

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи високопродуктивних
обчислень для моделювання фізичних процесів у розподілених
обчислювальних середовищах”

Виконав здобувач вищої освіти
IV курсу, групи КН-22
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Червоний Є. В.
« ____ » _____ 2026 р.

Керівник проекту
доктор технічних наук, професор
_____ Коваленко О. В.
« ____ » _____ 2026 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань . 12 "Інформаційні технології"
Спеціальність 122 "Комп'ютерні науки"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерні науки"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 06 » лютого 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Червоному Єгору Владиславовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення системи високопродуктивних обчислень для моделювання фізичних процесів у розподілених обчислювальних середовищах

2. Керівник роботи Коваленко Олександр Володимирович, доктор технічних наук, професор

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 116-02 від 06.02.2026 року

3. Строк подання студентом роботи до захисту 23.05.2026 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою роботи є розробка Програмного забезпечення системи високопродуктивних обчислень для моделювання фізичних процесів у розподілених обчислювальних середовищах

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

7. Дата видачі завдання « 06 » лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2026 р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.04.2026 р.	
8.	Попередній захист роботи	23.05.2026 р.	

Дата видачі завдання
« 06 » лютого 2026 р.

Підпис керівника

Коваленко О.В.
(прізвище та ініціали)

Завдання прийнято до виконання
« 06 » лютого 2026 р.

Підпис здобувача

Червоний Є.В.
(прізвище та ініціали)

АНОТАЦІЯ

Червоний Є.В. Програмне забезпечення системи високопродуктивних обчислень для моделювання фізичних процесів у розподілених обчислювальних середовищах. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи високопродуктивних обчислень для моделювання фізичних процесів у розподілених обчислювальних середовищах.

Метою розробки є програмне забезпечення системи високопродуктивних обчислень для моделювання фізичних процесів у розподілених обчислювальних середовищах.

Результат роботи – програмна реалізація системи високопродуктивних обчислень для моделювання фізичних процесів у розподілених обчислювальних середовищах.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерні науки, високопродуктивні обчислення, моделювання фізичних процесів, розподілені обчислювальні середовища

ABSTRACT

Chervonyi, Y.V. Software for high-performance computing systems for modelling physical processes in distributed computing environments. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

This bachelor's thesis presents software designed for a high-performance computing system used to simulate physical processes in distributed computing environments.

The goal of the development is to create software for a high-performance computing system designed to model physical processes in distributed computing environments.

The result of the work is the software implementation of a high-performance computing system for modeling physical processes in distributed computing environments.

During the development of the software model, an analysis of existing hardware and software tools was performed. All components of the developed software are fully described.

A user-friendly interface has been developed. Instructions for working with the software tools are provided.

The program can be used on PCs running Windows 10/11.

The program was developed in the Python environment.

Keywords: computer science, high-performance computing, modeling of physical processes, distributed computing environments

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ ...	3
ВСТУП.....	5
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	8
1.1 Призначення системи.....	8
1.2 Область застосування.....	10
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	13
2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	13
2.2 Обґрунтування вибору засобів для побудови системи	21
2.3 Розгорнута постановка завдання	50
3 ОПИС І ОБГРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ.....	52
3.1 Опис функціонування системи	52
3.2 Розробка структурної схеми.....	54
3.3 Розробка функціональної схеми	56
3.4 Розробка діаграми процесів.....	60
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ ТА ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ І ПРОГРАМНИХ РІШЕНЬ..	62
4.1 Розробка блок-схем та опис алгоритмів функціонування системи	62
4.2 Захист розробленого програмного забезпечення.....	72
5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	75
6 ОСНОВНІ ВИСНОВКИ.....	79
СПИСОК ЛІТЕРАТУРИ.....	84

					ВКРБ-122.26.0034.00.00.ПЗ 1			
Вим.	Арк.	№ докум.	Підп.	Дата	Програмне забезпечення системи високопродуктивних обчислень для моделювання фізичних процесів у розподілених обчислювальних середовищах	Літ.	Арквш	Арквшів
Розроб.	Червоний С.В.					Б	1	88
Перев.	Коваленко О.В.							
Н.контр.	Коваленко А.С.					ЦНТУ КН-22		
Затв.	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

HPC	- High Performance Computing - високопродуктивні обчислення
BOINC	- Berkeley Open Infrastructure for Network Computing - платформа розподілених обчислень
CERN	- European Organization for Nuclear Research - Європейська організація ядерних досліджень
LHC	- Large Hadron Collider - Великий адронний колайдер
LHC@Home	- Проєкт розподілених обчислень CERN на платформі BOINC
ATLAS	- A Toroidal LHC ApparatuS - експеримент CERN для дослідження елементарних частинок
ATLAS@Home	- Розподілений обчислювальний проєкт експерименту ATLAS
SixTrack	- Програмний комплекс для моделювання траєкторій частинок
CPDN	- Climateprediction.net - платформа кліматичного моделювання
OpenIFS	- Open Integrated Forecasting System - дослідницька версія системи прогнозування ECMWF
ECMWF	- European Centre for Medium-Range Weather Forecasts - Європейський центр середньострокових прогнозів погоди
CPU	- Central Processing Unit - центральний процесор
VM	- Virtual Machine - віртуальна машина
API	- Application Programming Interface - інтерфейс програмування застосунків
HTTP	- HyperText Transfer Protocol - протокол передачі гіпертексту
HTTPS	- HyperText Transfer Protocol Secure - захищений HTTP-протокол
TCP	- Transmission Control Protocol - протокол передачі даних
CGI	- Common Gateway Interface - механізм запуску зовнішніх програм веб-сервером

FastCGI	-	Протокол взаємодії веб-сервера із зовнішніми застосунками
fcgiwrap	-	Служба запуску CGI/FastCGI-застосунків у Linux
Nginx	-	Високопродуктивний веб-сервер та reverse proxy
Prometheus	-	Система збору та збереження метрик моніторингу
Grafana	-	Платформа візуалізації та аналізу метрик
Exporter	-	Модуль збору та передачі метрик у систему моніторингу
Python	-	Високорівнева мова програмування
Linux	-	Сімейство Unix-подібних операційних систем
Dashboard	-	Інформаційна панель моніторингу
Alerting	-	Система автоматичних сповіщень
Endpoint	-	Точка доступу до сервісу або API
Metrics	-	Метрики стану та продуктивності системи
Work Unit (WU)	-	Окрема обчислювальна задача у BOINC
Checkpoint	-	Контрольна точка відновлення виконання задачі
Scheduler	-	Компонент планування та розподілу задач
DevOps	-	Методологія автоматизації процесів розробки та адміністрування ПЗ
SSL/TLS	-	Криптографічні протоколи захищеного з'єднання

ВСТУП

Актуальність теми. У сучасних умовах стрімкого розвитку інформаційних технологій спостерігається постійне зростання складності обчислювальних задач та обсягів даних, які необхідно обробляти у наукових, технічних та прикладних дослідженнях. Особливо актуальною дана проблема є у сферах моделювання фізичних процесів, математичного аналізу, кліматичного прогнозування, біоінформатики, астрофізики та досліджень у галузі фізики високих енергій. Для виконання подібних задач необхідні значні обчислювальні ресурси, які не можуть бути ефективно забезпечені традиційними персональними комп'ютерами або окремими серверними системами.

У зв'язку з цим особливого значення набувають системи високопродуктивних обчислень (High Performance Computing, HPC), які дозволяють виконувати складні обчислювальні операції із використанням паралельної та розподіленої обробки даних. HPC-системи забезпечують можливість одночасного використання великої кількості обчислювальних вузлів, що дозволяє суттєво скоротити час виконання задач та підвищити ефективність використання апаратних ресурсів.

Одним із найбільш перспективних напрямків розвитку HPC є використання розподілених обчислювальних середовищ. На відміну від класичних суперкомп'ютерів, розподілені системи дозволяють об'єднати значну кількість незалежних комп'ютерів у єдину обчислювальну інфраструктуру. Такий підхід забезпечує масштабованість, відмовостійкість та можливість використання вже існуючих ресурсів без необхідності створення дорогих спеціалізованих центрів обробки даних. Крім того, розподілені обчислення дозволяють залучати до виконання задач ресурси звичайних користувацьких комп'ютерів, що значно розширює доступні обчислювальні потужності.

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		4

Особливу роль у розвитку розподілених обчислень відіграє платформа BOINC (Berkeley Open Infrastructure for Network Computing), яка є однією з найбільш відомих та поширених систем volunteer computing. Дана платформа забезпечує механізми автоматичного розподілу обчислювальних задач між вузлами, контролю виконання обчислень, перевірки результатів та централізованого керування інфраструктурою. За допомогою BOINC реалізовано велику кількість наукових проектів світового рівня, серед яких особливе місце займають проекти, що використовуються для моделювання фізичних процесів та кліматичних змін.

Сучасні HPC-системи дедалі частіше інтегруються із засобами віртуалізації, контейнеризації та автоматизованого моніторингу. Використання технологій віртуалізації дозволяє стандартизувати середовище виконання задач, забезпечити ізоляцію обчислювальних процесів та підвищити стабільність роботи інфраструктури. Одночасно із цим виникає потреба у використанні сучасних засобів спостережуваності (observability), що дозволяють контролювати навантаження системи, аналізувати продуктивність та оперативно виявляти помилки під час виконання обчислень.

Мета й завдання дослідження

Метою роботи є дослідження принципів побудови системи високопродуктивних обчислень для моделювання фізичних процесів у розподілених обчислювальних середовищах, а також аналіз підходів до організації ефективного розподілу обчислювального навантаження між вузлами обчислювальної системи.

Для досягнення мети під час роботи було вирішено наступні завдання:

- дослідити сучасні підходи до побудови систем високопродуктивних обчислень (HPC) та розподілених обчислювальних середовищ
- проаналізувати архітектуру та принципи роботи платформ розподілених обчислень, зокрема BOINC та аналогічних систем;

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		5

- визначити основні вимоги до програмного забезпечення для моделювання фізичних процесів у розподіленій інфраструктурі;
- дослідити механізми декомпозиції обчислювальних задач та їх розподілу між обчислювальними вузлами;
- проаналізувати підходи до організації обміну даними між вузлами системи та централізованим координатором;
- дослідити принципи моніторингу виконання задач та збору результатів обчислень;
- розглянути способи інтеграції математичних моделей фізичних процесів у розподілене середовище виконання;
- оцінити ефективність та масштабованість різних підходів до реалізації подібних систем.

Практична цінність отриманих результатів

Результати, отримані мають практичне значення для підготовки фахівців у сфері високопродуктивних та розподілених обчислень, оскільки реалізована інфраструктура на базі BOINC дозволяє виконувати реальні наукові задачі з моделювання фізичних процесів у рамках проектів LHC@home та ClimatePrediction.net, а також досліджувати ефективність використання обчислювальних ресурсів. Додатково впроваджена система моніторингу з використанням exporter та Grafana забезпечує контроль за навантаженням і виконанням задач у реальному часі, що формує практичні навички роботи з розподіленими системами та інструментами спостереження, а сама інфраструктура може бути використана як навчальний стенд або база для подальших досліджень у галузі НРС.

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Дана система призначена для організації високопродуктивних розподілених обчислень, орієнтованих на моделювання складних фізичних процесів, що потребують значних обчислювальних ресурсів та тривалого часу виконання. Її основне призначення полягає у створенні програмно-апаратного середовища, здатного ефективно використовувати доступні обчислювальні потужності для виконання наукових задач, які неможливо або економічно недоцільно розв'язувати на одному окремому комп'ютері. Такий підхід дає змогу значно розширити обчислювальні можливості дослідника без необхідності використання дорогих спеціалізованих суперкомп'ютерних систем.

На відміну від традиційних обчислювальних рішень, де всі операції виконуються на одному сервері або в межах обмеженого локального кластера, запропонована система ґрунтується на розподілі навантаження між великою кількістю вузлів, які можуть знаходитися у різних фізичних і мережевих середовищах. Це дозволяє поєднувати ресурси окремих машин у єдину інфраструктуру та досягати суттєвого збільшення загальної продуктивності. Завдяки цьому система є придатною для задач, що містять велику кількість незалежних підобчислень, а також для проектів, у яких важливу роль відіграє паралельна обробка великих масивів даних.

Особливістю системи є використання платформи BOINC як базового механізму розподілу задач і керування обчислювальними клієнтами. Така архітектура забезпечує автоматизовану взаємодію між серверною частиною та клієнтськими вузлами, дозволяючи централізовано формувати обчислювальні завдання, передавати вхідні дані, контролювати виконання задач та отримувати результати після завершення обчислень. Клієнтський модуль працює у фоновому

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		7

режимі, що забезпечує непомітну для користувача участь системи в обчислювальному процесі та дає змогу використовувати вільні ресурси пристрою без істотного впливу на його основну роботу.

Призначенням системи також є забезпечення повного життєвого циклу обчислювальної задачі. Це включає формування та розподіл завдань, доставку необхідних вхідних параметрів, виконання моделювання фізичних процесів, збереження проміжних і фінальних результатів, а також подальшу перевірку коректності отриманих даних. Такий цикл дозволяє реалізувати як тривалі симуляції, так і велику кількість коротких незалежних обчислень, що є типовим для наукових і інженерних досліджень.

Система має також важливе значення з точки зору відмовостійкості та масштабованості. Оскільки обчислювальні вузли можуть підключатися та відключатися динамічно, архітектура повинна підтримувати стабільне функціонування навіть за часткової втрати окремих ресурсів. Саме тому у системі передбачено механізми автоматичного повторного призначення задач, контролю виконання та збереження службової інформації. Це робить її придатною для тривалого використання в середовищах, де навантаження змінюється в часі або доступність вузлів не є постійною.

Окремим напрямом призначення системи є підтримка дослідницької та навчальної діяльності. Вона може використовуватись як інструмент для виконання наукових обчислень у галузі фізики, кліматичного моделювання, аналізу експериментальних даних та інших напрямів, де необхідне паралельне виконання великої кількості однотипних обчислювальних процедур. Крім того, система є придатною для демонстрації принципів побудови розподілених обчислювальних середовищ, що робить її цінною з погляду навчального процесу та практичної підготовки здобувачів освіти.

У цілому призначення розробленої системи полягає у створенні гнучкої, масштабованої та ефективної платформи для високопродуктивних обчислень, яка може застосовуватись у наукових, освітніх та прикладних задачах. Вона дозволяє

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		8

значно підвищити продуктивність обробки даних, зменшити час виконання складних обчислювальних операцій та забезпечити зручні умови для організації роботи в розподіленому середовищі.

1.2 Область застосування

Розроблена система високопродуктивних розподілених обчислень призначена для використання у сферах, де виникає потреба у виконанні великої кількості ресурсоємних обчислювальних операцій. Використання архітектури НРС у поєднанні з платформою BOINC дозволяє ефективно об'єднувати ресурси великої кількості обчислювальних вузлів у єдине програмно-апаратне середовище. Завдяки цьому система може використовуватись для реалізації як фундаментальних наукових досліджень, так і прикладних задач, пов'язаних із аналізом даних, математичним моделюванням та складними інженерними розрахунками.

Однією з основних областей застосування системи є наукові дослідження. Значна частина сучасних наукових задач потребує виконання тривалих чисельних обчислень, які неможливо ефективно реалізувати на одному комп'ютері. Розподілене середовище дозволяє виконувати паралельну обробку великої кількості незалежних задач, що особливо важливо для моделювання фізичних процесів, аналізу експериментальних даних та проведення комп'ютерних симуляцій. Система може застосовуватись у задачах фізики високих енергій, астрофізики, кліматології, молекулярної динаміки, квантових обчислень та дослідження складних математичних моделей.

У сфері фізичного моделювання система дозволяє реалізовувати обчислювальні експерименти, пов'язані з аналізом поведінки складних динамічних систем. Зокрема, можливе моделювання взаємодії частинок, теплових процесів, електромагнітних полів, аеродинамічних явищ та інших процесів, які вимагають виконання великої кількості математичних операцій.

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		9

Використання розподілених обчислень значно скорочує час виконання таких задач та дозволяє збільшити точність моделей шляхом підвищення кількості обчислювальних ітерацій.

Важливим напрямом застосування є інженерія та промисловість. Система може використовуватись для проведення інженерних розрахунків, аналізу міцності конструкцій, моделювання навантажень, теплових режимів та динамічних процесів у складних технічних системах. Подібні задачі широко застосовуються у машинобудуванні, енергетиці, авіаційній та автомобільній галузях. Завдяки можливості розподілу обчислень між великою кількістю вузлів система дозволяє виконувати моделювання значно швидше у порівнянні з традиційними локальними рішеннями.

Окремою сферою застосування є аналіз великих обсягів даних (Big Data). Сучасні наукові експерименти, телеметричні системи та сенсорні мережі формують значні масиви інформації, обробка яких потребує великих обчислювальних ресурсів. Розроблена система може використовуватись для паралельного аналізу даних, статистичної обробки, пошуку закономірностей та виконання ресурсоемних аналітичних операцій. Такий підхід дозволяє суттєво прискорити процес обробки інформації та підвищити ефективність використання апаратних ресурсів.

Система також може застосовуватись у задачах штучного інтелекту та машинного навчання. Навчання сучасних моделей штучних нейронних мереж часто вимагає виконання великої кількості однотипних математичних операцій над значними масивами даних. Використання розподіленої інфраструктури дозволяє паралельно виконувати окремі етапи обробки даних або розподіляти навчальні задачі між вузлами системи, що скорочує час обчислень та підвищує загальну продуктивність середовища.

Перспективним напрямом є використання системи для екологічних та кліматичних досліджень. Розрахунок кліматичних моделей передбачає аналіз великої кількості параметрів атмосфери, океанічних процесів, температурних

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		10

змін та екологічних факторів. Подібні задачі потребують значних обчислювальних потужностей через високу складність математичних моделей. Розподілене середовище дозволяє проводити довготривалі симуляції кліматичних процесів, прогнозувати зміни навколишнього середовища та оцінювати вплив різних факторів на екосистеми.

У сфері інформаційної безпеки система може використовуватись для виконання задач криптоаналізу, тестування криптографічних алгоритмів, аналізу захищеності інформаційних систем та моделювання навантажень. Розподілені обчислення дозволяють виконувати паралельний перебір великих наборів даних, що є актуальним для дослідження сучасних криптографічних механізмів та систем захисту інформації.

Розроблена система має також значний потенціал використання у навчальному процесі. Вона може застосовуватись як навчальний стенд для підготовки студентів спеціальностей, пов'язаних із комп'ютерними науками, високопродуктивними обчисленнями, DevOps, системним адмініструванням та мережевими технологіями. Практична робота із платформою BOINC, системами моніторингу, віртуалізацією та розподіленими обчисленнями дозволяє сформувати навички роботи з сучасною НРС-інфраструктурою та технологіями побудови масштабованих інформаційних систем.

Завдяки використанню відкритої архітектури та підтримці стандартних мережових протоколів система може інтегруватися з існуючими науковими проектами та зовнішніми обчислювальними платформами. Це дозволяє розширювати інфраструктуру без значних змін базової архітектури та забезпечує можливість подальшого масштабування системи шляхом підключення нових серверів або клієнтських вузлів.

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

Високопродуктивні обчислення (HPC) є сучасною технологічною основою для розв'язання складних наукових і інженерних задач. Варто зазначити, що на сьогодні відсутні повністю ідентичні програмні аналоги платформи BOINC у класичному вигляді добровільних розподілених обчислень, проте існує ряд фреймворків і систем, на базі яких можливо реалізувати подібні функціональні рішення. До таких належать платформи для розподіленої обробки задач, управління чергами обчислень та оркестрації ресурсів. HPC-системи базуються на використанні паралельних обчислень, де велика кількість обчислювальних елементів працює одночасно над однією задачею. Основною метою HPC є максимізація продуктивності та мінімізація часу отримання результату.

Обчислювальні вузли працюють паралельно, обробляючи різні частини задачі, що дозволяє значно прискорити обчислення. HPC-системи часто організовані у вигляді кластерів - груп серверів, об'єднаних у єдину систему

Розподілені обчислення - це підхід, при якому обчислювальні задачі виконуються на кількох незалежних комп'ютерах, які взаємодіють через мережу. У таких системах задача ділиться на незалежні підзадачі, які виконуються паралельно. Це особливо ефективно для задач типу *embarrassingly parallel* (наприклад, чисельне моделювання, Monte Carlo, обробка великих даних).

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		12

Аналіз сучасних програмних рішень для розподіленої обробки задач

Ray - це сучасний open-source фреймворк для розподілених обчислень, орієнтований на високопродуктивну обробку задач у Python-екосистемі. Його головна мета полягає у спрощенні створення масштабованих застосунків, які можуть ефективно працювати як на одному комп'ютері, так і в межах великого кластеру серверів або хмарної інфраструктури.

Ray реалізує універсальну модель розподілених обчислень, яка поєднує паралельне виконання задач, управління станом, розподілену пам'ять та автоматичне масштабування.

Розробник працює з абстракціями, які виглядають як звичайні функції та об'єкти, але виконуються у розподіленому середовищі. Це дозволяє приховати складність мережевої взаємодії, синхронізації та планування задач.

Система здатна автоматично масштабувати виконання задач від одного вузла до тисяч обчислювальних нод. Це досягається завдяки динамічному розподілу задач між ресурсами кластера.

Ray підтримує механізми відновлення після збоїв, включаючи повторне виконання задач і відновлення обчислювального стану. Це критично для довготривалих обчислень і production-середовищ.

