механічні коливання, які негативно впливають на стабільність зображення [4]. На рисунку 1.2 подано приклад замазаних зображень космічних об’єктів при поривах вітру.

Змазання зображень, отримане в результаті, є особливо критичним для зображень із високою роздільною здатністю, де навіть мінімальна вібрація може призвести до значного погіршення якості.

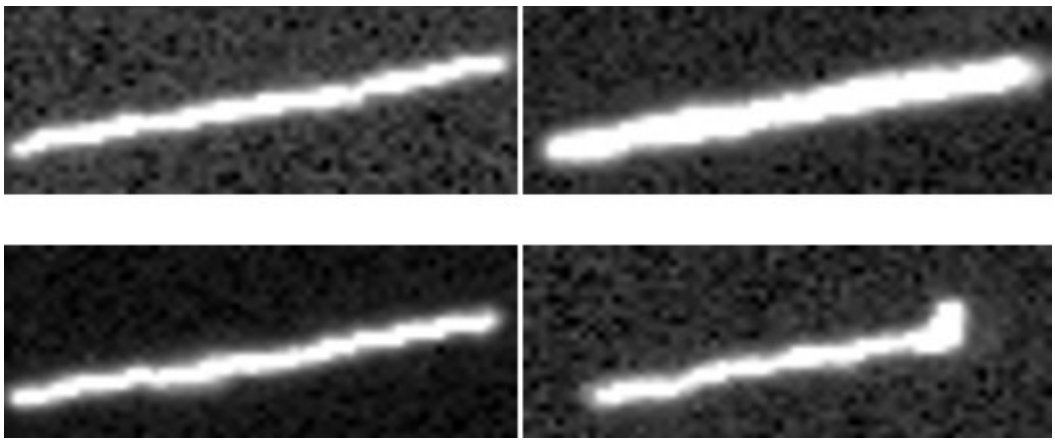


Рисунок 1.2 – Приклад змазаних зображень космічних об’єктів при поривах вітру

Великий вплив на погіршення якості зображень космічних об’єктів мають проблеми, пов'язані з супроводом космічних апаратів, викликані різними факторами, включаючи помилки в алгоритмах управління, зовнішні впливи або обмежені можливості бортових навігаційних систем, що в результаті призводить до того, що камера космічного апарату не може підтримувати стабільний фокус на обраній цілі, що, в свою чергу, призводить до появи артефактів на зображенні (див. 1.3).

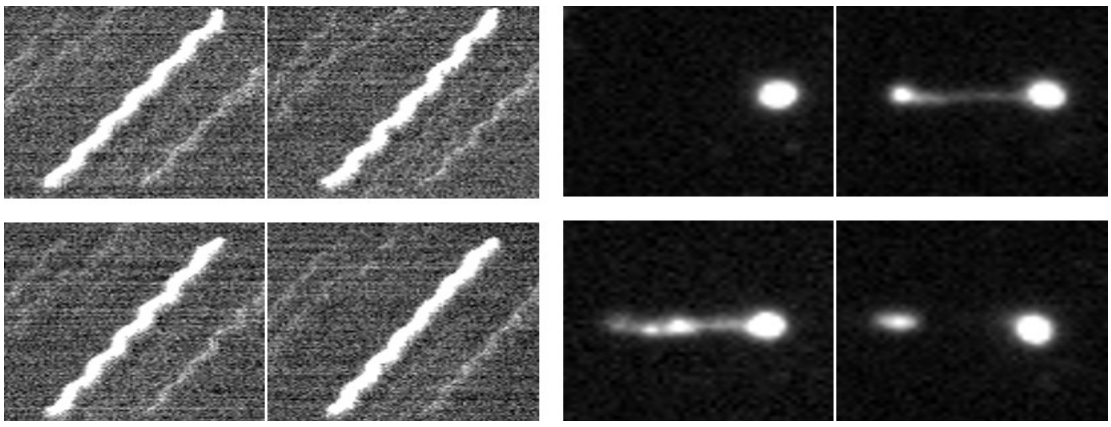


Рисунок 1.3 – Приклад змазаних зображень космічних об’єктів при супроводі космічних апаратів та при зриві добового ведення

Проблема, яка часто виникає під час спостережень за допомогою космічних апаратів і яка негативно впливає на якість астрономічних зображень,

пов’язана з недосконалістю систем точного позиціонування цих апаратів відносно Землі протягом доби. Ця проблема виникає внаслідок обмежень, які орбітальні характеристики накладають на здатність космічного апарату підтримувати постійну орієнтацію відносно поверхні Землі, особливо при роботі в режимі високої роздільної здатності. В результаті на зображеннях можуть з’являтися розмиті сегменти, показані на правій частині рисунку 1.3, що ускладнює процес ідентифікації конкретних об’єктів або регіонів космічного простору. Для забезпечення високої точності позиціонування КА потрібні складні системи управління та навігації, інтегровані в космічний апарат, що в багатьох випадках є дорогим рішенням.

Покращення якості супутникових зображень шляхом підвищення точності датчиків, удосконалення алгоритмів стабілізації зображення та оптимізації програмного забезпечення для обробки даних у реальному часі є стратегією, яка спрямована на покращення якості зображень на етапі їх отримання [5]. Цей підхід має ряд переваги, зокрема до яких можна віднести можливість отримання початкових даних вищої якості. Однак присутні також деякі значні недоліки, а саме низька доступність таких рішень та фінансові обмеження.

Перш за все слід зазначити, що вдосконалення фізичного обладнання та розробка складного програмного забезпечення вимагає значних капіталовкладень. Більш точні датчики та складні системи оптичної стабілізації можуть бути надзвичайно дорогими для розробки та виробництва. Ці витрати включають не лише фінанси, що витрачаються на початкову розробку та виробництво, але й витрати на інтеграцію цих технологій у космічні апарати, телескопи, тощо, а також витрати на випробування та сертифікацію.

По-друге, зростання складності апаратного та програмного забезпечення може призвести до більшої ймовірності операційних збоїв і, як наслідок, вищих витрат на технічне обслуговування та ремонт.

В свою чергу, постобробка з використанням алгоритмів для покращення фактично отриманих зображень забезпечує більш гнучкий та економічно

ефективний підхід [6]. Технології постобробки можуть покращити якість зображень після їх зйомки, включаючи зменшення шуму, підвищення різкості та контрастності. Це означає, що оригінальні зображення, що зняті навіть за допомогою застарілого обладнання, можна значно покращити за допомогою відповідного програмного забезпечення.

Крім того, розробка програмного забезпечення та алгоритмів покращення зображення може бути значно дешевшою та доступнішою, ніж розробка та виробництво фізичного обладнання. Програмне забезпечення можна легко оновлювати та розповсюджувати серед користувачів, що робить його більш гнучким для адаптації до нових завдань та умов роботи.

Отже, для обробки астрономічних зображень найбільш ефективним та економічно вигідним варіантом є використання програмного забезпечення, що здатне виправляти спотворення, спричинені атмосферною турбулентністю та іншими факторами. Такий підхід дозволяє значно підвищити якість астрономічних зображень, без потреби у вдосконаленні існуючого обладнання, та зробити астрономічні дослідження більш доступними та ефективними.

1.2 Огляд і порівняльний аналіз існуючих інструментів для роботи з астрономічними зображеннями

Для аналізу і порівняння інструментів, що застосовуються для роботи з астрономічними зображеннями і мають можливість їх відновлення, було виділено перелік популярних рішень, зосереджуючись на їх перевагах та недоліках у контексті раніше описаних потреб відновлення зображень.

Першим із знайдених рішень є програмне забезпечення AstroImageJ, що є адаптацією популярного програмного забезпечення для обробки зображень ImageJ, налаштованого спеціально для потреб астрономічної спільноти [7]. Це програмне забезпечення має багато інструментів для аналізу та обробки астрономічних даних, включаючи можливість виправлення розмитих зображень. До переваг цього інструменту можна віднести гнучкість в

налаштуванні під конкретні зображення, на рисунку 1.4 подано вигляд інтерфейсу для налаштувань аналізу та візуалізації даних астрономічних зображень.

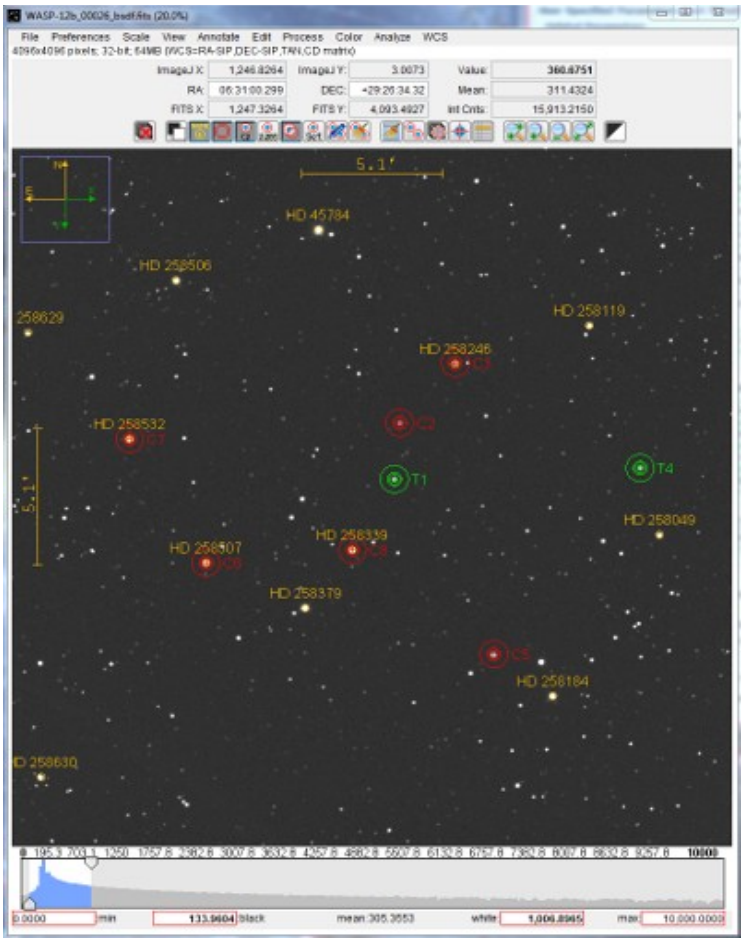


Рисунок 1.4 – Інтерфейс інструменту AstroImageJ

Проте в ході аналізу було знайдено ряд недоліків, до яких можна віднести наступні:

– відсутність масштабованості: хоча AstroImageJ ефективний для окремих зображень або невеликих наборів даних, його здатність обробляти великі набори астрономічних даних може бути обмеженою. Для дослідників, які працюють з великими обсягами даних, це може спричинити значні затримки;

– складність інтерфейсу і наявність великого переліку функцій потребують додатковий час для вивчення та адаптації.

Ще одним з ефективних інструментів для обробки та покращення якості астрономічних зображень є пакет IRAF (Image Reduction and Analysis Facility). Цей пакет надає перелік інструментів для попередньої обробки, класифікації, аналізу та візуалізації даних, проте він має один значний недолік, розповсюджений серед існуючих інструментів в астрономічній спільноті, він полягає в тому, що цей інструмент є застарілим, і як зазначено в документації, кодова база інструменту IRAF існує вже 40 років, і розробники не рекомендують починати нові проекти використовуючи цей інструмент і пропонують продукт Astropy, як альтернативу. Взаємодія з цим пакетом традиційно виконується за допомогою командного рядка, що в результаті може стати перешкодою у використанні користувачами, які віддають перевагу GUI або не мають достатнього досвіду роботи з командним рядком, а складність команд та процедур можуть сповільнити процес опанування інструмента.

При нормальній взаємодії пакет IRAF функціонує як система з декількома процесами [8]. Типова структура процесів IRAF показана на рисунку 1.5: процес CL забезпечує всю взаємодію з користувацьким інтерфейсом, який зазвичай є графічним терміналом. Всі прикладні програми виконуються як паралельні підпроцеси процесу CL. Якщо потрібен графічний ввід/вивід на інші пристрої, окрім графічного терміналу, процес графічного ядра підключається до CL як підпроцес.

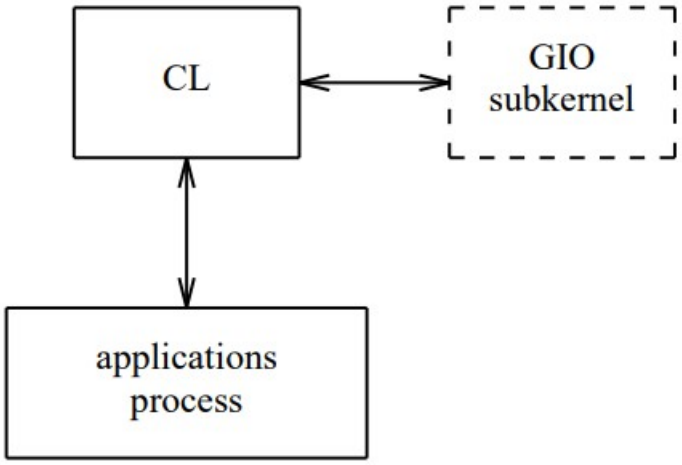


Рисунок 1.5 – Структура процесу IRAF

Як можна побачити, в пакеті IRAF розробники заклали базові принципи масштабування обробки астрономічних зображень за допомогою реалізації паралелізму в багатопроцесорному середовищі, це є перевагою цього пакету, проте зважаючи на відсутність повноцінної підтримки продукту є доцільним використання більш нових інструментів. Додатково слід зазначити, що не зважаючи на свою застарілість, пакет IRAF характеризується модульною архітектурою і може бути розширений за допомогою використання зовнішніх пакетів, що надає гнучкість у використанні цього інструменту за рахунок впровадження нових рішень та підходів в обробці та відновленні зображень.

Раніше згаданий продукт Astropy є потужною бібліотекою у мові програмування Python, що забезпечує широкий спектр інструментів для аналізу даних та моделювання. Бібліотека Astropy побудована на базі поширених бібліотек для наукових обчислень Python, таких як NumPy, SciPy та забезпечує продуктивне середовище для астрономічних досліджень. Активна спільнота Astropy та відкритий вихідний код підтримують безперервний розвиток і вдосконалення бібліотеки, а вичерпні навчальні посібники полегшують початок роботи для початківців.

Втім Astropy має суттєвий недолік, який полягає в тому, що це бібліотека, а не окремий незалежний сервіс або продукт, а отже вона є лише інструментом, який може бути інтегрований в інформаційні системи, або використаний для вузьких потреб в дослідницьких роботах у вигляді автоматизації окремих процесів або етапів обробки астрономічних зображень і відповідно потребує знання мови програмування Python для її використання.

В процесі аналізу вище зазначеного програмного забезпечення було помічено тенденцію, а саме поширеність десктопних застосунків в галузі обробки астрономічних зображень. Залежність від конкретних платформ помітно знижує потенціал для масштабування та обробки зображень через наявність обмеження ресурсів робочих машин, на якому ці застосунки встановлено.

Існують закриті або комерційні проекти, які надають потужні

інструменти для аналізу, обробки та відновлення астрономічних зображень за допомогою передових алгоритмів і великої обчислювальної потужності, що вирішує проблеми масштабування, якими володіють десктопні застосунки. Особливостями закритих проектів для обробки астрономічних зображень є наступні:

– використання хмарних технологій, що забезпечує значну обчислювальну потужність і дозволяє ефективно та швидко обробляти великі обсяги даних;

– наявність розширеного переліку алгоритмів, що використовують передові досягнення штучного інтелекту та машинного навчання для відновлення та покращення якості зображення та виявлення астрономічних об’єктів;

– можливість використання спеціалізованих інструментів для конкретних астрономічних завдань, таких як корекція атмосферних спотворень, калібрування зображення, виявлення об’єктів тощо.

Проте такі проекти мають наступні недоліки:

– вартість: для доступу до розширених функцій може знадобитися платна підписка або одноразовий платіж. Це може створити перешкоди для окремих дослідників або некомерційних організацій;

– конфіденційність даних: використання подібних сервісів вимагає передачі даних третім сторонам, що може викликати питання щодо безпеки та конфіденційності даних;

– залежність від постачальника сервісу: фактор, що проявляється в разі зміни умов обслуговування або припинення дії плану сервісу.

Отже, з огляду існуючих рішень, стає зрозуміло, що потреба в розробці інноваційних, більш доступних інструментів для обробки астрономічних зображень є важливою. Нові рішення повинні бути спрямовані на подолання існуючих обмежень, зокрема на забезпечення масштабованості, зниження вимог до технічної підготовки користувачів і покращення доступності. Також важливо зосередитися на розробці інтерфейсів, які дозволяють користувачам

легко адаптуватися до нових інструментів, скорочуючи час навчання та адаптації [9]. При цьому необхідно враховувати питання конфіденційності та безпеки даних і забезпечувати захист інформації під час обробки.

1.3 Аналіз ролі хмарних обчислень у процесах відновлення астрономічних зображень

Процес відновлення астрономічних зображень є потужним прикладом, який ілюструє як потенціал, так і проблеми, пов’язані з розширенням обробки великих обсягів даних. У контексті цього завдання можна виділити два основні аспекти, які вимагають масштабування: перший - це масштабування обчислювальних ресурсів, необхідних для ефективного відновлення зображень, а другий - це масштабування систем зберігання даних, які використовуються як для збереження вихідних зображень, так і для відновлених астрономічних зображень [10].

У цьому контексті масштабування обчислень передбачає використання високопродуктивних обчислювальних систем і алгоритмів паралельної обробки для ефективної обробки великих наборів даних астрономічних спостережень. Це дозволяє скоротити час обробки зображення та підвищити точність відновлення, що важливо для астрономічних досліджень, де кожен піксель може містити важливу інформацію.

Існує два ключових підходи для масштабування обчислювальної потужності за допомогою покращення інфраструктури обчислювальних систем а саме: горизонтальний та вертикальний.

Горизонтальне масштабування передбачає додавання більшої кількості вузлів до пулу ресурсів для паралельної обробки більшої кількості завдань. Особливо це стосується обробки астрономічних зображень, де кожне зображення можна обробляти незалежно, що дозволяє ефективно розподіляти навантаження між вузлами.

Вертикальне масштабування в свою чергу передбачає підвищення

продуктивності існуючих вузлів шляхом додавання більш потужних процесорів, більшої пам’яті або графічних прискорювачів (наприклад, GPU). В такому випадку можна підвищити ефективність алгоритмів відновлення зображень на одній обчислювальній машині без потреби в запуску нових вузлів.

Наочне порівняння двох типів масштабування подано на рисунку 1.6.

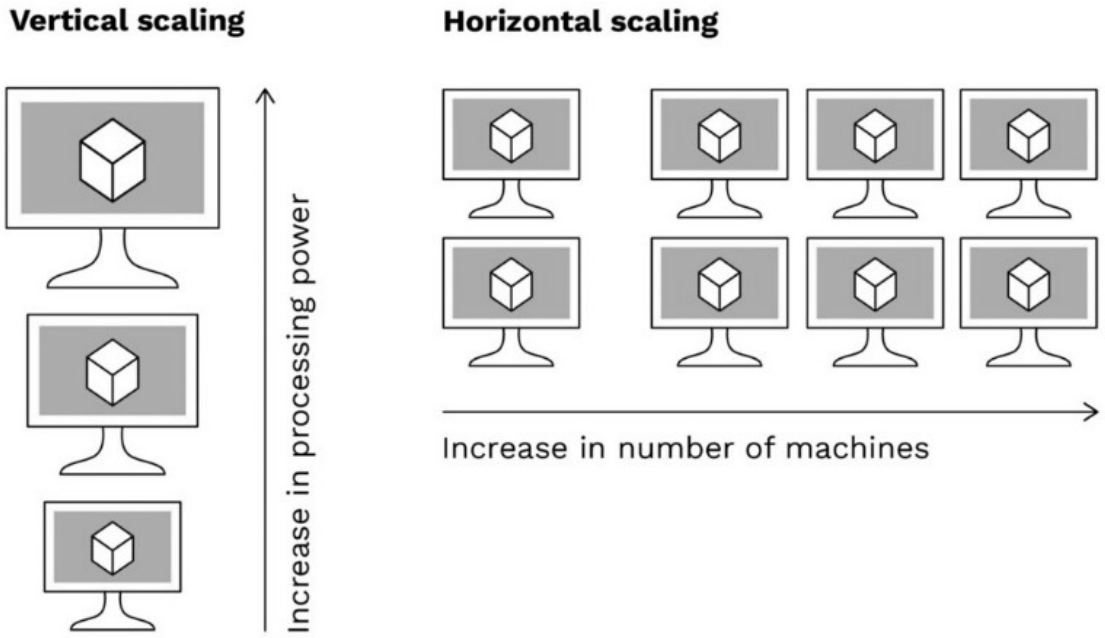


Рисунок 1.6 – Порівняння вертикального та горизонтального масштабування обчислювальних ресурсів

На додаток до проблеми масштабування обчислень, задача масштабування зберігання даних є не менш важливою у контексті відновлення астрономічних зображень [11]. Астрономічні дослідження генерують великі обсяги зображень з високою роздільною здатністю та об’ємом, які потребують не лише ефективної обробки, але й надійного масштабованого зберігання. Сучасні хмарні платформи пропонують різноманітні рішення для зберігання даних, які можна адаптувати до потреб астрономічних проектів різного масштабу.

