

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління
(повна назва)

Кафедра _____ електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти _____ другий (магістерський)

Методи аналізу швидкості паралельної обробки
інформації у GRID-системах

(тема)

Виконав:
студент _____ II курсу, групи _____ СПМ-21-1
Костенко О.С.
(прізвище, ініціали)

Спеціальність _____
123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма _____
Системне програмування
(повна назва освітньої програми)

Керівник: _____ Доц. Ільїна І.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2022 р.

Факультет _____ комп'ютерної інженерії та управління _____
Кафедра _____ електронних обчислювальних машин _____
Рівень вищої освіти _____ другий (магістерський) _____
Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)
Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)
Освітня програма _____ Системне програмування _____
(повна назва)

Харківський національний університет радіоелектроніки

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту _____ Костенку Олександр Сергійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Методи аналізу швидкості паралельної обробки інформації у GRID-системах

затверджена наказом по університету від “ 07 ” листопада 2021 р. № 1454 Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 13 грудня 2022 р.

3. Вхідні дані до роботи 1) дистрибутив PelicanNPC; 2) інфраструктура BOINC

4. Перелік питань, що потрібно опрацювати у роботі _____

1) аналіз залежності прискорення обчислень від архітектурних особливостей

2) розробка програмного комплексу для визначення прискорення обчислень

3) дослідження процесу навчання нейронних мереж із застосуванням GRID-систем

4) дослідження впливу багатоядерності обчислювальних вузлів на прискорення

Обчислень у GRID-системах

5) розробка узагальненого методу оцінки прискорення обчислень у GRID-системах

6) розробка імітаційної моделі GRID-систем для аналізу та імітації їх роботи

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) _____

Слайд-презентація – 26. _____

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд теоретичних відомостей про GRID та розподілені обчислення	08.11.22-14.11.22	
2	Вибір методики дослідження	15.11.22-17.11.22	
3	Дослідження методів аналізу швидкості паралельної обробки інформації у GRID-системах	18.11.22-20.11.22	
4	Розробка архітектури та моделі GRID-систем	21.11.22-23.11.22	
5	Розгортання обчислювального кластеру	24.11.22-26.11.22	
6	Проведення експерименту	27.11.22-30.11.22	
7	Оформлення матеріалів кваліфікаційної роботи	01.12.22-06.12.22	
8	Подання кваліфікаційної роботи керівникові та її попередній захист	07.12.22-08.12.22	
9	Подання кваліфікаційної роботи на рецензування	09.12.22-12.12.22	

Дата видачі завдання 07 листопада 2022 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

Доц. Ільїна І.В. _____
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 93 сторінок, 38 рисунків, 5 таблиць, 2 додатки, 21 джерел.

РОЗРОДІЛЕНІ ОБЧИСЛЕННЯ, ПРОЦЕСОР, НЕЙРОННІ МЕРЕЖІ , ВУЗЛИ, ОБРОБКА ДАНИХ, ІНФРАСТРУКТУРА.

Метою кваліфікаційної роботи є заповнення прогалин в аналізі та оцінці факторів, що впливають на швидкість паралельної обробки інформації у географічно розподілених інформаційних системах.

У ході виконання кваліфікаційної роботи було визначено залежності швидкості обчислень у GRID-системах від кількості обчислювальних вузлів, що дозволяє більш точно оцінювати тимчасові та ресурсні витрати для вирішення завдань у GRID-системах. Розроблено методи оцінки оптимальних у сенсі прискорення обчислень кількості обчислювальних вузлів у GRID-системах, враховуючі як параметри GRID-системи, так і параметри розв'язуваних завдань. Розроблено методи, що дозволяють проводити аналіз часу обчислень та прискорення обчислень у GRID-системах, завдяки чому можливе більш точне визначення необхідного рішення поставленої задачі часу за відомих параметрів системи.

ABSTRACT

Master's thesis: 93 pages, 38 figures, 5 tables, 2 appendices, 21 sources.

DISTRIBUTED COMPUTING, PROCESSOR, NEURAL NETWORKS,
NODES, DATA PROCESSING, INFRASTRUCTURE.

The purpose of the qualification work is to fill the gaps in the analysis and assessment of factors affecting the speed of parallel processing of information in geographically distributed information systems.

In the course of the qualification work, the dependence of the calculation speed in GRID-systems on the number of computing nodes was determined, which allows for a more accurate assessment of time and resource costs for solving tasks in GRID-systems. Methods for estimating the number of computing nodes in GRID systems that are optimal in the sense of speeding up calculations have been developed, taking into account both the parameters of the GRID system and the parameters of the tasks to be solved. Methods have been developed that allow for the analysis of calculation time and acceleration of calculations in GRID systems, thanks to which it is possible to more accurately determine the necessary solution to the given time problem under known system parameters.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
1 АНАЛІЗ ПРОБЛЕМАТИКИ В ОБЛАСТІ ОРГАНІЗАЦІЇ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ В GRID -СИСТЕМАХ. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ.....	11
1.1 Стан досліджень розробок у галузі GRID-систем. Шляхи розвитку розподілених систем.....	11
1.2 Особливості паралельних обчислень у GRID-системах	13
1.3 Класифікація архітектур GRID-систем.....	14
1.3.1 Класифікація GRID за типом обладнання на характеристиками обчислювальних вузлів.....	18
1.3.2 Класифікація за типом доступу до GRID-систем	20
1.4 Стек протоколів GRID-систем.....	21
1.5 Постановка задачі.....	24
2 ДОСЛІДЖЕННЯ ТА ВИКОРИСТАННЯ МЕТОДІВ АНАЛІЗУ ШВИДКОСТІ ПАРАЛЕЛЬНОЇ ОБРОБКИ ІНФОРМАЦІЇ У GRID - СИСТЕМАХ	26
2.1 Чинники, що уповільнюють процес паралельної обробки інформації	26
2.2 Узагальнена формула обмеження зростання продуктивності обчислювальної GRID-системи	29
2.3 Завдання мінімізації кількості обчислювальних вузлів у GRID- системах	31
2.4 Метод оцінки швидкості паралельної обробки в GRID-системах гомогенного типу	32

2.5	Метод оцінки швидкості паралельної обробки в GRID-системах гетерогенного типу	36
2.6	Аналіз продуктивності обчислювальних вузлів з багатоядерними процесорами.....	39
2.7	Аналіз навчання нейронних мереж у GRID-системах	42
2.8	Взаємозв'язок обсягу послідовних обчислень та часу обробки пакетів у GRID-системах.....	49
2.9	Динаміка характеристик завантаженості GRID-систем у процесі обробки даних.....	51
3	ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ВИКОРИСТАНИХ МЕТОДИК ОЦІНКИ ПРОДУКТИВНОСТІ GRID-СИСТЕМ. ПИТАННЯ ПРАКТИЧНОГО ЗАСТОСУВАННЯ	54
3.1	Розробка архітектури та структурної схеми моделюваної GRID-системи. Імітація вирішення завдань у GRID-системі.....	54
3.2	Розгортання високопродуктивного обчислювального кластера з урахуванням дистрибутива PelicanHPC.....	69
3.3	Розробка GRID-системи на основі інфраструктури BOINC.....	70
	ВИСНОВКИ.....	72
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	74
	ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	76
	ДОДАТОК Б Програма аналізу прискорення розподілених обчислень на алгоритмах кластеризації	91

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

БАК – великий адронний колайдер

OSI – базова еталонна модель відкритих систем

WDS – бездротова розподільча система (англ., Wireless Distribution System)

ГА – генетичні алгоритми

LCG – LHC Computing GRID

EC2 – Elastic Compute Cloud

UEC – Ubuntu Enterprise Cloud

EGEE – Enabling Grids for E-science

NGG – Next Generation GRID

CPU – процесор

PDA – кишеньовий персональний комп'ютер

NFS – близький безконтактний зв'язок

ВСТУП

В останні роки в ІТ-спільноті не слабшає інтерес до теоретичних та практичних питань організації паралельної обробки інформації в географічно розподілених обчислювальних системах. Географічно розподілені обчислення (англ. GRID computing) - технологія розподілених обчислень, в яких обчислювальна система («суперкомп'ютер») представлена в вигляді з'єднаних мережею обчислювальних вузлів, слабопов'язаних, гомогенних або гетерогенних комп'ютерів, що працюють разом для виконання великої кількості завдань. GRID застосовується для вирішення завдань, які потребують значних обчислювальних ресурсів. Спектр розв'язуваних завдань відноситься до багатьох областей, від розвідки нафти та газу, проблем надпровідності та генетики, до аналізу даних з марсоходу Curiosity та моделювання процесу формування планетарних систем. Вже кілька років інформація, що надходить з Великого Адронного Колайдера обробляється спеціально створеною GRID-системою LCG, що включає в собі 170 комп'ютерних центрів у 42 країнах світу та оброблює масив даних 25 Петабайт [21].