Центральний компонент, який відповідає за управління кластером, зберігання метаданих та координацію розподілу задач. Робочі вузли виконують безпосередні обчислення. Кожен вузол може одночасно виконувати кілька задач залежно від доступних ресурсів.

Ray є сучасною універсальною платформою для побудови розподілених систем, яка поєднує простоту Python із можливостями масштабованих кластерних обчислень. Завдяки своїй архітектурі та екосистемі він активно використовується у сферах машинного навчання, обробки даних та високонавантажених серверних застосунків.

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		13

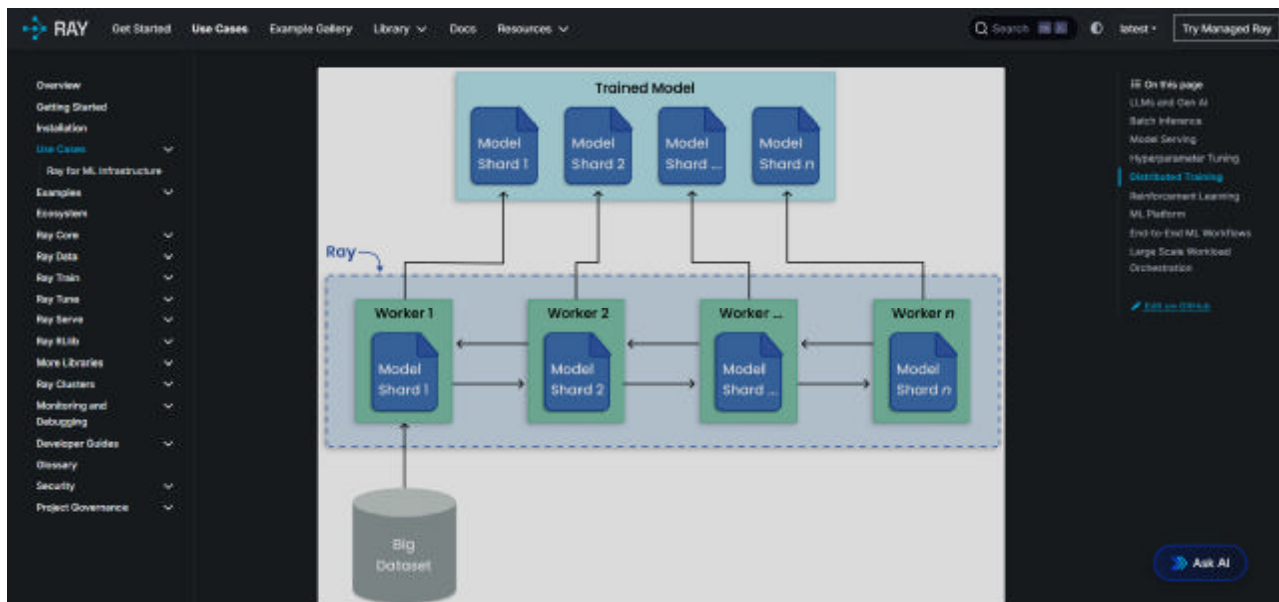


Рисунок 2.1 - Ray фреймворк для розподілених обчислень

Celery - фреймворк для розподіленого виконання задач у Python, який призначений для асинхронної обробки операцій поза основним потоком виконання застосунку. Його основна ідея полягає у винесенні тривалих або ресурсомістких задач у фонові процеси, що дозволяє основній системі залишатися швидкою та відзивною навіть під високим навантаженням. Celery широко застосовується у веб-додатках, мікросервісних архітектурах та системах обробки даних, де необхідна ефективна організація фонових обчислень.

Архітектура Celery базується на моделі взаємодії між компонентами через систему повідомлень. Основними елементами є застосунок, який створює задачі, брокер повідомлень, що відповідає за їх доставку, та робочі процеси, які виконують ці задачі. Такий підхід реалізує класичну модель виробника і споживача, де задачі не виконуються одразу, а передаються в чергу для подальшої обробки. Це дозволяє досягати високого рівня масштабованості та розподіленості системи.

Ключову роль у роботі Celery відіграє message broker, який забезпечує надійну передачу повідомлень між компонентами системи. У якості брокера можуть використовуватися різні технології, зокрема Redis або RabbitMQ, які

відповідають за збереження задач до моменту їх виконання та гарантують доставку навіть у випадку тимчасових збоїв. Завдяки цьому Celery може працювати в умовах нестабільної мережі та забезпечувати цілісність обробки задач.

Виконання задач здійснюється worker-процесами, які можуть запускатися на різних серверах або контейнерах. Це дозволяє горизонтально масштабувати систему шляхом додавання нових виконавчих вузлів. Кожен worker незалежно отримує задачі з черги та виконує їх, що забезпечує паралельність і ефективне використання обчислювальних ресурсів. Така модель особливо корисна у високонавантажених системах, де необхідна обробка великої кількості запитів одночасно.

Celery також підтримує механізми збереження результатів виконання задач. Результати можуть зберігатися у різних сховищах, включаючи бази даних або кеш-системи, що дозволяє отримувати їх пізніше або використовувати в інших частинах системи. Це робить фреймворк зручним для побудови складних асинхронних процесів, де результат однієї задачі може впливати на подальші обчислення.

Важливою особливістю Celery є підтримка планування задач. Система дозволяє виконувати операції як у відкладеному режимі, так і на регулярній основі, що імітує функціональність cron-систем. Це дає можливість автоматизувати рутинні процеси, такі як періодична обробка даних, відправка повідомлень або виконання системних перевірок.

Фреймворк також відзначається високим рівнем гнучкості у питаннях паралелізму. Він може використовувати багато процесну або багатопоточну модель виконання, а також асинхронні підходи, що дозволяє адаптувати його під різні типи навантажень. Це робить Celery універсальним інструментом як для CPU-інтенсивних, так і для I/O-інтенсивних задач.



Рисунок 2.2 - CELERY фреймворк для розподіленого виконання задач

HTCondor - відкрита система керування обчислювальними навантаженнями, яка призначена для організації високопродуктивних обчислень у середовищах з великою кількістю незалежних задач. Вона використовується для ефективного розподілу обчислювальних робіт між різними обчислювальними ресурсами, такими як кластери, ґрид-системи або хмарні платформи. Основна мета HTCondor полягає у максимальному використанні доступних обчислювальних ресурсів, зокрема тих, які простоюють, що робить систему особливо ефективною для задач високої пропускну здатності.

Система HTCondor була розроблена в Університеті Вісконсин-Медісон і з моменту своєї появи стала одним із ключових інструментів у сфері наукових та академічних обчислень. Вона базується на концепції розподіленого планування задач, де центральний компонент координує розподіл робіт між доступними обчислювальними вузлами. Такий підхід дозволяє ефективно використовувати

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		16

середовища, де різні машини можуть мати різні характеристики продуктивності та доступності.

Архітектура HTCondor побудована навколо моделі централізованого керування з кількома спеціалізованими компонентами. Центральна частина системи відповідає за узгодження задач і ресурсів, визначаючи, які обчислювальні вузли можуть виконувати конкретні завдання. Завдання надходять до системи через механізм черги, після чого розподіляються на виконавчі вузли відповідно до доступності ресурсів та заданих умов виконання.

Однією з ключових особливостей HTCondor є використання моделі обчислення, при якій система активно використовує вільні або недовантажені ресурси. Це означає, що обчислювальні вузли можуть виконувати задачі лише тоді, коли вони не зайняті локальними процесами, що дозволяє значно підвищити загальну ефективність використання інфраструктури без необхідності виділення спеціалізованих ресурсів.

HTCondor також підтримує складні політики планування задач, які дозволяють гнучко визначати пріоритети, вимоги до ресурсів та умови виконання. Завдяки цьому система може ефективно обробляти великі черги незалежних задач, оптимізуючи їх виконання відповідно до доступних ресурсів. Важливою особливістю є також підтримка механізмів контрольних точок, які дозволяють відновлювати виконання задач після збоїв або переривання роботи вузлів.

Система широко використовується у наукових дослідженнях, зокрема у проектах з високою інтенсивністю обчислень, таких як моделювання, симуляції та аналіз великих обсягів даних. Вона активно застосовується у міжнародних наукових колабораціях, включаючи експерименти у сфері фізики, де необхідно обробляти величезні масиви обчислювальних задач.

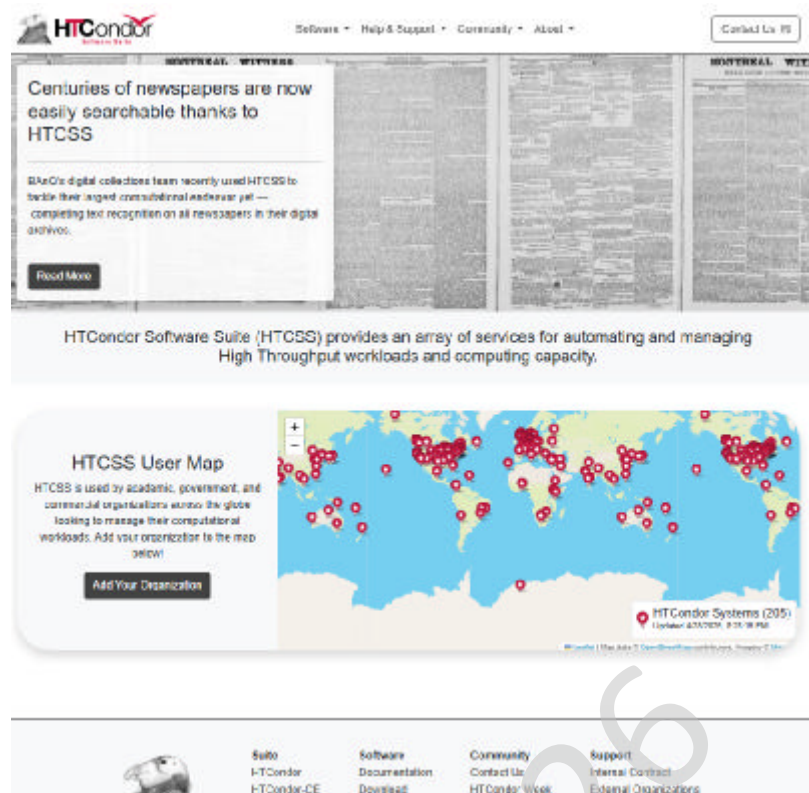


Рисунок 2.3 - HTCCondor система керування обчисленнями

Порівняльна характеристика та переваги BOINC

У порівнянні з такими системами, як Ray, Celery та HTCCondor, платформа BOINC представляє принципово інший підхід до організації розподілених обчислень, орієнтований на використання глобально розподілених, нецентралізованих ресурсів. Якщо Ray та Celery здебільшого застосовуються у межах керованої інфраструктури, а HTCCondor - у кластерних і ґрид-середовищах, то BOINC дозволяє залучати необмежену кількість добровільних вузлів, що функціонують у відкритому мережевому середовищі. Це формує іншу модель масштабування, де ключовим фактором є не контроль ресурсів, а їх масовість і доступність.

Порівняльний аналіз показує, що кожна з систем орієнтована на різні класи задач і сценарії використання. Ray забезпечує низьку затримку та гнучку модель виконання, що робить його придатним для інтерактивних обчислень і

машинного навчання. Celery спеціалізується на асинхронній обробці задач у веб-системах, де важлива інтеграція з backend-архітектурою. HTCondor орієнтований на високопродуктивні обчислення у контрольованих середовищах із чіткою політикою планування. У свою чергу BOINC ефективний у задачах, які допускають значну затримку, але можуть бути виконані паралельно у великій кількості незалежних екземплярів.

На відміну від вищезгаданих систем, BOINC забезпечує:

- Глобальну масштабованість обчислень, яка досягається шляхом об'єднання великої кількості персональних комп'ютерів, підключених через Інтернет, що дозволяє формувати обчислювальні потужності рівня суперкомп'ютерів без централізованої інфраструктури.

- Опортуністичне використання ресурсів, при якому обчислення виконуються лише у періоди простою пристроїв, що забезпечує ефективне використання доступних обчислювальних потужностей без впливу на користувачів.

- Підвищену відмовостійкість за рахунок дублювання задач та перевірки результатів, що є необхідним в умовах відсутності повного контролю над обчислювальними вузлами.

- Економічну ефективність, оскільки система не потребує значних капітальних витрат на створення та підтримку обчислювальної інфраструктури, використовуючи вже наявні ресурси користувачів.

- Орієнтацію на задачі з високою пропускнуою здатністю, які можуть бути розбиті на незалежні підзадачі, що виконуються паралельно без необхідності інтенсивної взаємодії між вузлами.

- Глобальну доступність та гнучкість розгортання, що дозволяє використовувати систему в міжнародних наукових проектах із великою кількістю учасників.

Таким чином, BOINC не конкурує безпосередньо з Ray, Celery чи HTCondor, а доповнює їх, пропонуючи альтернативну модель розподілених

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		19

обчислень, що базується на масовому залученні ресурсів. Такий підхід є особливо доцільним для задач наукового характеру, де критичною є загальна обчислювальна потужність, а не мінімальна затримка виконання.

2.2 Обґрунтування вибору засобів для побудови системи

Для реалізації розподіленої обчислювальної системи було обрано платформу BOINC, яка дозволяє ефективно організувати виконання великої кількості незалежних задач у глобально розподіленому середовищі. Основною причиною вибору даного рішення є можливість масштабування обчислень за рахунок використання великої кількості клієнтських вузлів, об'єднаних через мережу Інтернет. Такий підхід дозволяє суттєво підвищити продуктивність системи без необхідності використання дорогих суперкомп'ютерних комплексів.

BOINC підтримує клієнт-серверну архітектуру, у якій центральний сервер виконує функції генерації, розподілу та перевірки задач, а клієнтські вузли забезпечують безпосереднє виконання обчислень. Важливою перевагою платформи є підтримка різних операційних систем та апаратних конфігурацій, що дозволяє використовувати неоднорідні обчислювальні ресурси в межах єдиної інфраструктури.

Однією з причин вибору BOINC є наявність механізмів автоматичної валідації результатів та відмовостійкості. Система здатна повторно розподіляти задачі у випадку помилок або недоступності окремих вузлів, що забезпечує стабільність роботи та достовірність отриманих результатів. Також платформа підтримує механізми контрольних точок, які дозволяють продовжувати виконання задач після переривання без втрати проміжних результатів.

Для забезпечення стабільної роботи серверної частини було використано операційну систему Linux, яка характеризується високою надійністю, ефективним використанням ресурсів та широкими можливостями адміністрування.

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		20

Платформа розподілених обчислень BOINC

У якості основного програмного середовища для організації високопродуктивних розподілених обчислень було обрано платформу BOINC (Berkeley Open Infrastructure for Network Computing). Дана платформа є однією з найбільш відомих та поширених систем volunteer computing і використовується для реалізації масштабних наукових проектів, пов'язаних із математичним моделюванням, аналізом великих обсягів даних, фізичними симуляціями та біоінформатикою. Вибір BOINC обумовлений її відкритою архітектурою, підтримкою великої кількості клієнтських вузлів та наявністю механізмів автоматизованого розподілу обчислювального навантаження.

Платформа BOINC забезпечує повний цикл роботи із обчислювальними задачами. Серверна частина системи виконує генерацію work units, їх планування, розподіл між вузлами, приймання результатів, валідацію обчислень та збереження статистичної інформації у базі даних. Клієнтська частина BOINC встановлюється на обчислювальних вузлах та працює у фоновому режимі, періодично звертаючись до серверу для отримання нових задач і передачі результатів виконання.

Архітектура системи побудована за класичною клієнт-серверною моделлю. Центральний сервер виступає координатором усієї інфраструктури та забезпечує керування обчислювальним процесом. На сервері розміщується база даних, система планування задач, механізми аутентифікації клієнтів, веб-інтерфейс адміністрування та модулі валідації результатів. Клієнтські вузли можуть бути представлені як персональними комп'ютерами, так і серверними системами, що використовують власні апаратні ресурси для виконання обчислень.

Однією з ключових переваг BOINC є можливість ефективного використання ресурсів великої кількості незалежних пристроїв, об'єднаних через мережу Інтернет. Система дозволяє масштабувати обчислювальну інфраструктуру практично без обмежень, додаючи нові вузли без необхідності

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		21

суттєвої зміни архітектури. Це забезпечує можливість створення розподілених НРС-середовищ без використання дорогих суперкомп'ютерних комплексів.

Особливістю BOINC є підтримка асинхронної моделі виконання задач. Клієнт отримує work unit, виконує обчислення локально та після завершення передає результати назад на сервер. У процесі виконання не вимагається постійне мережеве з'єднання із сервером, що дозволяє використовувати систему навіть у нестабільних мережевих умовах. Крім того, клієнт може автоматично призупиняти виконання задач при підвищеному навантаженні на систему користувача або при роботі від акумуляторного живлення.

Важливою складовою архітектури BOINC є механізм дублювання задач та валідації результатів. Оскільки система функціонує у середовищі із потенційно ненадійними або неперевіреними вузлами, сервер може надсилати одну й ту саму задачу декільком клієнтам незалежно один від одного. Після завершення обчислень результати порівнюються спеціальним модулем validator. Якщо результати збігаються у межах допустимої похибки, задача вважається коректно виконаною.

У випадку невідповідності результатів система може автоматично повторно відправити задачу іншим вузлам для додаткової перевірки. Це дозволяє забезпечити стабільність роботи навіть при великій кількості клієнтів із різними апаратними характеристиками та рівнем надійності.

BOINC також підтримує механізми пріоритезації задач та адаптивного планування навантаження. Сервер аналізує продуктивність клієнтських вузлів, швидкість виконання попередніх задач, доступний обсяг пам'яті, кількість процесорних ядер та інші параметри системи. На основі цих даних виконується оптимальний розподіл work units між вузлами, що дозволяє підвищити ефективність використання ресурсів.

Додатковою перевагою BOINC є підтримка віртуалізованого виконання задач. Частина наукових проектів використовує Oracle VM VirtualBox для запуску спеціалізованих віртуальних машин із попередньо налаштованим

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		22

програмним середовищем. Це дозволяє забезпечити однакові умови виконання задач незалежно від конфігурації хост-системи та спрощує запуск складного наукового програмного забезпечення.

Для взаємодії між клієнтом та сервером BOINC використовуються мережеві протоколи HTTP та RPC, а службова інформація передається у форматі XML. Такий підхід забезпечує кросплатформеність системи та спрощує інтеграцію із зовнішніми сервісами та системами моніторингу.

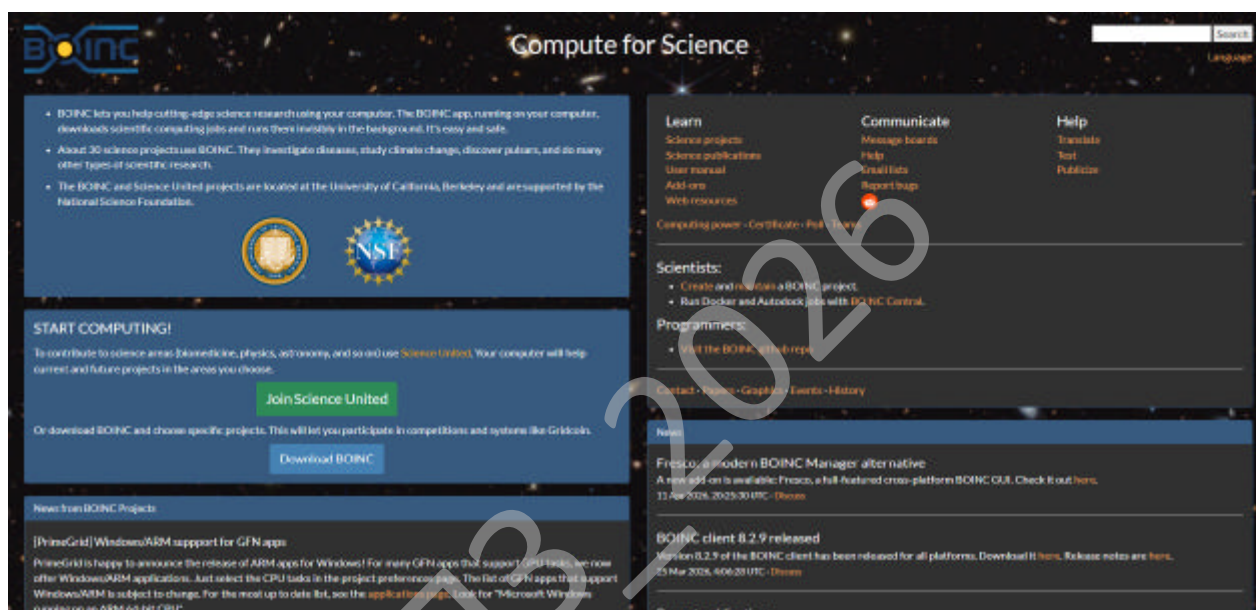


Рисунок 2.4 - Веб сайт BOINC

Одним із ключових наукових центрів сучасності, діяльність якого безпосередньо пов'язана з розвитком високопродуктивних та розподілених обчислювальних систем, є CERN. CERN (Conseil Européen pour la Recherche Nucléaire) - Європейська організація з ядерних досліджень, що розташована поблизу Женеви на кордоні Швейцарії та Франції. Організація була заснована у 1954 році з метою проведення фундаментальних досліджень у галузі фізики елементарних частинок, а також розвитку міжнародної наукової співпраці.