В рамках інформаційної системи відновлення зображень присутні два

основі типи даних: файли зображень у їх вихідному форматі та пов’язані з ними метадані, що можуть містити важливу інформацію, таку як деталі обробки зображень, історію змін і т.д. Додатково в інформаційних системах такого типу велика увага приділена зберіганню даних користувачів, і до цього висувається додатковий ряд безпекових вимог для запобігання втрати даних та несанкціонованого доступу до них.

На прикладі провайдеру хмарних сервісів Amazon Web Services (AWS) можна побачити різні підходи до масштабованого зберігання даних, найпопулярнішими з яких є Amazon Relational Database Service (RDS), Amazon Simple Storage Service (S3) і Elastic Block Store (EBS).

Серед викликів, пов'язаних з масштабуванням зберігання даних в інформаційних системах, Amazon Relational Database Service (RDS) пропонує ефективне рішення, спрямоване на оптимізацію процесів управління базами даних, забезпечуючи при цьому високу доступність і масштабованість. Amazon RDS дозволяє користувачам легко налаштовувати, керувати та масштабувати реляційні бази даних у хмарі, а також автоматизувати багато завдань, таких як резервне копіювання та реплікація [12]. Саме Amazon RDS є оптимальним вибором для зберігання метаданих та даних користувачів в контексті інформаційних систем, що займаються великими об'ємами даних і потребують високої надійності, доступності та масштабованості, що актуально для інформаційної системи відновлення зображень.

Говорячи про збереження файлів астрономічних зображень, це можна забезпечити за допомогою використання сервісу Amazon S3, що є високонадійним рішенням для зберігання даних, яке можна масштабувати та розширювати для великих обсягів даних без обмежень щодо розміру чи кількості файлів. Цей сервіс ідеально підходить для зберігання необроблених астрономічних зображень, оброблених результатів та інших типів наукових даних, які потребують високої доступності та тривалого зберігання.

Аналогом сервісу AWS S3 є сервіс Amazon EBS, який в свою чергу пропонує блочне сховище для використання з екземплярами Amazon EC2, що

дозволяє масштабувати сховище даних та його продуктивність незалежно від обчислювальних ресурсів. Типова взаємодія з сервісом Amazon EBS показана на рисунку 1.7.

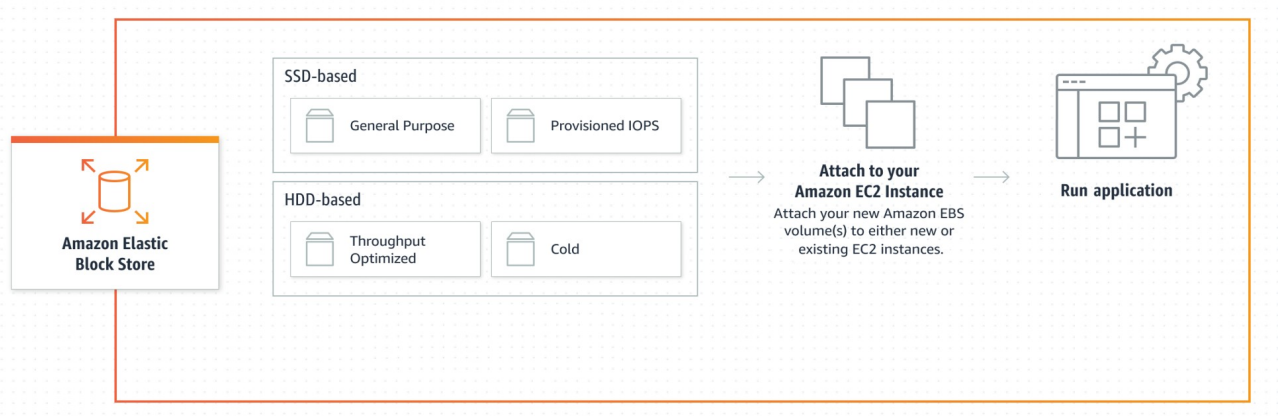


Рисунок 1.7 – Структура використання сервісу Amazon EBS

AWS EBS надає можливість швидшого запису та читання даних у порівнянні з AWS S3. Класичним випадком використання EBS є його використання у зв’язці з віртуальними машинами Elastic Compute Cloud (EC2), що в результаті надає можливості вертикального масштабування зосередженого на використанні однієї віртуальної машини.

Важливим аспектом масштабування сховищ даних в хмарних середовищах є можливість автоматичного керування ресурсами без втручання користувача в цей процес. Це включає в себе автоматичне розширення сховища в міру збільшення обсягу даних і оптимізацію витрат на зберігання за рахунок використання різних класів сховищ даних.

Використання хмарних обчислень для відновлення астрономічних зображень може бути економічно ефективним рішенням, оскільки воно може оптимізувати витрати на обчислювальні ресурси. Розрахункова модель сучасних провайдерів хмарних технологій дозволяє дослідницьким групам масштабувати обчислювальні ресурси відповідно до потреб проекту, усуваючи необхідність інвестувати в дорогі локальні центри обробки даних.

Хмарні платформи, такі як AWS, Google Cloud Platform і Microsoft Azure,

спрощують доступ до передових обчислювальних ресурсів, включаючи обчислювальні кластери на основі GPU та спеціалізоване програмне забезпечення. Інтеграція з хмарними сервісами забезпечує можливість швидкого аналізу та обробки астрономічних зображень, сприяючи прискоренню наукових відкриттів в астрономії. Завдяки потенціалу економічної ефективності та оптимізації витрат хмарні обчислення є ідеальним рішенням для проектів різного розміру, що дозволяє зменшувати обчислювальні ресурси відповідно до потреб досліджень.

Загалом, сучасні хмарні сервіси використовують загальні принципи, які надають можливість налаштування обчислювальних станцій та системи зберігання даних без зайвих витрат часу та фінансів. Це є одним з ключових факторів особливо для міжнародних наукових проектів і співпраць, де дослідники з різних куточків світу можуть легко отримувати доступ до необхідних даних і обчислювальних ресурсів.

Використовуючи хмарні технології, можливо не лише досягати високих результатів у відновленні та аналізі астрономічних зображень, але й забезпечувати ефективне зберігання та обробку величезних обсягів наукових даних, сприяючи розвитку астрономічної науки.

1.4 Визначення вимог до компонентів інформаційної системи відновлення зображень

Під час аналізу існуючих інструментів і систем для обробки астрономічних зображень було виявлено низку важливих недоліків, які потенційно можливо усунути в процесі розробки поточної інформаційної системи для відновлення зображень.

До основних недоліків можна віднести обмежені можливості масштабування, складність інтерфейсу, залежність від специфічного апаратного та програмного забезпечення, а також високу потребу в технічній підготовці користувачів. На основі цих спостережень були розроблені вимоги до

компонентів нової системи, включаючи як функціональні, так і нефункціональні аспекти.

В рамках розробки інформаційної системи відновлення зображень було виділено дві ролі користувачів: неавторизований користувач та авторизований користувач. Таке рішення було прийнято з метою спрощення системи контролю доступу та забезпечення базового рівня безпеки на ранніх етапах розробки проекту.

Рішення обмежити систему двома ролями було прийнято на основі аналізу потреб користувачів та рівня готовності проекту. Основними цілями на цьому етапі були забезпечення функціональності ядра системи та визначення основного напрямку розвитку. Обмеження доступу до проекту через систему ролей є ефективним засобом захисту інформації та ресурсів від потенційних загроз, а також дозволяє орієнтувати розвиток ключових функцій системи на вимоги та потреби авторизованих користувачів [13].

Такий підхід також спрощує процес управління доступом і безпекою на ранніх стадіях розробки проекту і забезпечує стабільну основу для подальшого розширення функціоналу та адаптації системи до конкретних потреб користувачів у майбутньому.

Для кожної з вище зазначених ролей були розроблені специфічні функціональні вимоги, які відображають їх можливості в межах системи. Відповідні вимоги для кожної ролі користувачів подано на Use Case діаграмі інформаційної системи відновлення зображення, яка подана у додатку А на рисунку А.1 [14].

Як раніше було зазначено, неавторизовані користувачі мають обмежений доступ до функцій системи. Користувачі цієї категорії мають наступні функції:

– реєстрація в системі: неавторизовані користувачі можуть створити новий обліковий запис, надавши необхідну інформацію, таку як електронна адреса та пароль;

– вхід в систему: після реєстрації користувачі можуть увійти в систему, використовуючи свої облікові дані, і отримати доступ до додаткових функцій,

доступних авторизованим користувачам.

Авторизовані користувачі мають доступ до широкого спектру функцій, які дозволяють їм ефективно використовувати систему для відновлення зображень. Авторизовані користувачі мають доступ до наступних функцій:

– створення нових завдань на відновлення зображень: авторизовані користувачі можуть запустити процес відновлення зображень, завантаживши зображення і вказавши необхідні параметри обробки за можливості;

– отримання списку всіх завдань: користувачі можуть переглянути список всіх завдань, створених в системі, і відстежувати стан їх обробки;

– отримання детальної інформації про кожне завдання: можливість перегляду детальної інформації про кожне завдання, включаючи параметри обробки та хід виконання;

– завантаження результатів обробки зображень: після завершення обробки користувач може завантажити результати у вигляді ZIP архіву;

– видалення завдання відновлення зображення;

– вихід з системи.

Розробка цих функціональних вимог ґрунтується на необхідності забезпечення базового рівня безпеки та контролю доступу до системи, одночасно надаючи користувачам гнучкі інструменти для роботи з астрономічними зображеннями. Обмеження доступу для неавторизованих користувачів дозволяє зменшити ризик несанкціонованого доступу до чутливих даних та системних ресурсів, тоді як надання розширених можливостей авторизованим користувачам спрямовано на максимізацію ефективності використання системи для наукових та дослідницьких цілей.

Для забезпечення ефективності, надійності та зручності інформаційної системи відновлення зображень були визначені наступні нефункціональні вимоги:

– платформа: система повинна бути реалізована у вигляді клієнт-серверного веб-застосунку;

– продуктивність та час відгуку: система повинна забезпечувати високу продуктивність та швидкий час відгуку при обробці завдань відновлення зображень. Важливо, щоб система могла ефективно використовувати обчислювальні ресурси, забезпечуючи оптимальну швидкість обробки даних;

– масштабованість: система повинна бути спроектована таким чином, щоб її можна було легко масштабувати по горизонталі та вертикалі при збільшенні кількості користувачів та обсягів даних. Це включає в себе можливість адаптації до зростаючих робочих навантажень без шкоди для продуктивності або якості обслуговування;

– безпека: важливо забезпечити конфіденційність, цілісність і доступність даних користувачів. Система повинна включати механізми аутентифікації, авторизації, шифрування даних і захисту від зовнішніх і внутрішніх загроз;

– надійність та доступність: системи повинні забезпечувати високий рівень надійності та безперервної роботи, з автоматизованим резервним копіюванням та відновленням;

– модульність і масштабованість: архітектура системи повинна бути модульною, що дозволяє легко додавати, модернізувати або замінювати компоненти системи без порушення роботи всієї системи;

– інтеграція: система повинна мати можливість інтегруватися з іншими додатками та сервісами за допомогою API та стандартних протоколів обміну даними;

– простота використання та підтримка: користувацький інтерфейс повинен бути інтуїтивно зрозумілим і простим у використанні для користувачів з різним рівнем кваліфікації;

– сумісність: система має бути сумісною з низкою операційних систем і платформ та доступною для широкого кола користувачів.

2 РОЗРОБКА МЕТОДУ ВІДНОВЛЕННЯ АСТРОНОМІЧНИХ ЗОБРАЖЕНЬ

2.1 Алгоритм відновлення астрономічних зображень на основі методу Люсі-Річардсона

У контексті проблеми розмиття астрономічних зображень, яка була детально розглянута раніше, виникає необхідність вибору алгоритму їх відновлення. Ця проблема дуже актуальна для астрономії, де точність і чіткість зображень мають вирішальне значення для наукових досліджень і освоєння космосу. Існує багато методів відновлення зображень, кожен зі своїми особливостями, перевагами та обмеженнями. Однак після ретельного аналізу та порівняння різних підходів алгоритм деконволюції Люсі-Річардсона був обраний як найбільш підходящий для цього завдання [15].

Вибір алгоритму Люсі-Річардсона залежить від кількох ключових факторів. По-перше, цей алгоритм спеціально розроблено для роботи із зображеннями, розмитими за допомогою відомої функції точкового розсіювання (ФТР), що є типовою ситуацією для астрономічних зображень. По-друге, цей алгоритм є ітераційним, що дозволяє контролювати процес відновлення зображення з метою досягнення оптимального балансу між чіткістю та збереженням природності зображення. Нарешті, алгоритм демонструє високу ефективність у видаленні розмиття без створення значних артефактів, що робить його ідеальним рішенням для відновлення астрономічних зображень, де кожна деталь має значення [16].

Зазначена функція точкового розсіювання описує візуальний вигляд точкового джерела світла (наприклад, далекої зірки, що спостерігається через телескоп) на пристрої для створення зображення, такому як CCD-камера, враховуючи оптичні ефекти системи та інші чинники [17]. Внаслідок цих ефектів світло не фіксується як ідеальна точка, а розсіюється, формуючи на зображенні нечітку пляму. Розмір і форма цієї плями, яка визначається через

функцію розсіювання, залежать від різних чинників:

– оптична система: недоліки в лінзах, дзеркалах та інших компонентах оптичної системи можуть спричинити розсіювання світла, що призводить до ширшої ФТР;

– атмосферні умови: для наземних телескопів, зміни в атмосфері (наприклад, турбулентність) можуть спотворювати вхідне світло, впливаючи на ФТР;

– інструментальні фактори: характеристики самого датчика зображення, включаючи розмір і форму пікселів, можуть впливати на ФТР;

– дифракція: хвильова природа світла змушує його дифрагувати навколо країв апертур телескопа, сприяючи формуванню ФТР.

У обробці зображень та техніках деконволюції знання ФТР дозволяє коригувати або значно зменшувати ці ефекти розмиття, маючи на меті відновлення зображення до його оригінального, нерозмитого стану.

Спостережуване зображення можна розкласти у вигляді суми окремих точок і виразити через матрицю переходу p, яка діє на базове зображення:

$$d_i = \sum_j p_{i,j} u_j, \tag{2.1}$$

де d_i – інтенсивність пікселя вихідного зображення;

$p_{i,j}$ – елемент матриці переходу, що виражає зміщення між початковим пікселем j та вихідним і;

u_j – інтенсивність пікселя вхідного зображення j.

Матриця переходу може бути виражена як здвиг між початковим пікселем та вихідним за наступною формулою:

$$p_{i,j} = P(i - j), \tag{2.2}$$

де $P(i - j)$ – функція точкового розсіювання.

У контексті алгоритму деконволюції Люсі-Річардсона ФТР є критично

важливим вхідним сигналом, оскільки він безпосередньо впливає на процес деконволюції. Алгоритм використовує ФТР для усунення ефектів розмиття шляхом повторного уточнення оцінки справжнього зображення. Алгоритм особливо добре підходить для астрономічних зображень, де першочерговою є потреба відновити якомога більше просторової інформації від віддалених небесних об'єктів.

Суть алгоритму Люсі-Річардсона полягає в його ітераційному підході для оцінки вихідного зображення шляхом мінімізації різниці між спостережуваним зображенням і зображенням, згорнутим із ФТР. Математично процес можна описати наступною формулою:

$$A_{deb}^{t+1} = A_{deb}^{t}\left(h'_{PSF} \otimes \frac{A_{out}}{h_{PSF} \otimes A_{deb}^{t}}\right), \tag{2.3}$$

де A_{deb}^{t} – відновлене зображення на t ітерації;

A_{out} – спотворене зображення;

h_{PSF} – функція точкового розсіювання;

h'_{PSF} – зворотна функція точкового розсіювання;

$\otimes$ – операція згортки.

Алгоритм починається з початкового наближення до справжнього зображення, часто із самого спостережуваного зображення. На кожній ітерації алгоритм уточнює це припущення, застосовуючи наведену вище формулу, ефективно підвищуючи різкість зображення шляхом компенсації відомих спотворень, внесених ФТР. Ітераційний процес триває, доки алгоритм не досягне стабільного рішення, яке зазвичай визначається попередньо визначеною кількістю ітерацій, або коли покращення між ітераціями падає нижче певного порогу.

Однією з сильних сторін алгоритму Люсі-Річардсона є його здатність покращувати зображення без значного посилення шуму, що часто зустрічається при деконволюції зображення. Ця властивість особливо цінна в астрономії, де

співвідношення сигнал/шум у зображеннях може бути низьким, а збереження цілісності даних має вирішальне значення.

Застосовуючи деконволюцію Люсі-Річардсона до астрономічних зображень, можна виявити деталі, які раніше були затемнені ефектом розмиття, підвищуючи наукову цінність спостережень. Цей алгоритм дозволяє виділяти дрібніші просторові деталі із зображень небесних об’єктів, сприяючи більш точним вимірюванням і сприяючи глибшому розумінню їхніх фізичних властивостей і поведінки.

Математичний модуль, що реалізує алгоритм Люсі-Річардсона скомпільований під платформу Linux у вигляді бінарного файлу, що використовує файл налаштувань поданий у лістингу 2.1.

```
<?xml version="1.0" encoding="UTF-8"?>
<lucy>
    <pathInFrame>/image-recovery/data/input.fits</pathInFrame>
    <pathInSegments></pathInSegments>
    <pathOutFrame>/image-recovery/data/output.fits</pathOutFrame>
     <pathOutSegments>/image-recovery/data/segmentsOut.bin</pathOut
Segments>
    <pathRes></pathRes>
</lucy>
```

Рисунок 2.1 – Файл налаштувань методу Люсі-Річардсона

Можна побачити, що у файлі налаштувань задаються шляхи до вхідних та вихідних файлів.

На рисунку 2.2 подано результат застосування методу алгоритму Люсі-Річардсона на астрономічному зображенні небесних тіл.

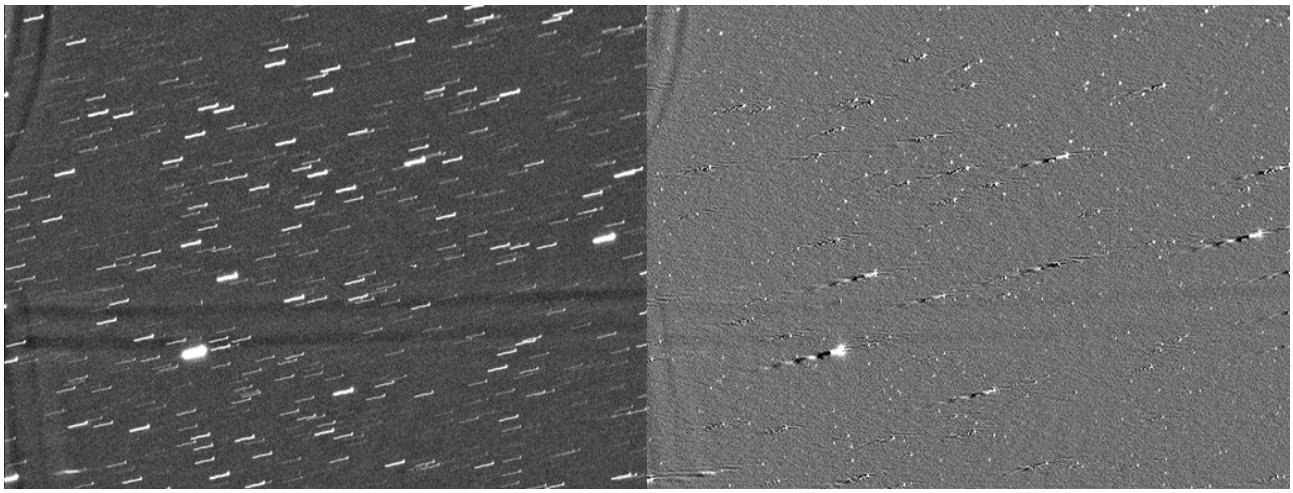


Рисунок 2.2 – Результат застосування алгоритму Люсі-Річардсона

Як можна побачити з отриманих результатів, ключовий негативний фактор погіршення якості зображення, а саме розмиття, було усунуто. Побічним ефектом, що з’явився в результаті відновлення виявилась поява бокових артефактів навколо зірок на астрономічному зображенні, які можна побачити на проекції 3D представлення відновленого астрономічного зображення (див. 2.3).