Останні дослідження в галузі GRID-систем зачіпають безліч проблем, пов'язаних з інфраструктурою (моніторинг та інтеграція систем) до впровадження та аналізу результатів роботи конкретних GRID-систем у сфері хмарних обчислень.

Однак, незважаючи на широке поширення GRID-систем та велику кількість присвячених їм публікацій залишається відносно мало освітленим питання аналізу продуктивності та визначення повного спектру факторів, що впливають на швидкість обробки даних у цих системах. Найближча до цієї теми публікація торкається питання аналізу прискорення обчислень на багатоядерних процесорах, де аналізується залежність прискорення обчислень від кількості ядер та потоків у процесорі, аналізу деяких з

алгоритмів обміну інформацією та виявлення сильних та слабких сторін цих алгоритмів. Розглянуті ресурси далеко не вичерпні у своїй спробі охопити низку основних аспектів проблематики в даній предметній галузі, проте в них повною мірою не висвітлено вплив специфічних особливостей GRID-систем на швидкість обробки даних. Ця робота значною мірою присвячена питанням аналізу прискорення паралельною обробки даних в географічно розподілених системах з інтенсивним міжпроцесорним обміном.

Метою цієї роботи є заповнення прогалин в аналізі та оцінці факторів, що впливають на швидкість паралельної обробки інформації у географічно розподілених інформаційних системах.

Предметна область дослідження визначається наступними задачами:

- проведення аналізу існуючих методів розрахунку часу та швидкості обчислень у GRID-системах;
- визначення залежності швидкості обчислень у GRID-системах від кількості обчислювальних вузлів;
- огляд методів оцінки оптимальної у сенсі прискорення обчислень кількості обчислювальних вузлів у GRID-системах;
- розробка методик, що дозволяють проводити оцінку часу обчислень та прискорення обчислень у GRID-системах.

1 АНАЛІЗ ПРОБЛЕМАТИКИ В ОБЛАСТІ ОРГАНІЗАЦІЇ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ В GRID-СИСТЕМАХ. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Стан досліджень розробок у галузі GRID-систем. Шляхи розвитку розподілених систем

Обсяги даних, які фігурують у різних обчислювальних задачах, що вимагають розподіленої обробки, становлять десятки терабайт, а для розрахунків, пов'язаних з цими задачами, задіяні комп'ютери у всьому світі. Обчислювальні потужності, які забезпечуються GRID системами, дуже великі і продовжують збільшуватись. Так, наприклад, обсяги даних, які планується отримати в ході експериментів у ВАК, становлять близько 500 Тб. Загальна кількість учасників у проекті SETI@HOME становить більше 1.4 мільйони людей із різних країн світу. Виникли такі масштабні проекти, як хмарна служба Amazon EC2 [2], UEC у Canonical Ubuntu 9.10 [3], LCG - обчислювальний GRID для великого адронного колайдера (ВАК) [1], ряд проектів добровільних обчислень "@HOME" [4-8] з охопленням аудиторії по всій планеті.

Зі збільшенням масштабів GRID-систем дедалі більше виявляють свій вплив специфічні для систем такого роду архітектурні особливості, що позначаються на швидкості обчислень та часу витраченого на вирішення завдань. Аналіз цих особливостей може дозволити визначити найбільш підходящі параметри систем для розв'язання конкретних завдань. Наукові публікації авторів, що досліджують питання побудови GRID-систем, присвячені аналізу паралельних обчислень загалом, або аналізу окремих аспектів GRID- систем (в основному, алгоритмів розподілу навантаження), але безлічі інших властивостей досі приділено недостатньо уваги.

Аналіз існуючих проектів щодо побудови GRID-систем дозволяє ряду

дослідників зробити висновок про три основні напрямки розвитку GRID-технологій і пов'язаною з ними класифікації GRID-систем:

- обчислювальні GRID (Computational GRID);
- GRID для інтенсивної обробки даних (Data GRID);
- Семантичний GRID для оперування даними з різних баз даних (Semantic GRID).

Метою першого напряму є досягнення максимальної швидкості обчислень за рахунок глобального розподілу операцій між безліччю комп'ютерів, а також, можливо серверами та суперкомп'ютерами. Задача розділяється між безліччю окремих систем, на яких проводиться обробка інформації, після чого результати надходять на основний сервер. Кожен з обчислювальних вузлів може розглядатися як самостійна обчислювальна система, що виконує поставлену роботу та має доступ до мережі. Обчислювальні GRID у багатьох випадках є більш рентабельнішими, ніж суперкомп'ютери рівної обчислювальної потужності.

Метою другого напряму є обробка величезних обсягів даних відносно нескладними програмами. Тому обчислювальні ресурси GRID-інфраструктури в цьому випадку часто є кластерами персональних комп'ютерів. А ось доставка даних для обробки та пересилання результатів у цьому випадку є досить складне завдання. Одним з найбільших проєктів, метою якого є створення GRID-системи для обробки наукових даних, є проєкт EGEE [11].

EGEE – проєкт, спрямований на побудову GRID-інфраструктури, яка зможе використовуватись у численних наукових дослідженнях у Європі. Консорціум учасників проєкту включає понад 70 інститутів із 27 європейських країн. Стартовим майданчиком для EGEE служить LHC Computing Grid, метою якого є надання обчислювальних ресурсів для аналізу даних, що надходять від ВАК. Почавши з цієї інфраструктури, EGEE додає ресурси з усіх частин світу і заохочує користувачів з інших спільнот, щоб сформувати найбільшу у світі GRID-інфраструктуру, яка могла б

використовуватися в численних галузях науки. В результаті науково-дослідна спільнота Європи отримує у своє розпорядження найпотужнішу обчислювальну платформу, послуги якої – це цілодобовий доступ до найбільших обчислювальних ресурсам. Доступ не залежить від розташування споживачів і ґрунтуватиметься на використанні наукових мереж Geant та NRNs.

Дві ключових особливості GRID систем третього покоління – наголос на прийнятті сервісно-орієнтованої моделі та зростаюча увага до метаданих. Фактично сервісно-орієнтований підхід припускає, що гнучке застосування ресурсів GRID у додатках Grid вимагатиме інформацію про функціональні можливості, характеристики та інтерфейси різних компонентів, і ця інформація повинна бути погодженою, щоб можна було її обробляти комп'ютером [7].

Були введені нові терміни «розподілене співробітництво» і «віртуальна організація». В третьому поколінні сформувалося більш цілісне уявлення про виконання GRID обчислень, і можна сказати, що справа йдеться скоріше про інфраструктуру для е-Науки, ніж про поліпшення уже розробленої існуючої технології. Очікуване використання обчислювальних засобів з масовим паралелізмом - тільки частина картини, що проступила, існує також величезна кількість користувачів, отже розподілені обчислення не були прерогативою тільки комп'ютерів з масовим паралелізмом.

GRID може забезпечити підтримку віддаленого спілкування вчених в реальному часі. Особливо важливою є інфраструктура для підтримки розподілених ресурсів – тут є багато ключових сервісів: безпека, масштабування і управління, реєстрація і пошук, та інтерфейси Web-сервісів, орієнтовані на повідомлення, для забезпечення могутніх механізмів співробітництва. Всі головні GRID-сервіси і інфраструктура забезпечують співпрацю і є вкрай значущими для суспільства.