На сьогодні CERN є найбільшим у світі науково-дослідним центром у сфері фізики високих енергій. У лабораторіях CERN функціонують складні

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		23

прискорювальні комплекси, детектори та обчислювальні інфраструктури, які використовуються для дослідження фундаментальних властивостей матерії, енергії, простору та часу. Найбільш відомим проектом організації є Великий адронний колайдер (Large Hadron Collider, LHC) - найбільший та найпотужніший прискорювач частинок у світі.

Основною задачею CERN є проведення експериментів зі зіткнення елементарних частинок на надвисоких енергіях. У процесі таких експериментів утворюються величезні обсяги наукових даних, які потребують складної математичної обробки, фільтрації, аналізу та довготривалого зберігання. Щороку експерименти CERN генерують десятки петабайт інформації, що створює надзвичайно високі вимоги до обчислювальних ресурсів та інфраструктури передачі даних.

Для вирішення проблеми обробки настільки великих обсягів інформації в CERN було створено глобальну розподілену обчислювальну систему Worldwide LHC Computing Grid (WLCG). Дана інфраструктура об'єднує сотні дата-центрів, університетів та наукових установ у різних країнах світу в єдину обчислювальну мережу. Архітектура WLCG побудована за багаторівневим принципом та включає центральні обчислювальні вузли CERN, регіональні дата-центри та локальні кластери, що забезпечують паралельну обробку наукових даних.

Особливістю CERN є активне використання технологій Grid Computing та Distributed Computing. На відміну від класичних суперкомп'ютерів, де всі ресурси фізично зосереджені в одному центрі обробки даних, розподілені системи CERN дозволяють використовувати обчислювальні потужності, розміщені у різних країнах та організаціях. Такий підхід забезпечує високу масштабованість, відмовостійкість та можливість ефективного розподілу навантаження між великою кількістю вузлів.

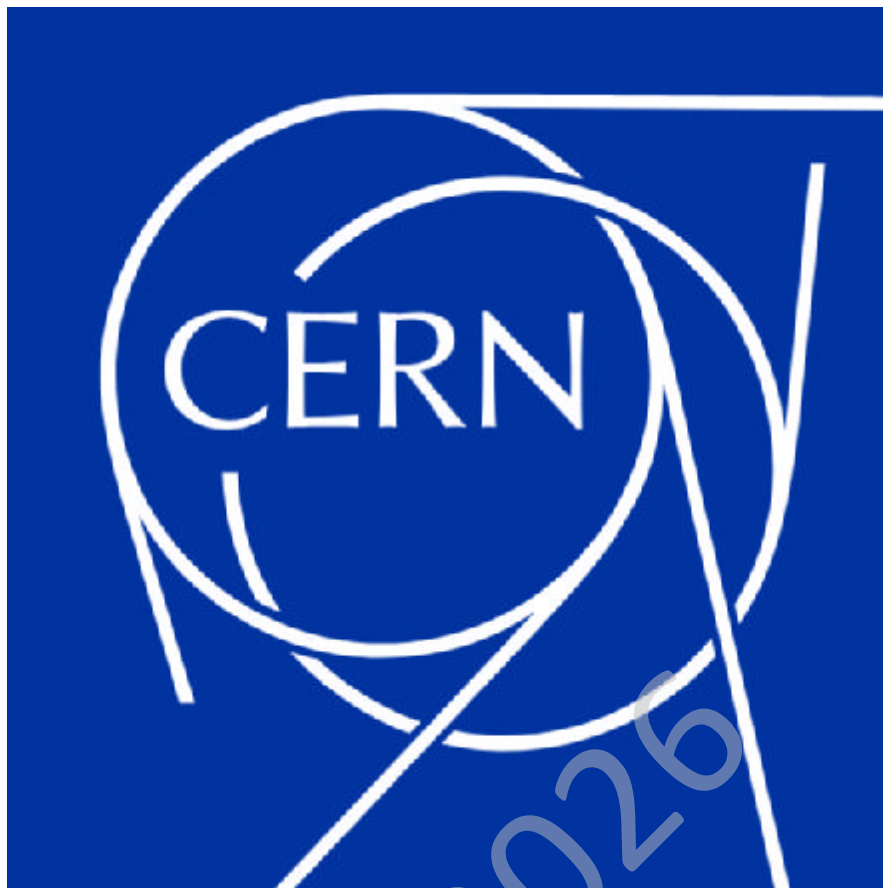


Рисунок 2.5 - Європейська організація з ядерних досліджень

Одним із найбільш відомих та технологічно складних проектів у сфері добровільних розподілених обчислень є LHC@home, створений за підтримки CERN. Даний проект призначений для виконання наукових обчислень, пов'язаних із дослідженнями Великого адронного колайдера (Large Hadron Collider, LHC), який є найбільшим у світі прискорювачем елементарних частинок. Основною метою проекту є використання розподілених обчислювальних ресурсів користувачів для виконання задач моделювання фізичних процесів високих енергій, аналізу поведінки пучків частинок та дослідження параметрів роботи прискорювального комплексу.

Великий адронний колайдер являє собою надскладну наукову установку, що генерує величезні обсяги експериментальних даних. У процесі роботи детектори формують потоки інформації, які потребують подальшої математичної

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		25

обробки та моделювання. Частина обчислень пов'язана із безпосереднім аналізом результатів експериментів, а інша - із симуляцією фізичних процесів, що дозволяє прогнозувати поведінку системи та перевіряти коректність теоретичних моделей. Такі задачі є надзвичайно ресурсоемними та вимагають використання високопродуктивних обчислювальних систем.

Традиційне використання централізованих суперкомп'ютерів для подібних задач пов'язане із високими фінансовими витратами на придбання обладнання, електроживлення, охолодження та обслуговування інфраструктури. Саме тому в межах CERN було реалізовано підхід використання добровільних розподілених обчислень, при якому тисячі користувачів по всьому світу надають власні обчислювальні ресурси для виконання частини наукових задач. Для реалізації такого підходу була обрана платформа BOINC, яка забезпечує механізми автоматичного розподілу задач, передачі вхідних даних, контролю виконання та валідації результатів.

Архітектурно LHC@home побудований за багаторівневою клієнт-серверною моделлю. Центральна серверна інфраструктура CERN виконує формування задач, розподіл work units між клієнтськими вузлами, збір результатів та їх перевірку. Клієнтські вузли представлені персональними комп'ютерами користувачів, на яких встановлено BOINC-клієнт. Після підключення до проекту клієнт автоматично отримує задачі, необхідні файли та параметри виконання, після чого запускає обчислення у фоновому режимі.

Однією з ключових особливостей LHC@home є використання віртуалізації для виконання частини задач. Значна кількість наукових застосунків CERN має складне програмне оточення та специфічні вимоги до операційної системи, бібліотек і залежностей. Для забезпечення однакових умов виконання незалежно від типу клієнтської системи використовується Oracle VM VirtualBox.

У такому випадку BOINC-клієнт автоматично завантажує образ віртуальної машини, запускає її локально та виконує обчислення всередині ізольованого середовища. Це дозволяє забезпечити сумісність між різними

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		26

платформами та мінімізувати ризик помилок, пов'язаних із відмінностями програмного забезпечення на вузлах користувачів.



Рисунок 2.6 - Large Hadron Collider

Також одним із наукових експериментів у галузі фізики високих енергій є ATLAS Experiment, який функціонує у складі дослідницької інфраструктури CERN. Назва ATLAS є скороченням від A Toroidal LHC ApparatuS. Експеримент створений для дослідження фундаментальних властивостей елементарних частинок та перевірки сучасних фізичних теорій, включаючи Стандартну модель фізики частинок, теорії суперсиметрії, дослідження темної матерії та пошук нових фізичних явищ.

ATLAS є одним із чотирьох основних детекторів Великого адронного колайдера (LHC). Детектор розташований у спеціальному підземному комплексі на глибині близько 100 метрів та являє собою складну багаторівневу систему сенсорів, магнітів, електронних модулів та обчислювальних компонентів. Загальні розміри установки становлять приблизно 46 метрів у довжину та 25 метрів у висоту, а маса перевищує декілька тисяч тонн. Такі масштаби роблять ATLAS одним із найбільших наукових приладів у світі.

Основний принцип роботи експерименту полягає у реєстрації результатів зіткнення протонних пучків, які розганяються у Великому адронному колайдері до надвисоких енергій. У момент зіткнення частинок утворюється велика кількість вторинних елементарних частинок, траєкторії та характеристики яких

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		27

фіксуються детекторами. Отримані сигнали аналізуються з метою виявлення закономірностей, що можуть свідчити про існування нових фізичних процесів або частинок.

Особливістю експерименту ATLAS є надзвичайно великий обсяг даних, який формується у процесі роботи детектора. Частота зіткнень частинок може досягати десятків мільйонів подій за секунду. Усі ці події генерують величезний потік інформації, який неможливо повністю зберігати або обробляти у реальному часі без використання спеціалізованих високопродуктивних обчислювальних систем.

Для попередньої обробки даних в ATLAS використовується багаторівнева система тригерів (Trigger System). Її основне завдання полягає у швидкому відборі найбільш важливих подій для подальшого аналізу. На першому рівні виконується апаратна фільтрація даних із використанням спеціалізованих електронних модулів та FPGA-процесорів. Далі програмні алгоритми здійснюють детальніший аналіз подій та визначають, чи потрібно зберігати конкретну інформацію для подальших досліджень.

Після первинної обробки дані передаються до глобальної розподіленої обчислювальної інфраструктури. Для цього використовується система Worldwide LHC Computing Grid (WLCG), яка забезпечує зберігання, реплікацію та обробку інформації у дата-центрах різних країн світу. Архітектура системи побудована за багаторівневим принципом Tier-0, Tier-1, Tier-2 та Tier-3.

Рівень Tier-0 розташований безпосередньо у CERN та виконує первинне збереження і початкову реконструкцію даних. Після цього інформація автоматично копіюється до великих регіональних дата-центрів рівня Tier-1, які забезпечують резервне зберігання та подальшу обробку. Центри Tier-2 використовуються для виконання аналітичних задач, симуляцій та тестових обчислень, а вузли Tier-3 зазвичай представлені локальними університетськими або дослідницькими кластерами.

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		28

Для ефективного використання обчислювальних ресурсів у проекті ATLAS застосовуються технології Grid Computing та Distributed Computing. Обчислювальні задачі автоматично розподіляються між тисячами серверів та обчислювальних вузлів залежно від доступних ресурсів, продуктивності та пріоритетів виконання. Такий підхід дозволяє виконувати складні математичні розрахунки паралельно на великій кількості систем.

Особливу роль у роботі ATLAS відіграють симуляційні задачі. Перед проведенням фізичних експериментів науковці виконують комп'ютерне моделювання поведінки частинок у прискорювачі та детекторі. Для цього використовуються спеціалізовані програмні пакети, які реалізують складні математичні моделі фізичних процесів. Такі симуляції потребують значних обчислювальних ресурсів та можуть виконуватись протягом тривалого часу.

У межах експерименту активно застосовуються технології віртуалізації та контейнеризації. Це дозволяє створювати стандартизовані середовища виконання для наукового програмного забезпечення незалежно від апаратної конфігурації конкретного вузла. Завдяки цьому забезпечується сумісність програмних компонентів та спрощується розгортання обчислювальної інфраструктури.

Для моніторингу стану системи використовуються спеціалізовані інструменти збору телеметрії, аналізу продуктивності та контролю навантаження. Контролюються параметри використання процесорних ресурсів, обсяги оперативної пам'яті, швидкість передачі даних, статус виконання задач та працездатність серверного обладнання. Системи моніторингу дозволяють оперативно виявляти помилки, перевантаження або відмови вузлів.

Одним із важливих напрямів розвитку обчислювальної інфраструктури ATLAS є інтеграція хмарних технологій та добровільних розподілених обчислень. Частина задач може виконуватись на зовнішніх обчислювальних ресурсах, включаючи університетські кластери, хмарні дата-центри та персональні комп'ютери учасників проекту через платформу LHC@home. Це

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		29

дозволяє додатково масштабувати обчислювальні потужності та зменшувати навантаження на центральну інфраструктуру CERN.

Результати експерименту ATLAS мають фундаментальне значення для сучасної науки. Саме завдяки цьому експерименту у 2012 році було підтверджено існування бозона Гігса - елементарної частинки, передбаченої Стандартною моделлю фізики. Це відкриття стало одним із найважливіших наукових досягнень XXI століття та підтвердило ефективність використання масштабних розподілених обчислювальних систем для вирішення складних наукових задач.

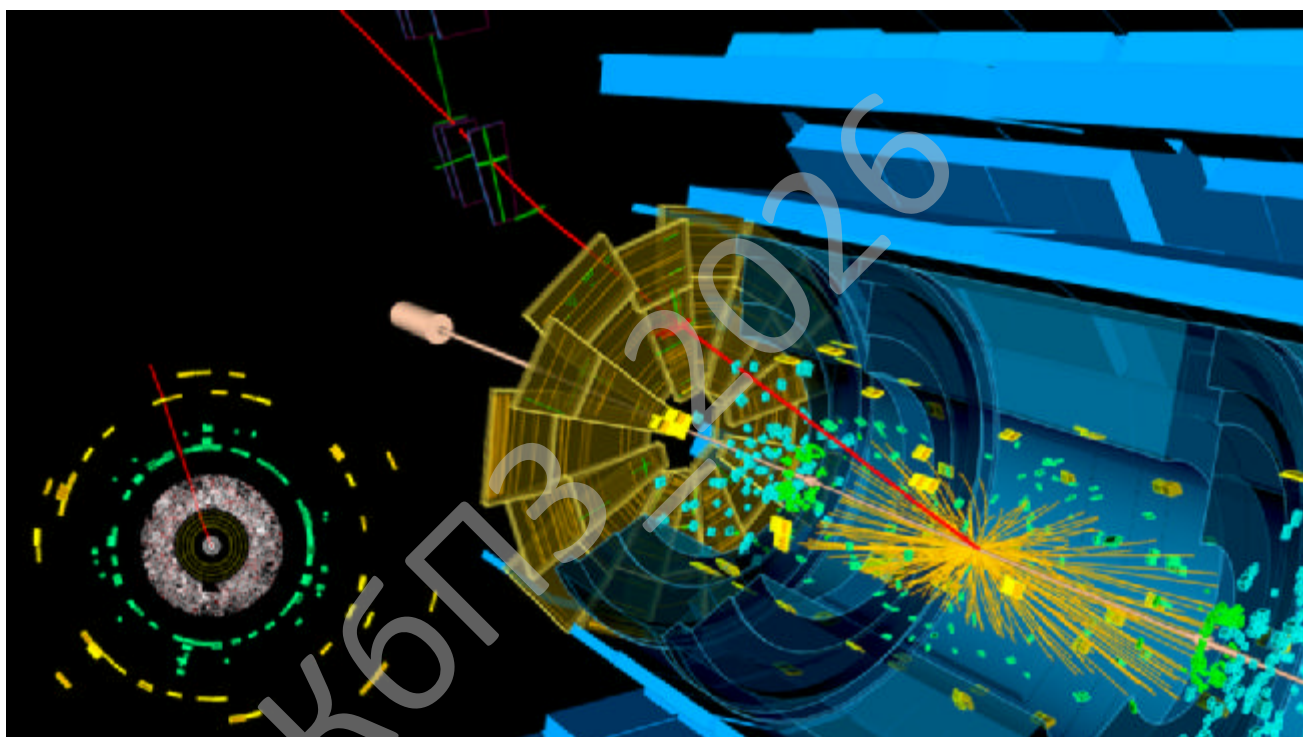


Рисунок 2.7 - A Toroidal LHC ApparatuS

Одним із ключових програмних компонентів, що використовується у наукових обчисленнях в межах інфраструктури CERN, є система SixTrack. Даний програмний комплекс призначений для моделювання траєкторій руху заряджених частинок у прискорювальних системах та використовується для дослідження стабільності пучків у Великому адронному колайдері (LHC) та інших прискорювачах.

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Основним призначенням SixTrack є виконання довготривалого чисельного моделювання динаміки частинок у магнітних структурах прискорювачів. Програма дозволяє аналізувати поведінку протонних пучків під впливом складних електромагнітних полів, оцінювати стабільність орбіт частинок, виявляти потенційні втрати пучка та прогнозувати ефективність роботи прискорювальних комплексів.

У сучасних прискорювачах елементарних частинок точність розрахунків має критичне значення, оскільки навіть незначні похибки в конфігурації магнітних систем можуть призвести до нестабільності пучка або аварійних ситуацій. Саме тому програмні комплекси типу SixTrack використовуються для проведення попередніх симуляцій та перевірки різних режимів роботи прискорювального обладнання ще до запуску реальних фізичних експериментів.

Алгоритми SixTrack базуються на математичних моделях нелінійної динаміки заряджених частинок. У процесі роботи програма виконує чисельне інтегрування рівнянь руху частинок через велику кількість елементів прискорювального кільця, включаючи дипольні магніти, квадрупольні системи, коригувальні магніти та інші компоненти магнітної оптики.

Особливістю SixTrack є необхідність виконання надзвичайно великої кількості повторюваних обчислень. Для отримання достовірних результатів необхідно моделювати рух великої кількості частинок протягом мільйонів або навіть мільярдів обертів у прискорювачі. Це створює значне навантаження на обчислювальні ресурси та робить задачу ідеально придатною для використання технологій високопродуктивних розподілених обчислень.

Програмний комплекс SixTrack активно використовується у межах проекту LHC@home, який реалізовано на платформі BOINC. У цьому випадку великі обчислювальні задачі автоматично розбиваються на окремі незалежні підзадачі (work units), які можуть виконуватись паралельно на великій кількості клієнтських вузлів.

З наукової точки зору SixTrack є важливим інструментом для аналізу довготривалої стабільності пучків частинок. Результати симуляцій використовуються для оптимізації конфігурацій прискорювача, оцінки ризиків втрати частинок та прогнозування ефективності майбутніх експериментів. Отримані дані допомагають дослідникам покращувати параметри роботи LHC та забезпечувати безпечне функціонування прискорювальних систем.

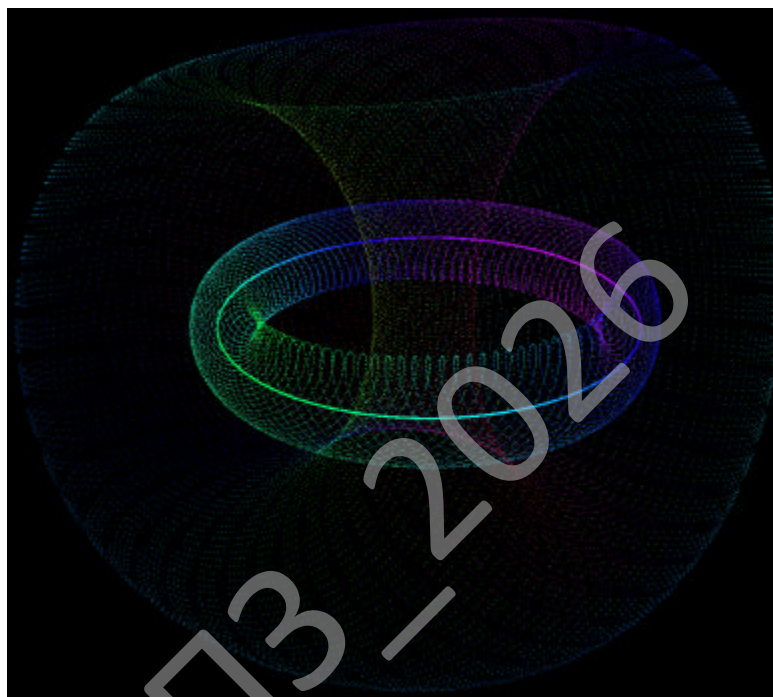


Рисунок 2.8 - Приклад SixTrack

Climateprediction.net є ще одним із найвідоміших міжнародних наукових проєктів у сфері розподілених обчислень, який спеціалізується на моделюванні кліматичних процесів та прогнозуванні глобальних змін клімату. проєкт був створений за участю науковців Оксфордського університету з метою використання обчислювальних ресурсів добровольців для проведення масштабних кліматичних експериментів, які потребують надзвичайно великої кількості обчислень. Основною ідеєю проєкту є залучення персональних

комп'ютерів користувачів у всьому світі до виконання складних математичних моделей атмосфери, океанів та кліматичних систем Землі.

На відміну від традиційних центрів високопродуктивних обчислень, де використовується обмежена кількість суперкомп'ютерів, Climateprediction.net реалізує концепцію volunteer computing, у межах якої обчислювальні ресурси тисяч незалежних вузлів об'єднуються в єдине розподілене середовище. Такий підхід дозволяє виконувати величезну кількість незалежних кліматичних симуляцій паралельно, що є особливо важливим для дослідження складних природних процесів із великою кількістю змінних параметрів.

Основним напрямком роботи проекту є створення ансамблевих кліматичних моделей. У межах такого підходу одна й та сама модель запускається багаторазово із незначними змінами початкових параметрів. Це дозволяє оцінити можливі сценарії розвитку кліматичних процесів та визначити ймовірність виникнення певних природних явищ. Подібні розрахунки є надзвичайно ресурсоемними, оскільки навіть одна повноцінна кліматична модель може виконуватись протягом декількох днів або тижнів.

Особливістю Climateprediction.net є орієнтація на довготривалі моделювання. На відміну від короткострокових погодних прогнозів, кліматичні симуляції можуть охоплювати періоди від десятків до сотень років. Під час виконання таких задач моделюються процеси циркуляції атмосфери, зміни температури поверхні океану, концентрації парникових газів, переміщення повітряних мас, процеси утворення опадів та інші фактори, що впливають на глобальний клімат.