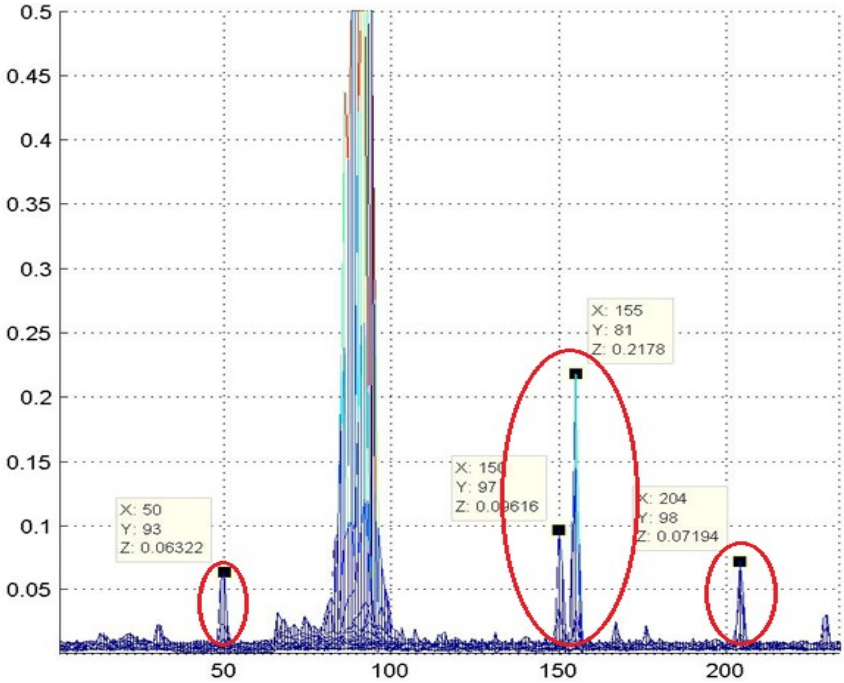


Рисунок 2.3 – Проекція 3D представлення відновленого зображення з визначеними боковими артефактами

2.2 Розробка комбінованого підходу до відновлення астрономічних зображень

У додаток до методу Люсі-Річардсона було використано попередню обробку у вигляді вирівнювання яскравості за допомогою медіанного фільтра. Медіанний фільтр є важливим інструментом для попередньої обробки зображень, зокрема для видалення фонового засвічення і неоднорідності яскравості, які неможливо ефективно виправити за допомогою стандартних методів калібрування. Проблема нерівномірності яскравості актуальна при обробці астрономічних зображень, де вимоги до рівномірності фону дуже високі, особливо в тих випадках, коли видима яскравість об'єктів слабка [18].

Медіанний фільтр дозволяє ефективно видаляти масштабний шум і відблиски, зберігаючи краї і деталі основних об'єктів. Він обробляє кожен піксель зображення, замінюючи його значення медіаною інтенсивності у визначеній локальній околиці. Загалом використання медіанного фільтру може бути описано наступною формулою:

$$A_{out}(m,n) = A_{\square}(m,n) - A_{med}(m,n), \tag{2.4}$$

де A_{out} – результуюче зображення;

$A_{\square}$ – вхідне зображення;

A_{med} – значення медіанного значення пікселів в оточенні пікселя m, n;

m та n – власне індекси пікселя.

Отримане зображення з вирівняною фоновою яскравістю може мати пікселі з від’ємним значенням, тож додатково слід виконати корекцію та відняти мінімальне значення з поміж усіх пікселів від кожного пікселя:

$$A_{out}(m,n) = A_{\square}(m,n) - A_{min} \tag{2.5}$$

Такий підхід дозволяє звести до мінімуму вплив випадкових відблисків

або нерівномірного освітлення на якість фону зображення і тим самим підвищити точність подальшого відновлення зображення за допомогою методу Люсі-Річардсона.

Використання медіанного фільтра перед застосуванням методу Люсі-Річардсона є виправданим, оскільки воно дозволяє знизити рівень шуму та збільшити відношення сигнал/шум, спростити завдання реконструкції зображення та підвищити якість кінцевих результатів. Медіанна фільтрація надає можливість виправити великі, але локалізовані спотворення яскравості, які можуть бути викликані різними зовнішніми факторами, особливо техногенним освітленням або характеристиками умов спостереження.

По аналогії з математичним модулем реалізації методу Люсі-Річардсона, медіанний фільтр скомпільований у вигляді бінарного файлу, що використовує файл налаштувань, який можна побачити у лістингу 2.2. На поточному етапі розмір вікна медіанної фільтрації було підібрано експериментальним шляхом.

```
<FhfSettings>
  <logAndInitFile>
      <FullNameLogFile>/image-recovery/data/output.fits.log</FullNameLogFile>
                <FullNameInitFile>/image-recovery/bin/initfhf.xml</FullNameInitFile>
                <FullNameResFile>/image-recovery/data/output.res</FullNameResFile>
  </logAndInitFile>
  <inversMedianFilter>
    <IsInversMedianFilter>1</IsInversMedianFilter>
    <FullNameFitInInversMedianFilter>/image-recovery/data/output.fits</FullNameFitInInversMedianFilter>
    <FullNameFitOutInversMedianFilter>/image-recovery/data/res_output.fits</FullNameFitOutInversMedianFilter>
    <InversMedianWindow>51</InversMedianWindow>
    <ColorOrBw>1</ColorOrBw>
    <SmallOrBig>0</SmallOrBig>
    <Binning>3</Binning>
    <HorizontalBandsFilter>false</HorizontalBandsFilter>
  </inversMedianFilter>
</FhfSettings>
```

Рисунок 2.4 – Файл налаштувань методу вирівнювання яскравості

Як можна побачити у поданому файлі налаштувань присутні параметри для використання медіанного фільтру та шлях до вхідних та вихідних файлів.

Як можна побачити з раніше описаних файлів налаштувань для методу

Люсі-Річадрсона і методу вирівнювання яскравості за допомогою медіанного фільтра, обидва вони використовують формат файлів FITS. Цей формат став стандартом де-факто для зберігання, передачі та обробки астрономічних зображень і даних, включаючи як самі зображення, так і пов’язані з ними метадані [19].

Конвеєр відновлення зображення з використанням вище описаних методів для зображення у форматі FITS подано на рисунку 2.5. Слід зауважити, що при такому підході алгоритм відновлення зображення застосовується одразу до його початковій версії у форматі FITS.

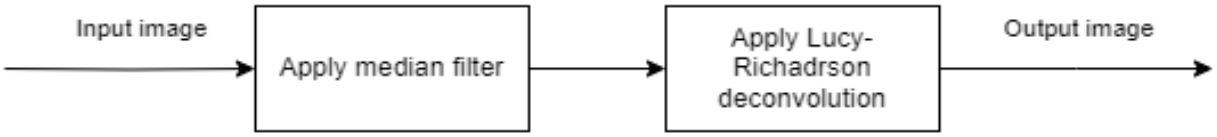


Рисунок 2.5 – Конвеєр обробки зображення у форматі FITS

Формат FITS відіграє важливу роль в астрономії та астрофізиці, його унікальність полягає у високій гнучкості та здатності зберігати різні типи даних, включаючи зображення, таблиці та векторні карти. Однією з головних переваг цього формату є можливість інтегрувати метадані безпосередньо з даними, що дозволяє користувачам зберігати детальну інформацію про астрономічні спостереження, таку як час, місцезнаходження та параметри інструментів.

Крім того, FITS також підтримує високоточні числові дані. Це важливо для астрономічних досліджень, де точність вимірювань впливає на важливі наукові результати. Така точність досягається використанням різних форматів для зберігання числових даних, що дозволяє вченим вибрати формат, який найкраще відповідає їхнім потребам.

Однак використання FITS як єдиного формату для астрономічних даних має недоліки, особливо з точки зору доступності та сумісності. Багато з

найпоширеніших програм для перегляду та обробки зображень не підтримують формат FITS, що в свою чергу ускладнює широкий доступ до астрономічних даних серед науковців, що не спеціалізуються на астрономії. Така обмежена сумісність може перешкоджати міждисциплінарним дослідженням і обміну знаннями.

Зважаючи на ці обмеження, адаптація алгоритму відновлення зображення для роботи з різними форматами стає важливим кроком до підвищення зручності використання астрономічних даних.

В ході дослідження було здійснено ретельний відбір відповідного інструменту для конвертації зображень з урахуванням необхідності збереження високої якості зображень та врахування специфіки існуючих розповсюджених форматів файлів.

Після ретельного аналізу існуючих інструментів було прийнято рішення на користь пакету ImageMagick для перетворення зображень [20]. Цей вибір обґрунтований його гнучкістю та широким спектром функціональних можливостей. ImageMagick надає підтримку різних форматів зображень, включаючи JPEG, PNG, TIFF і власне FITS, і дозволяє легко конвертувати зображення між різними форматами. Під час конвертації ImageMagick забезпечує збереження високої якості зображень, що є критично важливим для точності наукового аналізу та інтерпретації астрономічних даних.

На рисунку 2.6 показано оновлений конвеєр обробки зображення з урахуванням початкової конвертації звичайного формату зображення у формат FITS, і вихідної конвертації з формату FITS в початковий формат зображення.

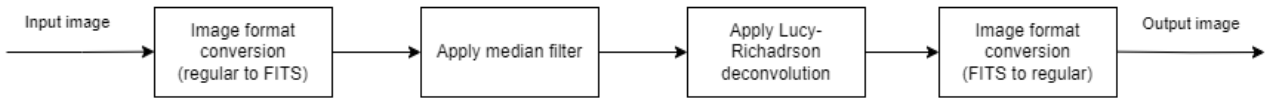


Рисунок 2.6 – Конвеєр обробки зображення у звичайному форматі

3 ПРОЄКТУВАННЯ ТА РОЗРОБКА АРХІТЕКТУРИ КОМПОНЕНТІВ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВІДНОВЛЕННЯ ЗОБРАЖЕНЬ

3.1 Проєктування компонентів інформаційної системи відновлення зображень

З метою визначення структури інформаційної системи та уточнення функціональних вимог для подальшого проектування архітектури системи було проведено об’єктне та структурне моделювання системи за допомогою графічної мови UML. Використання UML для проектування компонентів системи відновлення зображень дозволило детально описати різні аспекти системи, такі як структуру системи у вигляді класів та зв’язків між ними, поведінку об’єктів в системі, та динаміку процесів, що протікають у системі.

Для більш детального розуміння процесу відновлення зображення у системі, на основі раніше розробленої діаграми прецедентів, було створено перелік діаграм послідовностей (Sequence Diagram), які дозволяють визначити сутності, що присутні в системі, та данні, що циркулюють у системі.

На рисунку 3.1 подана діаграма послідовностей для процесу створення нового завдання відновлення зображення. Діаграма ілюструє виклики методів об’єктів виділених класів та повідомлення, що передаються у системі між ними, в ході створення нового завдання за допомогою використання веб-клієнту, REST API та пов'язаних з ними сервісів.

Створення завдання починається з ініціації процесу авторизованим користувачем шляхом відправки форми, що містить інформацію про нове завдання та зображення, які повинні бути відновлені, за допомогою веб-клієнту у браузері.

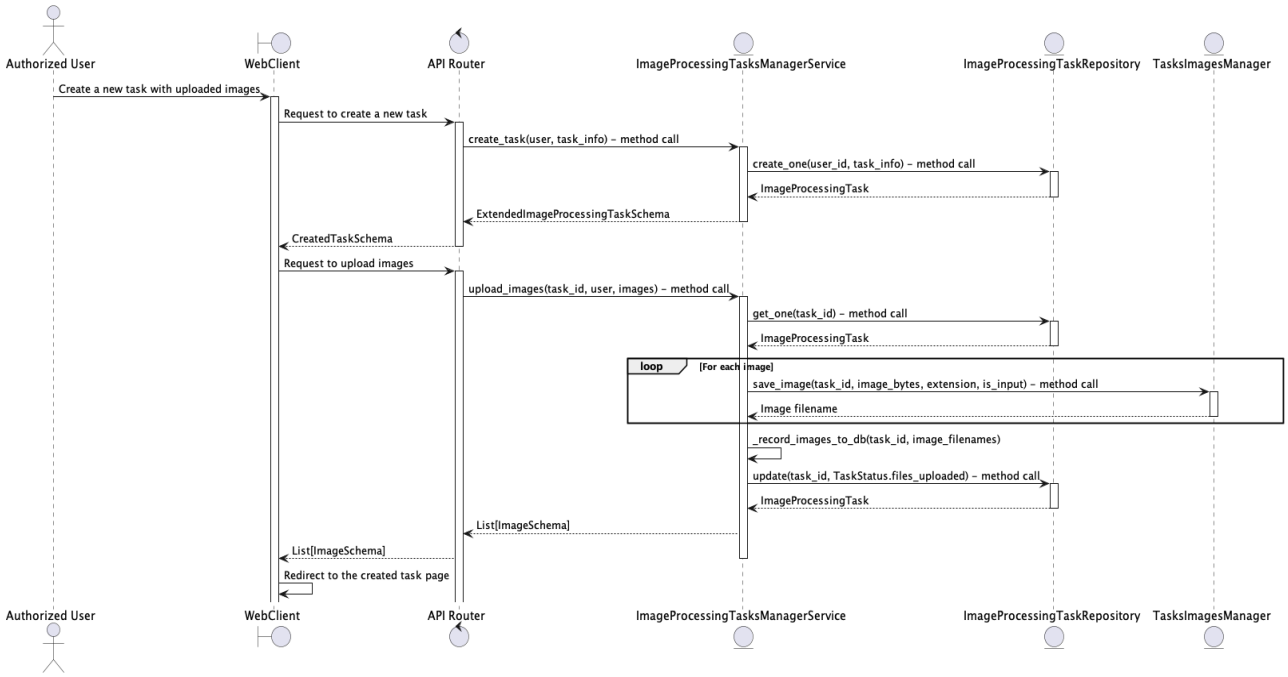


Рисунок 3.1 – Діаграма послідовностей для процесу «Створення нового завдання відновлення зображень в системі»

Алгоритм створення завдання в системі виглядає наступним чином:

– запит на створення завдання: авторизований користувач надсилає запит на створення нового завдання, включно з завантаженими зображеннями через веб-клієнт;

– збереження завдання: «API Router» надсилає запит до сервісу «ImageProcessingTasksManagerService», який викликає метод «create_task» для створення завдання в системі. Цей сервіс, у свою чергу, взаємодіє з репозиторієм «ImageProcessingTaskRepository» для збереження завдання в базі даних за допомогою методу «create_one»;

– реєстрація завдання: після створення завдання репозиторій повертає об'єкт класу «ImageProcessingTask» назад до сервісу «ImageProcessingTasksManagerService», який потім повертає розширену схему завдання «ExtendedImageProcessingTaskSchema» в «API Router», яка містить додані метадані. В результаті дані про створене завдання повертаються назад до веб-клієнта, включно з ідентифікатором завдання, використовуючи який в подальшому буде виконано завантаження зображень;

– завантаження зображень: веб-клієнт надсилає новий запит для завантаження зображень для завдання. Цей запит перенаправляється до сервісу "ImageProcessingTasksManagerService", який знову звертається до репозиторію щоб отримати раніше створене завдання. Кожне зображення зберігається за допомогою сервісу «TasksImagesManager», в результаті чого, кожен файл зображення отримує унікальне ім'я та ідентифікатор в системі;

– оновлення статусу завдання: після збереження зображень "ImageProcessingTasksManagerService" оновлює статус завдання. В результаті назад до веб-клієнта передається інформація про збереженні зображення та їх ідентифікатори;

– перенаправлення користувача: веб-клієнт перенаправляє користувача на створену сторінку завдання, завершуючи процес.

Ця діаграма демонструє взаємодію між різними компонентами системи та наочно демонструє послідовність кроків, необхідних для створення завдання відновлення зображень в системі. Слід зазначити важливість присутності кількох запитів між клієнтом та сервером в ході створення завдання для відновлення зображень, а саме: запит на створення завдання та запит на завантаження зображень. Цей підхід було використано з огляду на надійність та потенційну можливість усунення помилок: у разі помилок під час завантаження зображень немає потреби заново створювати завдання. Запит на завантаження зображень може бути повторений без зміни базової інформації про завдання з відновлення зображень, що знижує складність усунення помилок та підвищує загальну надійність системи.

Для більш детального розуміння процесу початку обробки завдань було розроблено діаграму послідовностей, яка описує процес переходу завдань від стану, коли вони створюються в системі, до стану, коли вони є запланованими для виконання (див. 3.2). Головною інформацією, яку надає ця діаграма, є факт використання планувальника завдань, який кожну секунду перевіряє наявні у системі завдання та визначає ті, які перебувають в стані, коли користувач завантажив зображення, проте завдання ще не було відправлено на виконання.

Тобто система визначає перелік завдань, що очікують виконання, та відправляє їх в чергу завдань. Важливою в даному контексті є присутність черги завдань, завдяки якій система має потенціал до масштабування, що полягає в створенні механізму обробки завдань з черги.

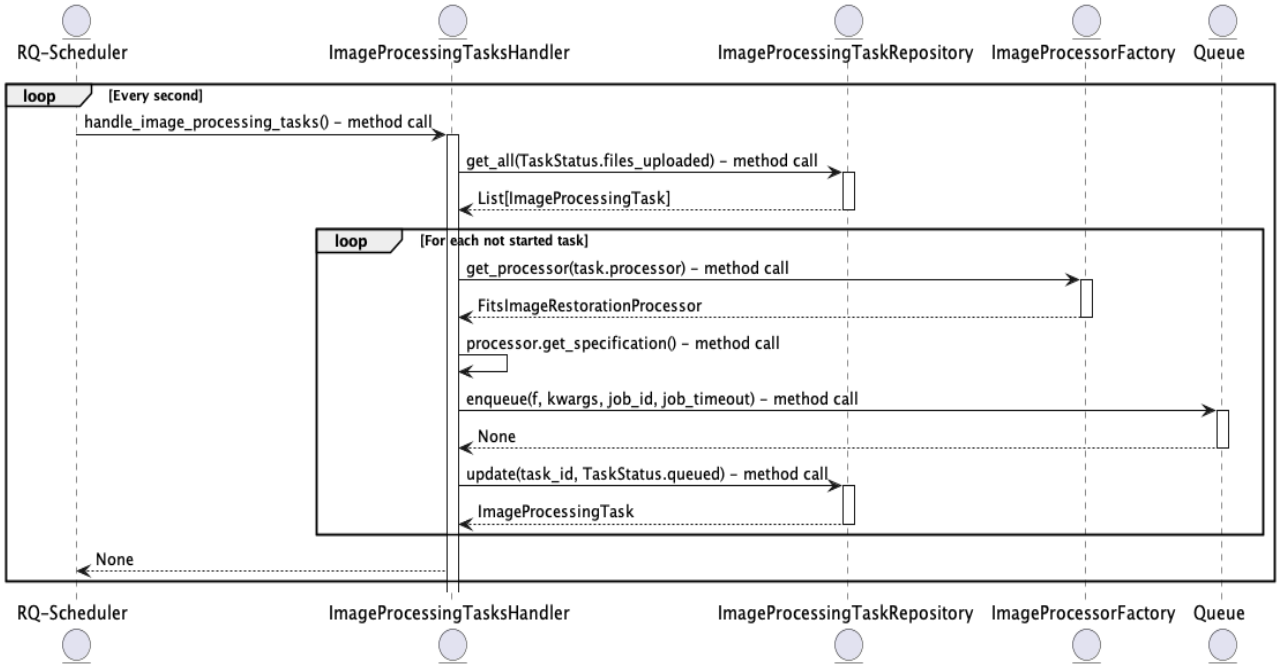


Рисунок 3.2 – Діаграма послідовностей для процесу «Розміщення завдань відновлення зображень у черзі виконання завдань»

Для розуміння процесу виконання завдань з відновлення зображень було створено діаграму послідовностей, яка подана на рисунку 3.3. Ця діаграма надає інформацію про те, як завдання можуть бути виконані у системі та які ресурси для цього потрібні. Процес виконання завдання можна розглядати як незалежний від системи процес, який може бути виконаний в будь якому середовищі, що відповідає наступним вимогам:

– середовище має доступ до читання та запису в базу даних, яка зберігає інформацію про завдання відновлення зображень та метадані, що пов’язані з зображеннями;

– середовище має доступ до читання та запису в сховище даних, де містяться зображення, для отримання вхідних зображень та збереження

результату.