1.2 Особливості паралельних обчислень у GRID-системах

Використання GRID-систем актуально в задачах, алгоритми рішення яких натякають на можливість розпаралелювання більшої частини обчислень з обміном даними між обчислювальними вузлами та сервером [12].

Визначають два ключові фактори, які ускладнюють здійснення паралельних обчислень у GRID-середовищі [10]:

- нестабільність роботи великої кількості обчислювальних вузлів та мережі, через яку передаються пакети з інформацією;
- недостатня передбачуваність часу відгуку на запит, пов'язана з проходженням пакетів через безліч мережевих пристроїв та з відмінностями в протоколах зв'язку використовуваних у зовнішніх комп'ютерних мережах та для міжпроцесорного обміну. Також дається взнаки обмеження швидкості обміну пакетами.

Серед інших параметрів GRID-систем слід виділити масштабованість, можливу сегментованість, високу інтенсивність та складність зовнішнього інформаційного обміну, роботу на високих рівнях моделі OSI доступність, налаштованість, керованість параметрів. Можна виділити і безліч інших параметрів, що належать до таких аспектів організації GRID-систем, як безпека, архітектура побудови, специфіка розв'язуваних завдань та ін. В даній роботі необхідну увагу приділено архітектурним особливостям та факторам, що позначаються на динаміці інформаційних потоків.

Стабільна організація паралельних обчислень з інтенсивним обміном інформацією між обчислювальними вузлами GRID-системи є нетривіальним завданням, можливість рішення якої доцільно розглядати для кожної системи окремо. При цьому якщо об'ємне завдання, яке необхідно вирішити, може бути розбите на велику кількість невеликих, незалежних частин, то GRID-технологія виявляється ефективним та відносно дешевим рішенням на відміну від суперкомп'ютерів, які виявляються невиправданими дорогим і

найчастіше неефективним рішенням.

Прикладами таких задач являються:

- масова обробка потоків експериментальних, статистичних чи інших даних великого обсягу;
- візуалізація великих наборів даних (окремі області візуального подання обробляються незалежно, а потім об'єднуються);
- складні бізнес-програми з великими обсягами обчислень.

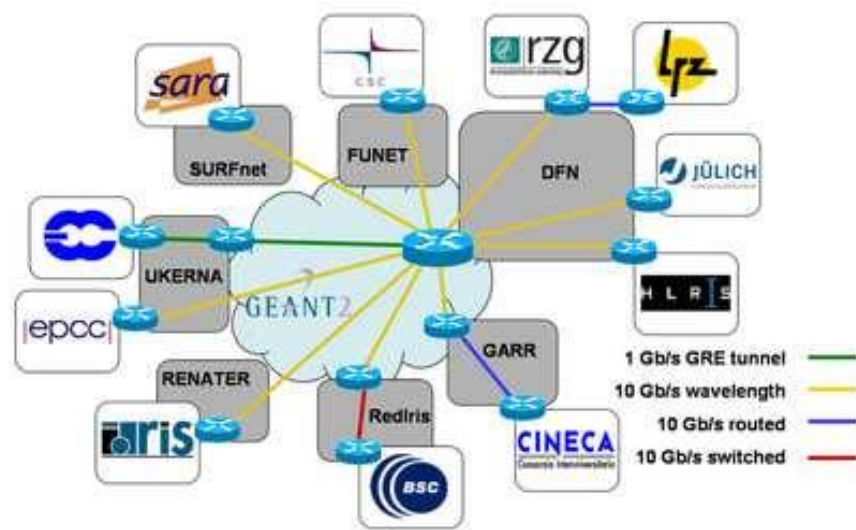


Рисунок 1.1 — Схема інфраструктури DEISA

Інтерес представляє застосування GRID-систем для координації використання різних суперкомп'ютерів та кластерів. Проект DEISA, схема якого представлена (рисунок 1.1) [13], може бути прикладом цього напрямку, в якому проводиться об'єднання суперкомп'ютерних центрів. Фінансований Європейською Комісією з 2002 року, цей проект об'єднує 11 національних суперкомп'ютерних центрів у 7 країнах Європи.

1.3 Класифікація архітектур GRID-систем

Як свідчать дослідження [12, 14], фундаментальна різниця між GRID -

технологіями і перспективним баченням GRID наступного покоління NGG полягає в розстановці пріоритетів між проникністю та здатністю до самоврядування. Під проникністю тут розуміється складна властивість, що залежить від інших примітивних атрибутів, серед яких головними є: доступність, орієнтація на користувача або налаштованість та динамічна взаємодія.

Представлена класифікація GRID-систем на основі чотирьох ознак – доступність, налаштованість, інтерактивність та керованість параметрів (таблиця 1.1).

Таблиця 1.1 – Класифікація GRID-систем за доступністю, налаштовуваності інтерактивності та керованості параметрів [15]

Ціль проекту	Категорії новітніх систем	Підкатегорії	Головна відмінність від традиційних GRID-систем	Приклади проектів
1	2	3	4	5
Доступ	GRID доступу	Ad hoc GRID	Без зумовлених точок входу	OurGrid (www.ourgrid.org) и myGrid (www.mygrid.org.uk)
		Мобільні GRID	Підтримка мобільності клієнтів. Служб, або всього разом	Akogrimo (www.mobilegrids.org) ISAM (інфраструктура мобільних додатків www.inf.ufrgs.br/~isam/English)
		Бездротові GRID	Підтримка бездротового зв'язку між GRID-вузлами та інтерфейсом	Home Grid and Office Grid

Продовження таблиці 1.1

1	2	3	4	5
Інтерактив	Інтерактивні GRID	Явні інтерактивні GRID	Підтримка явної взаємодії з користувачами в режимі реального часу	CrossGrid (www.crossgrid.org) and edutain@grid (www.edutaingrid.eu)
		Контекст чутливі GRID	Взаємодія з середовищем, побудова контексту та адаптація поведінки	RUNES (Reconfigurable Ubiquitous Networked Embedded Systems, www.istrunes.org), SENSE Embedded (Smart Network of Sensing Entities, www.sense-ist.org), Hydra(Networked Embedded System Middleware for Heterogeneous Physical Devices in a Distributed Architecture, www.hydra.eu.com), and MORE (Network-centric Middleware for Group communication and Resource Sharing across Heterogeneous Embedded Systems, www.ist-more.org)

Продовження таблиці 1.1

1	2	3	4	5
Налаштовув аність	Налаштов увані GRID	Персональ ні GRID	Використов увані індивідуаль но чи особисно приналежні	Personal Grid
		Персоналіз овані GRID	Реалізують глобально наладштовув ані GRID- портали	Akogrimo (www.mobilegrids.org) and MyGrid (www.mygrid.org.uk)
Керованість	Керовані GRID	Автономні GRID	Зі спробою моделюван ня поведінки автономної нервової системи	IBM OptimalGrid and AutoMAGI8
		GRID знань	Використов ують технології знань для підтримки самокерува ння	OntoGrid (www.ontogrid.net/ontogrid/index.jsp), InteliGrid (www.inteligrid.com), and K-Wf Grid (Knowledge-Based Workflow System for Grid Applications, www.kwfgrid.eu)
		Органічні GRID	Використов ують абстракції мурашника для підтримки самоуправл іння	Organic Grid9

Класифікація GRID систем за розміром, типом рішення, моделі доступу, інтерактивності, ступеня персоналізації та схеми управління

представлена на рисунку 1.2 [15].