Для забезпечення точності результатів використовуються складні математичні рівняння чисельного моделювання. Основою подібних моделей є системи диференціальних рівнянь, які описують фізичні процеси теплообміну, динаміки рідин та газів, радіаційного балансу та взаємодії між різними шарами атмосфери. Обчислення виконуються дискретно у часових кроках, що потребує

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		33

значної кількості операцій із плаваючою комою та великого обсягу оперативної пам'яті.

Важливим аспектом роботи проекту є дослідження екстремальних погодних явищ. За допомогою Climateprediction.net проводяться експерименти, спрямовані на аналіз причин виникнення ураганів, посух, аномальної спеки, сильних опадів та інших небезпечних природних процесів. Завдяки великій кількості паралельних симуляцій дослідники можуть оцінювати вплив окремих факторів на ймовірність виникнення певних подій та прогнозувати можливі наслідки глобальних кліматичних змін.

Types of Experiments:

With CPDN, scientists can conduct diverse experiments to explore weather and climate-related questions. These experiments may involve:

- Running climate models to simulate future climate scenarios and assess their impacts on various aspects of the environment and society.
- Analyzing historical climate data to identify trends, patterns, and drivers of climate variability and change.
- Integrating climate models with other Earth system models to study feedback loops and interactions between different components of the Earth's climate system.



Рисунок 2.9 - Climateprediction.net

OpenIFS є спеціалізованою версією інтегрованої системи прогнозування погоди ECMWF Integrated Forecasting System (IFS), адаптованою для використання у наукових дослідженнях та освітніх проектах. Дана система розроблена Європейським центром середньострокових прогнозів погоди (ECMWF) і використовується для чисельного моделювання атмосферних процесів, аналізу погодних явищ та дослідження динаміки кліматичних систем.

OpenIFS є спрощеним і більш доступним варіантом повної системи IFS, яка використовується одним із провідних світових центрів прогнозування погоди. Незважаючи на адаптацію для наукових досліджень, OpenIFS зберігає основні

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докum.	Підпис	Дата		34

математичні моделі та алгоритми, що застосовуються у професійних метеорологічних обчисленнях. Це дозволяє використовувати систему для моделювання атмосферних процесів із високою точністю.

Основою OpenIFS є складний набір фізичних та математичних моделей, які описують процеси руху атмосфери, теплообміну, утворення хмарності, випадіння опадів, турбулентності та взаємодії між атмосферою й поверхнею Землі. Для цього використовуються системи рівнянь гідродинаміки та термодинаміки, які чисельно розв'язуються на багатовимірній просторовій сітці.

Особливістю OpenIFS є підтримка ансамблевого прогнозування. У межах такого підходу формується велика кількість прогнозів із незначними варіаціями початкових умов. Це дозволяє оцінювати ступінь невизначеності прогнозу та визначати ймовірність розвитку певних погодних сценаріїв. Саме ансамблеві обчислення є одним із найбільш ресурсоемних типів задач у сучасній метеорології.

Для виконання великомасштабних ансамблевих симуляцій OpenIFS інтегрується з інфраструктурою Climateprediction.net у межах проекту OpenIFS@Home. У даному випадку модель прогнозування розподіляється між великою кількістю добровольчих обчислювальних вузлів через платформу BOINC. Кожен вузол виконує окремий варіант погодного сценарію, після чого результати передаються до центральної системи для подальшого аналізу.

Одним із важливих напрямків використання OpenIFS є моделювання тропічних циклонів, сильних опадів, атмосферних фронтів та інших небезпечних природних процесів. Велика кількість незалежних симуляцій дозволяє оцінювати чутливість прогнозу до незначних змін параметрів атмосфери та досліджувати межі передбачуваності погодних процесів.

У процесі виконання OpenIFS використовує значні апаратні ресурси, включаючи центральний процесор, оперативну пам'ять та дискову підсистему. Обчислення виконуються у багатопотоковому режимі з активним використанням чисельних методів та операцій із плаваючою комою. Частина задач може

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		35

працювати у віртуалізованому середовищі, що забезпечує сумісність програмного забезпечення незалежно від конфігурації клієнтської системи.

Важливою перевагою OpenIFS є його наукова достовірність. Оскільки система базується на професійній метеорологічній платформі ECMWF, результати моделювання можуть використовуватись не лише у навчальних цілях, але й у реальних наукових дослідженнях. Це робить OpenIFS цінним інструментом для аналізу атмосферних процесів, тестування нових методів прогнозування та дослідження впливу глобальних кліматичних змін.

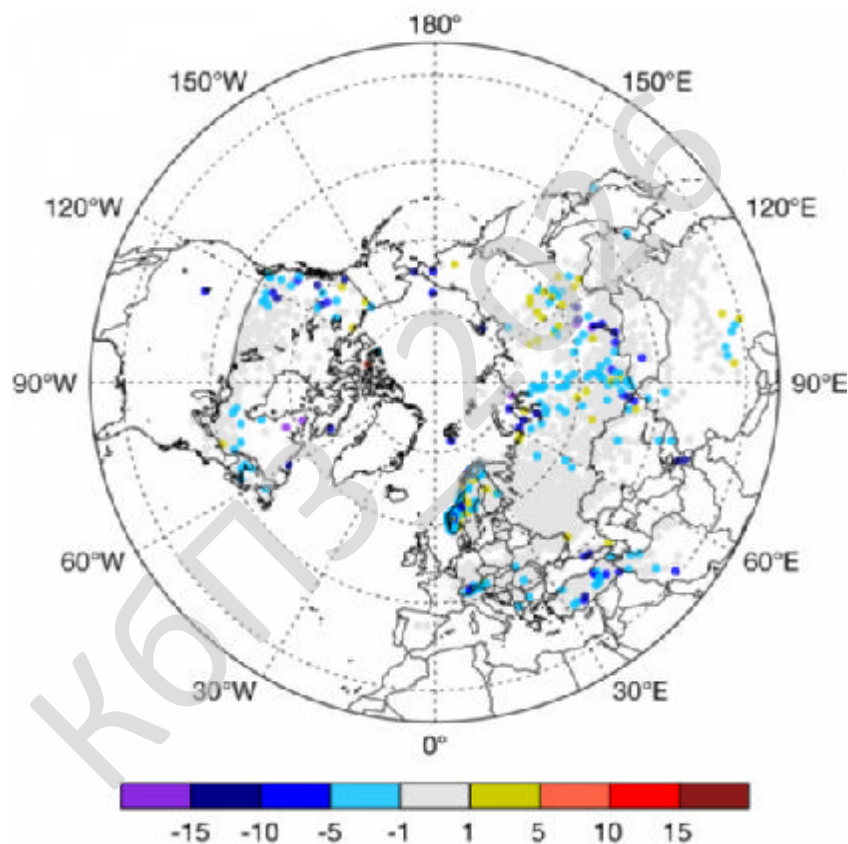


Рисунок 2.10 – Приклад роботи OpenIFS

Архітектура та алгоритми роботи системи

Архітектура платформи BOINC побудована за класичною багаторівневою клієнт-серверною моделлю, у межах якої центральний сервер виконує функції координатора обчислювального середовища, а клієнтські вузли забезпечують

безпосереднє виконання обчислювальних задач. Такий підхід дозволяє організувати розподілену систему високопродуктивних обчислень без необхідності використання спеціалізованих суперкомп'ютерних кластерів, використовуючи при цьому ресурси великої кількості незалежних пристроїв.

Серверна частина системи включає комплекс програмних модулів, відповідальних за формування обчислювальних задач (work units), керування чергами виконання, збереження проміжних та фінальних результатів, а також взаємодію з клієнтськими вузлами через мережеві протоколи TCP/IP та HTTP(S). До складу серверної інфраструктури входять модулі планування (scheduler), генерації задач (work generator), модулі передачі файлів, підсистема перевірки результатів (validator), база даних та веб-інтерфейс адміністрування. Кожен із компонентів виконує окрему функцію у загальному циклі роботи системи, що забезпечує модульність та масштабованість архітектури.

Після формування задач сервер виконує їх декомпозицію на незалежні підзадачі невеликого розміру. Такий підхід дозволяє ефективно розподіляти навантаження між великою кількістю клієнтських вузлів. Кожна задача містить набір вхідних даних, службову інформацію про параметри виконання, вимоги до апаратних ресурсів та контрольні параметри для перевірки коректності результатів. Сформовані work units зберігаються у серверній базі даних та очікують призначення обчислювальним вузлам.

Клієнтська частина BOINC реалізована у вигляді фонових сервісів, який працює на обчислювальному вузлі та періодично взаємодіє із сервером проекту. Після встановлення з'єднання клієнт передає інформацію про характеристики системи, включаючи модель процесора, кількість доступних ядер, обсяг оперативної пам'яті, наявність графічних прискорювачів, параметри операційної системи та поточний рівень навантаження. На основі отриманих даних сервер приймає рішення щодо призначення задач конкретному вузлу.

Алгоритм роботи клієнта передбачає циклічне виконання декількох основних етапів: отримання задач, завантаження вхідних файлів, локальне

планування виконання, запуск обчислювальних процесів, контроль стану задач, формування результатів та передачу обчислених даних назад до серверної частини системи. Планувальник BOINC забезпечує динамічний розподіл ресурсів між задачами з урахуванням пріоритетів, доступності процесорного часу та налаштувань користувача.

Особливістю системи є підтримка декількох режимів виконання обчислень. Частина задач може запускатись безпосередньо на ресурсах хост-системи, використовуючи центральний процесор та оперативну пам'ять вузла. Для задач, які потребують ізольованого середовища або спеціалізованого програмного забезпечення, використовується підсистема віртуалізації на базі Oracle VM VirtualBox. У такому випадку BOINC автоматично створює та запускає віртуальну машину, всередині якої виконується необхідний науковий застосунок. Це дозволяє забезпечити однакове програмне середовище незалежно від типу операційної системи клієнтського вузла.

Алгоритмічна модель системи базується на концепції *embarrassingly parallel computing* - класі задач, які можуть виконуватись незалежно одна від одної без необхідності постійного обміну даними між вузлами. Такий підхід значно спрощує масштабування системи, оскільки збільшення кількості клієнтських вузлів практично лінійно підвищує загальну продуктивність обчислювального середовища. Це особливо ефективно для задач математичного моделювання, аналізу фізичних процесів, статистичних обчислень та наукових симуляцій.

Для забезпечення надійності функціонування BOINC використовує механізми відмовостійкості та контролю цілісності результатів. Оскільки клієнтські вузли можуть бути нестабільними або тимчасово недоступними, сервер постійно контролює статус виконання задач. У випадку перевищення допустимого часу виконання або втрати зв'язку із вузлом задача автоматично повертається до черги та призначається іншому клієнту. Це дозволяє уникнути

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		38

втрати обчислювальних даних та забезпечує завершення обчислень навіть у випадку відмов окремих елементів системи.

Важливим компонентом архітектури є алгоритми перевірки результатів. Для підвищення достовірності обчислень одна й та сама задача може надсилатись декільком незалежним вузлам. Після завершення виконання серверний validator порівнює отримані результати та визначає їх відповідність за принципом консенсусу. Якщо результати співпадають у межах допустимої похибки, задача вважається успішно завершеною. У випадку виявлення розбіжностей сервер повторно генерує додаткові копії задачі та надсилає їх іншим клієнтам для повторної перевірки.

Додатково в системі використовуються механізми контрольних точок (checkpointing), які дозволяють зберігати проміжний стан виконання задачі. У разі аварійного завершення роботи операційної системи або вимкнення живлення обчислення можуть бути продовжені з останньої збереженої точки без необхідності повторного запуску задачі з початку. Це дозволяє значно зменшити втрати процесорного часу при виконанні довготривалих наукових обчислень.

Мови програмування та технологічний стек

Для реалізації системи високопродуктивних розподілених обчислень на базі платформи BOINC було використано комплекс сучасних мов програмування, бібліотек та програмних засобів, які забезпечують стабільну роботу обчислювальної інфраструктури, підтримку розподіленого виконання задач, збір телеметрії та адміністрування системи. Вибір технологічного стеку здійснювався з урахуванням вимог до продуктивності, масштабованості, надійності та сумісності з різними операційними системами.

Основною мовою програмування обчислювального ядра BOINC є C та C++. Використання цих мов обумовлено їх високою швидкістю, мінімальними накладними витратами під час роботи з оперативною пам'яттю та можливістю безпосередньої взаємодії з апаратними ресурсами системи. Для систем високопродуктивних обчислень це має критичне значення, оскільки значна

частина часу витрачається на виконання складних математичних операцій, обробку масивів даних, моделювання фізичних процесів або виконання наукових розрахунків.

Мова С використовується у низькорівневих компонентах системи, які відповідають за роботу з мережею, файловою системою та внутрішніми структурами даних. С++ застосовується для реалізації більш складної логіки BOINC-клієнта, механізмів планування задач, роботи з потоками виконання та підтримки кросплатформеності. Завдяки об'єктно-орієнтованому підходу мови С++ забезпечується модульність архітектури та спрощується підтримка програмного коду.

Важливою перевагою використання С/С++ є підтримка багатопотоковості та паралельного виконання обчислень. Це дозволяє ефективно використовувати багатоядерні процесори серверів та робочих станцій. BOINC-клієнт може одночасно запускати декілька задач, розподіляючи навантаження між ядрами процесора, що суттєво підвищує загальну продуктивність системи.

Засоби моніторингу та спостережуваності

Сучасні розподілені обчислювальні системи характеризуються високою складністю архітектури, великою кількістю взаємодіючих компонентів та динамічним характером навантаження. У таких умовах критично важливим є забезпечення повної видимості процесів, що відбуваються всередині системи, оскільки без належного контролю неможливо гарантувати стабільність, продуктивність та коректність виконання обчислень. Саме для цього використовуються засоби моніторингу та спостережуваності (observability), які дозволяють у режимі реального часу отримувати інформацію про стан системи та аналізувати її поведінку.

Моніторинг у контексті розподілених систем передбачає безперервний збір, обробку та візуалізацію метрик, що характеризують роботу окремих компонентів. До таких метрик належать показники завантаження процесора, використання оперативної пам'яті, дискових ресурсів, мережевої активності, а

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		40

також специфічні показники прикладного рівня, наприклад кількість активних задач, швидкість їх виконання, час відгуку або кількість помилок. Завдяки цьому адміністратор або дослідник отримує можливість об'єктивно оцінювати стан системи та виявляти потенційні проблеми на ранніх етапах їх виникнення.

Поняття спостережуваності є більш широким і включає не лише збір метрик, але й аналіз логів, трасування виконання процесів та кореляцію подій між різними компонентами системи. Якщо моніторинг відповідає на питання «що відбувається в системі», то спостережуваність дозволяє зрозуміти «чому це відбувається». Такий підхід є особливо важливим у розподілених обчисленнях, де одна задача може проходити через декілька вузлів, і проблема на одному з етапів може впливати на кінцевий результат у зовсім іншій частині системи.

Застосування інструментів моніторингу дозволяє значно підвищити надійність системи. По-перше, стає можливим оперативне виявлення перевантажень окремих вузлів та балансування навантаження. По-друге, можна відстежувати збої у виконанні задач, перезапуски процесів або деградацію продуктивності. По-третє, аналіз історичних даних дозволяє прогнозувати поведінку системи та планувати масштабування ресурсів. Таким чином, моніторинг виконує не лише діагностичну, але й прогностичну функцію.

Ще однією важливою перевагою є можливість візуалізації даних. Графічне представлення метрик у вигляді графіків, діаграм та дашбордів значно спрощує аналіз великих обсягів інформації. Замість перегляду сирих логів користувач отримує узагальнену картину роботи системи, де чітко видно пікові навантаження, періоди простою, стабільність виконання задач та інші ключові параметри. Це особливо важливо для систем високопродуктивних обчислень, де кількість подій може бути дуже великою.

З точки зору розподілених систем, моніторинг також виконує функцію контролю життєвого циклу задач. Він дозволяє відстежувати, на якому етапі знаходиться кожна обчислювальна одиниця: чи вона очікує виконання, виконується, завершена або потребує повторного запуску. Це забезпечує

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		41

прозорість роботи системи та дозволяє ефективно керувати чергами задач у реальному часі.

Окремо слід відзначити важливість моніторингу для забезпечення відмовостійкості. У разі виходу з ладу окремих вузлів або деградації їх продуктивності система може автоматично реагувати на такі події, перенаправляючи задачі або змінюючи пріоритети виконання. Це зменшує ризик втрати обчислювальних результатів та підвищує загальну стабільність інфраструктури.

Особливістю платформи BOINC є відсутність повноцінних вбудованих засобів моніторингу та спостережуваності, які б відповідали сучасним вимогам адміністрування високонавантажених розподілених систем. Стандартні механізми BOINC дозволяють отримувати лише базову інформацію про стан задач, активність клієнта та процес передачі результатів. Проте такі засоби не забезпечують централізованого збору метрик, довготривалого зберігання статистики, аналізу навантаження у режимі реального часу та побудови комплексної системи контролю стану інфраструктури.

У зв'язку з цим для реалізації підсистеми моніторингу було використано сучасний стек спостережуваності, побудований на основі систем Prometheus та Grafana. Даний підхід дозволив реалізувати повноцінний механізм збору, обробки, збереження та візуалізації телеметричних даних про роботу BOINC-клієнтів, серверної інфраструктури та віртуалізованого середовища виконання задач.

Основою підсистеми моніторингу є система Prometheus - програмний комплекс із відкритим вихідним кодом, призначений для збору часових рядів та моніторингу розподілених систем. Архітектура Prometheus побудована за моделлю pull monitoring, відповідно до якої сервер моніторингу самостійно виконує періодичне опитування вузлів інфраструктури та отримує необхідні метрики через HTTP-запити.

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		42

Сервер Prometheus виконує циклічне опитування exporter-модуля із заданим часовим інтервалом. Під час кожного циклу опитування здійснюється отримання актуальних значень метрик та їх збереження у внутрішній часовій базі даних TSDB (Time Series Database). Особливістю даної бази є висока ефективність роботи із часовими рядами, оптимізоване зберігання великої кількості метрик та підтримка швидкого виконання аналітичних запитів.

Для аналізу накопичених даних у Prometheus використовується спеціалізована мова запитів PromQL (Prometheus Query Language). Дана мова дозволяє виконувати математичні операції над часовими рядами, обчислювати середні значення, будувати агреговану статистику, визначати швидкість зміни параметрів та формувати складні аналітичні вирази. Завдяки цьому стало можливим реалізувати контроль завантаження процесора, середнього часу виконання задач, кількості завершених work units та частоти виникнення помилок у системі.

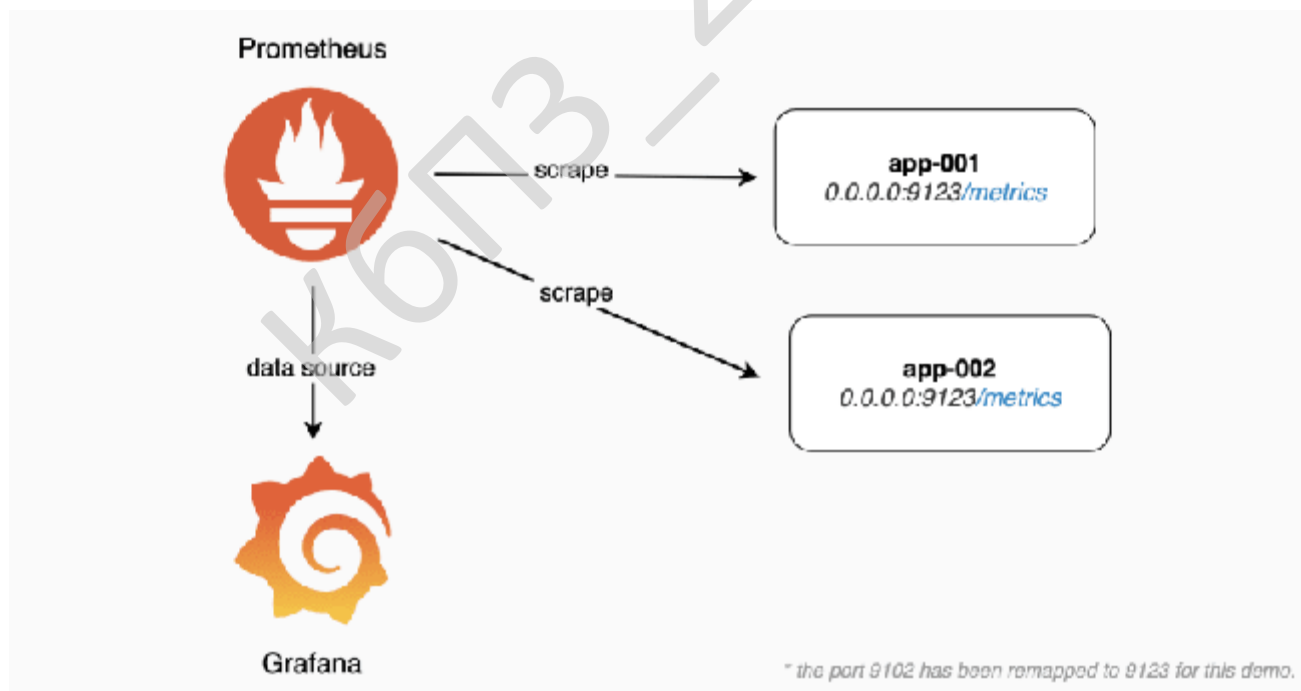


Рисунок 2.11 - Система збору метрик Prometheus

Для візуалізації отриманих даних було використано систему Grafana. Grafana являє собою багатфункціональну платформу візуалізації та аналітики, яка підтримує інтеграцію із різними джерелами даних, включаючи Prometheus. Основним призначенням Grafana є побудова інтерактивних інформаційних панелей (dashboard) для моніторингу стану інфраструктури у режимі реального часу.