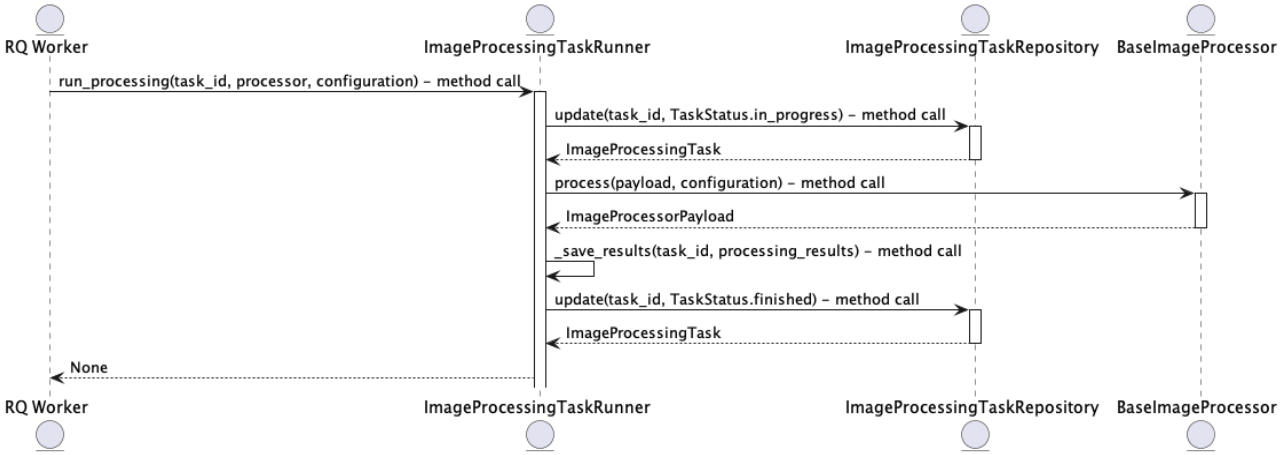


Рисунок 3.3 – Діаграма послідовностей для процесу «Виконання завдання відновлення зображень»

З раніше розроблених діаграм послідовностей можна визначити ключові стани завдання відновлення зображень в системі від моменту появи завдання системі. Усі можливі стани завдання відновлення зображень та переходи між ними можна побачити на діаграмі станів (див. 3.4). Було виділено наступні стани, в яких можуть перебувати завдання:

– «Created» – створено: завдання отримує стан створено при початковому збереженні в системі, в цей момент завдання не має жодних пов’язаних з ним файлів зображень;

– «Files uploaded» – файли завантажено: завдання переходить в цей стан в момент, коли користувач завантажив файли зображень до системи;

– «Queued» – розміщено в черзі: завдання переходить в цей стан коли система розміщує його у черзі задач для подальшого виконання;

– «In progress» - в процесі виконання: в цей стан завдання переходить після виключення з черги завдань та початку обробки зображень;

– «Finished» – вдало завершено: стан, який свідчить про успішне завершення виконання завдання з відновлення зображення;

– «Failed» – завершено з помилкою: стан, що інформує про невдале

виконання відновлення зображення, в цей стан завдання може перейти із стану «In progress», якщо в процесі обробки зображень виникнуть неочікувані помилки системи, а також в цей стан завдання може перейти із стану «Queued», коли воно відображається в базі даних як розміщено в черзі, проте в черзі завдань воно відсутнє, цей випадок є крайовим і може зустрітися досить рідко.

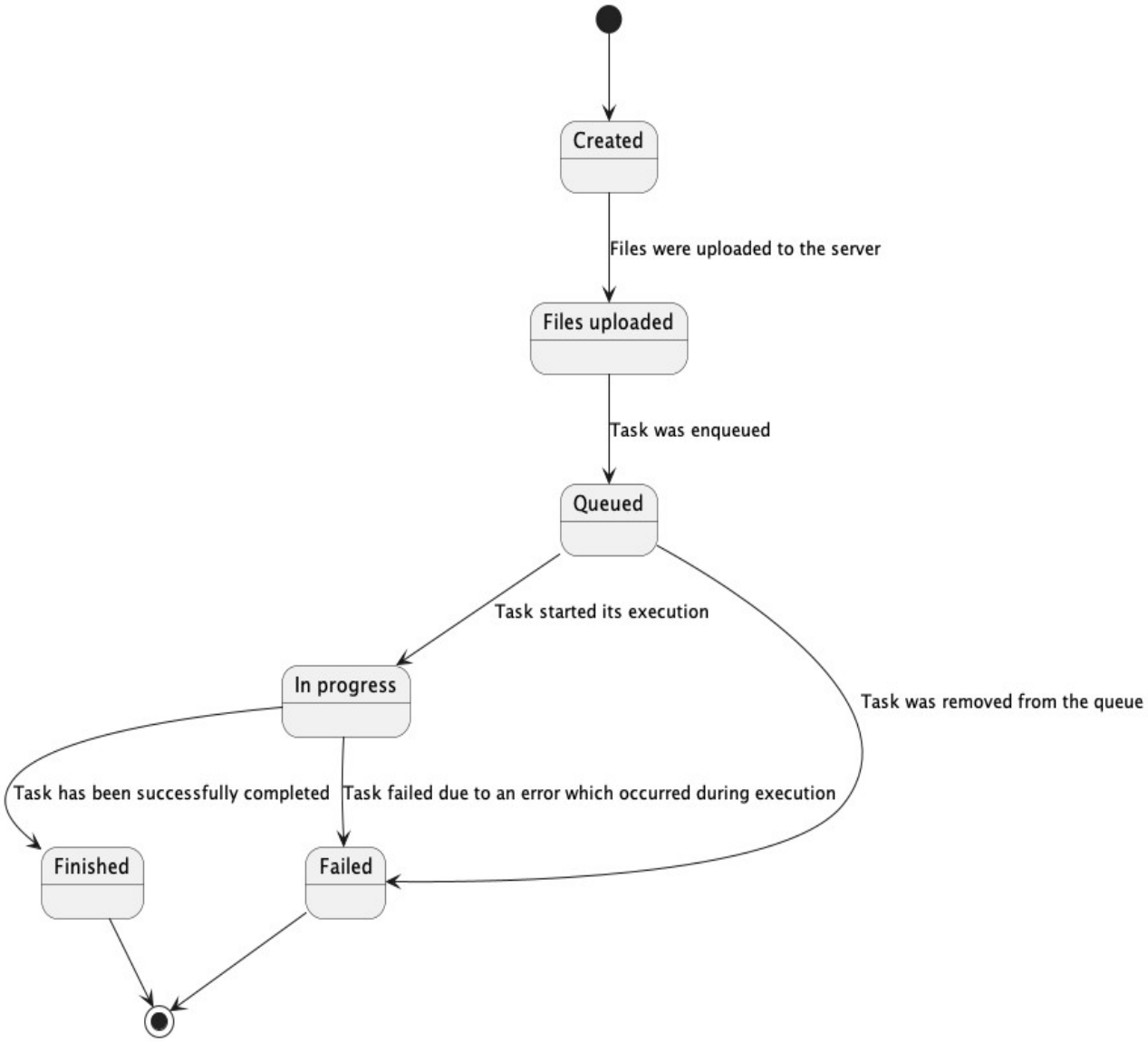


Рисунок 3.4 – Діаграма станів завдання обробки зображення

В ході проєктування було проведено структурний аналіз системи та розроблено UML діаграми класів, які демонструють структуру системи, її компоненти та взаємодію між ними. Діаграми класів було розроблено з

урахуванням стеку технологій, визначених для подальшої реалізації системи, а саме стек, що будується на базі мови програмування Python та її екосистеми. В ході аналізу було розроблено 3 діаграми класів у відповідності з діаграмами послідовностей, щоб акцентувати увагу на різних елементах інформаційної системи та поліпшити зрозумілість моделей, що проектуються.

На рисунку 3.5 подана діаграма класів, що пов’язані із створенням завдань з обробки зображень. На цій діаграмі зображено класи, що використовуються в ході створення завдання в системі.

Рисунок 3.5 – Діаграма класів пов’язаних із створенням завдань обробки зображень

Ця діаграма включає кілька основних компонентів: «APIRouter»,

«ImageProcessingTasksManagerService», «TasksImagesManager», «AbstractRepository» і його реалізації «SQLAlchemyRepository» та «ImageProcessingTaskRepository» і «StorageService» та його реалізацію «LocalStorageService».

Розглянемо визначені в системі класи більш детально:

– «APIRouter» – цей клас відповідає за обробку HTTP-запитів, пов'язаних з задачами обробки зображень. Він використовує методи «ImageProcessingTasksManagerService» для управління завданнями відновлення зображень в системі;

– «ImageProcessingTasksManagerService» – клас, що управляє завданнями обробки зображень в системі. Він використовується для створення завдань, завантаження зображень, видалення завдань та отримання інформації про існуючі завдання в системі;

– «TasksImagesManager» – клас, який зосереджений на таких операціях як: видалення, збереження та читання зображень. Він використовує одну з реалізацій абстрактного класу «StorageService» для низькорівневого доступу до сховища зображень;

– «StorageService» – абстрактний клас, що визначає інтерфейс для доступу до сховища зображень;

– «LocalStorageService» – клас, що є реалізацією абстрактного класу «StorageService» і призначений для доступу до локального сховища зображень, що знаходиться у середовищі де розгорнута інформаційна система;

– «AbstractRepository» – абстрактний клас, який визначає інтерфейс для репозиторіїв, що дозволяє взаємодіяти з базою даних через стандартні CRUD-операції. Цей клас є базовим для специфічних репозиторіїв, таких як «ImageProcessingTaskRepository»;

– «SQLAlchemyRepository» та «ImageProcessingTaskRepository» – ці класи реалізують абстрактні методи «AbstractRepository» для конкретних моделей, використовуючи «SQLAlchemy» як ORM для взаємодії з базою даних.

Важливо зазначити, що запропонована структура системи передбачає гнучкість у заміні компонентів, що забезпечується використанням спроектованих абстракцій.

Клас «StorageService» і його реалізації, такі як «LocalStorageService», демонструють, що система здатна адаптуватися до змін в технологіях зберігання даних. Завдяки абстракції, можливо з легкістю інтегрувати хмарні сховища, такі як, наприклад, AWS S3, що забезпечить не лише розширення ємкості сховища даних, але й збільшить доступність та надійність зберігання. Це дозволить системі ефективно масштабуватися, реагуючи на зростаючі вимоги до об'єму даних.

Класи побудовані на базі інтерфейсу «AbstractRepository», надають змогу легко замінювати одну базу даних на іншу без необхідності зміни клієнтського коду. Це важливо для масштабування, оскільки в майбутньому може з’явитися потреба в переході на більш продуктивні системи управління базами даних, що забезпечить кращу продуктивність та розширюваність при зростанні кількості даних та користувачів.

Для раніше описаної діаграми послідовностей процесу розміщення завдань відновлення зображень у черзі виконання завдань (див. 3.2.) було створено відповідну діаграму класів, яка подана на рисунку 3.6. Окрім раніше зазначеного «ImageProcessingTaskRepository» було виділено наступні класи:

– «BaseImageProcessor» – абстрактний клас, що визначає інтерфейс до обробника зображень. Цей клас містить методи для завантаження та зберігання зображень, а також ряд абстрактних методів, що відповідають за надання інформації про обробник та, власне, обробку зображень. Ці абстрактні методи перевизначені в класах нащадках, опис яких буде наведено далі;

– «ImageProcessingTasksHandler» – службовий клас для керування завданнями обробки зображень. Цей клас створено для запуску нових завдань та їх контролю;

– «ImageProcessorFactory» – фабрика, яка створює екземпляри обробників зображень, що є нащадками базового класу «BaseImageProcessor»;

– «Queue» – клас, який керує чергою завдань. Використовується для взаємодії з асинхронною чергою завдань відновлення зображень;

– «ImageProcessingTaskRunner» – клас для виконання обробки зображень. Він отримує завдання, запускає відповідний обробник з використанням переданих йому налаштувань і зберігає результати.

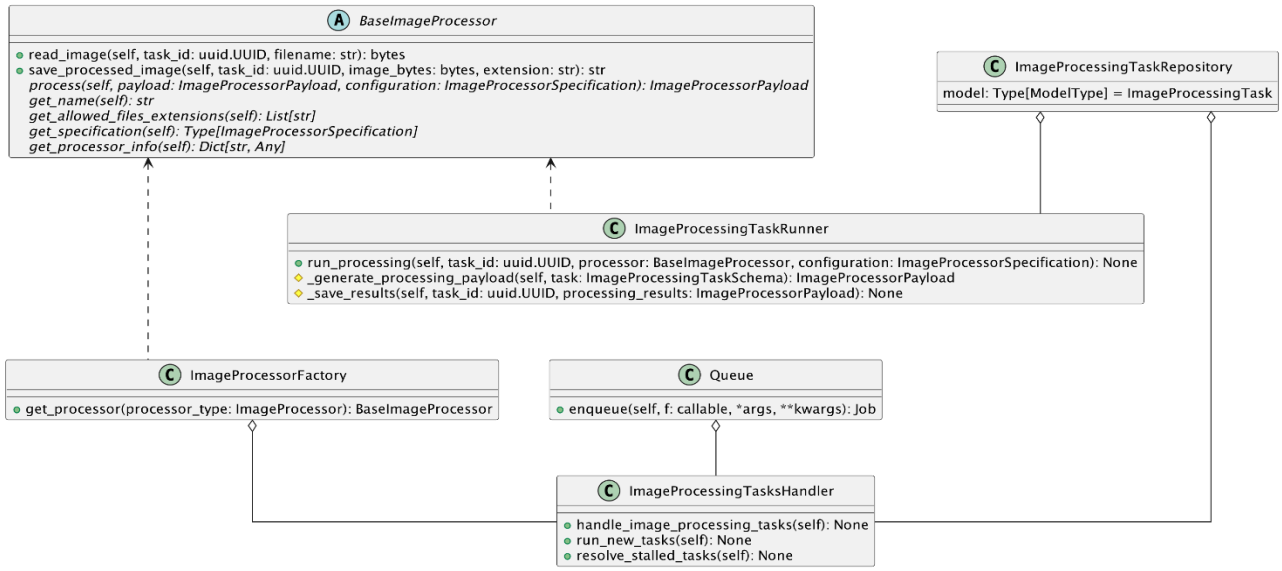


Рисунок 3.6 – Діаграма класів пов’язаних із розміщенням завдань відновлення зображень у черзі

Наступним кроком було створено діаграму класів, яка подана на рисунку 3.7. Ця діаграма демонструє присутню у системі ієрархію класів, що відповідають за обробку зображень. Ця ієрархія відповідає раніше визначеним конвеєрам відновлення зображення поданим на рисунках 2.3 та 2.4. Було виділено наступні класи:

– «BaseImageProcessor» – клас, що був описаний раніше, є абстрактним базовим класом для усіх обробників зображень у системі;

– «BatchImageProcessor» – абстрактний клас, який розширює базовий клас «BaseImageProcessor». Він розширює базову функціональність пакетною обробкою зображень, що дозволяє одночасно відновлювати декілька зображень.

– «XMLConfiguredImageProcessorMixin» та «BinaryHostedImageProcesorMixin» це класи-міксини, що надають додаткову функціональність обробнику зображень, наприклад, обробка за допомогою зовнішніх двійкових скриптів або керування параметрами конфігурації через XML файл.

– «LucyFilterProcessor», «BrightnessEqualisationProcessor», «ImageFormat ConversionProcessor», «FitsImageRestorationProcessor», «RegularImage RestorationProcessor» – це реалізації обробників зображень, кожна з яких спеціалізується на певному типі обробки. Слід зазначити, що класи «FitsImageRestorationProcessor» та «RegularImageRestorationProcessor» є класами верхнього рівня, через те, що вони створюють конвеєри відновлення зображень за допомогою агрегації класів-обробників нижчого рівня.

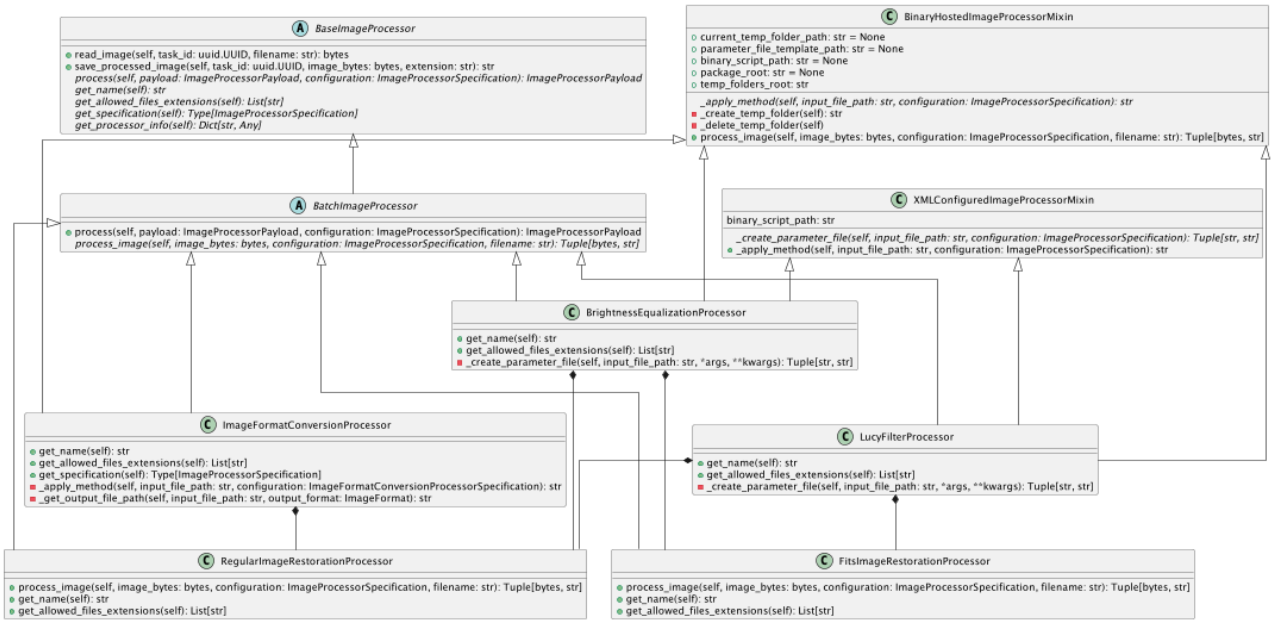


Рисунок 3.7 – Діаграма класів пов’язаних із виконанням завдань з обробки зображень

Отже, при проєктуванні структури класів, що використовуються для відновлення зображень, було використано модульний підхід, при якому модулі вищого порядку, а саме модулі які виконують відновлення зображень, складаються з модулів нижчого порядку, що реалізують окремі методи обробки зображень.

Використання такого підходу підвищує незалежність різних модулів обробки зображень, та надає можливість їх легкого комбінування та заміни, що загалом надає можливість повторного використання кодової бази.

В процесі проєктування для інформаційної системи відновлення зображення було обрано реляційну модель даних. За допомогою CASE засобу All Fusion Data Modeler була створена логічна модель даних, яка подана на рисунку 3.8.

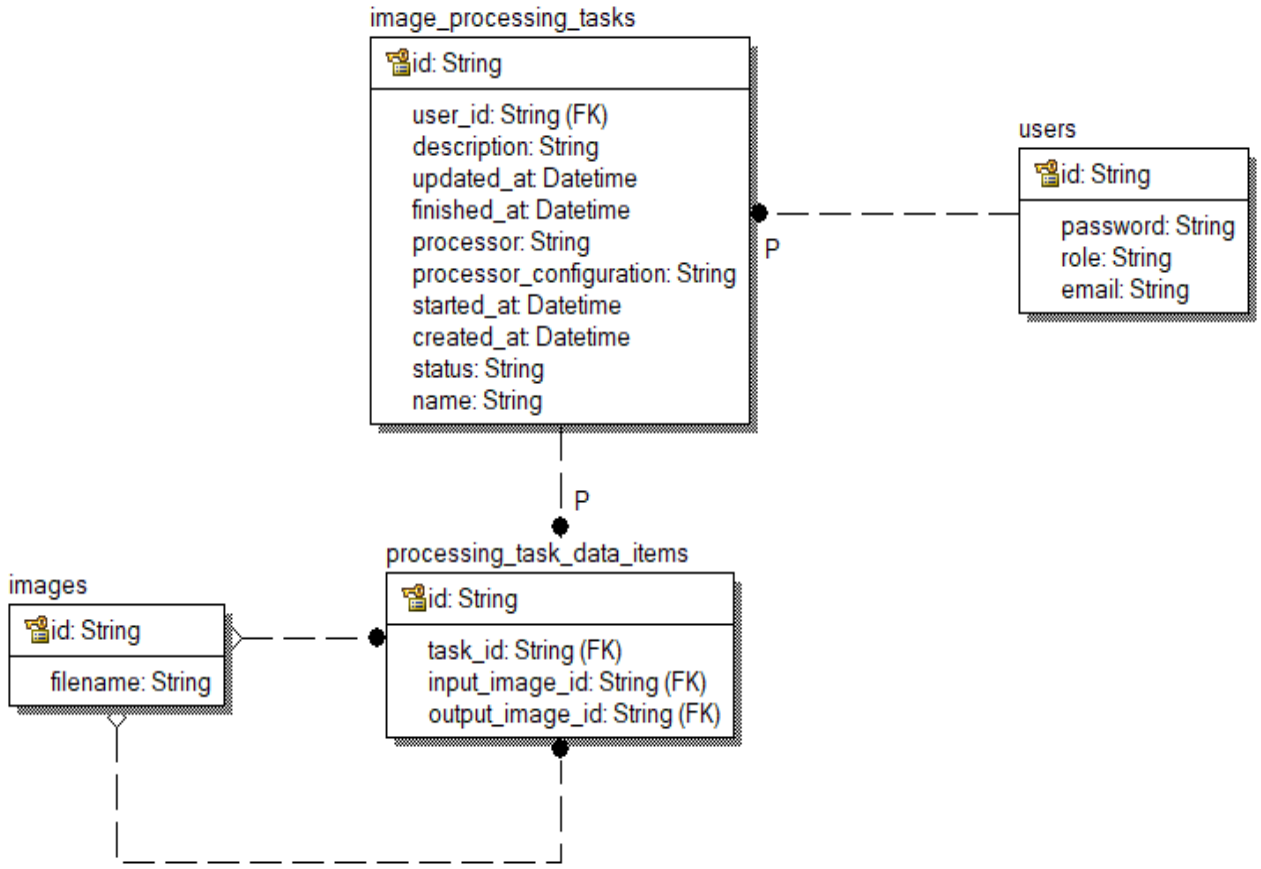


Рисунок 3.8 – Логічна модель даних інформаційної системи

В ході аналізу було виділено наступні сутності:

– сутність «users» – сутність, що відповідає за зберігання мінімальної інформації користувачів, що надає можливість створення облікового запису та подальшої авторизації на основі збережених даних, а саме електронної адреси та пароля;

– сутність «images» – сутність, що містить інформацію про зображення, це можуть бути як вхідні зображення, що користувач завантажив для обробки, так і вихідні зображення, що пройшли етап відновлення;

– сутність «image_processing_tasks» – сутність, що містить інформацію про завдання обробки зображень, а саме інформацію про час створення завдання, час його виконання та оновлення, ці дані надають інформацію про загальний час виконання завдання. Додатково ця сутність зберігає налаштування методу обробки зображення та статус завдання;

– сутність «processing_task_data_items» – сутність, що є проміжною між сутностями зображення та завдання, ця сутність надає можливість збереження асоціації зображень із завданнями, таким чином кожне завдання може мати одне або декілька вхідних та вихідних зображень, що надає гнучкий інструмент, потенціал якого може бути використано в майбутньому, що дозволить використання методів обробки зображень, які отримують на вхід декілька зображень, і надають в результаті єдине зображення, або навпаки, коли з одного вхідного зображення можна отримати декілька вихідних зображень.