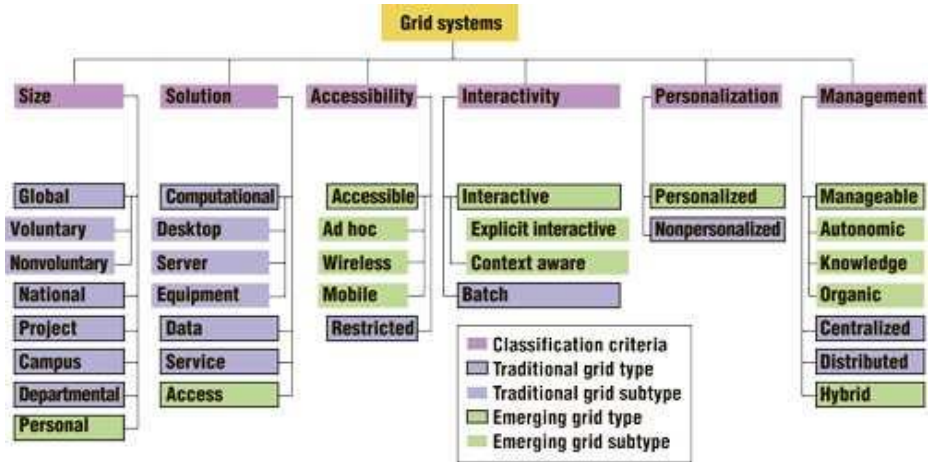


Рисунок 1.2 – Класифікація традиційних та сучасних GRID-систем (NGG-Next Generation grids)

Кожна GRID-система може ставитись до кількох критеріїв класифікації.

Моделі GRID-систем, досліджуваних у цій роботі, містять в якості обов'язкового елемента єдиний сервер, який розсилає обчислювальним вузлам завдання та приймаючий результати обчислень. У зв'язку з цим, результати роботи можуть бути застосовні для будь-яких GRID-систем з вказаних у класифікації в Табл. 1.1 та рисунку 1.2 крім розподілених та гібридних за схемою управління (Management - Distributed/Hybrid на рисунку .3). Для розподілених та гібридних схем управління необхідна модифікація результатів роботи або перетворення архітектури GRID-системи шляхом умовного поєднання серверів.

1.3.1 Класифікація GRID за типом обладнання на характеристиками обчислювальних вузлів

У обчислювальних GRID-системах ключовими параметрами є цикли CPU (central processing unit, центральний процесорний пристрій) та GPU (graphics processing unit, графічний процесор). У залежності від обладнання, на якому вони розгорнуті, обчислювальні GRID-системи поділяються на [12]:

- настільні;
- серверні;
- інструментальна GRID.

У настільних GRID велику частину ресурсів складають ресурси настільних комп'ютерів, що не використовуються, тоді як ресурси серверні GRID зазвичай обмежуються ресурсами, доступними на серверів.

Інструментальні GRID включають нетривіальний елемент обладнання, яке керується віддалено і передає дані для аналізу.

GRID служб або GRID утиліт надають комерційні комп'ютерні послуги (цикли CPU та дискову пам'ять), які, працюючи в дослідницьких або промислових областях фахівці, можуть за необхідності купити.

Глобальні GRID встановлюються в Інтернеті, надаючи окремим користувачам потужність незалежно від того, де ці користувачі географічно локалізовані.

GRID, що вільно масштабуються, дають можливість користувачам інтернету надати свої комп'ютерні ресурси для колективного виконання завдань, які потребують великих обчислювальних потужностей.

Невільно масштабовані GRID містять тільки ті машини, які їм визначено заздалегідь.

Національні GRID мають тільки ті комп'ютерні ресурсами, які доступні у межах відповідної держави.

Проектні GRID подібні до національних, але щодо об'єднання ресурсів вони не обмежуються ресурсами однієї держави, а об'єднують багато географічних та адміністративних областей. Вони доступні лише для зареєстрованих членів та організацій, що співпрацюють.

Інтра-GRID або університетські GRID, в яких обсяг ресурсів обмежується ресурсами єдиної організації, призначені для користувачів, які є членами головної організації.

Відомчі GRID доступні тільки для співробітників конкретного відомства. Персональним GRID властива дуже обмежена організаційна

структура. Доступ до них дозволено на персональному рівні самим власникам, або деяким об'єднанням власників.

1.3.2 Класифікація за типом доступу до GRID-систем

Ad-hoc GRID – система інтерактивних взаємодіючих гетерогенних обчислювальних вузлів в логічне співтовариство без попередньо конфігурованої фіксованої інфраструктури і з мінімальними адміністративними вимогами [16]. GRID інфраструктура розширена так, що починає включати динамічні приєднання, не вимагаючи наявності формальних, чітко визначених, вхідних точок GRID, наявність яких підтверджено. Приклад системи представлений (рисунок 1.3).

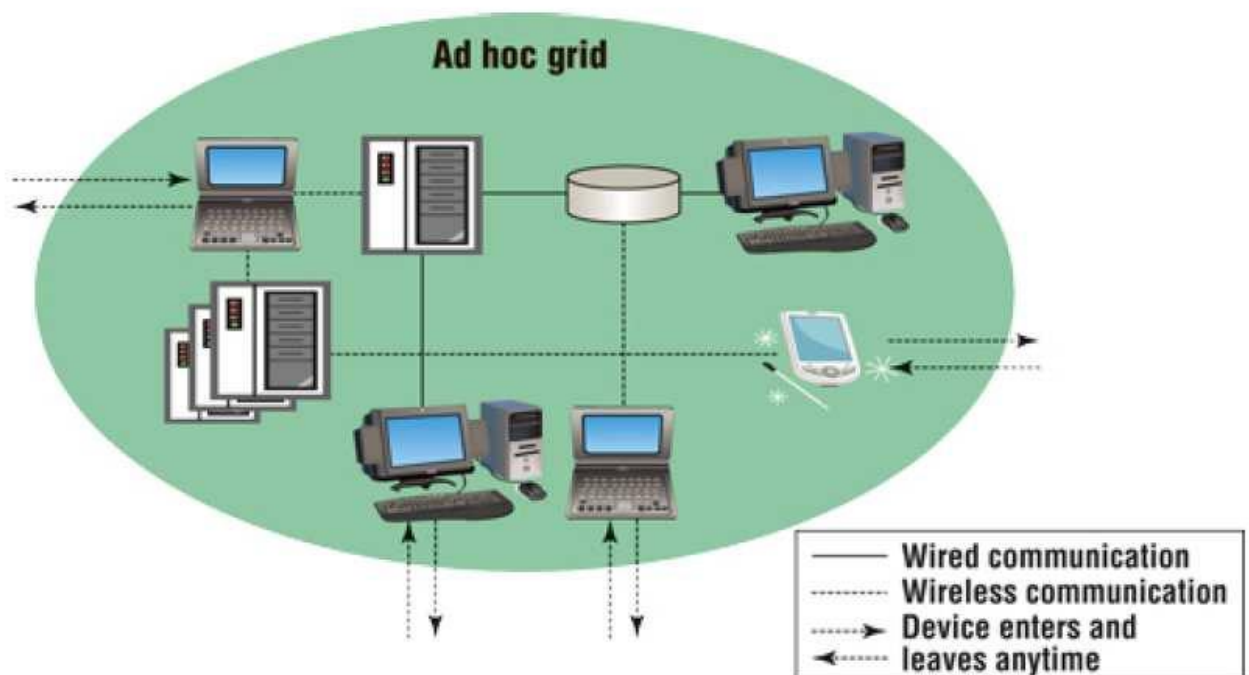


Рисунок 1.3 – Ad-hoc GRID-система

Бездротові GRID розширюють номенклатуру GRID-ресурсів включенням бездротових пристроїв різних розмірів та властивостей, таких як датчики, мобільні телефони, ноутбуки, спеціальні інструменти та мережеві інтерфейсні пристрої [17].

Мобільні GRID-системи надають доступ до GRID-служб через такі

мобільні пристрої, як PDA та смартфони. Мільйони мобільних пристроїв, що продаються, дають високу обчислювальну потужність нарівні з мобільністю. У надзвичайних ситуаціях, таких, як природні катастрофи та активні військові операції, бездротові мобільні пристрої можуть опинитися єдиним доступним засобом комунікації та обчислювального обслуговування. приклад системи представлено (рисунок 1.4).

Прикладами існуючих GRID-проектів із розглянутими схемами доступу можуть бути проекти Akogrimo (www.mobilegrids.org), ISAM (www.inf.ufrgs.br/~isam/English) і MADAM (www.intermedia.uio.no/display/madam/Home).