Grafana дозволяє будувати графіки, гістограми, таблиці, часові шкали та інші типи візуалізацій. Для кожного параметра можуть задаватися окремі умови відображення, масштаби часу, механізми агрегації та автоматичного оновлення даних. Це забезпечує можливість комплексного аналізу роботи системи та спрощує виявлення аномалій у роботі обчислювальної інфраструктури.

Важливою особливістю використання Grafana є підтримка системи сповіщень (alerting). У випадку перевищення критичних значень навантаження, аварійного завершення задач або втрати зв'язку із BOINC-клієнтом система може автоматично формувати повідомлення для адміністратора. Це дозволяє оперативно реагувати на несправності та забезпечувати безперервність виконання розподілених обчислень.

Grafana побудована за модульним принципом та підтримує велику кількість плагінів і джерел даних, що дозволяє інтегрувати її практично з будь-якою сучасною системою моніторингу. У рамках даної роботи Grafana використовується як центральний компонент візуалізації, який об'єднує дані про стан обчислювальних вузлів, активність BOINC-клієнтів, виконання задач та використання системних ресурсів. Завдяки цьому користувач отримує єдину точку доступу до всієї інформації про функціонування інфраструктури.

Однією з ключових переваг Grafana є підтримка роботи з часовими рядами (time-series data). Оскільки більшість метрик моніторингу змінюються у часі, система дозволяє не лише відображати поточний стан вузлів, але й аналізувати динаміку змін навантаження, продуктивності та стабільності роботи. Це особливо

важливо для систем високопродуктивних обчислень, де продуктивність окремих вузлів може суттєво змінюватися залежно від типу виконуваних задач.

Окрему роль відіграє механізм dashboard-панелей. Кожна інформаційна панель може бути налаштована під конкретні задачі моніторингу. Наприклад, одна панель може відображати загальне навантаження системи, інша - детальну статистику виконання BOINC-задач, а окрема панель - стан мережевої взаємодії або використання ресурсів VirtualBox. Це дозволяє логічно розділити інформацію та спростити її аналіз.

Grafana також підтримує механізми фільтрації та агрегації даних. Користувач може обирати часові інтервали аналізу, переглядати історичні дані, виконувати групування метрик за вузлами або типами задач та формувати порівняльні графіки. Такий підхід значно полегшує аналіз продуктивності системи та дозволяє виявляти вузькі місця інфраструктури.

Важливою перевагою Grafana є підтримка автоматичного оновлення даних у реальному часі. Після надходження нових метрик від Prometheus інформація на dashboard оновлюється автоматично без необхідності ручного перезавантаження сторінки. Це забезпечує актуальність інформації та дозволяє спостерігати за станом системи практично без затримок. Для забезпечення зручності адміністрування Grafana підтримує механізми керування доступом та систему ролей користувачів.

Це дозволяє обмежувати доступ до окремих панелей, змін конфігурації або історичних даних. Подібний функціонал є важливим при використанні системи у багатокористувацькому середовищі або в межах корпоративної інфраструктури.

Ще однією важливою особливістю Grafana є можливість експорту та збереження конфігурацій dashboard. Це дозволяє швидко переносити налаштування між серверами, створювати резервні копії панелей моніторингу та повторно використовувати підготовлені шаблони для інших проектів або вузлів.

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		45



Рисунок 2.12 - Система візуалізації Grafana

Для реалізації веб-частини системи моніторингу було використано веб-сервер Nginx. Даний веб-сервер забезпечує обробку HTTP-запитів, маршрутизацію звернень до exporter-модуля та надання доступу до метрик системи моніторингу. Вибір Nginx обумовлений його високою продуктивністю, стабільністю роботи та широким використанням у сучасних серверних і DevOps-інфраструктурах.

Nginx є високопродуктивним веб-сервером та reverse проху-системою, яка використовує асинхронну подієво-орієнтовану модель роботи. На відміну від класичних веб-серверів, які створюють окремий процес або потік для кожного з'єднання, Nginx здатний обробляти велику кількість одночасних запитів із мінімальним використанням системних ресурсів. Завдяки цьому він широко застосовується у високонавантажених системах, хмарних середовищах та інфраструктурах моніторингу.

Основними перевагами Nginx є висока швидкодія, низьке споживання оперативної пам'яті, простота конфігурації та підтримка великої кількості додаткових функцій. Веб-сервер підтримує роботу як HTTP-сервер, reverse проху,

балансувальник навантаження, SSL/TLS-шлюз та засіб кешування даних. Також Nginx має розвинену систему логування та механізми контролю доступу, що робить його зручним для побудови серверної інфраструктури.

У розробленій системі Nginx використовується для забезпечення доступу до endpoint `/metrics`, через який система моніторингу Prometheus отримує статистичні дані про роботу BOINC-клієнта. Для цього було реалізовано власний Python-exporter `boinc_exporter.py`, який збирає інформацію про стан задач, використання процесора, оперативної пам'яті, статус віртуальних машин та інші параметри роботи системи.

Веб-сервер Nginx виконує роль проміжного рівня між Prometheus та Python-exporter. Під час надходження HTTP-запиту до `/metrics` Nginx передає його до скрипта `boinc_exporter.py` за допомогою механізму FastCGI та служби `fcgiwrap`. Після виконання exporter формує набір метрик у форматі Prometheus та повертає їх клієнту у текстовому вигляді.



Рисунок 2.13 - Вебсервер Nginx

Для реалізації допоміжних компонентів системи моніторингу було використано мову програмування Python. Вибір Python обумовлений його простотою, високою швидкістю розробки, великою кількістю бібліотек та

широким використанням у сфері системного адміністрування, DevOps, автоматизації та аналізу даних.

Python є високорівневою інтерпретованою мовою програмування загального призначення, яка підтримує процедурний, об'єктно-орієнтований та функціональний підходи до розробки програмного забезпечення. Однією з головних переваг мови є зрозумілий синтаксис, що дозволяє швидко створювати та підтримувати програмні модулі навіть у складних інфраструктурних проєктах.

Завдяки великій кількості вбудованих та сторонніх бібліотек Python активно використовується для розробки серверних застосунків, автоматизації адміністрування, обробки даних, систем моніторингу та мережевих сервісів. Мова добре інтегрується з Linux-середовищем та дозволяє ефективно працювати із системними процесами, файлами конфігурації, мережевими з'єднаннями та API сторонніх сервісів.

У розробленій системі Python використовується для реалізації власного exporter-модуля `boinc_exporter.py`, який забезпечує збір метрик із BOINC-клієнта. Основним завданням exporter є отримання інформації про стан системи, аналіз активних обчислювальних задач та формування метрик у форматі, сумісному із Prometheus.

Під час роботи `boinc_exporter.py` виконує опитування BOINC-клієнта, аналізує список активних та завершених work units, визначає статуси задач, збирає дані про використання процесора, оперативної пам'яті та стан віртуальних машин VirtualBox. Після цього отримана інформація перетворюється у текстовий формат метрик Prometheus та передається через HTTP-інтерфейс `/metrics`.

Використання Python дозволило реалізувати exporter із мінімальними витратами часу та забезпечити просту модифікацію коду у майбутньому. Додатковою перевагою є кросплатформеність мови та можливість інтеграції з різними системами моніторингу, базами даних і мережевими сервісами.

У поєднанні з веб-сервером Nginx та системою Prometheus Python забезпечує побудову гнучкої та масштабованої підсистеми моніторингу, яка

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		48

дозволяє централізовано контролювати стан BOINC-інфраструктури та аналізувати ефективність виконання розподілених обчислень у режимі реального часу.



Рисунок 2.14 - Скриптова мова програмування Python

2.3 Розгорнута постановка завдання

В процесі розробки бакалаврської роботи на тему системи високопродуктивних обчислень для моделювання фізичних процесів у розподілених обчислювальних середовищах необхідно виконати наступний обсяг робіт:

- обґрунтувати актуальність використання розподілених обчислювальних систем для вирішення задач моделювання фізичних процесів, а також визначити область їх застосування у наукових дослідженнях та освітньому процесі;
- провести аналіз існуючих платформ розподілених обчислень, визначити їх переваги та недоліки, а також обґрунтувати вибір платформи BOINC як базового рішення для реалізації системи;

- виконати налаштування та підключення сервісу розподілених обчислень системи BOINC, забезпечивши коректну взаємодію з інфраструктурою проекту через мережеві протоколи, включаючи отримання обчислювальних задач, їх виконання та передачу результатів; взаємодії;
- підключити та налаштувати на сервері наукові обчислювальні проекти, зокрема LHC@home та ClimatePrediction.net, що використовуються для моделювання фізичних процесів;
- дослідити особливості виконання обчислювальних задач у середовищі віртуалізації, зокрема використання Oracle VM VirtualBox для запуску спеціалізованих віртуальних машин, у яких виконуються задачі зазначених проектів;
- встановити та налаштувати exporter для збору метрик роботи системи, включаючи інформацію про виконання задач, стан клієнтів та використання обчислювальних ресурсів;
- інтегрувати систему моніторингу із використанням Grafana, забезпечивши візуалізацію ключових показників роботи розподіленої системи;
- налаштувати інформаційну панель (dashboard) у Grafana для відображення стану виконання задач, навантаження на процесор, пам'ять та інші ресурси системи у реальному часі;
- провести тестування системи під навантаженням, включаючи запуск обчислювальних задач різної складності, з метою оцінки продуктивності, масштабованості та стабільності роботи;
- проаналізувати отримані результати виконання задач та ефективність використання ресурсів, визначити можливі вузькі місця та шляхи оптимізації системи;
- підготувати опис використаних технологій, архітектури системи, алгоритмів розподілу задач та засобів моніторингу для включення у пояснювальну записку;

- сформулювати висновки щодо ефективності використання розподілених обчислювальних середовищ для моделювання фізичних процесів та надати рекомендації щодо подальшого розвитку системи.

К6ПЗ_2026

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

3 ОПИС І ОБГРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Проаналізувавши існуючі підходи до організації розподілених обчислень та високопродуктивних систем, було прийнято рішення використати платформу BOINC для реалізації системи моделювання фізичних процесів у розподіленому середовищі. Система спрямована на ефективне використання доступних обчислювальних ресурсів шляхом їх об'єднання у єдину інфраструктуру, що дозволяє виконувати складні наукові задачі без необхідності використання централізованих суперкомп'ютерів.

Основна функція даної системи полягає у розподілі обчислювальних задач між клієнтськими вузлами, їх виконанні та подальшій агрегації результатів. У якості прикладного навантаження використовуються наукові проекти, зокрема LHC@home та ClimatePrediction.net, які дозволяють виконувати моделювання фізичних процесів різної складності. Виконання задач у даних проектах часто здійснюється у середовищі віртуалізації із застосуванням Oracle VM VirtualBox, що забезпечує ізоляцію та уніфікацію обчислювального середовища.

Основними критеріями функціонування системи є:

- ініціалізація клієнта: встановлення та налаштування клієнтського програмного забезпечення BOINC на сервері та обчислювальних вузлах, підключення до обраних наукових проектів;
- отримання задач: клієнт періодично звертається до серверів проектів для отримання нових обчислювальних задач (work units), які формуються на основі наукових моделей;
- виконання обчислень: запуск задач у фоновому режимі з використанням доступних ресурсів процесора, оперативної пам'яті та, за необхідності, віртуалізованого середовища;

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		52

- використання віртуалізації: застосування Oracle VM VirtualBox для виконання задач, які потребують спеціалізованого середовища, що дозволяє стандартизувати процес обчислень незалежно від апаратної платформи;
- передача результатів: після завершення обчислень результати автоматично передаються на сервери відповідних проектів для подальшої обробки та валідації;
- валідація результатів: перевірка коректності обчислень шляхом порівняння результатів, отриманих від різних клієнтів, що забезпечує достовірність наукових даних;
- моніторинг системи: збір метрик роботи системи за допомогою exporter-а та їх візуалізація у Grafana для аналізу продуктивності та навантаження;
- аналіз навантаження: відображення у реальному часі показників використання ресурсів, таких як завантаження процесора, використання пам'яті та кількість активних задач;
- управління виконанням: можливість регулювання параметрів клієнта, включаючи обмеження використання ресурсів, час виконання задач та пріоритети обчислень.

Даний алгоритм забезпечує безперервний цикл обчислень, у якому клієнтські вузли отримують задачі, виконують їх та повертають результати для подальшого аналізу. Завдяки цьому система дозволяє ефективно використовувати наявні ресурси та досягати високої продуктивності при виконанні складних наукових розрахунків.

Важливою особливістю реалізованої системи є відсутність повноцінного вбудованого моніторингу у BOINC, що потребує використання додаткових інструментів. Для вирішення цієї задачі було розроблено власний exporter, який збирає інформацію про стан виконання задач та передає її до системи візуалізації Grafana. Це дозволяє створити зручну панель моніторингу (dashboard), яка відображає ключові параметри роботи системи у реальному часі.

З точки зору масштабованості система може бути розширена шляхом підключення нових клієнтських вузлів або додавання нових наукових проектів. Такий підхід дозволяє адаптувати систему до різних типів задач, включаючи моделювання фізичних процесів, кліматичні дослідження та обробку великих масивів даних.

Таким чином, використання BOINC у поєднанні з інструментами моніторингу, такими як Grafana, дозволяє створити ефективну, масштабовану та гнучку систему високопродуктивних обчислень, придатну для виконання складних наукових задач у розподіленому середовищі.

3.2 Розробка структурної схеми

Програмна частина системи високопродуктивних обчислень побудована на базі платформи BOINC та додаткових інструментів моніторингу. Архітектура системи включає серверну частину, клієнтські вузли, підключені наукові проекти та підсистему моніторингу.

Логіка роботи системи базується на взаємодії декількох ключових функціональних компонентів, які забезпечують повний цикл обробки обчислювальних задач - від їх формування до отримання та аналізу результатів.

Центральну роль у системі відіграє сервер керування задачами, який виконує функції координатора. Він відповідає за формування обчислювальних задач, їх розподіл між клієнтськими вузлами та збереження отриманих результатів. Планування виконання задач здійснюється з урахуванням доступних ресурсів обчислювальних вузлів, їх продуктивності та типу задач. Для цього використовуються відповідні алгоритми розподілу work units. Сервер також взаємодіє з базою даних, у якій зберігається інформація про задачі, клієнтів, результати обчислень та статистичні показники. Додатково реалізована підсистема валідації, що забезпечує перевірку результатів шляхом їх порівняння між кількома незалежними вузлами.

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		54

Клієнтські обчислювальні вузли представлені програмним забезпеченням BOINC, яке встановлюється на сервері університету та працює у фоновому режимі. Клієнт періодично взаємодіє з сервером проекту, отримуючи нові задачі та передаючи результати виконання. Обчислення виконуються із використанням доступних ресурсів процесора, оперативної пам'яті та, за необхідності, дискової підсистеми. При цьому передбачено механізми обмеження використання ресурсів, що дозволяє уникнути негативного впливу на основні процеси системи та забезпечує стабільність роботи обчислювального вузла.

Джерелом обчислювальних задач виступають зовнішні наукові проекти, інтегровані через інфраструктуру BOINC. У межах даної роботи використовуються проекти LHC@home та ClimatePrediction.net, які генерують задачі, пов'язані з моделюванням фізичних процесів. Отримані результати застосовуються у наукових дослідженнях, зокрема у галузях фізики високих енергій та кліматичного моделювання.

Важливою складовою системи є підсистема віртуалізації, що забезпечує виконання частини задач у ізолюваному середовищі. Для цього використовується платформа Oracle VM VirtualBox, яка дозволяє запускати попередньо налаштовані віртуальні машини з необхідним програмним забезпеченням. Такий підхід гарантує однакові умови виконання задач незалежно від апаратної конфігурації обчислювального вузла та підвищує відтворюваність результатів.

Окремо реалізована система моніторингу та збору метрик, що компенсує відсутність розвинених вбудованих засобів спостереження у BOINC. Для цього було налаштовано exporter, який збирає інформацію про стан клієнта, кількість активних задач, рівень завантаження процесора та використання пам'яті. Зібрані дані передаються до системи візуалізації, що дозволяє здійснювати централізований контроль роботи обчислювальної інфраструктури.

Система візуалізації реалізована на базі Grafana, де створено спеціалізовану панель моніторингу. Вона забезпечує відображення ключових

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		55

показників роботи системи у реальному часі, включаючи статус клієнтів, кількість виконуваних задач та ефективність використання ресурсів. Це дозволяє оперативно аналізувати продуктивність системи, виявляти потенційні проблеми та приймати рішення щодо оптимізації роботи розподіленого обчислювального середовища.

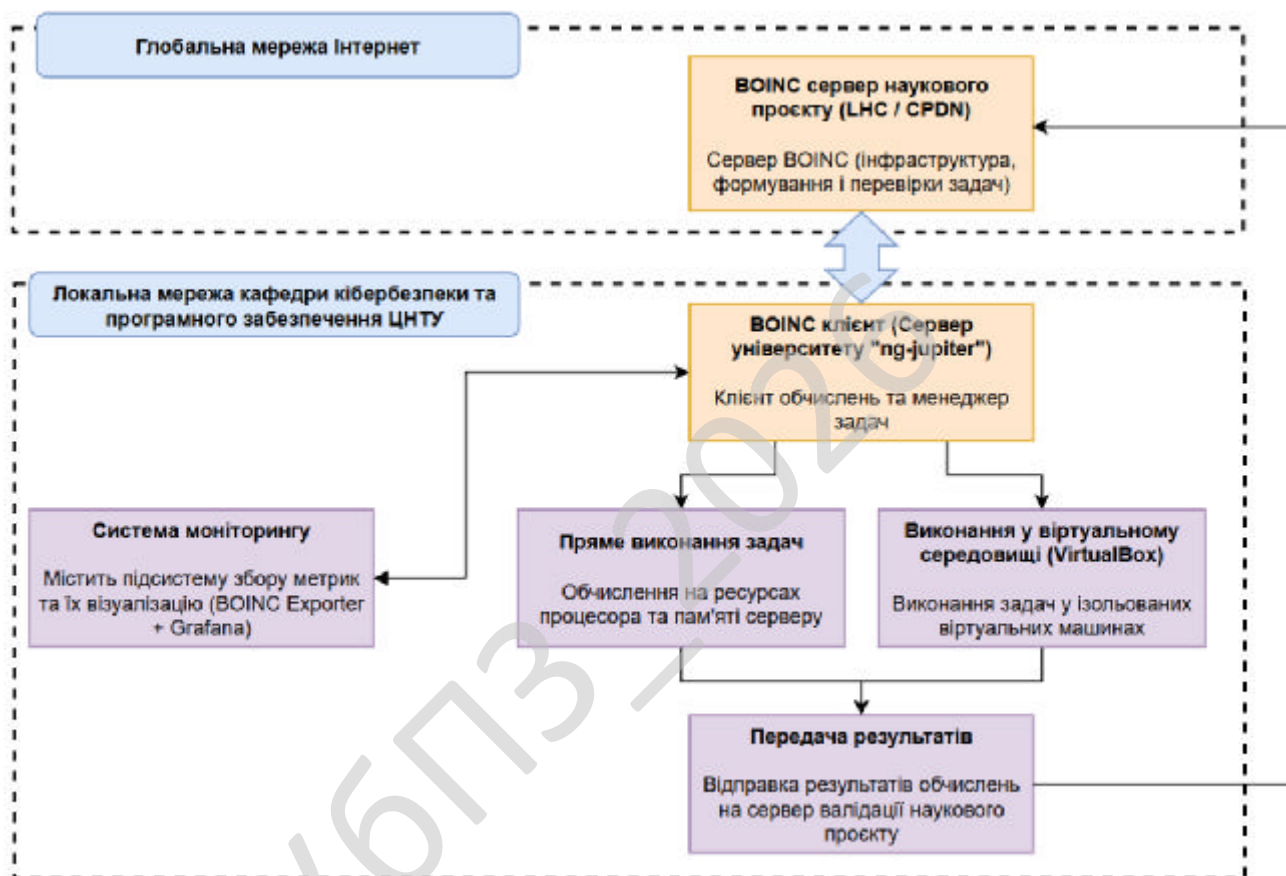


Рисунок 3.1 - Структурна схема системи

3.3 Розробка функціональної схеми

Функціональна схема (рис. 3.2) демонструє логіку взаємодії між обчислювальним вузлом, сервером наукового проекту, підсистемою виконання задач та системою моніторингу.