Для кожного атрибуту сутностей було визначено типи даних, які визначають тип інформації, що зберігається в цьому атрибуті.

Типи даних атрибутів сутності «users»:

– атрибут «id» – рядковий тип даних;
– атрибут «password» – рядковий тип даних;
– атрибут «role» – рядковий тип даних;
– атрибут «email» – рядковий тип даних.

Типи даних атрибутів сутності «image_processing_tasks»:

– атрибут «id» – рядковий тип даних;
– атрибут «user_id» – рядковий тип даних;
– атрибут «description» – рядковий тип даних;
– атрибут «updated_at» – дата та час;
– атрибут «finished_at» – дата та час;

– атрибут «started_at» – дата та час;
– атрибут «created_at» – дата та час;
– атрибут «processor» – рядковий тип даних;
– атрибут «processor_configuration» – рядковий тип даних;
– атрибут «status» – рядковий тип даних;
– атрибут «name» – рядковий тип даних.

Типи даних атрибутів сутності «images»:
– атрибут «id» – рядковий тип даних;
– атрибут «filename» – рядковий тип даних.

Типи даних атрибутів сутності «processing_task_data_items»:
– атрибут «id» – рядковий тип даних;
– атрибут «task_id» – рядковий тип даних;
– атрибут «input_image_id» – рядковий тип даних;
– атрибут «output_image_id» – рядковий тип даних.

3.2 Обґрунтування обраних технологій та середовища розробки

Для розробки веб-застосунку для відновлення астрономічних зображень було обрано набір технологій, що забезпечує ефективну обробку великих обсягів даних, високу доступність системи, масштабованість та гнучкість, що є ключовими потребами в обробці астрономічних даних.

Ключовими інструментами для реалізації серверної частини застосунку було обрано мову програмування «Python» та веб-фреймворк «FastAPI» через їх простоту та зрозумілість, що сприяє швидкій розробці та тестуванню складних систем, таких, як поточна система, що розробляється [21]. Крім того, запропонований фреймворк є інструментом, що дозволяє з легкістю розробляти складні системи, які надають інтерфейс у вигляді REST API, з можливістю автоматичної генерації документації.

Для реалізації клієнтської частини застосунку було обрано фреймворк

«React». Компонентний підхід, що пропонується даним фреймворком, дозволяє розробляти користувацькі інтерфейси для складних системи, які можуть ефективно оновлювати та відображати великі обсяги даних без втрати швидкодії застосунку.

Для зберігання даних у структурованому вигляді було обрано реляційну СУБД PostgreSQL, що забезпечує високу надійність та швидкодію, що є дуже критичним при обробці астрономічних даних у високонавантаженій системі.

Для підтримки механізму черг завдань було використано СУБД Redis з наступних причин:

– масштабованість: черги завдань у Redis спрощують масштабування системи шляхом додавання додаткових робочих процесів, які виконують обробку завдань у черзі;

– надійність і відмовостійкість: Redis має вбудовані механізми реплікації та збереження даних, які забезпечують підтримку черги завдань навіть у разі збою системи. Це критично важливо для забезпечення безперервності обробки даних в астрономічних дослідженнях.

– швидкість обробки: оскільки Redis працює в основному в пам’яті, час доступу до даних значно нижчий, ніж у традиційних дискових системах зберігання. Це дозволяє швидко додавати та видаляти завдання з черги, що покращує загальну продуктивність системи [22, 23].

З метою спрощення розгортки застосунку в різних середовищах, було використано такі інструменти як «Docker» та «Docker-Compose» [24]. Вони дозволяють контейнеризувати застосунки та їх залежності, що полегшує переміщення та масштабування систем між різними середовищами. Крім того, зважаючи на те, що система складається з декількох сервісів, використання інструменту «Docker-Compose» є обумовленим потребою у координації та автоматизації розгортки декількох контейнерів [25].

У вигляді веб-серверу було обрано Caddy через легкість налаштування, високий рівень продуктивності та автоматичне управління SSL сертифікатами, що є критично важливим при публікації застосунку у мережі інтернет.

3.3 Розробка масштабованої архітектури інформаційної системи відновлення зображень

У контексті розробки веб-додатку для відновлення астрономічних зображень головними проблемами є ефективна обробка великих обсягів даних і забезпечення високої доступності та масштабованості системи. Ці проблеми можна успішно вирішити шляхом впровадження архітектури, заснованої на використанні черг завдань.

В процесі розробки архітектури на базі черги завдань було створено дві черги завдань і відповідно два пули робітників, як показано на рисунку 3.9.

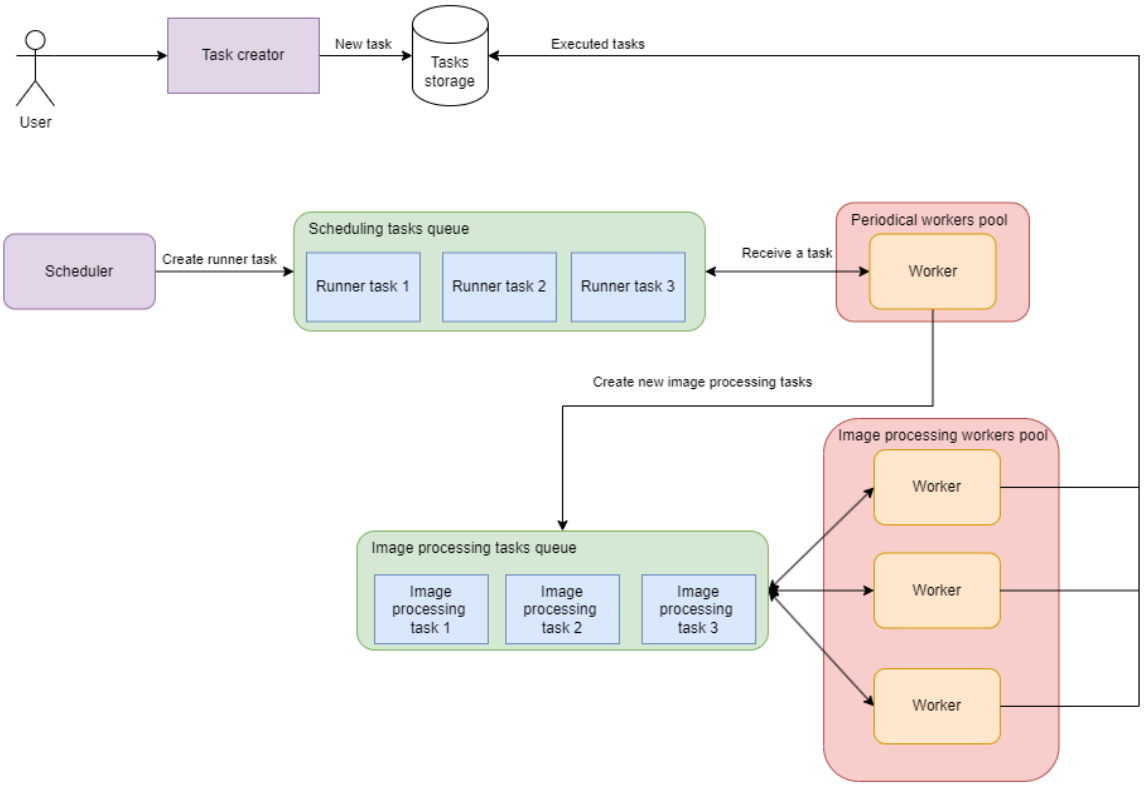


Рисунок 3.9 – Діаграма використання черги завдань у системі

З поданої діаграми можна побачити, що перша черга – є чергою періодичних завдань. В цю чергу з визначеною періодичністю, яка може бути змінена, надсилаються завдання, що в свою чергу полягають в визначенні нових необроблених завдань з відновлення зображень і переміщення їх в другу

чергу, що відповідає за збереження цих завдань і подальше їх виконання робочими процесами.

Порівняно з прямим додаванням завдань у чергу в момент їх створення користувачем, запропонований підхід є виправданим з кількох причин. По-перше, періодичне сканування на наявність нових завдань і подальше їх додавання в чергу забезпечує більш рівномірний розподіл навантаження на систему. У системах, де користувачі можуть одночасно відправляти велику кількість запитів, пряме додавання завдань у чергу може призвести до затримок або навіть блокування обробки через перевантаження системи. Періодичний підхід зменшує ризик таких сценаріїв, оскільки система сама контролює час та обсяг додавання завдань у чергу, забезпечуючи більш плавне та ефективне виконання процесів.

Ще одною причиною є забезпечення більшого контролю над потоком оброблюваних даних. Система може адаптуватися до змін доступності ресурсів, динамічно регулюючи частоту виконання періодичного завдання залежно від поточного навантаження та пріоритетів. Такий підхід дозволяє оптимізувати використання обчислювальних ресурсів і пам'яті, знизити витрати і підвищити ефективність системи.

Додавання нових робітників (робочих-процесів), які виконують завдання з черги, є важливим елементом підвищення продуктивності та масштабованості поточної системи. У поточній ітерації дослідження використання додаткових робітників тісно пов’язане з вертикальним масштабуванням, яке передбачає підвищення продуктивності за рахунок збільшення обчислювальних ресурсів (наприклад, процесора, оперативної пам’яті) на існуючих машинах.

Під час вертикального масштабування, додавання робітників дозволяє більш ефективно використовувати збільшені обчислювальні ресурси машини. Вертикальне масштабування було обрано у вигляді ефективного підходу на початкових етапах розвитку системи. Оскільки кожен робітник може обробляти завдання незалежно, додавання додаткових робітників на потужнішій машині призводить до зменшення часу обробки завдань та збільшення загальної

пропускної спроможності системи.

При вертикальному масштабуванні системи, необхідність використання більш потужних машин обумовлена наступними причинами:

– обчислювальна потужність: більша кількість робітників вимагає відповідної кількості ЦП ресурсів для паралельної обробки завдань;

– пам'ять: кожен робітник використовує кількість оперативної пам'яті. Зі збільшенням кількості робітників зростає і загальна потреба у оперативній пам'яті;

– операції читання та запису: більш потужні робочі станції мають кращі можливості для обробки великої кількості операцій читання та запису, що критично важливо для систем, які активно працюють з диском або мережевими ресурсами.

Архітектура інформаційної системи була створена з використанням технологічного стеку, що був зазначений раніше, до складу якого входять мова програмування «Python», фреймворк для клієнтської частини застосунку «React», фреймворк для серверної частини застосунку «FastAPI», СУБД «Redis» та «PostgreSQL», бібліотека для взаємодії з чергою «Redis Queue» та інструменти для контейнеризації застосунку «Docker» і «Docker-Compose».

Кожен елемент з зазначеного стеку технологій виконує окремі функції, важливі для протікання процесів в інформаційній системі. Загальна архітектура інформаційної системи відновлення зображень показана на рисунку 3.10.

У запропонованій архітектурі інформаційної системи відновлення астрономічних зображень для ефективної обробки даних і розподілу ресурсів використовується клієнт-серверний підхід, в якому клієнт ініціює запити до сервера, який обробляє ці запити, та надсилає відповідь.

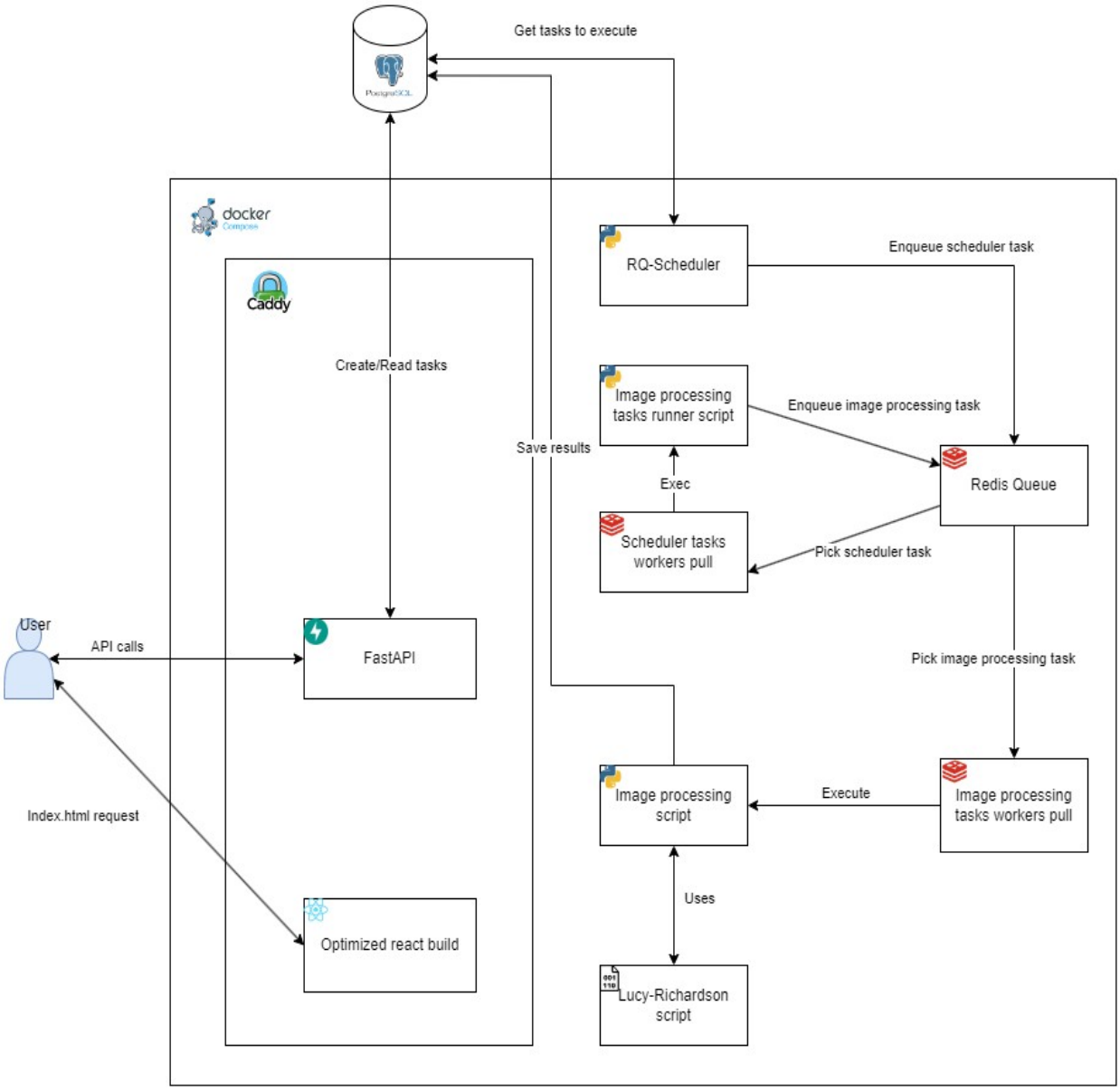


Рисунок 3.10 – Архітектура інформаційної системи відновлення зображення

Такий підхід має декілька важливих переваг:

– масштабованість: сервер може обробляти запити від великої кількості клієнтів одночасно, що дозволяє масштабувати систему в міру зростання потреб в обробці;

– централізоване управління: всі ключові завдання та процеси управління даними централізовані на сервері, що спрощує оновлення, обслуговування та безпеку системи;

– економія ресурсів: оскільки основна обробка відбувається на сервері,

клієнти не потребують значних обчислювальних ресурсів. Це особливо важливо для пристроїв з обмеженою обчислювальною потужністю;

– гнучка взаємодія: оскільки вимоги клієнта обмежуються надсиланням запитів на сервер і отриманням відповідей, система може бути адаптована до різних клієнтських пристроїв і платформ;

– оптимізація ресурсів сервера: використання пам'яті, потужність процесора та інші ресурси можуть бути оптимізовані шляхом налаштування сервера в залежності від рівня навантаження.

При проектуванні архітектури інформаційної системи було використано масштабований підхід до зберігання зображень. Замість зберігання зображень у вигляді бінарних даних в БД, зберігаються шляхи або посилання на місцезнаходження зображень у файловій системі. Таке рішення має кілька переваг, включаючи гнучкість, масштабованість та ефективне управління ресурсами.

Зберігання шляхів до зображень, а не самих зображень, дозволяє легко переходити на альтернативні рішення для зберігання, такі як хмарні сховища. Ця гнучкість дозволяє адаптуватися до змінних вимог до зберігання зображень та безперешкодно інтегруватися з існуючою хмарною інфраструктурою, забезпечуючи оптимальну продуктивність та надійність.

Більш того, зберігання шляхів до зображень зменшує обсяг зберігання у базі даних, що призводить до підвищення продуктивності бази даних та зниження витрат на зберігання. Це також спрощує процес управління даними та їх резервного копіювання, оскільки в базі даних потрібно зберігати лише метадані та посилання на зображення.

Такий підхід також покращує портативність та доступність даних, оскільки зображення можуть бути збережені в будь-якому місці, доступному для інформаційної системи, незалежно від того, знаходяться вони локально чи у хмарному сховищі.

На основі логічної схеми даних описаної раніше (див. 3.8) було

реалізовано фізичну модель даних засобами СУБД «PostgreSQL», яка подана на рисунку 3.11.

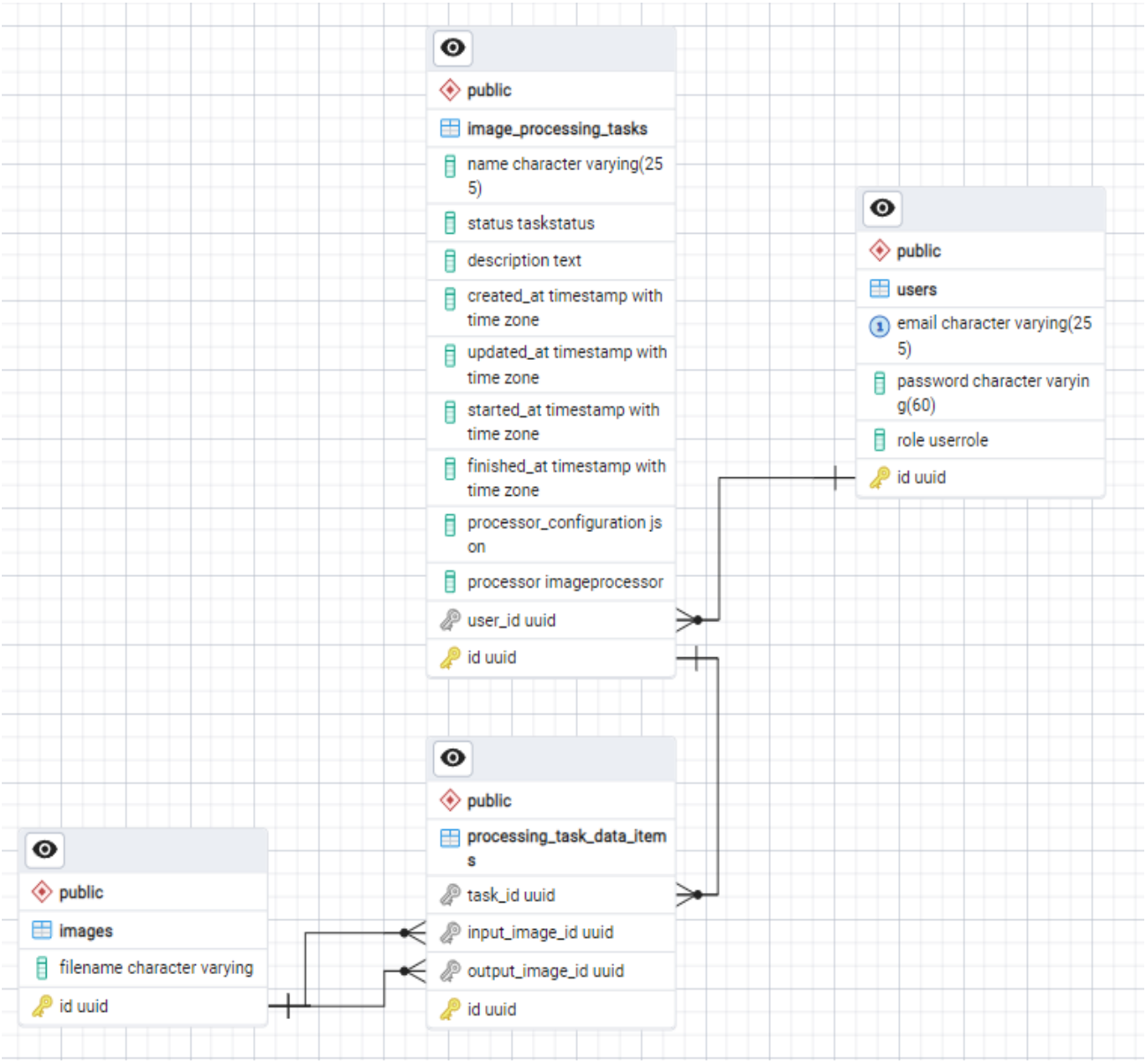


Рисунок 3.11 – Фізична модель даних

Для повного розуміння інформації, що зберігається в таблицях бази даних, нижче наведено опис усіх полів для кожної зазначеної таблиці.