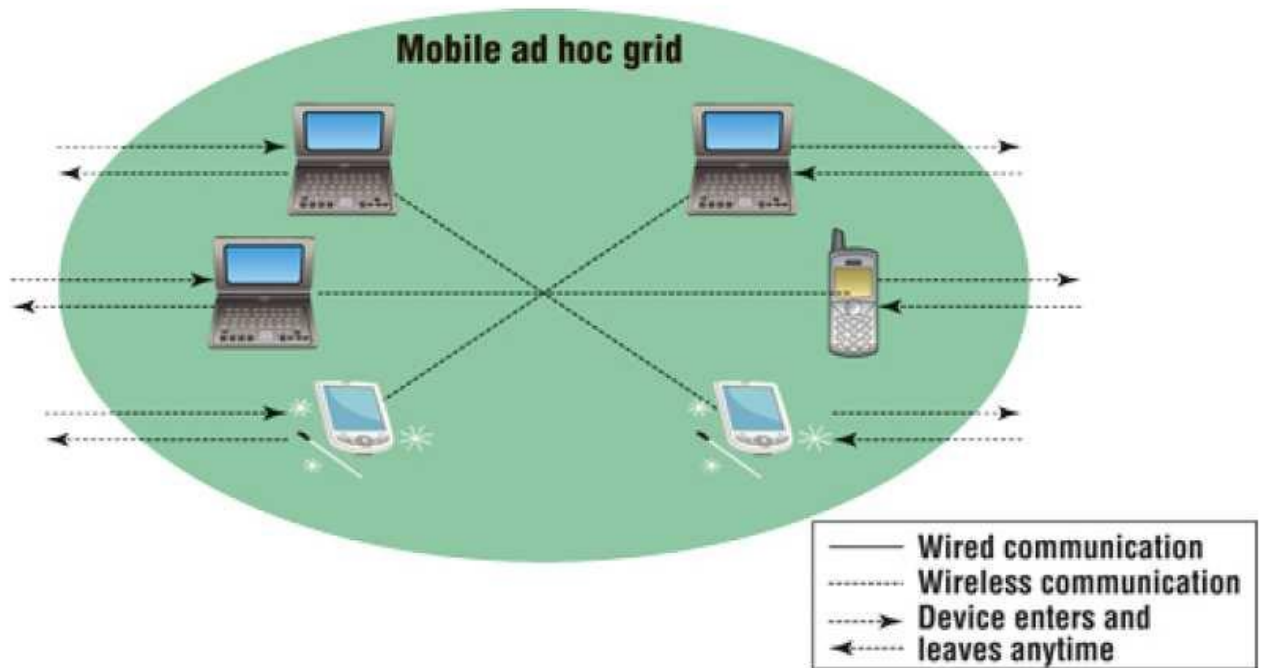


Рисунок 1.4 – Мобільна Ad-hoc GRID-система

1.4 Стек протоколів GRID-систем

Загальна структура GRID-систем описується в вигляді стека протоколів [14]. У такій моделі кожен рівень призначений для вирішення вузького кола завдань і використовується для надання послуг для вищих рівнів. Верхні рівні ближче до користувача та працюють із найбільш абстрактними об'єктами, тоді як нижні рівні сильно залежать від фізичної реалізації GRID-

ресурсів. Стек GRID-протоколів включає 5 рівнів:

- прикладний рівень (Application Layer) описує користувацькі додатки, що працюють у середовищі віртуальної організації. Додатки функціонують, використовуючи протоколи, визначені на нижчих рівнях;
- колективний (Collective Layer) рівень відповідає за координацію використання наявних ресурсів;
- ресурсний рівень (Resource Layer) - це багаторівневе ядро системи, протоколи якого взаємодіють із ресурсами, використовуючи уніфікований інтерфейс і не відрізняючи архітектурні особливості конкретного ресурсу;
- зв'язуючий рівень (Connectivity Layer) складають протоколи, які забезпечують обмін даними між компонентами базового рівня та протоколи аутентифікації;
- апаратний рівень (Fabric Layer) складають протоколи, яким відповідні служби безпосередньо працюють з ресурсами.

Для завдань, які вирішуються в даній роботі, необхідна інформація як про архітектуру GRID-системи, так і про середовище обміну даними, тобто, що надається на апаратному і зв'язуючому рівнях стека протоколів.

Апаратний рівень: керування локальними ресурсами. На цьому рівні забезпечується доступ до розподілених ресурсів, необхідний протоколам вищого рівня. Спектр можливих ресурсів досить широкий - це можуть бути комп'ютери, пристрої зберігання даних, каталоги, мережеві ресурси тощо. При цьому ресурс може бути логічною сутністю (розподіленою системою) або фізичною (кластером комп'ютерів). Реалізація такого ресурсу може включати внутрішні протоколи (наприклад, NFS), однак такі протоколи не включаються до стеку GRID. Компоненти апаратного рівня реалізують локальні операції, специфічні для кожного даного ресурсу (логічного чи фізичного). Цей рівень є набір інтерфейсів управління локальними ресурсами.

Зв'язуючий рівень: комунікації та безпека. Комунікаційні протоколи зв'язуючого рівня повинні забезпечувати надійний транспорт та

маршрутизацію повідомлень, а також присвоєння імен об'єктам мережі, а протоколи аутентифікації цього рівня, ґрунтуючись на комунікаційних, надають криптографічні механізми для ідентифікації та перевірки автентичності користувачів та ресурсів. Інфраструктура підтримки включає централізовану видачу сертифікатів, управління сертифікатами та ключами тощо.

Ресурсний рівень: спільне використання ресурсів. На цьому рівні з допомогою комунікаційних і автентифікаційних протоколів, що входять до нижчого зв'язуючого рівня, проводиться узгодження методів безпеки, ініціалізація та моніторинг ресурсів, та управління ними. Для доступу до локальних ресурсів та подальшого управління ресурсний рівень викликає відповідні функції апаратного рівня. Зауважимо, що протоколи ресурсного рівня призначені виключно для роботи з локальними ресурсами, вони не враховують глобальний стан системи. Цим займається колективний рівень, що знаходиться вище. Ресурсний рівень включає два основні класи протоколів:

- інформаційні протоколи, призначені для отримання інформації про структуру та стан ресурсу, його конфігурацію, поточне завантаження та політику (тобто умови) надання ресурсів (наприклад, вартість їх використання);

- протоколи управління, які забезпечують узгодженість доступу до ресурсу і визначають необхідні операції, які ресурс повинен виконати (Скажімо, ініціалізація процесу або доступу до даних).

Колективний рівень: координація ресурсів. Протоколи колективного рівня відповідають за взаємодію всіх елементів пулу ресурсів, як і відбито у самій назві. В якості прикладу глобальних функцій та сервісів, що реалізуються протоколами цього рівня, можна назвати службу каталогів, розподіл ресурсів, планування та брокерські послуги, служби моніторингу, діагностики, реплікації даних колективної авторизації.

Прикладний рівень: запуск програм у GRID-середовище. Цей вищий

рівень GRID-архітектури включає користувацькі додатки, які виконуються в середовищі об'єднаних ресурсів. У процесі виконання програми використовують протоколи нижченаведених рівнів, що забезпечують доступ до необхідних служб, а також прикладні програмні інтерфейси, відповідні даним протоколам. Програми можуть викликатися через досить складні обгортки та бібліотеки. Ці обгортки самі можуть визначати протоколи, сервіси та прикладні програмні інтерфейси, однак подібні надбудови не відносяться до фундаментальних протоколів та сервісів, визначаючим архітектуру GRID-систем.