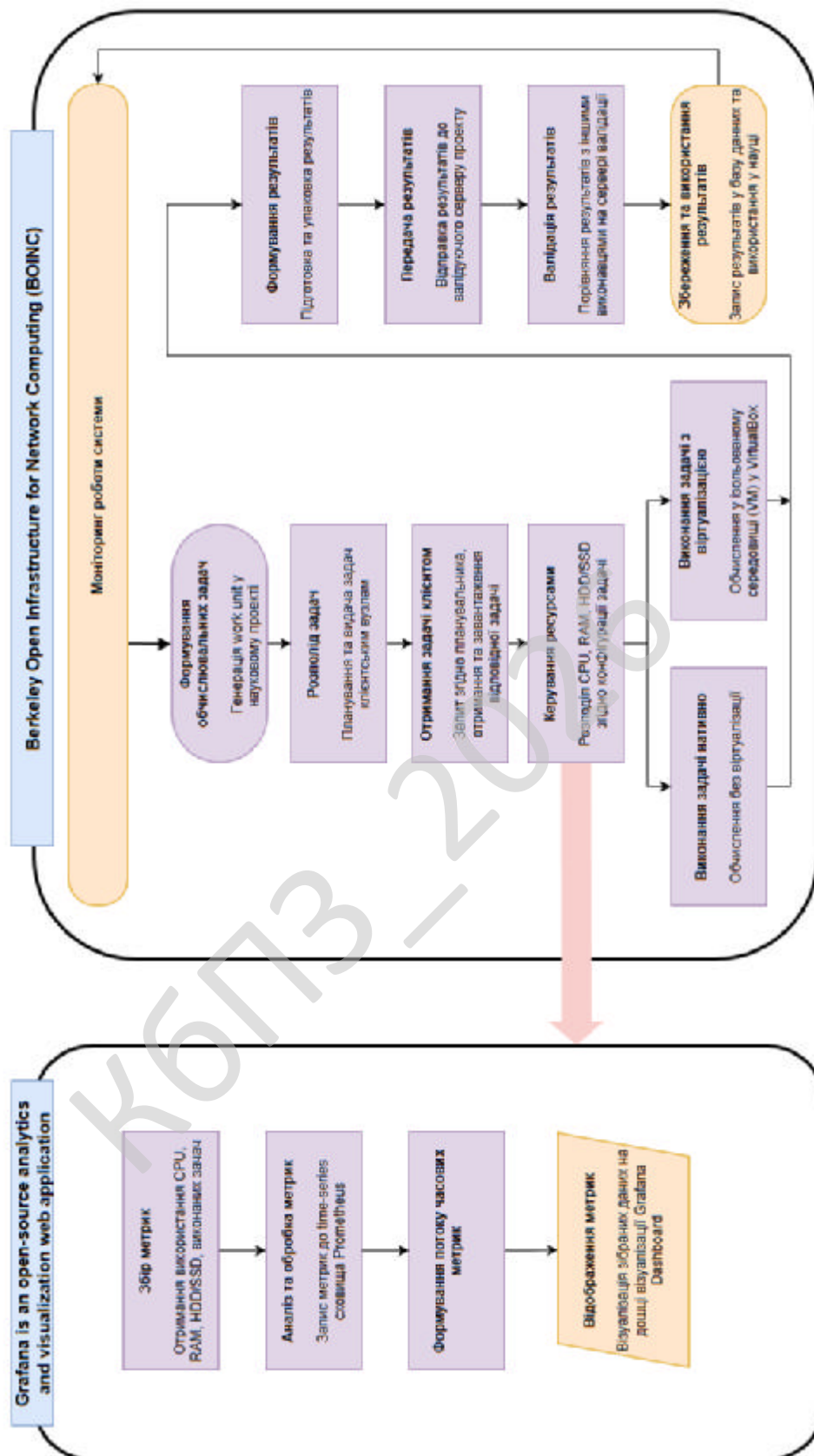


Рисунок 3.2 - Функціональна схема системи

Інформаційний обмін у системі базується на мережевій взаємодії клієнта з сервером проекту, використанні механізмів планування задач BOINC, а також передачі результатів обчислень через стандартні мережеві протоколи.

При запуску системи відбувається автоматична ініціалізація робочого середовища:

- клієнт BOINC, розгорнутий на сервері університету, встановлює з'єднання із сервером наукового проекту та проходить процедуру автентифікації;

- після підключення клієнт виконує запит до сервера для отримання обчислювальних задач (work units), які формуються відповідно до специфіки наукового проекту;

- у випадку використання задач, що потребують ізолюваного середовища, автоматично ініціюється запуск віртуальних машин, у яких розгортається необхідне програмне забезпечення;

- система моніторингу паралельно починає збір метрик стану клієнта, включаючи навантаження на процесор, використання пам'яті та статус виконання задач.

Основна логіка функціонування системи передбачає безперервний цикл обробки задач:

- отримання задач: клієнт періодично звертається до сервера проекту з метою отримання нових обчислювальних задач, при цьому враховується поточне навантаження системи та доступні ресурси;

- виконання задач: обчислення виконуються або безпосередньо на ресурсах сервера, або у віртуалізованому середовищі, залежно від вимог конкретного проекту;

- контроль ресурсів: клієнт динамічно регулює використання процесора та пам'яті, що дозволяє уникнути перевантаження системи;

- формування результатів: після завершення обчислень формується результат, який підготовлюється до передачі на сервер;

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		58

- передача результатів: результати автоматично відправляються на сервер наукового проекту для подальшої перевірки та використання;
- валідація: сервер проекту виконує перевірку отриманих результатів шляхом їх порівняння з результатами інших вузлів.

Окрему роль у системі відіграє підсистема моніторингу:

- Exporter здійснює збір інформації про стан клієнта, включаючи кількість активних задач, їх статус та використання ресурсів;
- зібрані дані передаються до системи візуалізації, де формуються графіки та панелі моніторингу;
- користувач отримує можливість у реальному часі відслідковувати ефективність роботи системи та навантаження на обчислювальний вузол.

Таким чином, функціональна схема відображає повний життєвий цикл обчислювальної задачі - від моменту її отримання до передачі результатів та їх подальшої обробки. Реалізована модель забезпечує ефективне використання ресурсів, масштабованість та можливість інтеграції з різними науковими проектами, що робить систему придатною для задач високопродуктивних розподілених обчислень

3.4 Розробка діаграми процесів

Розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Початковим етапом є запуск системи, після чого відбувається ініціалізація клієнтського програмного забезпечення BOINC на обчислювальному вузлі. Основна взаємодія користувача із системою здійснюється через модуль керування клієнтом, який відповідає за налаштування параметрів обчислень, підключення до наукових проектів та запуск процесу обробки задач.

Після налаштування параметрів система переходить до етапу отримання задач від серверів наукових проектів.

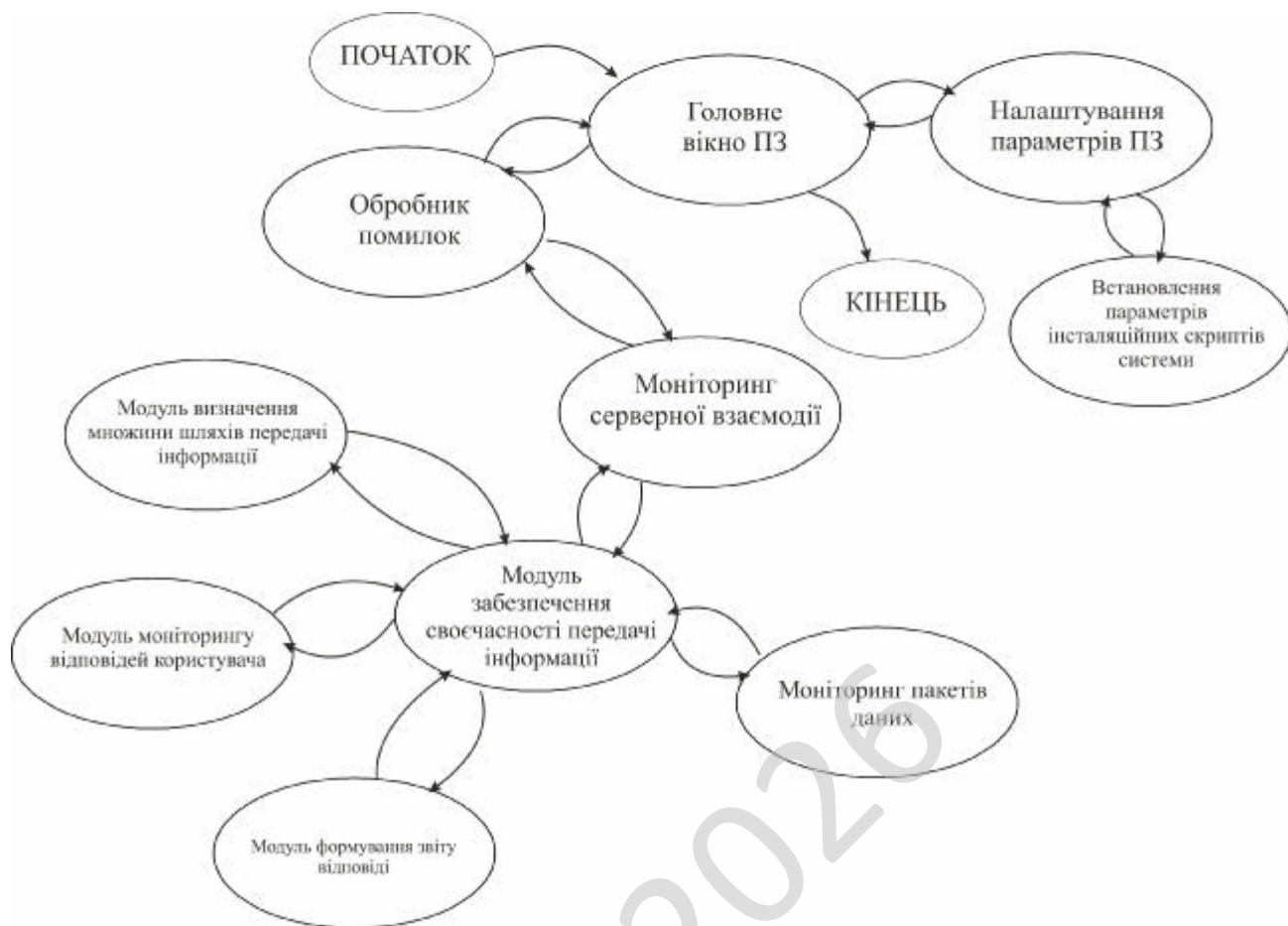


Рисунок 3.3 - Діаграма процесів

Отримані задачі передаються до модуля виконання, який визначає спосіб їх обробки - або безпосередньо на ресурсах хост-системи, або у віртуалізованому середовищі. У випадку використання віртуалізації, активується модуль керування віртуальними машинами, що забезпечує запуск ізольованого середовища виконання.

Паралельно з виконанням задач функціонує модуль моніторингу, який здійснює збір інформації про стан системи, включаючи завантаження процесора, використання оперативної пам'яті, кількість активних задач та їхній статус.

Ці дані передаються до модуля експорту метрик, який забезпечує їх подальшу передачу у систему візуалізації.

Модуль візуалізації (Grafana) обробляє отримані метрики та відображає їх у вигляді інтерактивних панелей, що дозволяє користувачу здійснювати аналіз продуктивності системи у реальному часі. Таким чином реалізується зворотний зв'язок між обчислювальним процесом та оператором системи.

Важливим елементом є модуль обробки помилок, який реагує на збої у виконанні задач, проблеми з мережею або перевищення ресурсних обмежень. У разі виникнення помилки система може повторно ініціювати виконання задачі або скоригувати параметри роботи клієнта.

Завершальним етапом процесу є передача результатів обчислень на сервер наукового проекту. Після успішної відправки результатів клієнт автоматично запитує нові задачі, забезпечуючи безперервність обчислювального процесу.

К6ПЗ-2026

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		61

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ ТА ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ І ПРОГРАМНИХ РІШЕНЬ

4.1 Розробка блок-схем та опис алгоритмів функціонування системи

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми. Для забезпечення безперервної та стабільної роботи системи високопродуктивних розподілених обчислень у складі програмного комплексу використовується системна служба boinc-client.service, яка функціонує під керуванням менеджера служб systemd операційної системи Linux. Дана служба забезпечує автоматичний запуск клієнта BOINC під час старту операційної системи, контроль його стану, перезапуск у випадку збоїв та підтримку довготривалого виконання обчислювальних задач у фоновому режимі.

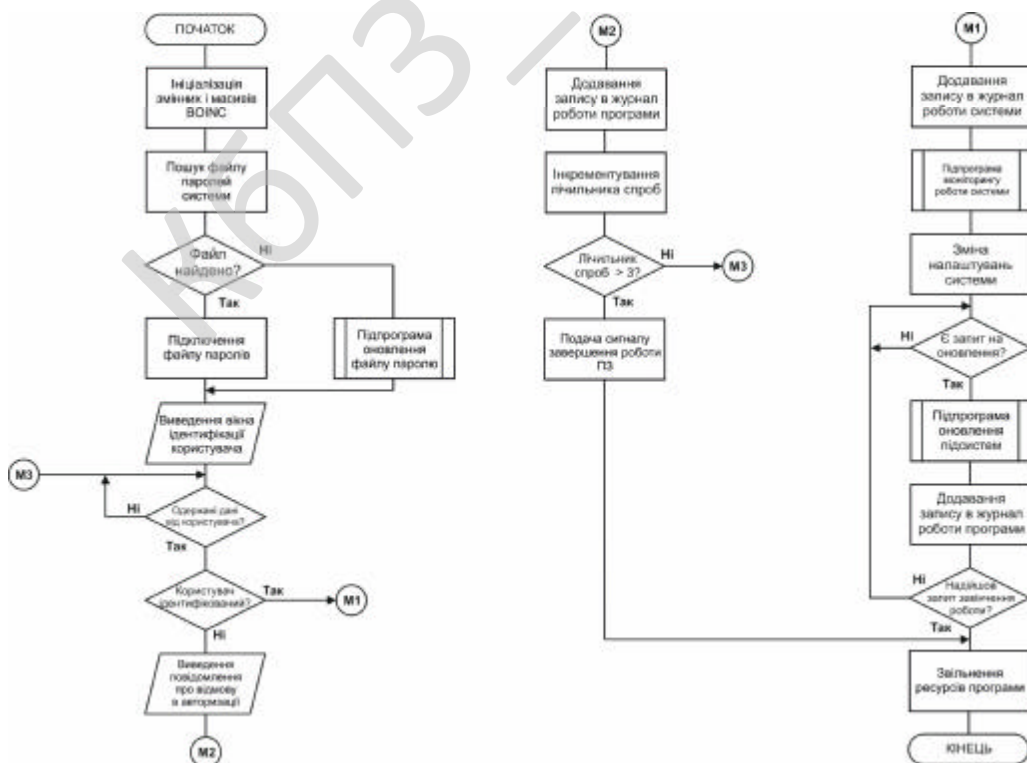


Рисунок 4.1 - Блок-схема основної програми

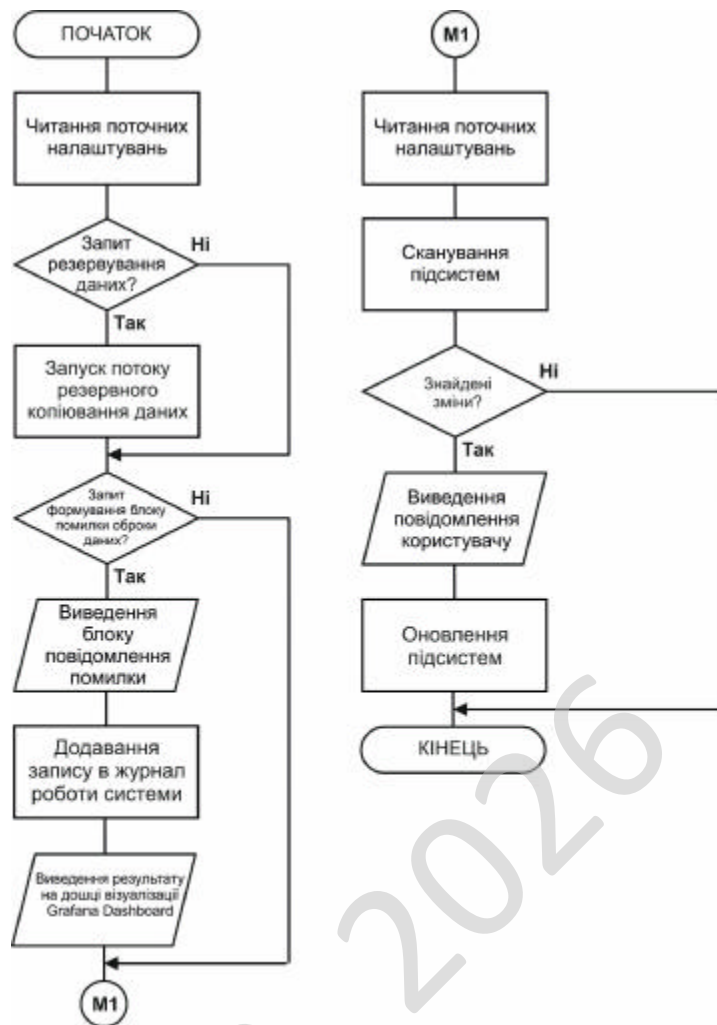


Рисунок 4.2 - Блок-схема роботи підпрограми

Робота системи починається з ініціалізації служби boinc-client.service. Після запуску виконується завантаження конфігураційних параметрів клієнта, перевірка наявності необхідних системних файлів та підключення до локальних директорій збереження службових даних. На даному етапі система формує внутрішні структури даних, ініціалізує параметри підключення до серверів наукових проектів та активує механізми керування обчислювальними задачами.

Одним із ключових елементів алгоритму функціонування є контроль цілісності конфігураційного середовища BOINC-клієнта. У випадку відсутності необхідних конфігураційних файлів або службових параметрів система автоматично переходить до процедури їх створення чи відновлення. Це дозволяє

забезпечити стійкість програмного комплексу до аварійного завершення роботи, пошкодження службових даних або перезапуску операційної системи.

Після завершення етапу ініціалізації служба переходить у режим активного функціонування (active (running)), у якому виконується безперервна обробка обчислювальних задач, отриманих від серверів розподілених наукових проектів. Клієнт BOINC здійснює періодичне встановлення мережових з'єднань із віддаленими серверами, отримує нові задачі, виконує їх обробку з використанням ресурсів центрального процесора та оперативної пам'яті, після чого передає результати обчислень назад до серверної частини системи.

Служба boinc-client.service також забезпечує контроль використання системних ресурсів. У процесі роботи виконується моніторинг завантаження процесора, використання оперативної пам'яті, кількості активних потоків та загального часу виконання задач. Отримані параметри використовуються для аналізу продуктивності системи та подальшого відображення у підсистемі моніторингу Grafana.

Завдяки використанню механізмів systemd забезпечується автоматичне відновлення роботи служби після виникнення помилок або аварійного завершення процесу. Це дозволяє підтримувати безперервний цикл високопродуктивних обчислень без необхідності ручного втручання адміністратора системи.

```
● boinc-client.service - Berkeley Open Infrastructure Network Computing Client
   Loaded: loaded (/lib/systemd/system/boinc-client.service; enabled; preset: enabled)
   Active: active (running) since Thu 2026-04-30 08:02:48 EEST; 8h ago
     Docs: man:boinc(1)
  Main PID: 2078 (boinc)
    Tasks: 358 (limit: 154301)
   Memory: 47.76
        CPU: 6d 23h 10min 39.176s
   CGroup: /system.slice/boinc-client.service
```

Рисунок 4.3 - Systemd service boinc-client.service

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		64

Після завершення ініціалізації виконується запуск модуля автентифікації та перевірки доступу до наукових проектів. BOINC-клієнт здійснює мережеву взаємодію із серверами проектів LHC@home та ClimatePrediction.net через протоколи TCP/IP та HTTP(S). Під час цієї взаємодії виконується передача ідентифікаційних параметрів вузла, після чого сервер формує пакет обчислювальних задач для конкретного клієнта.

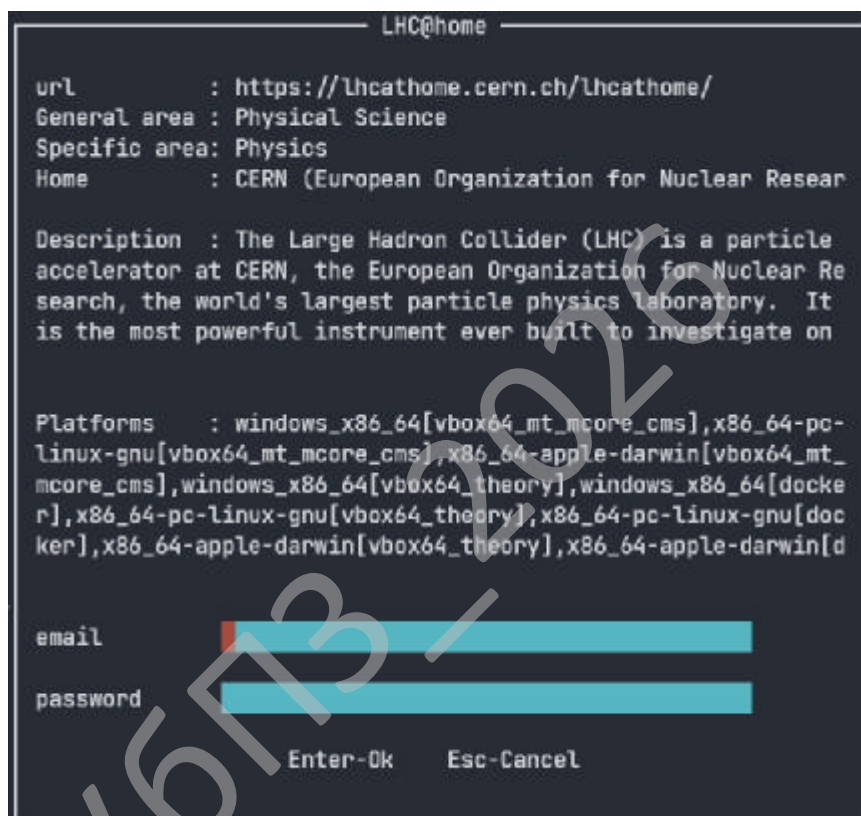


Рисунок 4.4 - Вікно авторизації у проект LHC@home

Наступним етапом алгоритму функціонування системи є обробка отриманих обчислювальних задач типу work unit, які надходять від серверів розподілених наукових проектів через мережу Інтернет. Після встановлення з'єднання із сервером проекту BOINC-клієнт виконує процедуру синхронізації, у ході якої здійснюється перевірка доступних задач, отримання службової інформації про проект, завантаження необхідних вхідних даних та формування

локальної черги обчислень. Усі отримані задачі автоматично розподіляються відповідно до параметрів конфігурації клієнта, доступних системних ресурсів та типу обчислювального навантаження.