Таблиця «users» – користувачі:

– поле «id» – ідентифікатор користувача;

– поле «password» – пароль користувача;

– поле «role» – роль користувача в системі;

– поле «email» – електронна пошта користувача.

Таблиця «image_processing_tasks» – задачі обробки зображень:

– поле «id» – ідентифікатор задачі;
– поле «user_id» – ідентифікатор користувача, який створив задачу;
– поле «description» – опис задачі;
– поле «updated_at» – час останнього оновлення задачі;
– поле «finished_at» – час завершення задачі;
– поле «started_at» – час початку виконання задачі;
– поле «created_at» – час створення задачі;
– поле «processor» – обробник, що використовується для задачі;
– поле «processor_configuration» – конфігурація обробника;
– поле «status» – статус задачі;
– поле «name» – назва задачі.

Таблиця «images» – зображення:

– поле «id» – ідентифікатор зображення;
– поле «filename» – ім'я файлу зображення.

Таблиця «processing_task_data_items» – елементи даних задач обробки:

– поле «id» – ідентифікатор елемента даних;
– поле «task_id» – ідентифікатор задачі, до якої належить елемент;
– поле «input_image_id» – ідентифікатор вхідного зображення;
– поле «output_image_id» – ідентифікатор вихідного зображення.

4 РОЗГОРТКА ТА ТЕСТУВАННЯ КОМПОНЕНТІВ ІНФОРМАЦІЙНОЇ СИСТЕМИ

4.1 Розгортка компонентів інформаційної системи засобами Cloud-технологій

Для розгортки інформаційної системи відновлення астрономічних зображень було обрано хмарний провайдер AWS. До ключових факторів, що вплинули на його вибір можна віднести наявність потрібних сервісів для підтримки роботи інформаційної системи та простоту масштабування ресурсів.

Інформаційна система була розгорнута засобами сервісу AWS EC2, який надає можливість створення віртуальних машин у вигляді екземплярів, які можуть бути пов’язані з іншими сервісами AWS.

Вертикальне масштабування системи при використанні EC2 екземплярів полягає в зміні типу або розміру екземпляра. Кожен тип в AWS EC2 має відповідну конфігурацію, що використовується для запуску віртуальної машини, ця конфігурація може бути збільшена у відповідності до наявних потреб, тобто використовуючи більший розмір обраного типу екземпляру або більш потужний тип екземпляру, можна збільшити продуктивність системи.

Такий підхід не тільки забезпечує високу доступність і продуктивність системи, але і дозволяє оптимізувати витрати за рахунок приведення інфраструктури у відповідність з реальними потребами системи.

На рисунку 4.1 поданий перелік екземплярів EC2 після авторизації нового акаунту в AWS Console.

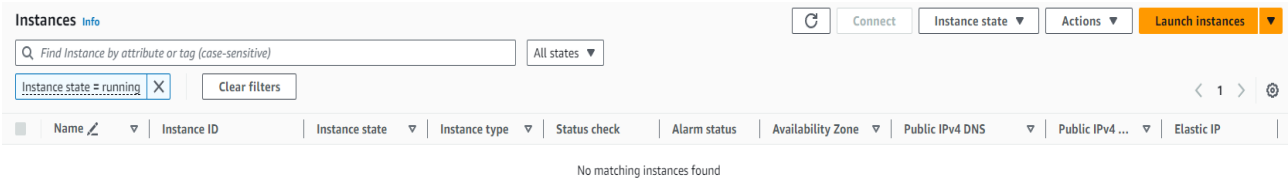


Рисунок 4.1 – Порожній перелік екземплярів EC2

Для створення нового екземпляру EC2 було виконано наступні кроки. По-перше, було обрано тип операційної системи, а саме Ubuntu 24.04 LTS на архітектурі x86, як показано на рисунку 4.2. Цю операційну систему було обрано через її сумісність з математичними модулями, які були використані для відновлення зображень.

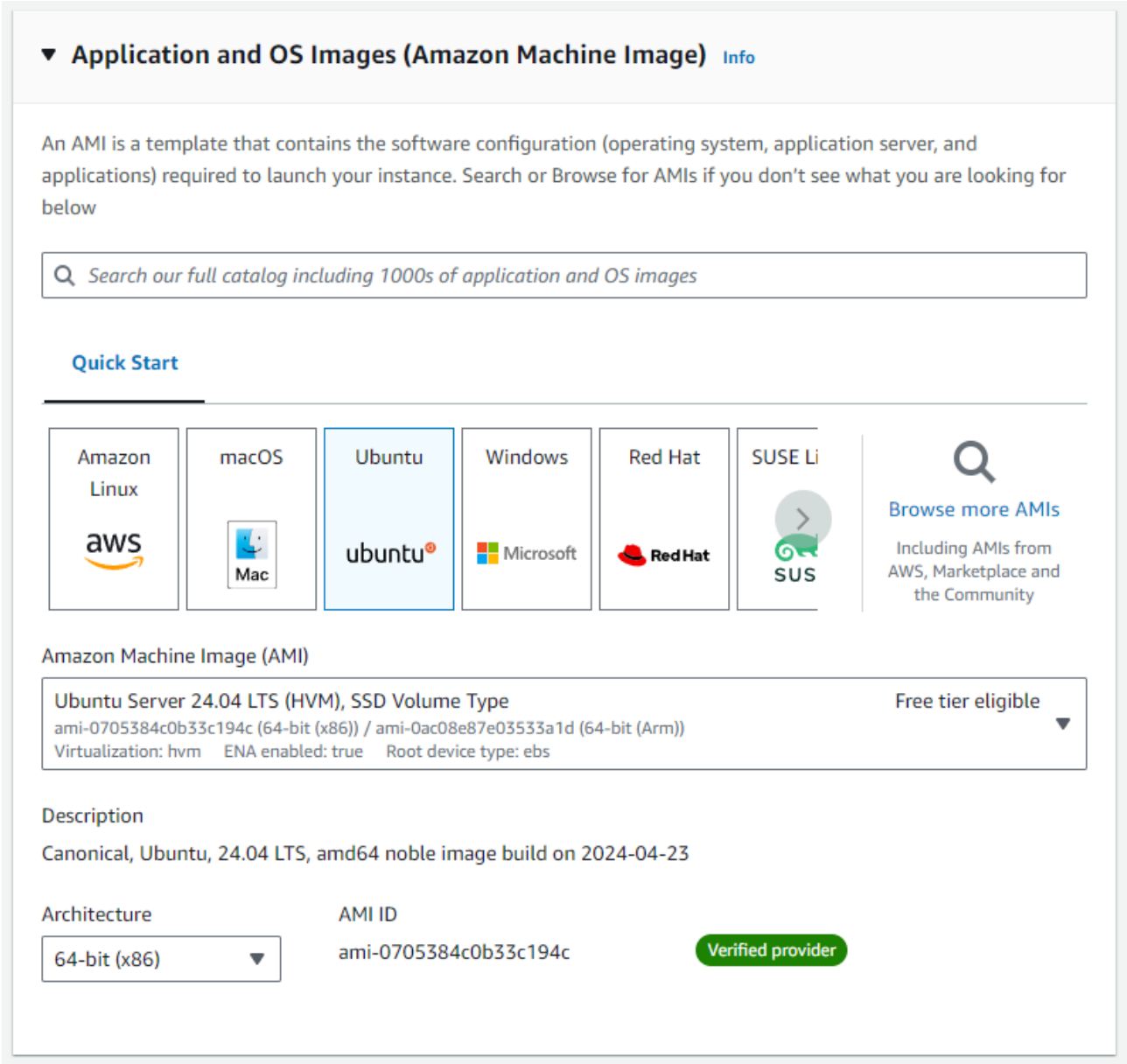


Рисунок 4.2 – Вибір операційної системи для віртуальної машини

Наступним кроком було обрано тип екземпляру EC2, що визначає конфігурацію та властивості віртуальної машини. Для поточної інформаційної

системи було обрано тип «C5», який оптимізований під обчислювальні задачі з використанням центральних процесорів. На рисунку 4.3 можна побачити порівняльні характеристики обраного типу «C5» та типу «M5», який розглядався як альтернативний варіант.

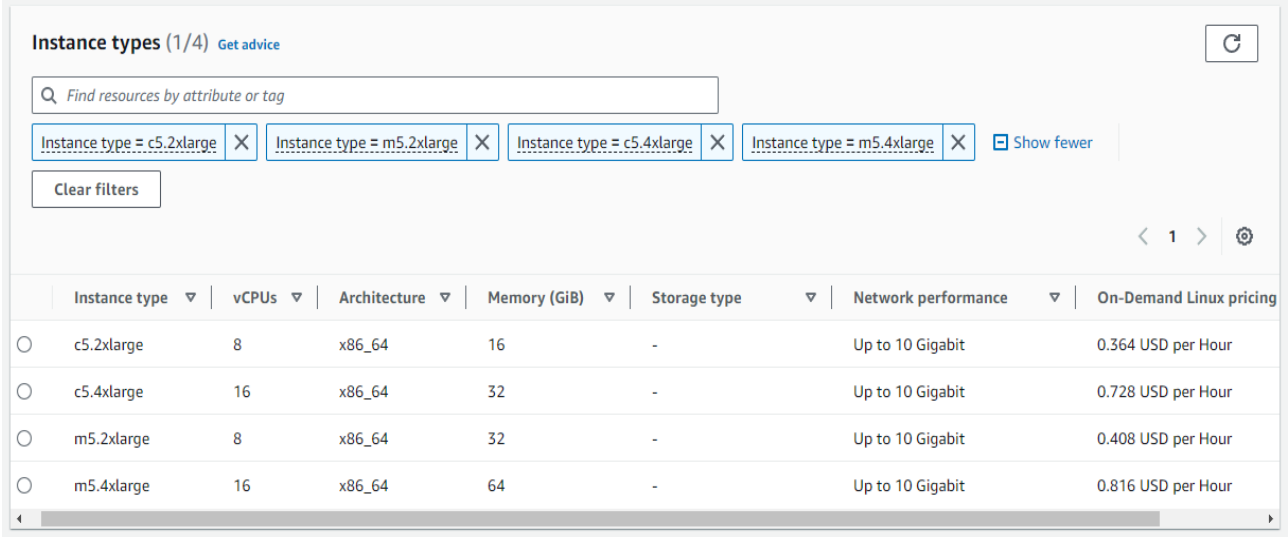


Рисунок 4.3 – Порівняльна таблиця різних типів екземплярів EC2

Можна побачити, що тип «M5» надає більший об’єм оперативної пам’яті, при цьому кількість віртуальних ядер залишається однаковим, у порівнянні з типом «C5». Зважаючи на те, що для поточної задачі кількість ядер та обчислювальної потужності є критичним показником, на відміну від кількості оперативної пам’яті, було обрано тип «C5», як показано на рисунку 4.4.

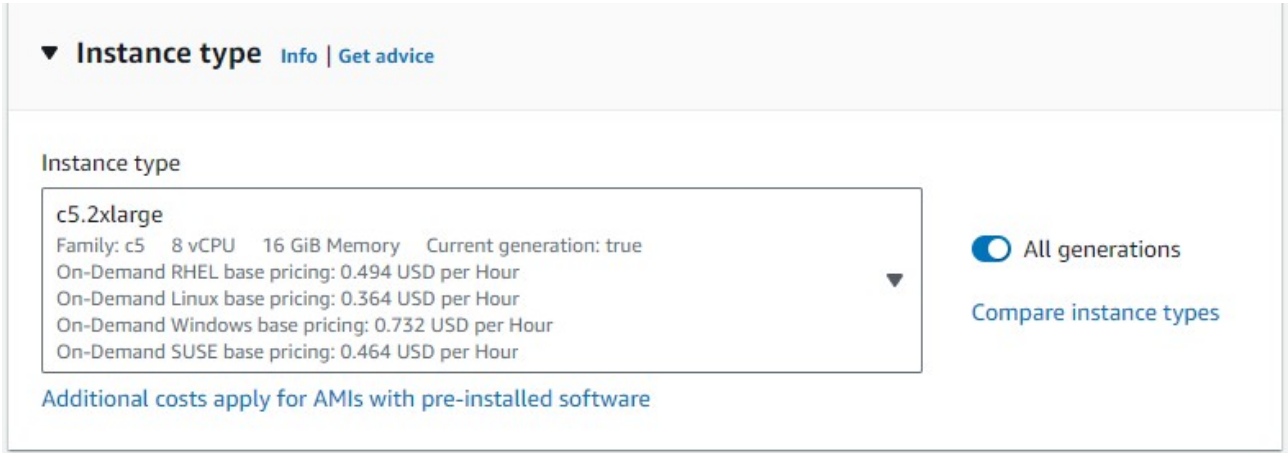


Рисунок 4.4 – Налаштування типу екземпляра EC2

Початково було обрано тип «c5.2xlarge», що має 8 віртуальних процесорів, та 16 GiB оперативної пам’яті. Після вибору типу екземпляра EC2 та операційної системи, що буде встановлена на віртуальну машину було виконано її створення, що підтверджується повідомленням поданим на рисунку 4.5.

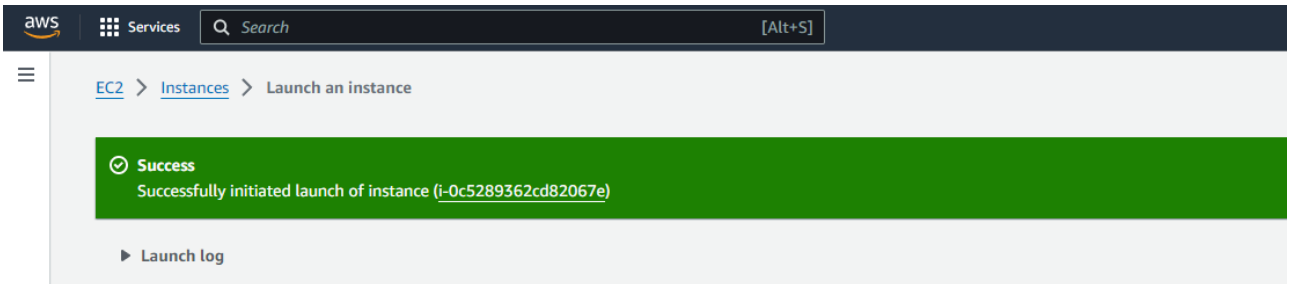


Рисунок 4.5 – Успішне створення екземпляра EC2

Віртуальні машини в AWS у вигляді EC2 мають властивість, яка полягає в тому, що IP адреса для віртуальної машини не є постійною. При перезапуску системи віртуальна машина кожен раз отримує нову IP-адресу, що є небажаною поведінкою при довгостроковій роботі інформаційної системи.

Вирішенням цієї проблеми є налаштування еластичної адреси «Elastic IP», що прив’язується до екземпляра EC2 і не змінюється протягом часу. За допомогою еластичної адреси екземпляри EC2 отримують статичну IP-адресу, що не змінюється при перезавантаженні віртуальної машини.

На рисунку 4.6 можна побачити еластичну адресу, яка була створена для раніше запущеної віртуальної машини. Еластична адреса не має великої кількості налаштувань і є простою публічною адресою в мережі інтернет.

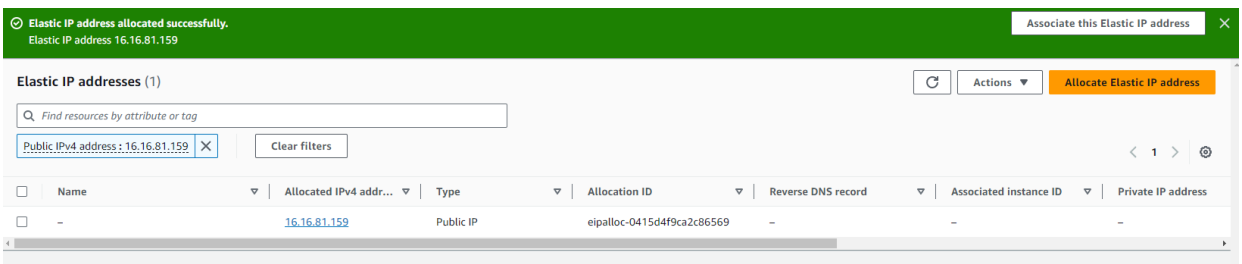


Рисунок 4.6 – Успішне створення еластичної адреси

Важливим кроком у налаштуванні статичної адреси для екземпляра EC2 є його зв’язок з еластичною адресою, що називається асоціацією. Тобто, адреса призначається екземпляру EC2, після чого вона вважається пов’язаною з віртуальною машиною і може бути використана для публічного доступу до віртуальної машини. На рисунку 4.7 можна побачити налаштування для зв’язку езкемпляра EC2 та еластичної адреси.

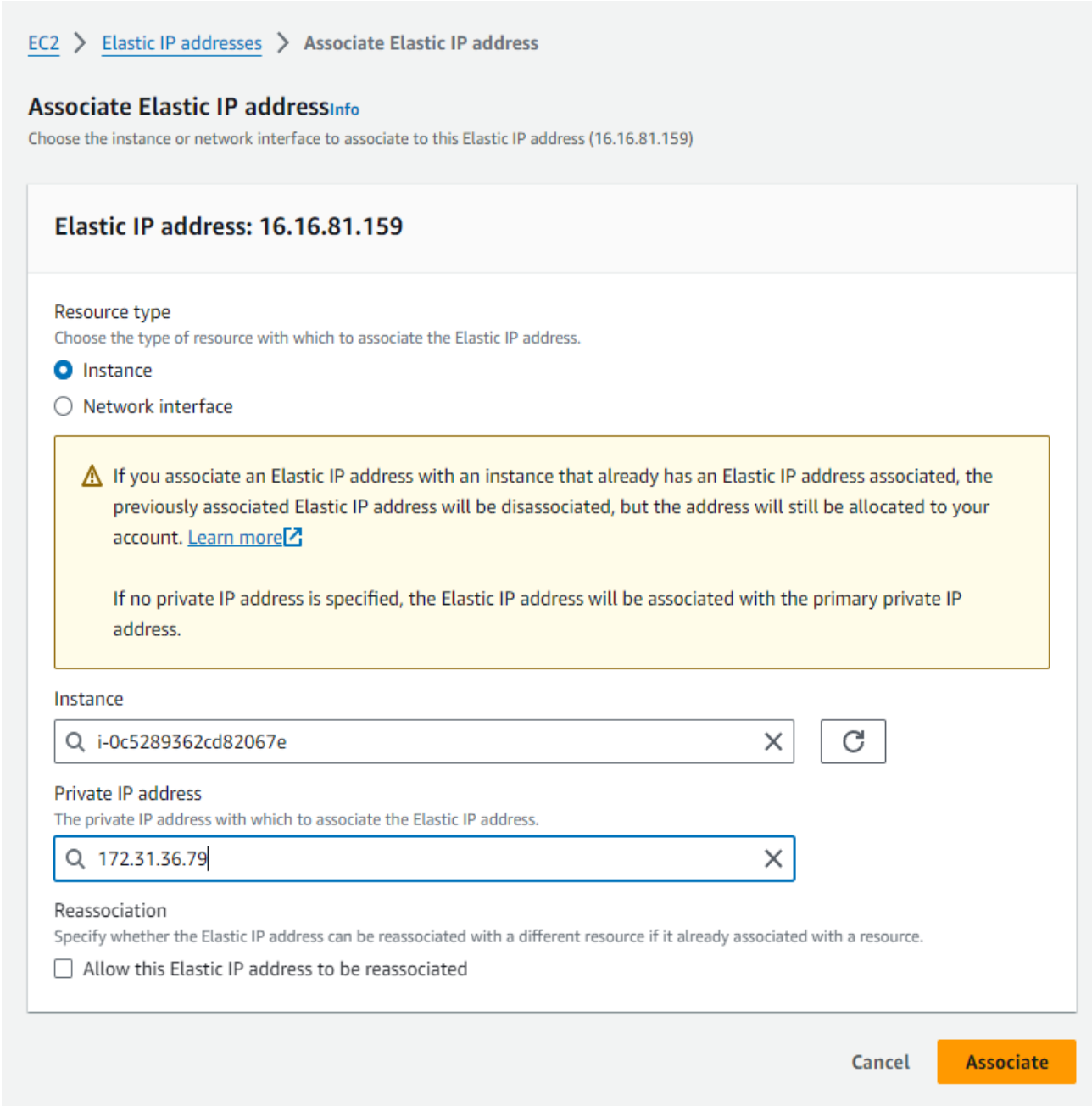


Рисунок 4.7 – Налаштування зв’язку між еластичною адресою та екземпляром EC2

Додатково для простоти пошуку еластичної адреси їй було призначено ім’я, як показано на рисунку 4.8.