1.5 Постановка задачі

Метою цієї роботи є аналіз та оцінка факторів прискорення обчислень у географічно розподілених інформаційних системах (GRID) з метою виявлення оновлених методик розрахунку швидкості обчислень та суміжних показників GRID-систем. Сформульовано такі основні завдання:

- провести аналіз залежності прискорення обчислень від архітектурних особливостей GRID-систем з визначенням оптимальної в сенсі прискорення обчислень кількості обчислювальних вузлів у конкретних GRID-системах;
- розробити програмний комплекс для визначення прискорення обчислень у GRID-системах з урахуванням розроблених методів;
- розробити математичну модель клієнт-серверної взаємодії з урахуванням особливостей архітектури та організації GRID-систем;
- дослідити процес навчання нейронних мереж із застосуванням GRID-систем;
- дослідити вплив багатоядерності обчислювальних вузлів на прискорення обчислень у GRID-системах;
- розробити узагальнений метод оцінки прискорення обчислень у GRID-системі гомогенного та гетерогенного типів;

2 ДОСЛІДЖЕННЯ ТА ВИКОРИСТАННЯ МЕТОДІВ АНАЛІЗУ ШВИДКОСТІ ПАРАЛЕЛЬНОЇ ОБРОБКИ ІНФОРМАЦІЇ У GRID-СИСТЕМАХ

2.1 Чинники, що уповільнюють процес паралельної обробки інформації

Асимптотичні властивості закону Амдала не можуть розглядатися без урахування факторів, здатних зробити істотний внесок у швидкість процесу обчислень при великій кількості обчислювальних вузлів у GRID-системі [11]. Відмінною особливістю GRID-систем (при порівнянні з кластерами та іншими системами, призначеними для паралельних обчислень) є географічна віддаленість обчислювальних вузлів і, як наслідок, поява тимчасових затримок під час обміну даними.

Додаткові затримки, що виникають у GRID-системах при передачі даних, поділяються на кілька типів:

- встановлення з'єднання для початку передачі;
- передача даних для розрахунків від сервера до обчислювального вузла або результатів обчислень від обчислювального вузла до сервера;
- виникнення черги обчислювальних вузлів до сервера отримання та обробку результатів.

Перші два типи затримок зменшують на певне значення прискорення системи. Третій тип затримки, як буде показано, може звести прискорення системи нанівець.

Припустимо, що під час обміну даними між обчислювальним вузлом та сервером обчислювальний вузол одноразово проводить перевірку активності сервера, після чого відправляє розраховані дані та отримує новий пакет. Для кожного з пакетів у задачі тимчасова затримка складе:

$$T1 = q \cdot k, \tag{2.1}$$

де q - загальна кількість пакетів у завданні, k - час затримки для перевірки активності сервера та обміну системною інформацією.

Після з'єднання сервера та обчислювального вузла починається обмін даними зі швидкістю, що є меншою між сервером та обчислювальним вузлом. Однак, враховуючи те, що сервер може обробляти кілька обчислювальних вузлів одночасно та кількість обчислювальних вузлів досягає достатньо великих значень, можна вважати, що обмін відбувається зі швидкістю, максимально можливою для сторони сервера. У випадку, якщо середньостатистична швидкість буде меншою за швидкість сервера, розрахунки затримки на обмін даними визначатимуть верхню можливу межу прискорення, але не зможуть бути меншими від реального значення прискорення системи. Таким чином, затримка на передачу даних складе:

$$T_2 = q \frac{r}{e}, \quad (2.2)$$

де q – загальна кількість пакетів у завданні,

r – обсяг даних для обміну (результати обчислювального вузла та новий пакет від сервера),

e – швидкість обміну з боку сервера (біт/сек).

В обсяг даних для обміну включені і результати обчислювального вузла і новий пакет від сервера в силу послідовності обміну даними, схема якої наведена на рисунку 2.1. Стрілками позначено напрямки передачі даних. Дужкою об'єднані дії, що становлять повторюваний протягом усіх розрахунків блок операцій.

Як видно зі схеми послідовності обміну даними, після першого з'єднання та передачі пакета виконується одна і та ж послідовність дій: з'єднання, передача результату, передача пакету. Таким чином, між передачею результату та передачею пакета відсутні додаткові тимчасові затримки, та обсяг даних підсумовується. Не дивлячись на те, що обмін відбувається паралельно, передбачається, що передача завжди відбувається

послідовно. Це пов'язано з тим, що серверу може знадобитися попередня перевірка/обробка результатів перед надсиланням нового пакета.

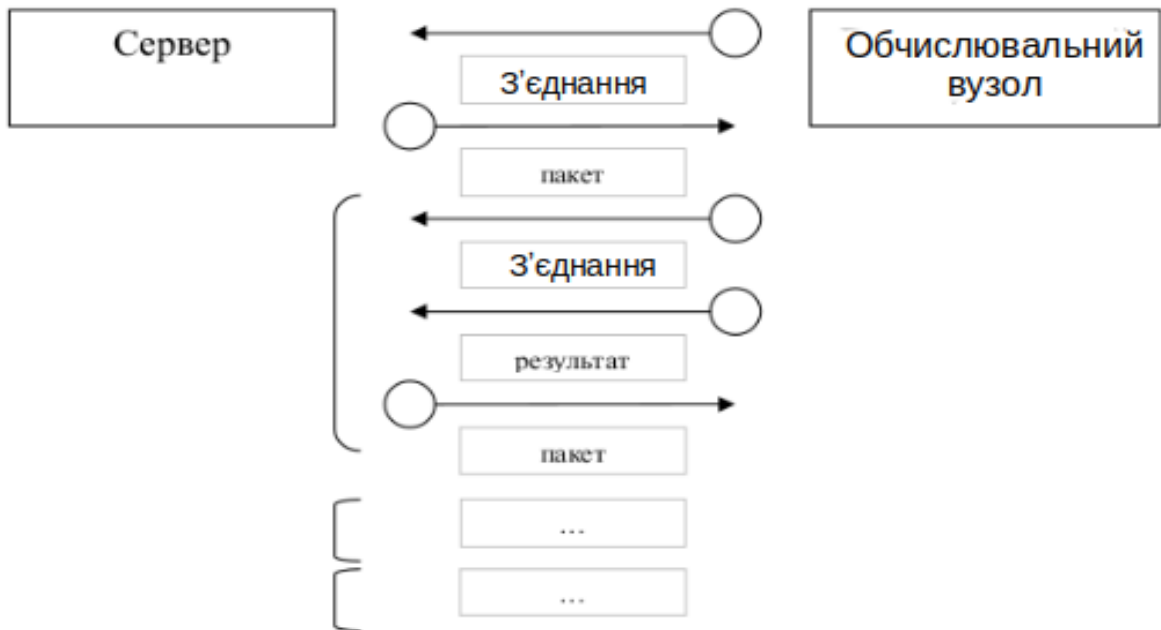


Рисунок 2.1 – Послідовність обміну даними між обчислювальним вузлом та сервером

При великих кількостях обчислювальних вузлів затримка на очікування в черзі до сервера буде найбільш суттєвим. Вона виникає у разі, коли обчислювальні вузли вишиковуються в чергу до сервера для обміну даними. Для аналізу ймовірності виникнення черги та її довжини необхідно визначити частоту звернень обчислювальних вузлів до сервера:

$$Z = q \frac{c \cdot s}{t} \quad (2.3)$$

де c – кількість обчислювальних вузлів,

s – час обробки пакета сервером,

t – час обробки пакета на обчислювальному вузлі.

Час обробки пакету на сервері – час, що витрачається сервером між закінченням прийому результатів обчислень та початком передачі нового

пакета. Час обробки пакета на обчислювальному вузлі — час, що витрачається обчислювальним вузлом між закінченням прийому пакета та початком з'єднання з сервером для надсилання результатів обчислень.

Довжина черги до сервера складе[8]:

$$m(z, n) = \frac{\frac{z^{n+1}}{n \cdot n! \left(1 - \frac{z}{n}\right)^2}}{\sum_{k=0}^n \frac{z^k}{k!} + \frac{z^{(n+1)}}{n! \max((n-z), 0)}} \quad (2.4)$$

де z – частота звернень обчислювальних вузлів до сервера (звернень у секунду),

n – кількість каналів сервера для обробки запитів обчислювальних вузлів (од.).