Контроль стану виконання задач реалізовано за допомогою текстового інтерфейсу boinctui, який дозволяє у режимі реального часу відображати перелік активних, завершених та призупинених задач. У головному вікні інтерфейсу відображаються назви наукових проектів, стан виконання кожного work unit, відсоток завершення обчислень, поточний час роботи задачі, очікуваний час завершення, а також інформація про використання системних ресурсів. Крім цього, інтерфейс дозволяє контролювати процес передачі результатів, завантаження нових задач та стан мережевої взаємодії із серверами BOINC-проектів.

Після завантаження задач система виконує визначення типу їх виконання. Частина work units запускається безпосередньо у середовищі хост-операційної системи Linux із використанням ресурсів центрального процесора, оперативної пам'яті та локальної файлової системи серверу. Такий режим використовується для задач, які не потребують спеціалізованого ізольованого програмного середовища та можуть бути виконані нативно засобами BOINC-клієнта.

Для окремих категорій наукових задач застосовується підсистема віртуалізації Oracle VM VirtualBox. Необхідність використання віртуальної машини обумовлена специфікою окремих наукових застосунків, які потребують строго визначеного програмного оточення, спеціальних бібліотек або модифікованої конфігурації операційної системи. У даному випадку BOINC-клієнт автоматично ініціює запуск попередньо налаштованої віртуальної машини та передає їй параметри виконання задачі.

Алгоритм роботи віртуалізованої підсистеми включає декілька послідовних етапів. На початковому етапі здійснюється запуск служби VirtualBox та створення екземпляра віртуальної машини. Після цього виконується ініціалізація гостьової операційної системи, монтування службових каталогів

BOINC та передача вхідних даних задачі до ізолюваного середовища виконання. Далі відбувається запуск наукового застосунку, який безпосередньо виконує математичні обчислення або моделювання відповідно до параметрів отриманого work unit.

У процесі виконання задачі BOINC-клієнт здійснює постійний моніторинг стану віртуальної машини. Контролюється рівень використання процесорного часу, обсяг оперативної пам'яті, стан файлової системи, активність мережевих інтерфейсів та коректність виконання обчислювального процесу. Додатково виконується перевірка відповіді гостьової операційної системи та контроль наявності аварійних завершень процесів VirtualBox.

Після завершення обчислень результати автоматично експортуються з віртуальної машини до локального каталогу BOINC-клієнта, де проходять процедуру перевірки цілісності та підготовки до передачі на сервер проекту. Далі BOINC-клієнт формує мережевий пакет передачі результатів та виконує синхронізацію із сервером розподілених обчислень.

У випадку виникнення помилки виконання, зависання віртуальної машини або аварійного завершення процесу VirtualBox система автоматично формує запис у журналі подій Linux та внутрішньому журналі BOINC-клієнта. Після фіксації помилки виконується процедура завершення некоректного процесу, очищення тимчасових даних та повторної ініціалізації задачі. Якщо кількість помилок перевищує допустиме значення, задача переводиться у стан помилки та надсилається повідомлення до підсистеми моніторингу.

Для забезпечення стабільності роботи системи використовується механізм автоматичного керування пріоритетами задач. BOINC-клієнт динамічно регулює навантаження на процесор залежно від поточного стану серверу, доступності ресурсів та встановлених обмежень користувача. Завдяки цьому забезпечується баланс між виконанням розподілених обчислень та стабільною роботою основної серверної інфраструктури.

Паралельно із виконанням розподілених обчислень у системі функціонує окрема підсистема моніторингу та збору даних, призначена для контролю стану BOINC-клієнта, аналізу навантаження серверної інфраструктури та візуалізації параметрів виконання наукових задач у режимі реального часу. Необхідність створення власної системи моніторингу обумовлена тим, що стандартний BOINC-клієнт не містить розширених засобів централізованого збору метрик та не забезпечує інтеграцію із сучасними системами моніторингу серверної інфраструктури.

Алгоритм функціонування підсистеми моніторингу реалізовано за принципом циклічного опитування стану обчислювального середовища. Основним елементом даної підсистеми є програмний модуль типу exporter, написаний мовою Python. Даний модуль виконує автоматизований збір інформації про стан BOINC-клієнта, параметри активних задач, використання системних ресурсів та роботу віртуалізованих компонентів Oracle VM VirtualBox.

Робота exporter-модуля починається із формування запиту до локального інтерфейсу BOINC-клієнта. Після отримання відповіді система виконує аналіз списку активних work units, визначає стан кожної задачі, її приналежність до конкретного наукового проекту, поточний прогрес виконання, час обробки та очікуваний час завершення обчислень. Додатково виконується аналіз задач, які перебувають у стані очікування, передачі результатів або завершення із помилкою.

Окремий блок алгоритму призначений для збору системних параметрів серверу. Exporter здійснює зчитування поточного завантаження центрального процесора, кількості активних потоків, використання оперативної пам'яті, обсягу кешованих даних, стану файлової системи та рівня мережевої активності. Для задач, що виконуються у середовищі VirtualBox, додатково виконується перевірка стану віртуальних машин, використання виділених ресурсів та контроль активності процесів гостьових операційних систем.

Після завершення збору даних exporter виконує перетворення отриманої інформації у формат метрик системи моніторингу Prometheus. Кожен параметр представлено у вигляді окремої часової метрики, яка містить числове значення та набір службових міток (labels), що дозволяють ідентифікувати конкретний проект, тип задачі або обчислювальний вузол. Такий підхід забезпечує можливість детального аналізу продуктивності системи та побудови гнучких механізмів фільтрації й агрегації статистичних даних.

Для забезпечення доступу до метрик у системі було реалізовано окремий HTTP-інтерфейс /metrics, який працює через веб-сервер Nginx. Конфігурація сервера передбачає використання механізму FastCGI та служби fcgiwrap, що дозволяє виконувати Python-exporter як CGI-застосунок. Під час звернення до порту 9116 веб-сервер автоматично запускає скрипт boinc_exporter.py, який генерує актуальний набір метрик та передає його клієнту у текстовому форматі Prometheus.

Приклад файлу конфігурації nginx:

```
server {  
    listen 9116;  
    server_name _;  
  
    location /metrics {  
        gzip off;  
        include fastcgi_params;  
  
        fastcgi_param SCRIPT_FILENAME /usr/local/bin/boinc_exporter.py;  
        fastcgi_param DOCUMENT_ROOT /usr/local/bin;  
        fastcgi_param SCRIPT_NAME /boinc_exporter.py;  
        fastcgi_pass unix:/var/run/fcgiwrap.socket;  
    }  
}
```

У конфігурації веб-сервера визначено окремий блок location /metrics, який виконує маршрутизацію HTTP-запитів до Python-exporter. Для забезпечення сумісності із FastCGI використовуються службові параметри SCRIPT_FILENAME, DOCUMENT_ROOT та SCRIPT_NAME, що визначають місце розташування виконуваного скрипта та параметри його запуску. Передача

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		69

запитів здійснюється через Unix-сокет /var/run/cgiwrap.socket, що дозволяє зменшити накладні витрати у порівнянні з використанням TCP-з'єднань.

Після отримання метрик система моніторингу Prometheus виконує їх періодичне опитування та збереження у часовій базі даних. Надалі накопичені дані передаються до системи візуалізації Grafana, де реалізовано спеціалізовану інформаційну панель (dashboard) для контролю роботи BOINC-інфраструктури. На інформаційній панелі відображаються графіки завантаження процесора, використання оперативної пам'яті, кількість активних задач, швидкість виконання work units, стан мережевих підключень та активність віртуальних машин.

Особливістю реалізованої системи є можливість деталізованого аналізу задач за окремими науковими проектами. Grafana дозволяє відображати статистику для кожного BOINC-проекту окремо, включаючи кількість активних задач, середній час виконання, відсоток успішно завершених обчислень та кількість помилок. Це дозволяє оцінювати ефективність використання обчислювальних ресурсів та визначати найбільш навантажені компоненти системи.

Крім функцій візуалізації, підсистема моніторингу реалізує механізми виявлення аварійних ситуацій. У випадку перевищення допустимого рівня навантаження, зависання віртуальної машини, припинення роботи BOINC-клієнта або виникнення помилок передачі результатів система автоматично формує відповідний запис у журналі подій та може ініціювати процедуру сповіщення адміністратора. Завдяки цьому забезпечується своєчасне виявлення несправностей та підвищується загальна надійність роботи розподіленої обчислювальної інфраструктури.

Важливою складовою алгоритму є обробка помилок. Система постійно аналізує стан виконання задач і контролює стабільність мережевого підключення. У випадку втрати зв'язку із сервером проекту активується механізм повторної спроби підключення. Якщо помилка виникає під час виконання задачі,

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		70

Task ID	Task Name	Task Description	Task Status	Task Priority	Task Assignee	Task Due Date	Task Progress	Task Comments
1	Task 1	Task 1 Description	Completed	High	John Doe	2023-10-26	100%	Task 1 completed successfully.
2	Task 2	Task 2 Description	In Progress	Medium	Jane Smith	2023-10-27	50%	Task 2 is currently in progress.
3	Task 3	Task 3 Description	Pending	Low	Mike Johnson	2023-10-28	0%	Task 3 is pending approval.
4	Task 4	Task 4 Description	Completed	High	Sarah Lee	2023-10-29	100%	Task 4 completed successfully.
5	Task 5	Task 5 Description	In Progress	Medium	David Kim	2023-10-30	75%	Task 5 is currently in progress.
6	Task 6	Task 6 Description	Pending	Low	Emily White	2023-10-31	0%	Task 6 is pending approval.
7	Task 7	Task 7 Description	Completed	High	Chris Brown	2023-11-01	100%	Task 7 completed successfully.
8	Task 8	Task 8 Description	In Progress	Medium	Alex Green	2023-11-02	60%	Task 8 is currently in progress.
9	Task 9	Task 9 Description	Pending	Low	Mia Black	2023-11-03	0%	Task 9 is pending approval.
10	Task 10	Task 10 Description	Completed	High	Noah Grey	2023-11-04	100%	Task 10 completed successfully.

Окремим елементом алгоритму є підсистема оновлення. Вона періодично перевіряє наявність нових версій компонентів системи або змін конфігурації. У разі виявлення оновлення виконується процедура зупинки окремих служб, заміни файлів та повторної ініціалізації модулів без повного перезапуску серверу. Такий підхід дозволяє мінімізувати простій системи та забезпечує безперервність обчислювального процесу.

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

4.2 Захист розробленого програмного забезпечення

Одним із важливих аспектів функціонування розподілених обчислювальних систем є забезпечення захисту переданих даних, автентичності вузлів та цілісності програмного забезпечення. Оскільки платформа BOINC функціонує у глобальному мережевому середовищі та передбачає взаємодію великої кількості клієнтських вузлів із центральним сервером через мережу Інтернет, особливого значення набуває використання сучасних механізмів криптографічного захисту інформації.

Для забезпечення безпечної передачі даних у системі використовуються протоколи SSL/TLS (Secure Sockets Layer / Transport Layer Security). Дані технології забезпечують шифрування мережевого трафіку між клієнтськими вузлами, веб-серверами та сервісами моніторингу, що дозволяє запобігти перехопленню або модифікації інформації під час передачі мережею.

Основним призначенням SSL/TLS є створення захищеного каналу зв'язку між сторонами. Під час встановлення з'єднання виконується процедура криптографічного “рукоостискання” (handshake), у процесі якої сторони узгоджують алгоритми шифрування, перевіряють сертифікати автентичності та генерують сеансові ключі для подальшого обміну даними. Після завершення цього процесу весь трафік передається у зашифрованому вигляді.

У рамках роботи BOINC використання TLS-з'єднань дозволяє забезпечити:

- захист облікових даних користувачів
- безпечне завантаження work units
- захищену передачу результатів обчислень
- захист службових XML-конфігурацій
- підтвердження автентичності серверів проєктів
- захист від атак типу “man-in-the-middle”

Особливо важливим використання TLS є для наукових проєктів CERN та LHC@Home, де клієнтські вузли регулярно взаємодіють із центральною

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		72

інфраструктурою проєкту та отримують великі обсяги обчислювальних даних. Шифрування дозволяє гарантувати, що завантажені задачі не були змінені сторонніми особами, а результати обчислень надходять саме до офіційного сервера проєкту.

У системах, що використовують віртуалізацію, таких як ATLAS@Home або OpenIFS@Home, захист мережевого трафіку також є критично важливим через передачу образів віртуальних машин, конфігураційних файлів та проміжних результатів моделювання. Використання TLS дозволяє уникнути компрометації даних та забезпечити цілісність середовища виконання задач.

Окремо механізми SSL/TLS були використані у підсистемі моніторингу. Під час роботи Grafana, Prometheus та exporter-компонентів можливе передавання службової інформації про навантаження вузлів, активні задачі та стан інфраструктури. Захищене HTTPS-з'єднання дозволяє уникнути несанкціонованого доступу до метрик, а також забезпечує безпечне адміністрування системи моніторингу через веб-інтерфейс.

Для реалізації захищеного доступу у системі застосовувався веб-сервер Nginx, який підтримує сучасні механізми TLS-шифрування та роботу із цифровими сертифікатами. Nginx виконує функції термінації TLS-з'єднань, забезпечує обробку HTTPS-запитів та дозволяє централізовано керувати параметрами безпеки. Додатково можуть використовуватись механізми автоматичного оновлення сертифікатів та примусового перенаправлення HTTP-запитів на HTTPS.

Використання SSL/TLS також позитивно впливає на загальну надійність інфраструктури, оскільки сучасні криптографічні алгоритми дозволяють забезпечити:

- конфіденційність даних
- цілісність переданої інформації
- автентичність серверів та клієнтів
- захист від підміни пакетів

- безпечну роботу у відкритих мережах

Таким чином, використання механізмів SSL/TLS у розробленій системі є важливою складовою забезпечення кібербезпеки розподіленої обчислювальної інфраструктури. Реалізовані засоби захисту дозволяють гарантувати безпечний обмін даними між компонентами системи, забезпечують стабільність функціонування сервісів та підвищують рівень довіри до результатів виконаних обчислень.

К6ПЗ_2026

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		74

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Процес впровадження системи передбачає підготовку серверної частини, встановлення клієнтського програмного забезпечення, налаштування механізмів моніторингу та перевірку працездатності інфраструктури під навантаженням. Реалізована система дозволяє організувати обчислення як на окремих вузлах, так і у масштабованому кластерному середовищі із залученням великої кількості клієнтів.

На першому етапі впровадження було виконано встановлення та налаштування BOINC-клієнта. Після запуску клієнт автоматично підключається до вибраних наукових проєктів, отримує список доступних задач та виконує їх у фоновому режимі із використанням доступних ресурсів процесора та оперативної пам'яті.

На рисунку 5.1 відображено головне вікно BOINC TUI, у якому відображається список активних задач, статус їх виконання, назви підключених проєктів, прогрес обчислень та поточний стан клієнта.

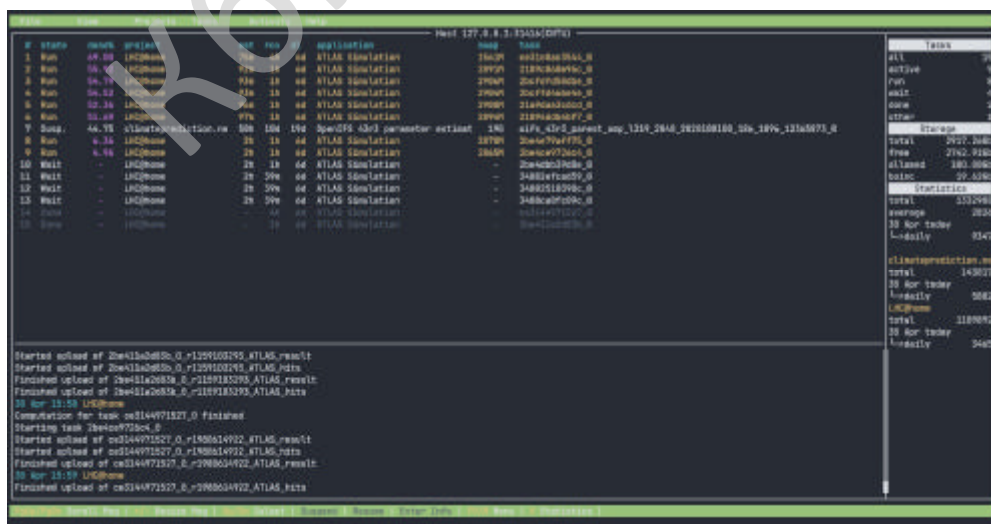


Рисунок 5.1 - Головне вікно

Однією із важливих особливостей системи є можливість збору статистики виконання обчислень. Після завершення задач сервер проєкту нараховує користувачеві обчислювальні кредити, які використовуються для оцінки внеску вузла у виконання розподілених обчислень.

На рисунку 5.2 наведено статистичну інформацію щодо виконаних задач та отриманих кредитів у різних BOINC-проєктах. Представлені дані дозволяють оцінити продуктивність вузла, інтенсивність виконання задач та ефективність використання обчислювальних ресурсів.

Statistics [host]			
	Summary	climatepre	LHC@home
30 Apr	9346	5881	3465
29 Apr	0	0	0
23 Apr	11763	11763	0
22 Apr	11763	11763	0
21 Apr	5881	5881	0
20 Apr	5874	5874	0
16 Apr	5874	5874	0
15 Apr	0	0	0
14 Apr	4286	0	4286
13 Apr	11568	0	11568
9 Apr	0	0	0
8 Apr	0	0	0
7 Apr	0	0	0
6 Apr	0	0	0
2 Apr	0	0	0
1 Apr	0	0	0

Рисунок 5.2 - Статистична інформація отриманих кредитів

Для детального аналізу параметрів окремих задач використовується режим перегляду службової інформації Task Info Raw View. У даному режимі відображаються технічні параметри задачі, включаючи ідентифікатор work unit, параметри запуску, використані ресурси, інформацію про checkpoint, статус виконання та журнали роботи.

На рисунку 5.3 показано приклад детальної інформації про задачу BOINC-проекту. Аналіз таких параметрів дозволяє виконувати діагностику помилок, контролювати процес виконання та оцінювати стабільність роботи клієнта.

```
Task Info Raw View

PROJECT NAME      LHC@home
APP NAME          ATLAS Simulation

name              2189c860e96c_0
wu_name           2189c860e96c
platform          x86_64-pc-linux-gnu
version_num       301
plan_class        native_mt
project_url       https://lhcathome.cern.ch/lhcathome/
final_cpu_time    0.000000 (0s)
final_elapsed_time 0.000000 (0s)
exit_status       0
state             2 (downloaded)
report_deadline   1778152059.000000 (Thu May  7 14:07:39 2026)
received_time     1777547260.142614 (Thu Apr 30 14:07:40 2026)
estimated_cpu_time_remaining 5495.257115 (1h 31m 35s)

active_task_state 1 (executing)
app_version_num   301
slot              5
pid               161897
scheduler_state   2 (scheduled)
checkpoint_cpu_time 27659.710000 (7h 40m 59s)
fraction_done     0.564363 (56.44 %)
current_cpu_time  27659.710000 (7h 40m 59s)
elapsed_time      7119.043962 (1h 58m 39s)
swap_size         3031482368.000000 (2.823GB)
working_set_size  2382537728.000000 (2.14GB)
```

Рисунок 5.3 - Вікно з детальним описом задачі

Для забезпечення контролю стану інфраструктури було реалізовано систему моніторингу на базі Prometheus та Grafana. Збір метрик здійснюється через exporter, який отримує дані від BOINC-клієнта та формує їх у форматі Prometheus metrics.

У процесі впровадження було налаштовано веб-сервер Nginx та механізм FastCGI для забезпечення доступу до endpoint /metrics. Отримані дані передаються до Prometheus, після чого візуалізуються у Grafana у вигляді інтерактивних dashboard-панелей.

Панель Grafana, яка відображає стан обчислювальних вузлів, активність задач, навантаження на процесор, використання пам'яті, кількість завершених задач та інші показники роботи системи.