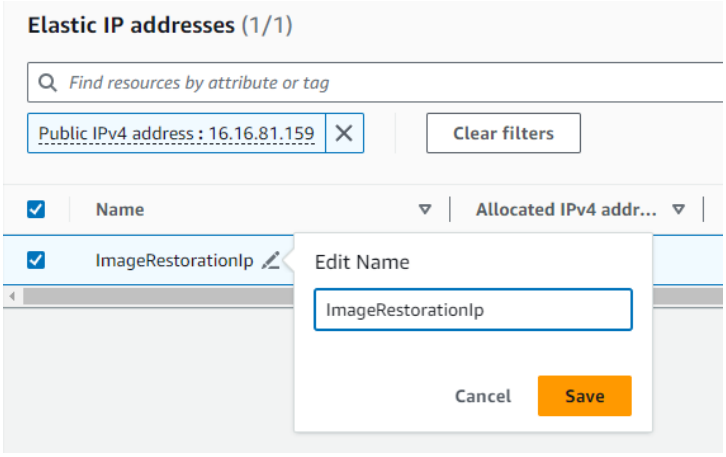


Рисунок 4.8 – Призначення ім’я еластичній адресі

В результаті після усіх налаштувань в списку екземплярів EC2 можна побачити створений екземпляр для розгортки інформаційної системи, як показано на рисунку 4.9. Графіки моніторингу, що подані у властивостях екземпляра EC2 свідчать про те, що сервер приймає данні та готовий до запуску компонентів інформаційної системи.

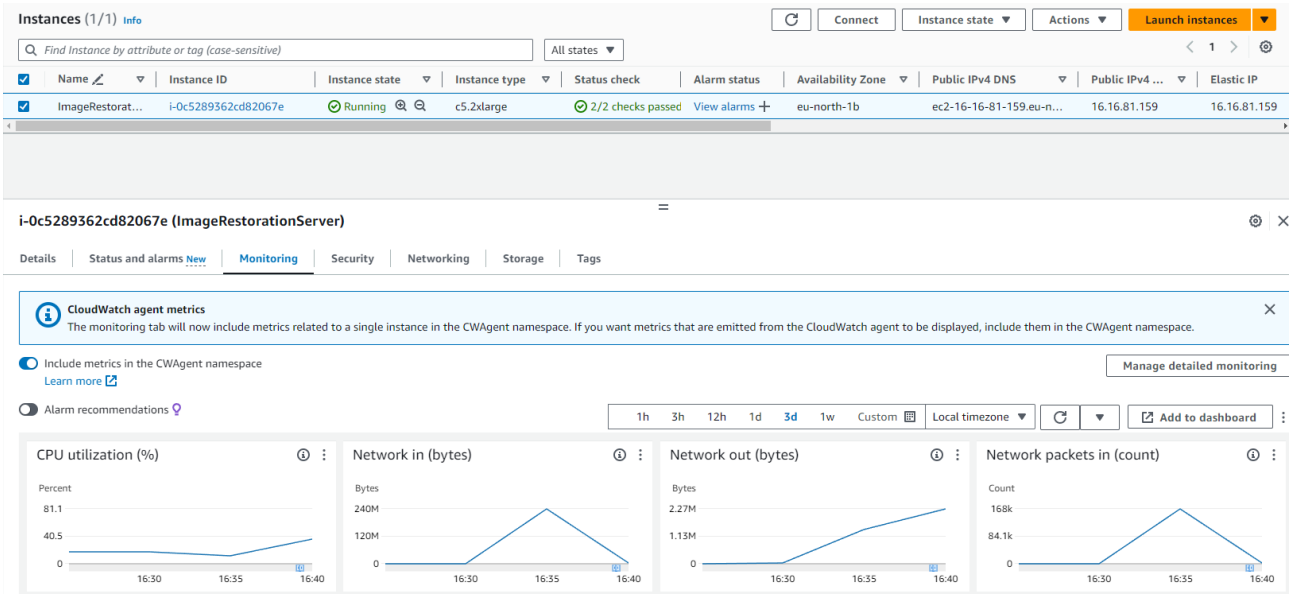


Рисунок 4.9 – Відображення створеного екземпляра EC2

Детальні інструкції щодо налаштування застосунку на віртуальній машині подані в додатку Б у розділі «Підготовка до роботи». Результатом розгортки застосунку засобами віртуальної машини, що представлена екземпляром EC2 є веб інтерфейс, що доступний за посиланням, яке подано на рисунку 4.10. З переліком доступних функцій можна ознайомитися у розділі «Правила по експлуатації програмного засобу» у додатку Б.

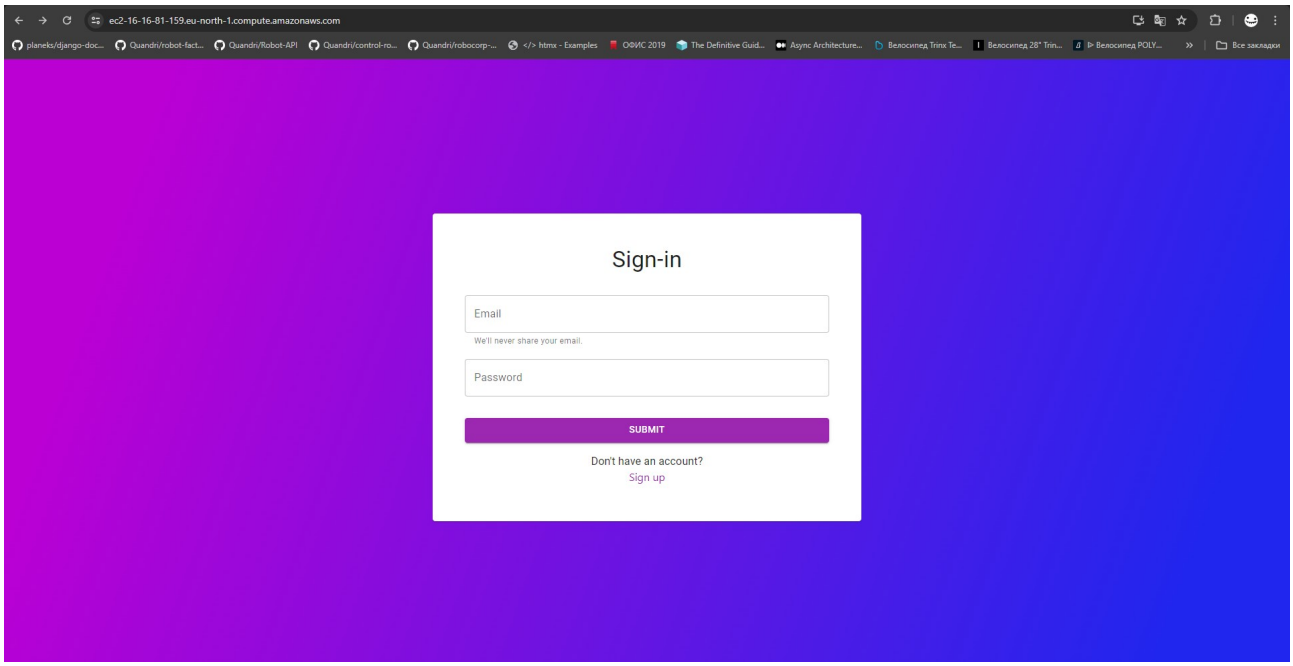
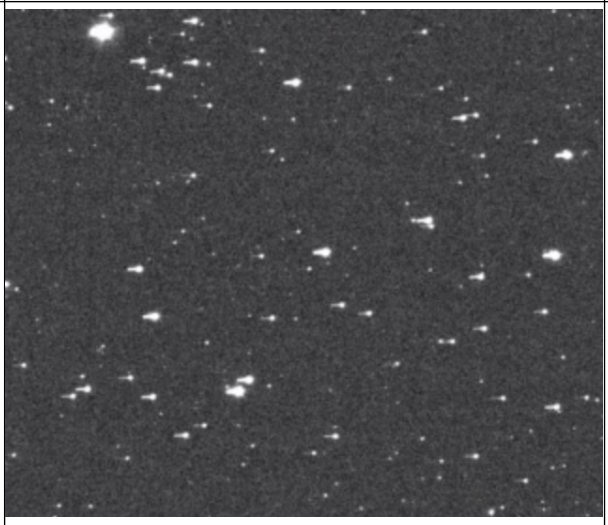
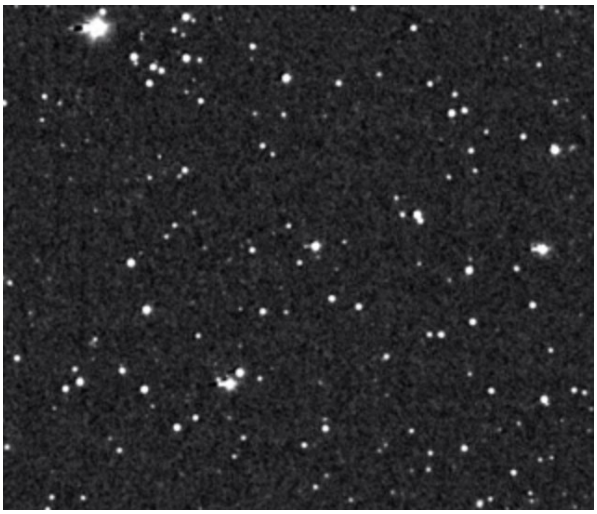
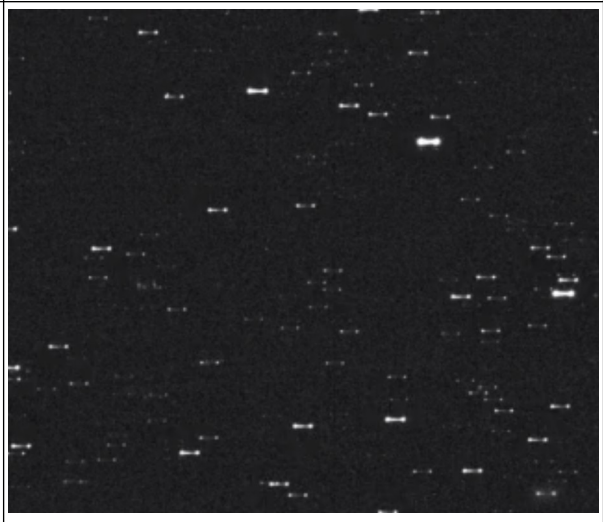
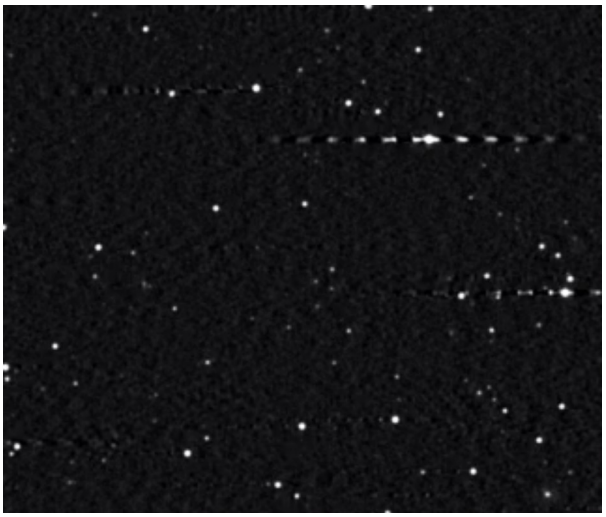
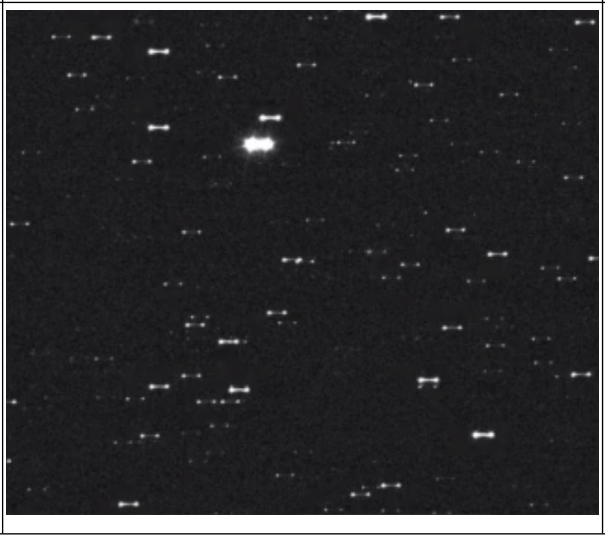
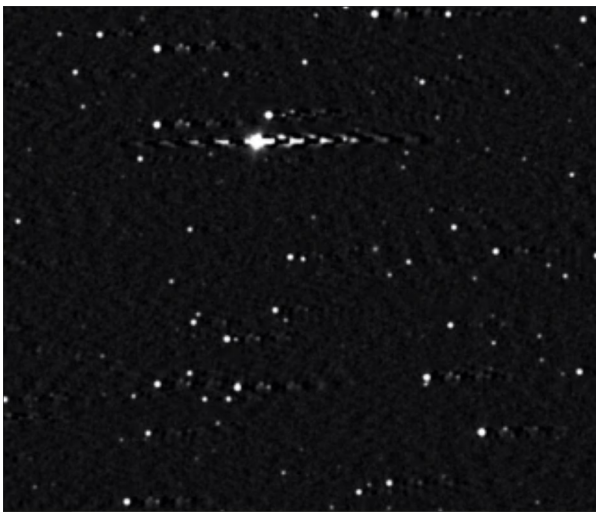


Рисунок 4.10 – Вигляд розгорнутого застосунку

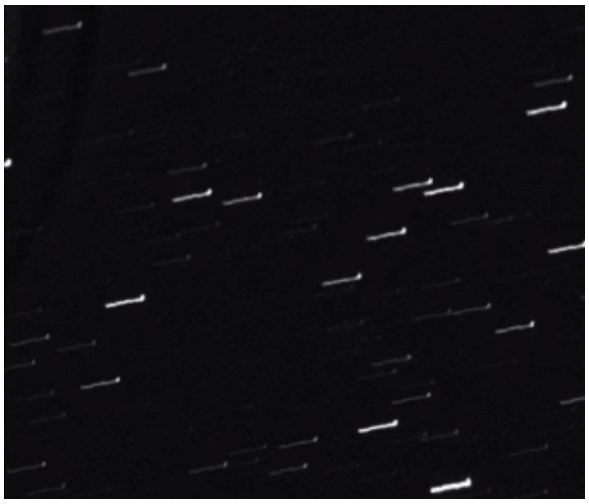
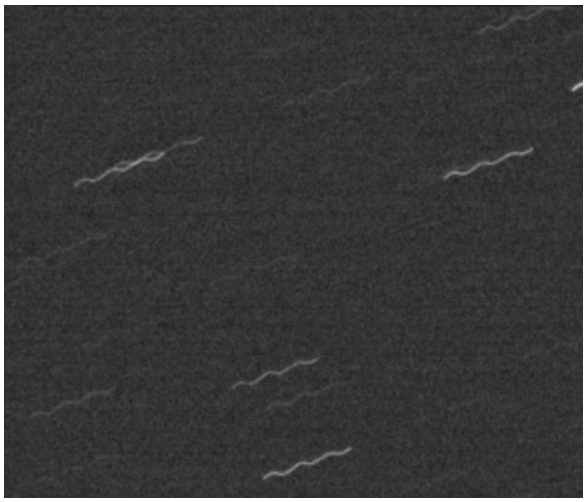
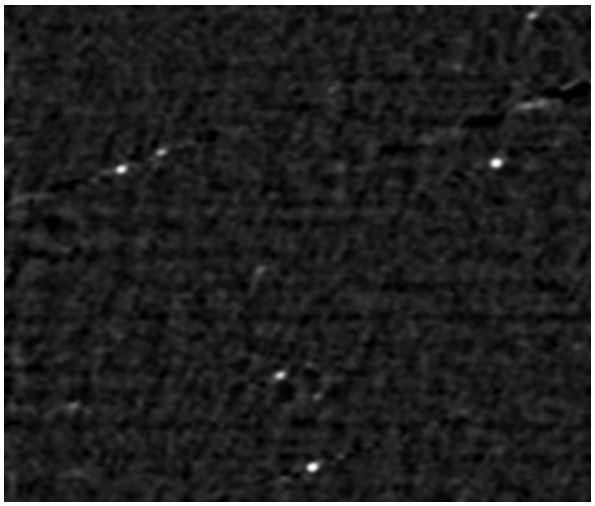
4.2 Тестування модуля відновлення зображення

З метою перевірки працездатності модуля відновлення зображень було використано декілька кадрів та створено завдання в системі. У таблиці 4.1 подано кадри до та після відновлення. Можна побачити, що ключова мета була досягнута, боковий зсув зображень та змазаність була усунута, натомість в деяких зображеннях можна побачити побічний ефект у вигляді появи артефактів.

Таблиця 4.1 — Результати відновлення кадрів

№	Вхідний кадр	Вихідний кадр
1		
2		
3		

Продовження таблиці 4.1

№	Вхідний кадр	Вихідний кадр
4		
5		

4.2 Перевірка продуктивності розроблених компонентів інформаційної системи

З метою перевірки продуктивності розроблених компонентів інформаційної системи та можливостей масштабування було реалізовано скрипт, що дозволяє створити визначену кількість завдань з відновлення зображення в системі.

Цей підхід дозволяє визначити середній час обробки зображень, що є важливим показником продуктивності системи. Порівняння цього часу при різних конфігураціях системи дозволяє оцінити ефективність масштабування та оптимізації ресурсів. На рисунку 4.11 подано результат запуску скрипту, як можна побачити в даному випадку було створено 4 завдання, для кожного з

яких було завантажено файл зображення розміром 800x800 пікселів.

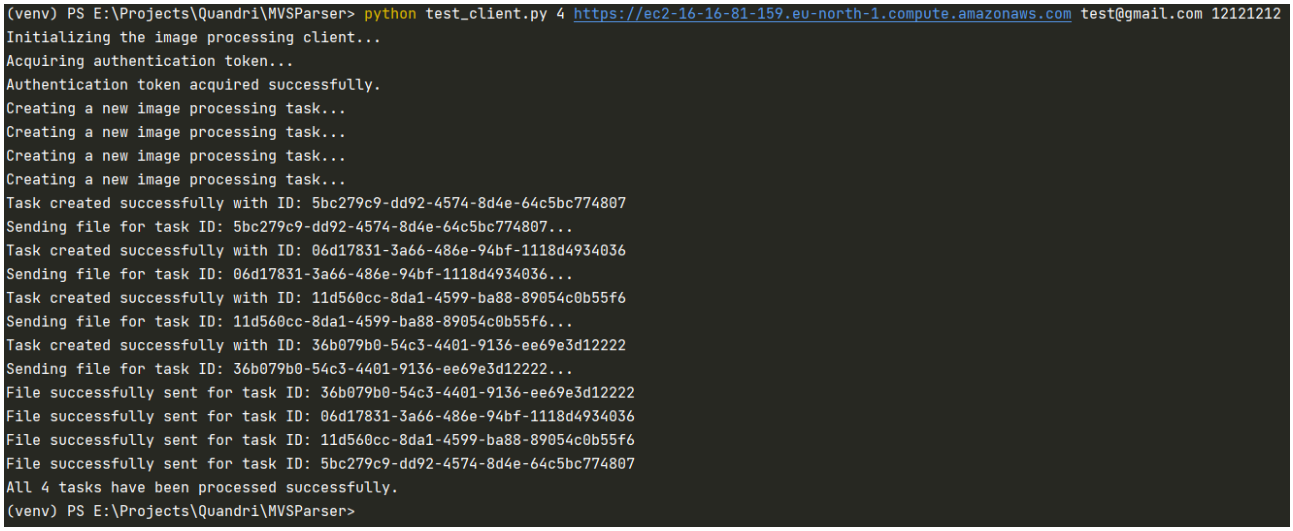


Рисунок 4.11 – Приклад запуску скрипту для перевірки продуктивності системи

Щоб переконатись, що завдання дійсно було створено, можна відкрити перелік створених завдань за допомогою веб-інтерфейсу, як показано на рисунку 4.12. Бачимо, що було створено 4 завдання, які одразу були відправлені на виконання, про це можна свідчити зі статусу «In progress» кожного завдання.