Імовірність виникнення черги становитиме[8].

$$p(z, n) = \min \left(1 - \frac{\sum_{k=0}^n \frac{\left(\frac{z^k}{n!}\right)}{\sum_{k=0}^n \frac{z^k}{k!} + \frac{z^{n-1}}{n! \max((n-z), 0)}}, 1 \right) \quad (2.5)$$

2.2 Узагальнена формула обмеження зростання продуктивності обчислювальної GRID-системи

Тепер, підсумувавши всі додаткові затримки та врахувавши їх вплив на уповільнення розрахунків у системі можна навести узагальнену формулу обмеження зростання продуктивності обчислювальної GRID-системи при суттєвому зростанні кількості обчислювачів [18, 19]:

$$S(a, c, k, t, e, r, n) = \frac{1}{\left(a + (1-a) \cdot \left(\frac{1}{c} + \frac{k}{t} + \frac{r}{e \cdot t} \cdot \left(1 + p\left(z\left(c, \frac{r}{e}, t\right), n\right) \cdot m\left(z\left(cm, \frac{r}{e}, t\right), n\right) \right) \right) \right)} \quad (2.6)$$

де a – частка послідовних обчислень (значення в інтервалі від 0 до 1);

c – кількість обчислювальних вузлів;

k – затримка перед початком передачі (с);

t – час розрахунку одного пакета на одному обчислювальному вузлі (с);

r – обсяг пакета (біт);

e – швидкість передачі з боку сервера (біт/с);

n – кількість каналів обробки обчислювальних вузлів на сервері;

m – довжина черги до сервера (2.4);

p – ймовірність виникнення черги до сервера (2.5).

Точка максимуму прискорення обчислень в залежності від кількості обчислювачів визначається за формулою (2.7).

$$\frac{\partial}{\partial c} S(a, c, k, t, e, r, n) = 0, \quad (2.7)$$

Вплив кожної з тимчасових затримок показано на рисунку 2.2. Як приклад для ілюстрації впливу тимчасових затримок були вибрані такі параметри:

- частка послідовних обчислень – 0.2;
- кількість обчислювальних вузлів варіюється;
- затримка перед початком передачі - 0.5 с;
- час розрахунку одного пакета на одному обчислювальному вузлі – 70 с;
- обсяг пакету – 30 Кбіт;
- швидкість передачі з боку сервера – 30 кілобіт на секунду;
- кількість каналів обробки обчислювальних вузлів – 1.

Наведені значення параметрів дозволяють наочно показати ступінь падіння прискорення при порівняно невеликій кількості обчислювальних вузлів. Тимчасова затримка у черзі до сервера для певної кількості обчислювальних вузлів, як видно з (рисунок 2.2), утворює точку максимуму, а надалі значення прискорення прямує до нуля.

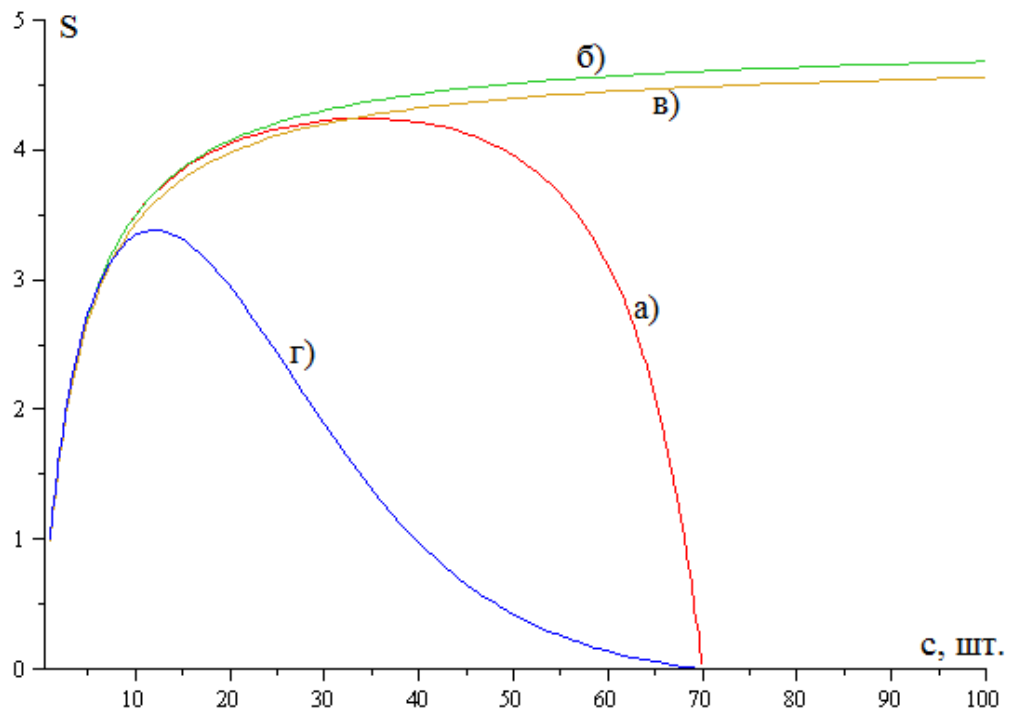


Рисунок 2.2 – Ілюстрація впливу кожної з тимчасових затримок у формулі (2.6): а) $T_{1,2,3}$ (2.6); б) T_1 (2.1); в) T_2 (2.2); г) T_3 (2.4,2.5)

2.3 Завдання мінімізації кількості обчислювальних вузлів у GRID-системах

Практичний інтерес представляє розгляд завдання про мінімізацію кількості обчислювальних вузлів GRID-системи для заданого часу обчислень. Для певного часу обчислень T необхідно знайти оптимальну у сенсі прискорення кількість обчислювальних вузлів. Інші параметри системи вважаються фіксованими. Самому процесу оптимізації піддається розпаралелюється частина завдання, отже, для цієї підзадачі послідовна частина дорівнює 0. І прискорення відповідно до закону Амдала дорівнюватиме кількості працюючих обчислювачів. Якщо задача вирішується t_1 секунд часу на одному обчислювачі, то на c обчислювачах час знадобиться $T_a = t_1 / c$. Значення прискорення S розраховується за $a=0$.

Кожен розпаралелений етап опишемо наступними характеристиками:

- частка послідовних обчислень $a = 0$;

- час розрахунку одного пакету на одному обчислювальному вузлі змінюється від етапу до етапу;
- обсяг пакету змінюється від етапу до етапу;
- інші параметри обчислювального середовища залишаються постійними протягом всього розв'язання задачі.

Слід також згадати, що на швидкість розрахунку може впливати і структура системи: гомогенна за характеристиками обчислювальних вузлів або гетерогенна.

Далі під гетерогенними GRID-системами мається на увазі неоднорідність вузлів по їх технічним характеристикам та можливостям, що приводить до різних затрат часу на різних вузлах при ідентичних пакетах. Під гомогенними системами мається на увазі окремий випадок мереж, при якому всі вузли абсолютно ідентичні за характеристиками. Різноманітність характеристик вузлів призводить до ускладнення оптимізації, обчислень з точки зору кількості одночасно задіяних вузлів. На кожному розпаралеленому етапі утворюються пакети даних для розрахунків задачі на обчислювальних вузлах. Оптимальною за швидкістю розв'язання задачі є ситуація, коли кількість пакетів дорівнює кількості задіяних на етапі обчислювальних вузлів.

2.4 Метод оцінки швидкості паралельної обробки в GRID-системах гомогенного типу

Розглянемо гомогенну GRID-систему. Якщо кількість необхідних обчислювальних вузлів, що надалі позначається як C_{need} , менше загальної кількості вузлів у мережі, то розв'язуваний етап роздається C_{need} вузлам. Якщо C_{need} більше загальної кількості вузлів у мережі, то деяким вузлам доведеться розраховувати кілька пакетів, і вся система буде очікувати на результати розрахунку. Можлива ситуація, коли пакети будуть роздаватися обчислювальним вузлам не коло за колом відразу всім, а по мірі рішення

кожним із вузлів. У такому разі час на розрахунки завдання в GRID-системі зменшиться.

У гомогенних GRID-системах розсилка Q пакетів проводиться по C_{need} обчислювальним вузлам ($Q \geq C_{\text{need}}$) і у випадку, якщо сервер розіслав першу чергу пакетів. буде очікувати повернення всіх результатів, то час T_A на розрахунок етапу виражатиметься формулою:

$$T_A = t \cdot \left[\frac{Q}{C_{\text{max}}} \right] + t \cdot \begin{cases} 1, \left\{ \frac{Q}{C_{\text{max}}} \right\} > 0 \\ 0, \left\{ \frac{Q}{C_{\text{max}}} \right\} = 0 \end{cases} \quad (2.8)$$

де квадратними дужками позначена ціла частина від виразу, а фігурними – дрібна. Q – кількість пакетів, C_{max} – загальна кількість обчислювальних вузлів, t – час розрахунку одного пакета. Залежність часу розрахунку етапу кількості пакетів показано (рисунок 2.3).