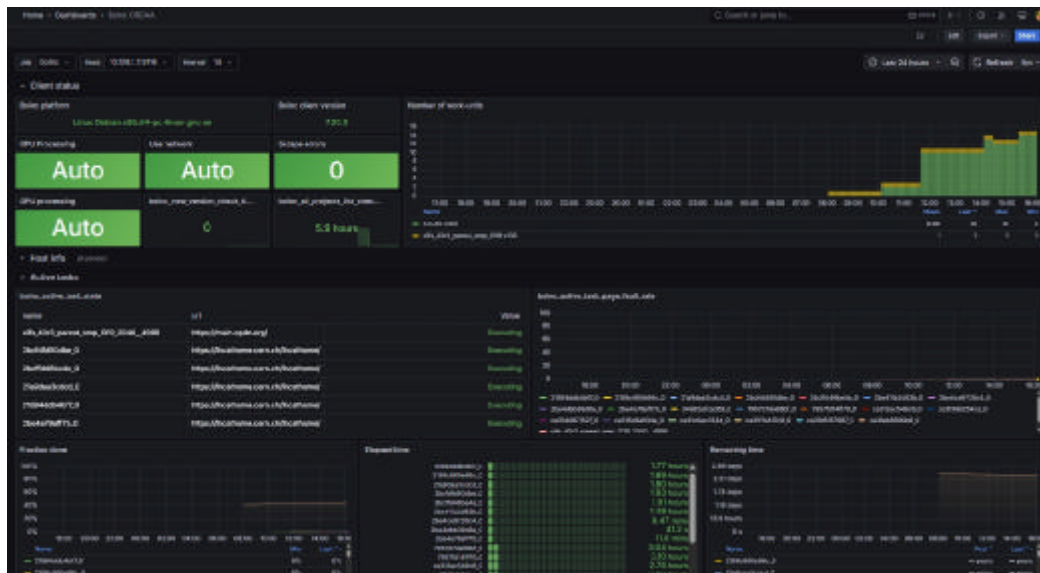


Рисунок 5.4 - Панель візуалізації у Grafana

У процесі розробки також було реалізовано механізми захисту програмного забезпечення та авторського права. Система використовує захищені протоколи передачі даних, а програмні компоненти мають власну структуру конфігурації та організації доступу.

На рисунку 5.5 наведено приклад відображення авторської інформації програмного забезпечення.

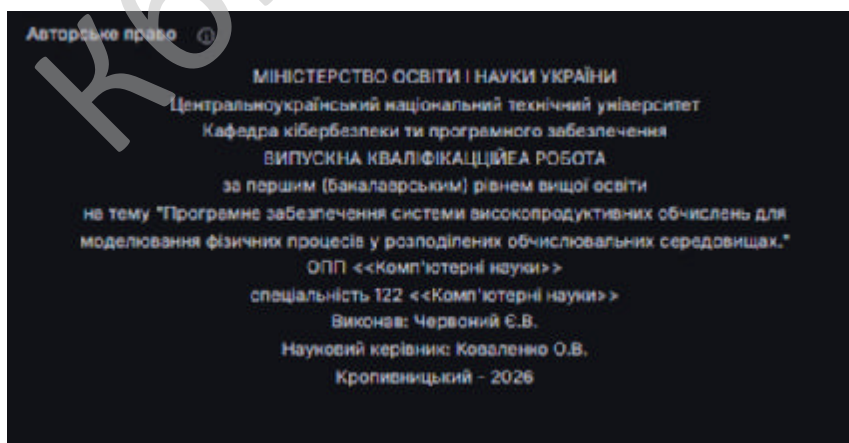


Рисунок 5.5 - Авторське право

6 ОСНОВНІ ВИСНОВКИ

У результаті виконання випускної кваліфікаційної роботи було проведено комплексне дослідження принципів побудови систем високопродуктивних розподілених обчислень та реалізовано програмно-апаратну інфраструктуру на базі платформи BOINC для моделювання фізичних процесів у розподіленому середовищі. У процесі роботи було розглянуто сучасні підходи до організації НРС-систем, методи розподілу обчислювального навантаження між вузлами, особливості виконання обчислювальних задач у середовищах із великою кількістю незалежних клієнтів, а також засоби моніторингу та контролю роботи інфраструктури.

Проведений аналіз сучасного стану високопродуктивних обчислень показав, що традиційні підходи, засновані виключно на використанні локальних серверів або окремих суперкомп'ютерних систем, мають значні обмеження щодо вартості масштабування, енергоспоживання та доступності ресурсів. У зв'язку з цим особливого значення набувають розподілені обчислювальні середовища, які дозволяють об'єднувати велику кількість незалежних вузлів у єдину систему для спільного виконання задач. Такий підхід забезпечує можливість ефективного використання наявних ресурсів та значно знижує витрати на побудову високопродуктивної обчислювальної інфраструктури.

У ході роботи було детально досліджено платформу BOINC як одну із найбільш поширених систем добровільних розподілених обчислень. Визначено її основні архітектурні особливості, принципи роботи серверної та клієнтської частин, алгоритми розподілу задач та механізми перевірки достовірності результатів. Встановлено, що BOINC є ефективним інструментом для реалізації embarrassingly parallel задач, у яких окремі підзадачі можуть виконуватись незалежно одна від одної без необхідності постійної синхронізації між вузлами.

У рамках дослідження було проаналізовано ряд сучасних наукових

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		79

проектів, які використовують платформу BOINC для проведення масштабних обчислень. Особливу увагу приділено проектам LHC@Home, ATLAS@Home, SixTrack та Climateprediction.net. Дослідження показало, що використання добровільних розподілених обчислень дозволяє виконувати складні наукові задачі глобального масштабу із залученням великої кількості персональних комп'ютерів користувачів у всьому світі.

Під час аналізу проекту LHC@Home було встановлено, що система дозволяє моделювати фізичні процеси, пов'язані із роботою Великого адронного колайдера, використовуючи розподілені ресурси тисяч вузлів. Окремо було досліджено експеримент ATLAS, який є одним із найбільших наукових експериментів CERN та використовується для дослідження фундаментальних частинок і взаємодій між ними. Проаналізовано особливості використання віртуалізації у задачах ATLAS@Home, де виконання задач здійснюється у спеціалізованих віртуальних середовищах для забезпечення сумісності програмного забезпечення та ізоляції обчислювального процесу.

Також було розглянуто програмний комплекс SixTrack, призначений для моделювання траєкторій руху заряджених частинок у прискорювачах. Встановлено, що даний програмний комплекс є прикладом високопродуктивних наукових обчислень, які ефективно масштабуються у розподіленому середовищі. Дослідження проекту Climateprediction.net дозволило проаналізувати використання BOINC у задачах кліматичного моделювання та прогнозування погодних процесів. Окремо розглянуто використання OpenIFS - дослідницької версії моделі ECMWF Integrated Forecasting System, яка забезпечує моделювання атмосферних процесів із використанням великих ансамблів прогнозів.

Практична частина роботи включала побудову та налаштування власної обчислювальної інфраструктури на базі Linux та BOINC. Було встановлено та налаштовано клієнтське програмне забезпечення, реалізовано механізми автоматичного отримання та виконання задач, а також проведено тестування роботи системи у режимі тривалих обчислень. Використання Linux-середовища

забезпечило стабільність роботи, ефективне використання ресурсів та можливість інтеграції із сучасними засобами моніторингу.

У процесі реалізації системи було досліджено механізми виконання задач як безпосередньо на хост-системі, так і у середовищі Oracle VM VirtualBox. Використання віртуалізації дозволило забезпечити підтримку задач, які потребують спеціалізованого програмного оточення, а також підвищити рівень ізоляції та безпеки виконання обчислювальних процесів. Було встановлено, що застосування віртуальних машин є важливим компонентом сучасних наукових розподілених платформ, оскільки дозволяє стандартизувати середовище виконання незалежно від конфігурації клієнтського вузла.

Окрему увагу у роботі було приділено питанням моніторингу та спостережуваності інфраструктури. У результаті дослідження встановлено, що базові засоби BOINC не забезпечують достатнього рівня деталізації та централізованого контролю стану системи, тому виникла необхідність використання зовнішніх систем моніторингу.

Для вирішення даної задачі було розроблено власний exporter мовою Python, який забезпечує збір метрик із BOINC-клієнта та операційної системи. Exporter реалізує формування метрик у форматі Prometheus та дозволяє отримувати інформацію про стан задач, навантаження на процесор, використання оперативної пам'яті, активність віртуальних машин, кількість виконаних задач та інші параметри функціонування системи.

Для забезпечення доступу до метрик було використано веб-сервер Nginx та механізм FastCGI через службу fcgiwrap. Реалізовано окремий endpoint /metrics, через який система Prometheus здійснює опитування exporter та збір статистичної інформації. Використання Nginx дозволило забезпечити стабільну роботу HTTP-інтерфейсу, маршрутизацію запитів та інтеграцію із CGI/FastCGI-застосунками.

У рамках роботи також було досліджено платформу Prometheus як систему збору та збереження часових рядів метрик. Встановлено, що Prometheus є

ефективним інструментом для моніторингу динамічних інфраструктур та забезпечує зручні механізми агрегації, аналізу та обробки даних. Для візуалізації зібраних метрик було використано Grafana, яка дозволила реалізувати інтерактивні dashboard-панелі для аналізу стану інфраструктури у режимі реального часу.

Розроблені панелі Grafana забезпечують відображення інформації про продуктивність вузлів, кількість активних задач, навантаження на CPU та RAM, швидкість виконання work units, стан підключення клієнтів та інші параметри роботи системи. Додатково було налаштовано механізми alerting, які дозволяють автоматично формувати повідомлення у випадку перевищення критичних значень або виникнення аварійних ситуацій.

У роботі також було розглянуто питання інформаційної безпеки та захисту програмного забезпечення. Проаналізовано використання криптографічних протоколів SSL/TLS для забезпечення захищеної передачі даних між клієнтами та серверами проєктів. Визначено, що використання шифрування є необхідною умовою забезпечення конфіденційності даних, захисту від перехоплення інформації та перевірки автентичності серверів.

Додатково було досліджено механізми перевірки достовірності результатів у BOINC, які базуються на принципах дублювання задач та консенсусної валідації. Такий підхід дозволяє мінімізувати ризик отримання некоректних результатів у середовищах із ненадійними або частково контрольованими вузлами.

Під час тестування системи було підтверджено стабільність роботи клієнтської частини BOINC, працездатність механізмів моніторингу та коректність функціонування системи збору метрик. Проведені експерименти показали ефективність використання розподілених обчислень для виконання ресурсоємних задач та можливість подальшого масштабування інфраструктури шляхом підключення нових вузлів.

Практична цінність отриманих результатів полягає у створенні

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		82

повноцінного навчально-дослідницького середовища для вивчення високопродуктивних обчислень, систем моніторингу, розподілених платформ та сучасних підходів DevOps. Розроблена інфраструктура може використовуватись як основа для подальших наукових досліджень, тестування алгоритмів розподілу навантаження, аналізу ефективності використання ресурсів та побудови масштабованих НРС-систем.

Отримані результати підтверджують перспективність використання платформ добровільних розподілених обчислень для вирішення сучасних наукових і прикладних задач. Реалізована система демонструє можливість побудови ефективної та масштабованої інфраструктури високопродуктивних обчислень із використанням відкритих програмних технологій, сучасних засобів моніторингу та розподіленої архітектури.

КБПЗ_2026

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Друк.	№ докум.	Підпис	Дата		83

СПИСОК ЛІТЕРАТУРИ

1. BOINC [Електронний ресурс]. - Режим доступу:
<https://boinc.berkeley.edu/>
2. BOINC - Projects [Електронний ресурс]. - Режим доступу:
<https://boinc.berkeley.edu/projects.php>
3. BOINC - User manual [Електронний ресурс]. - Режим доступу:
<https://github.com/BOINC/boinc/wiki/User-manual>
4. BOINC - Web Resources [Електронний ресурс]. - Режим доступу:
<https://github.com/BOINC/boinc/wiki/Web-Resources>
5. LHC@home [Електронний ресурс]. - Режим доступу:
<https://lhathome.web.cern.ch/>
6. SixTrack and numerical simulations [Електронний ресурс]. - Режим доступу:
<https://lhathome.web.cern.ch/projects/sixtrack/sixtrack-and-numerical-simulations>
7. CERN [Електронний ресурс]. - Режим доступу: <https://home.cern/>
8. ATLAS Experiment [Електронний ресурс]. - Режим доступу:
<https://atlas.cern/>
9. ATLAS@Home [Електронний ресурс]. - Режим доступу:
<https://lhathome.web.cern.ch/projects/atlas>
10. Climateprediction.net [Електронний ресурс]. - Режим доступу:
<https://www.climateprediction.net/>
11. OpenIFS [Електронний ресурс]. - Режим доступу:
<https://www.ecmwf.int/en/research/projects/openifs>
12. ECMWF [Електронний ресурс]. - Режим доступу:
<https://www.ecmwf.int/>
13. Grafana [Електронний ресурс]. - Режим доступу:
<https://grafana.com/oss/grafana/>

14. Grafana Technical documentation [Електронний ресурс]. - Режим доступу: <https://grafana.com/docs/>
15. Prometheus - Docs [Електронний ресурс]. - Режим доступу: <https://prometheus.io/docs/introduction/overview/>
16. Grafana support for Prometheus [Електронний ресурс]. - Режим доступу: <https://prometheus.io/docs/visualization/grafana/>
17. Oracle VM VirtualBox - Technical documentation [Електронний ресурс]. - Режим доступу: https://www.virtualbox.org/wiki/Technical_documentation
18. Debian - Documentation [Електронний ресурс]. - Режим доступу: <https://www.debian.org/doc/>
19. LHC@home - How it works [Електронний ресурс]. - Режим доступу: <https://lhcathe.web.cern.ch/about/how-it-works>
20. Barney B. Introduction to Parallel Computing [Електронний ресурс]. - Режим доступу: <https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial>
21. Grama A. Introduction to Parallel Computing / A. Grama, A. Gupta, G. Karypis, V. Kumar. – 2nd ed. – Boston : Addison-Wesley, 2003.
22. Foster I. The Grid: Blueprint for a New Computing Infrastructure / I. Foster, C. Kesselman. – San Francisco : Morgan Kaufmann Publishers, 2004.
23. Sterling T. High Performance Computing: Modern Systems and Practices / T. Sterling, M. Anderson, M. Brodowicz. - Cambridge : Morgan Kaufmann, 2017.
24. Gropp W. Beowulf Cluster Computing with Linux / W. Gropp, T. Sterling, P. Becker, D. Savarese. – 2nd ed. – Cambridge : MIT Press, 2003.
25. Rauber T. Parallel Programming: for Multicore and Cluster Systems / T. Rauber, G. Rünger. – Berlin : Springer, 2013.
26. MPI Forum. MPI: A Message-Passing Interface Standard [Електронний ресурс]. – Режим доступу: <https://www.mpi-forum.org/docs/>
27. OpenMP Architecture Review Board. OpenMP Application Programming Interface [Електронний ресурс]. – Режим доступу:

<https://www.openmp.org/specifications/>

28. Wilkinson B. Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers / B. Wilkinson, M. Allen. – 2nd ed. – New Jersey : Prentice Hall, 2005.

29. Pacheco P. An Introduction to Parallel Programming / P. Pacheco. – Burlington : Morgan Kaufmann, 2011.

30. Tanenbaum A. Distributed Systems: Principles and Paradigms / A. Tanenbaum, M. van Steen. – 2nd ed. – Upper Saddle River : Pearson Education, 2007.

31. Buyya R. Grid Computing: Making the Global Infrastructure a Reality / R. Buyya, K. Krauter. – Hoboken : Wiley, 2009.

32. Schwiegelshohn U. Parallel Computing / U. Schwiegelshohn. – New York : Springer, 2019.

33. Stallings W. Cryptography and Network Security: Principles and Practice / W. Stallings. – 7th ed. – Boston : Pearson, 2017.

34. Silberschatz A. Operating System Concepts / A. Silberschatz, P. Galvin, G. Gagne. – 10th ed. – Hoboken : Wiley, 2018.

35. Patterson D. Parallel Computer Architecture: A Hardware/Software Approach / D. Patterson, J. Hennessy. – San Francisco : Morgan Kaufmann, 1998.

36. Python Documentation [Електронний ресурс]. – Режим доступу: <https://docs.python.org/3/>

37. Nginx Documentation [Електронний ресурс]. – Режим доступу: <https://nginx.org/en/docs/>

38. FastCGI Specification [Електронний ресурс]. – Режим доступу: <https://fastcgi-archives.github.io/>

39. OpenSSL Documentation [Електронний ресурс]. – Режим доступу: <https://www.openssl.org/docs/>

40. Скакун П.П., Коваленко А.С., Коваленко О.В., Ткачук Р.О. «Комплексний аналіз підходів до управління ІТ-проєктами в Україні» The scientific heritage (Budapest, Hungary), 2025, Volume 166, Pages 88-92. Режим

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		86

доступу: <https://doi.org/10.5281/zenodo.16763561> (Закордонне фахове видання, країна ОЕСР).

41. Вінтенко Б.Ю., Миронець І.В., Смірнов О.А. Коваленко О.В., Смірнов С.А., Буравченко К.О., Якименко Н.М. «Дослідження інформаційного забезпечення та технологічних регламентів процесів керування критичною інфраструктурою енергоблоку АЕС з реактором типу ВВЕР-1000». Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка». 2024. № 1(25), С. 253-278. Режим доступу: <https://doi.org/10.28925/2663-4023.2024.25.253278> <https://csecurity.kubg.edu.ua/index.php/journal/article/view/655> (Фахове видання. Категорія «Б»)

42. Смірнова Т.В., Моторін Ю.Ю., Буравченко К.О., Бочуля Т.В., Коваленко О.В. «Вибір оптимальної технології побудови хмарної інформаційно-комунікаційної системи автоматизації виробничих процесів» Вимірювальна та обчислювальна техніка в технологічних процесах, № 1 (2022). С. 15-26. 2022. Режим доступу: <http://vottp.khmnpu.edu.ua/index.php/vottp/article/view/30/36> (Фахове видання. Категорія «Б»)

43. Вінтенко, Б.Ю., Миронець, І.В., Смірнов, О.А., Коваленко, О.В., Усік, П.С., Буравченко, К.О., Лисенко, І.А. «Логіко-структурна модель комп'ютерно-орієнтованої процедури системи підтримки оперативного персоналу АЕС». Кібербезпека: освіта, наука, техніка. 2025. Том 2 № 30. С. 413-427, 2025. <https://doi.org/10.28925/2663-4023.2025.30.984> Режим доступу: <https://www.csecurity.kubg.edu.ua/index.php/journal/article/view/984> (Фахове видання. Категорія «Б»)

44. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». Системи управління, навігації та зв'язку, 2023, вип. 2(72), С. 170-178. Режим доступу: <https://doi.org/10.26906/SUNZ.2023.2.170> (Фахове видання. Категорія «Б»)

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Дрк.	№ докум.	Підпис	Дата		87

45. Вінтенко Б.Ю., Смірнов О.А., Миронець І.В., Смірнова Т.В., Коваленко О.В., Мацуй А.М. «Модель шляхів отримання вхідних даних комп'ютерної інтелектуальної системи підтримки оперативного персоналу АЕС». Центральнотраїнський науковий вісник. Технічні науки. 2025. Вип. 11(42), ч. II. С.52-62. Режим доступу: [https://mapiea.kntu.kr.ua/pdf/11\(42\)_II/11\(42\)_II_2025.pdf](https://mapiea.kntu.kr.ua/pdf/11(42)_II/11(42)_II_2025.pdf) (Фахове видання. Категорія «Б»)

46. Khudov H., Baranik O., Kovalenko O., Yakovenko Y., Chahan Y. «The Information Technology for Determining Vehicle Route Based on Ant Colony Algorithms» International Journal of Emerging Technology and Advanced Engineering, 2022, 12(12), Pages 117-128. Режим доступу: <https://www.scopus.com/record/display.uri?eid=2-s2.0-5130069202&origin=resultslist> (Scopus).

47. Serhii Pohasii, Serhii Yevseiev, Oleksandr Zhuchenko, Oleksandr Milov, Volodymyr Lysechko, Oleksandr Kovalenko, Maryna Kostiak, Andrii Volkov, Aleksandr Lezik, Vitalii Susukailo «Development of crypto-code constructs based on LDPC codes» Eastern-European Journal of Enterprise Technologies 2/9 (116), 2022, Pages 44-59. Режим доступу: <https://www.scopus.com/record/display.uri?eid=2-s2.0-85130069202&origin=resultslist> (Scopus).

48. Hennadii Khudov, Oleksandr Kostianets, Oleksandr Kovalenko, Oleh Maslenko, Yuriy Solomonenko «Using softwaredefined radio receivers for determining the coordinates of low-visible aerial objects» Eastern-European Journal of Enterprise Technologies Vol. 5 No. 9 (124), 2023, Pages 61-73. Режим доступу: <https://www.scopus.com/record/display.uri?eid=2-s2.0-5172343893&origin=resultslist> (Scopus).

49. Hennadii Khudov, Volodymyr Bashynskyi, Oleksandr Kovalenko, Kristina Tahyan, Oleksii Fakadii « The methods for improving the quality of detection of inconspicuous aerial objects through the use of external radiation sources» International Journal of Emerging Technology and Advanced Engineering, 2023, Volume 13, Issue 3, Pages 91-100. Режим доступу: https://doi.org/10.46338/ijetae0323_09 (Закордонне

фахове видання).

50. Kovalenko O., Khudov H., Mynko P., Ikhsanov S., Diakonov O., Solomonenko Y., Drob Y., Kharun O., Cherkashyn S., Serdiuk O. «Development A Method For Determining The Coordinates Of Air Objects By Radars With The Additional Use Of Multilateration Technology» Eastern-European Journal of Enterprise Technologies Volume 5, 2021, Pages 6-16. Режим доступу: <https://www.scopus.com/record/display.uri?eid=2-s2.0-85119667497&origin=resultslist> (Scopus).

К6ПЗ_2026

					ВКРБ-122.26.0034.00.00.ПЗ	Арк.
Вим.	Друк.	№ докум.	Підпис	Дата		89