Після виконання завдань, можна побачити, що завдання у відповідному списку отримали новий статус «Finished», що підтверджує успішність їх виконання, як показано на рисунку 4.13. Кожне завдання має інформацію, що свідчить про час виконання а саме:

– поле «Processing time spent» – час, що було витрачено на виконання відновлення зображення, в цей час не включено очікування завдання на початок виконання;

– поле «Total time spent» – час, що включає як час виконання завдання, так і час очікування завдання в черзі, є очевидним що цей час завжди більший за час виконання завдання і залежить від завантаженості черги завдань.

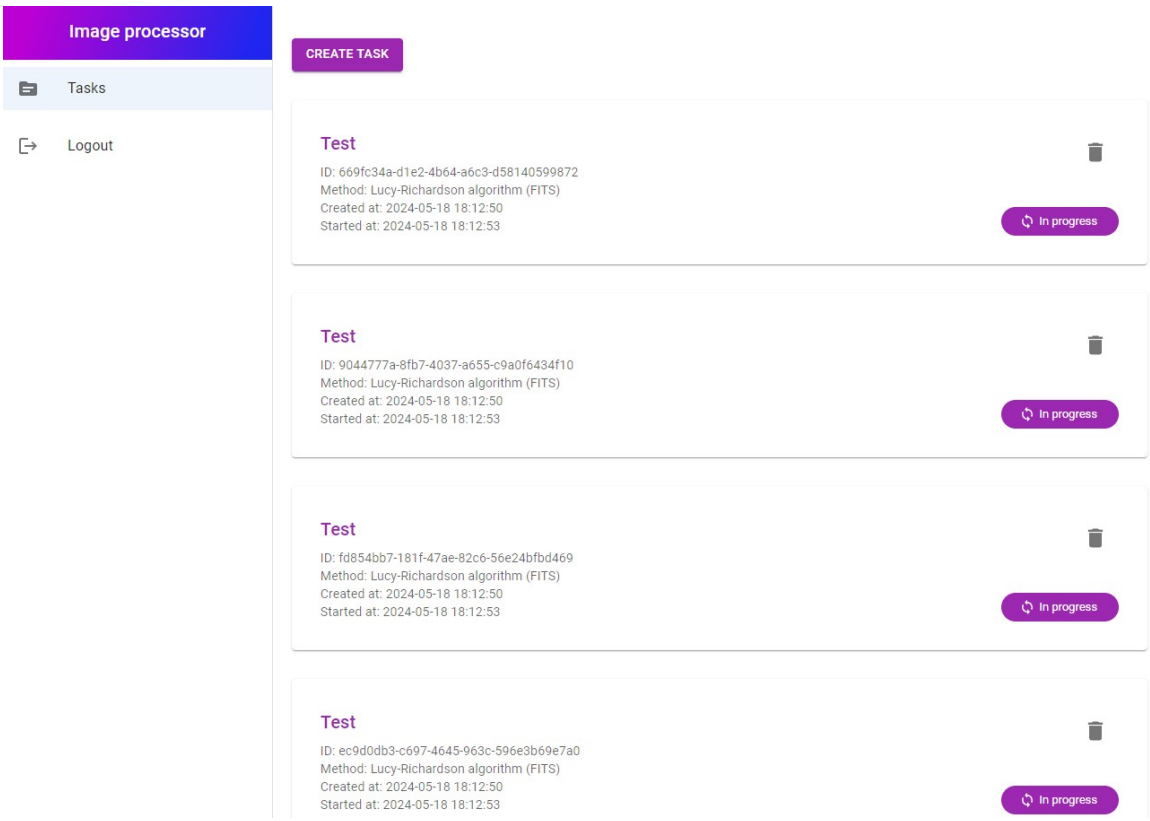


Рисунок 4.12 – Перелік створених завдань

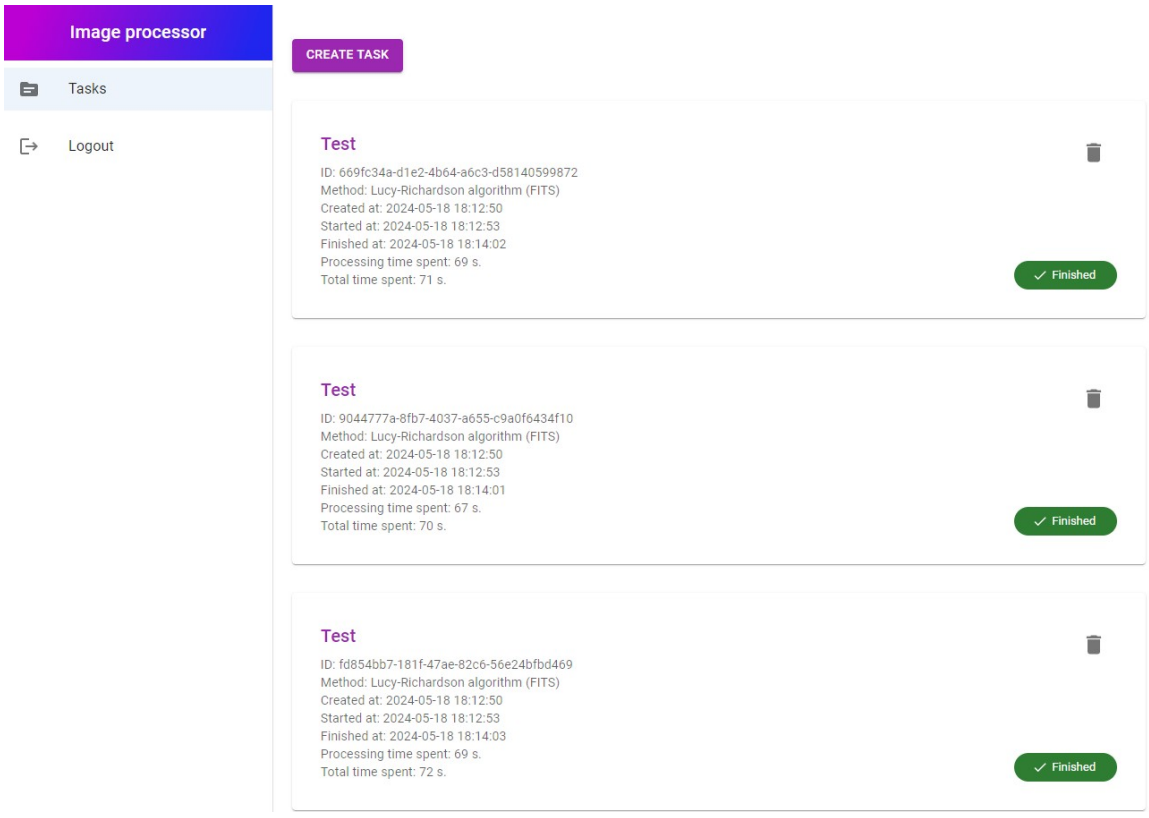


Рисунок 4.13 – Перелік виконаних завдань

Ключовим показником для перевірки продуктивності системи було обрано середній час виконання завдань в системі, який визначається за наступною формулою:

$$T_{avg} = \frac{\sum_{i=1}^{n} T_{proc}}{n} \tag{3.1}$$

де T_{avg} – середній час обробки зображень в системі;

n – кількість завдань для обробки зображення;

T_{proc} – час обробки і-ого завдання.

Перевірку продуктивності системи було виконано для двох конфігурацій екземпляра EC2, а саме «c5.2xlarge» та «c5.4xlarge». Для кожної конфігурації було проведено 8 тестів з різною кількістю завдань (від 4 до 32 з кроком 4). Як раніше зазначалось, зміна конфігурації екземпляра EC2 за допомогою зміни типу екземпляра. На рисунку 4.14 можна побачити процес зміни типу екземпляра EC2, який було виконано після перевірки продуктивності першої конфігурації системи.

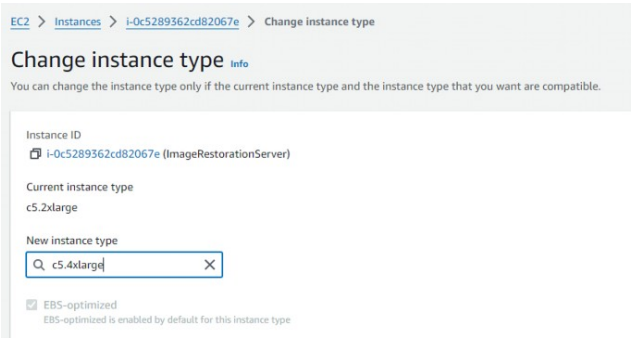


Рисунок 4.14 — Процес зміни типу екземпляра EC2

Результати перевірки продуктивності розробленої системи відновлення зображень подано у таблиці 4.1, де було зазначено середній час обробки завдань для певної кількості завдань у системі для різних конфігурацій екземплярів EC2.

Таблиця 4.2 – Результати перевірки продуктивності системи

Кількість завдань	Середній час обробки, с c5.2xlarge 8 vCPU (4 core) 16 GiB	Середній час обробки, с c5.4xlarge 16 vCPU (8 core) 32 GiB
4	69	64
8	118,5	68,37
12	177,68	88
16	236,75	115,87
20	293,95	144,55
24	355,37	173,04
28	422,07	202,03
32	476,25	230,25

Відповідні результати було представлено у вигляді графіку, який показано на рисунку 4.15.

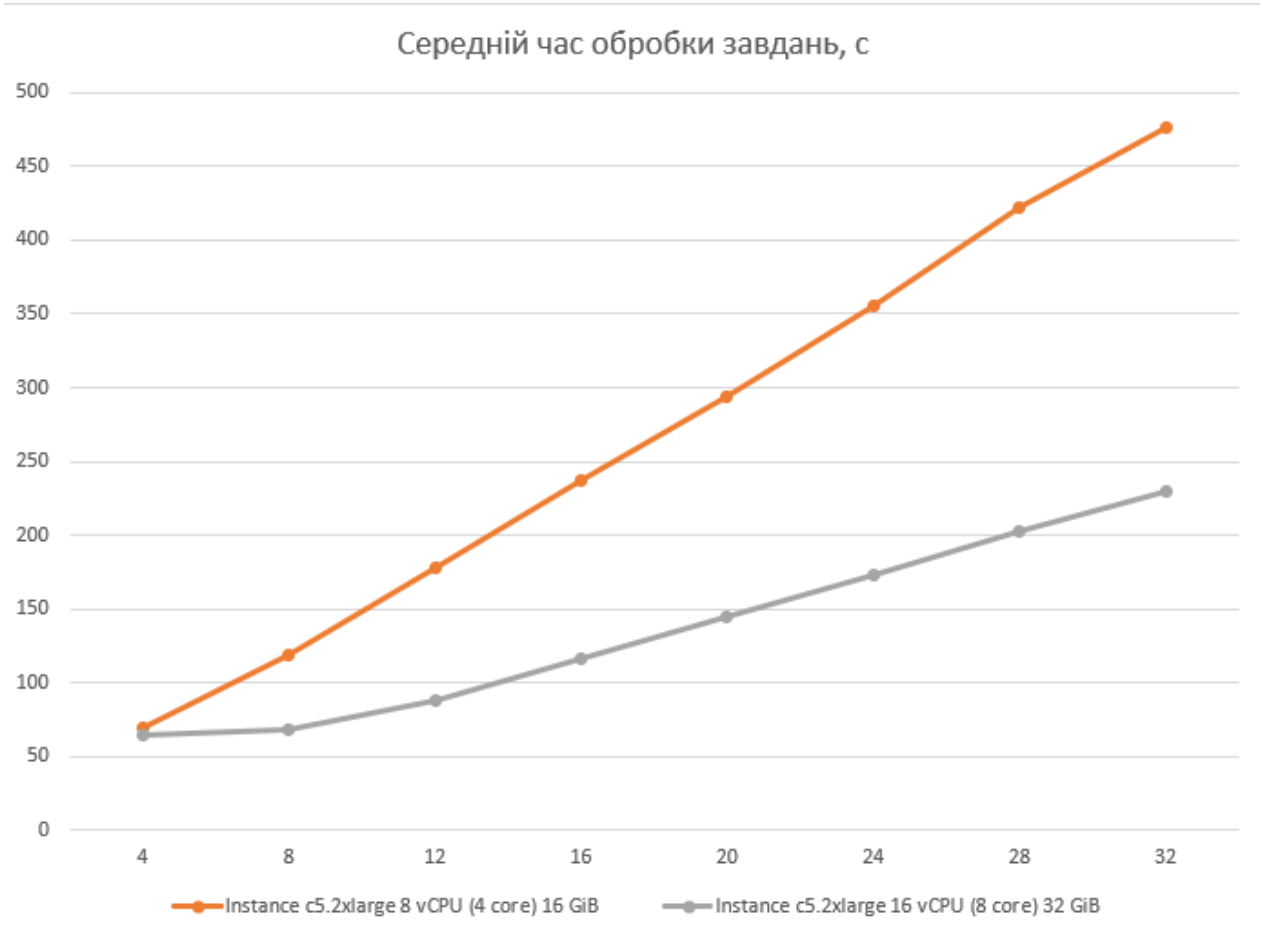


Рисунок 4.15 – Графік середнього часу обробки завдань для різних конфігурацій системи

Результати оцінювання продуктивності інформаційної системи свідчать

про те, що продуктивність системи напряму залежить від обсягу оброблюваних завдань.

Детальний аналіз показав, що конфігурація «c5.2xlarge» є оптимальною для випадків з меншою кількістю завдань в системі, забезпечуючи високий рівень продуктивності при мінімальних витратах.

У той же час, конфігурація «c5.4xlarge» продемонструвала вищу ефективність у ситуаціях із значним навантаженням. Поглиблений аналіз продуктивності обробки зображень виявив, що більш потужна конфігурація EC2 значно перевершує менш потужні налаштування. Зокрема, при обробці 32 завдань середній час обробки для конфігурації «c5.4xlarge» становить 230,25 секунд, тоді як для конфігурації «c5.2xlarge» цей показник складає 476,25 секунд, що в 2,06 рази більше. Це підтверджує існування сильної кореляції між збільшенням кількості віртуальних процесорів у віртуальній машині та зменшенням середнього часу виконання завдань: подвоєння кількості процесорів сприяє зменшенню часу виконання завдань удвічі.

Крім того, результати дослідження свідчать про важливість вибору відповідної конфігурації для досягнення оптимальної продуктивності системи. Вибір конфігурації залежить від специфічних вимог до навантаження та обсягу оброблюваних даних. Так, для сценаріїв з високими вимогами до продуктивності, доцільним є використання конфігурацій з більшим обсягом ресурсів, таких як «c5.4xlarge». Натомість, для завдань із меншим обсягом обробки, оптимальним вибором є конфігурації типу «c5.2xlarge», що дозволяє зменшити витрати при збереженні високого рівня продуктивності.

Отже, результати цього дослідження підтверджують важливість ретельного підбору конфігурацій EC2 для забезпечення ефективної роботи інформаційної системи в залежності від обсягу та складності оброблюваних завдань.

ВИСНОВКИ

Дослідження відновлення астрономічних зображень відіграють ключову роль у покращенні якості астрономічних даних, що важливі для наукового дослідження Всесвіту. Аналіз поточного стану засобів обробки астрономічних зображень виявив низку недоліків, включаючи обмежені можливості масштабування, складність інтерфейсів та високу залежність від спеціалізованого апаратного та програмного забезпечення. З цього випливає важливість розробки нових підходів і методів, які дозволять більш ефективно вирішувати ці проблеми. У дослідженні було запропоновано новий підхід до відновлення астрономічних зображень за допомогою використання інформаційної системи розробленої із застосуванням хмарних технологій.

До складу компонентів розробленої інформаційної системи входять математичні модулі, що реалізують алгоритм Люсі-Річардсона для відновлення астрономічних зображень з використанням методів попередньої обробки даних для покращення якості зображень.

Особлива увага приділена оптимізації процесу обробки зображень із збереженням максимальної ефективності відновлення зображень при високих навантаженнях у системі за рахунок проведення горизонтального масштабування засобами хмарних технологій.

Ефективність горизонтального масштабування в поточній задачі підтверджена дослідженнями, які показали, що використання хмарних технологій в запропонованій системі дозволяє зберегти високу продуктивність та надійність при збільшенні об’єму оброблюваних даних.

Результати роботи було впроваджено при виконанні молодіжної держбюджетної науково-дослідної роботи №347 «Розробка обчислювальних методів виявлення об'єктів з близьконульовим та локально незмінним рухом оптико-електронними засобами», що демонструє її практичну значимість та можливість використання в реальних умовах (див. додаток Г).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Hadzhyiev, E. R. Development of an information system for processing astronomical images using cloud technologies: 28-й Міжнародний молодіжний форум «Радіоелектроніка та молодь XXI столітті». Зб. матеріалів форуму. Т.6. Конференція «Інформаційні інтелектуальні системи». Харків: ХНУРЕ. 2024. С. 834-835.

2. Khlamov S., Savanevych V., Vlasenko V., Hadzhyiev E. Automated data mining of the single objects from blurred astronomical CCD frames using the Lucy-Richardson deconvolution: CEUR Workshop Proceedings, 2024. Vol. 3664. P. 178 – 191.

3. Roddier F. Adaptive Optics in Astronomy. Cambridge University Press, 1999. 411 p.

4. Starck J.-L., Murtagh F. Astronomical Image and Data Analysis. 2nd ed. Springer, 2006. 352 p.

5. Bellanger M., Benjamin A. Digital Signal Processing: Theory and Practice, 10th Edition. Wiley, 2024. 400 p.

6. Berry R., Burnell J. The Handbook of Astronomical Image Processing. Willmann-Bell, Inc., 2005. 684 p.

7. AstroImageJ (AIJ) - ImageJ for Astronomy. URL: https://www.astro.louisville.edu/software/astroimagej/ (дата звернення: 12.03.2024).

8. IRAF Community Distribution home page. URL: https://iraf-community.github.io/ (дата звернення: 15.03.2024).

9. Cooper A., Reimann R., Cronin D. About Face: The Essentials of Interaction Design. 4th ed. Wiley, 2014. 720 p.

10. Kleppmann M. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. O'Reilly Media, 2017. 616 p.

11. Burns B. Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services. O'Reilly Media, 2018. 166 p.

12. Wittig A., Wittig M. Amazon Web Services in Action. Manning

Publications, 2018. 528 p.

13. Остапов С.Е., Євсеєв С. П., Король О.Г. Технології захисту інформації : навч. посіб. Х.: Вид. ХНЕУ, 2013. 476 с.

14. Розенберг Д. Применение объектного моделирования с использованием UML и анализ прецедентов. М.:ДМК-издательство, 2002. 169 с.

15. Khlamov S., Savanevych V., Briukhovetskyi O., Pohorelov A. CoLiTec software–detection of the near-zero apparent motion. Proceedings of the International Astronomical Union: Cambridge University Press, 2017. Vol. 12. P. 349 – 352.

16. He Z., Wang H., A Deconvolutional Reconstruction Method Based on Lucy–Richardson Algorithm for Joint Scanning Laser Thermography. IEEE Transactions on Instrumentation and Measurement, 2021. Vol. 70. P. 1-8.

17. Howell S. B. Handbook of CCD Astronomy. 2nd ed. Cambridge University Press, 2006. 218 p.

18. Khetkeeree S. Optimization of Lucy-Richardson Algorithm Using Modified Tikhonov Regularization for Image Deblurring, J. Phys.: Conf. Ser. 2020. Vol. 1438.

19. Faaique M. Overview of Big Data Analytics in Modern Astronomy, Int. Journal of Mathematics, Statistics, and Computer Science. 2023. Vol. 2. P. 96–113.

20. ImageMagick – Convert, Edit, or Compose Digital Images. URL: https://imagemagick.org/index.php (дата звернення: 17.03.2024).

21. Python 3.12.2 documentation. URL: https://docs.python.org/3 (дата звернення: 17.03.2024).

22. Redis Queue documentation. URL: https://redis.com/glossary/redis-queue/ (дата звернення: 17.03.2024).

23. Python RQ documentation. URL: https://python-rq.org/docs/ (дата звернення: 17.03.2024).

24. Docker-Compose overview. URL: https://docs.docker.com/compose/ (дата звернення: 18.03.2024).

25. Ibrahim M. H., Sayagh M., and Hassan A. E. A study of how Docker Compose is used to compose multi-component systems, Empirical Software Engineering, 2021. Vol. 26, P. 128.