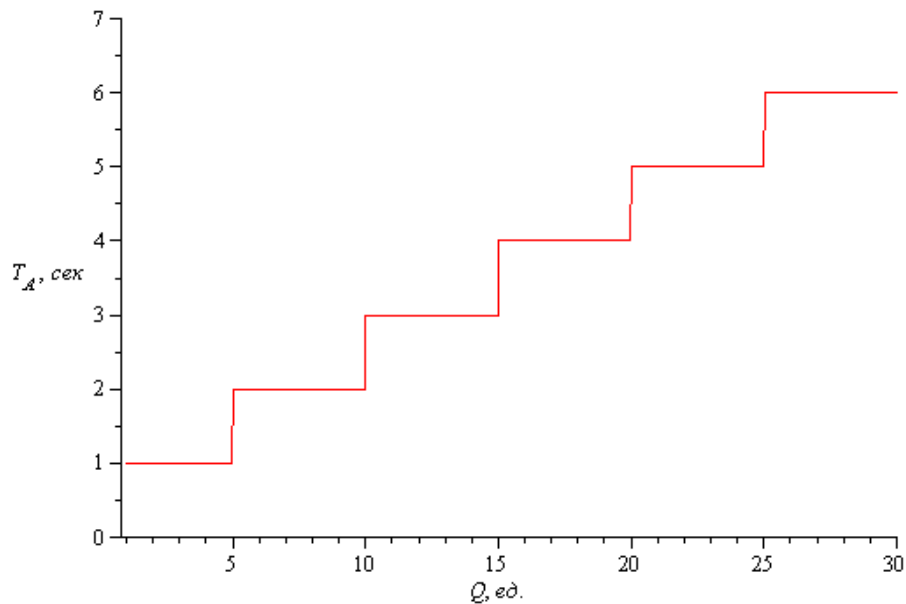


Рисунок. 2.3 – Ілюстрація залежності часу розрахунку етапу задачі від кількості пакетів за $t=1$, $C_{\text{max}}=5$

Критерій розподілу етапу на C_{need} вузлів:

$$s(0, c, t, e, r, n)^{c: [1, \dots, C_{\text{max}}]} \rightarrow \max$$

Введемо позначення

$$q \equiv \frac{r}{e \cdot t}$$

а також прийmemo число каналів обробки запитів $n=1$. Тоді:

$$S(0, c, t, e, r, 1) = \frac{1}{\frac{1}{c} + \frac{k}{t} + q \cdot (1 + p(qc, 1) \cdot m(qc, 1))} \quad (2.9)$$

$$m(qc, 1) = \frac{\frac{q^2 c^2}{1 \cdot 1 (1 - qc)^2}}{1 + qc + \frac{q^2 c^2}{\max(1 - qc, 0)}} \quad (2.10)$$

При $qc < 1$ довжина черги до сервера складе:

$$m(qc, 1) = \frac{q^2 c^2}{1 - qc} \quad (2.11)$$

Імовірність виникнення черги становитиме:

$$m(qc, 1) = 1 - \sum_{h=0}^1 \frac{\left(\frac{(qc)^h}{h!}\right)}{\sum_{k=0}^1 \frac{(qc)^k}{k!} + \frac{(qc)^{(1+1)}}{1! \cdot (1 - qc)}}$$

$$p(qc, 1) = 1 - \left(\frac{1}{1 + qc + \frac{q^2 c^2}{1 - qc} + 1 + qc + \frac{q^2 c^2}{1 - qc}} \right)$$

$$p(qc, 1) = q^2 c^2. \quad (2.12)$$

Таким чином, формула (2.6) набуде вигляду:

$$S_{\text{homogeneous}} \left(0, c, t, e, r, 1 = \frac{1}{\frac{1}{c} + \frac{k}{t} + q \cdot \left(1 + q^2 c^2 \cdot \frac{q^2 c^2}{1 - qc} \right)} \right)$$

$$\frac{\partial S_{\text{homogeneous}}}{\partial c} \rightarrow \text{max: } c = \frac{t \cdot e}{2 \cdot r} \quad (2.13)$$

Якщо врахувати, що між етапами змінюються лише значення e та t , то кількість обчислювальних вузлів, що дає найбільший приріст прискорення обчислень буде визначатися, як [10, 19]:

$$C_{\text{homogeneous}}(t, e, r, C_{\text{max}}) = \left\{ \begin{array}{l} 1, t < \frac{2 \cdot r}{e} \\ \frac{e}{2r} \cdot t, t \in \left[\frac{2 \cdot r}{e}, \dots, \frac{2 \cdot r}{e} \cdot C_{\text{max}} \right] \\ C_{\text{max}}, t > \frac{2 \cdot r}{e} \cdot C_{\text{max}} \end{array} \right\} \quad (2.14)$$

Використання такої кількості обчислювальних вузлів в порівнянні з використанням усіх обчислювальних вузлів при різних вхідних параметрах GRID-системи відображено (рисунок 2.4 та 2.5).

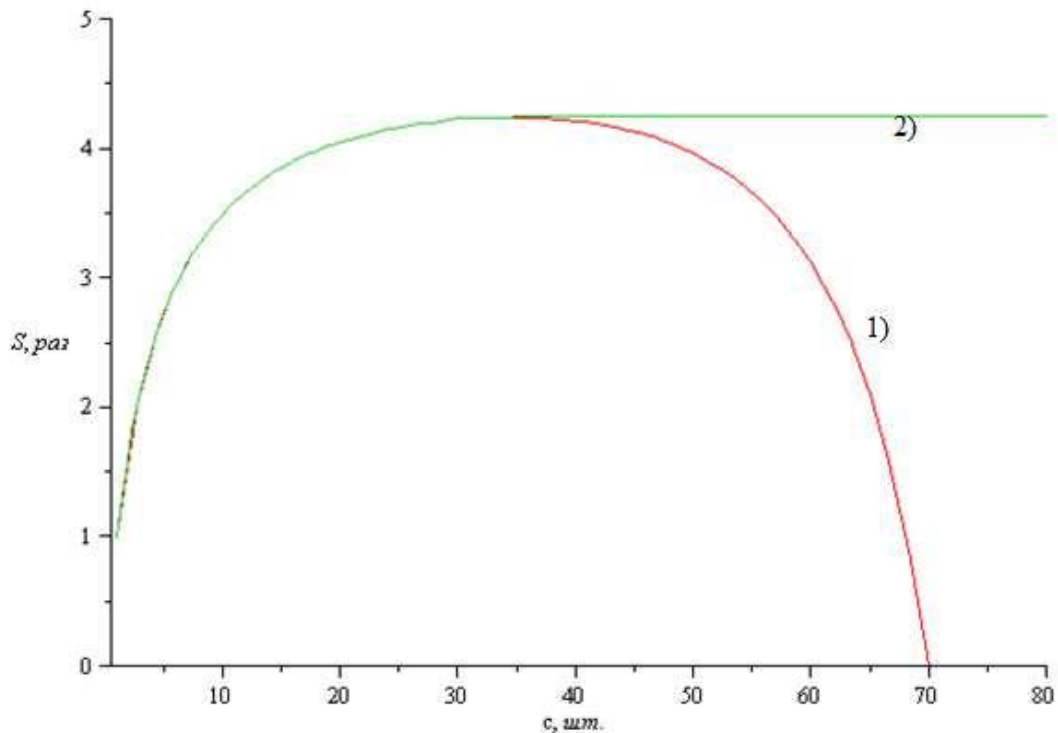


Рисунок 2.4 – Ілюстрація прикорення обчислень

ь при використанні: 1) всіх обчислювальних вузлів; 2) кількості обчислювальних вузлів згідно

(2.14) ($a=0.2, k=0.5, t=70, e=30000, r=30000, n=1$)

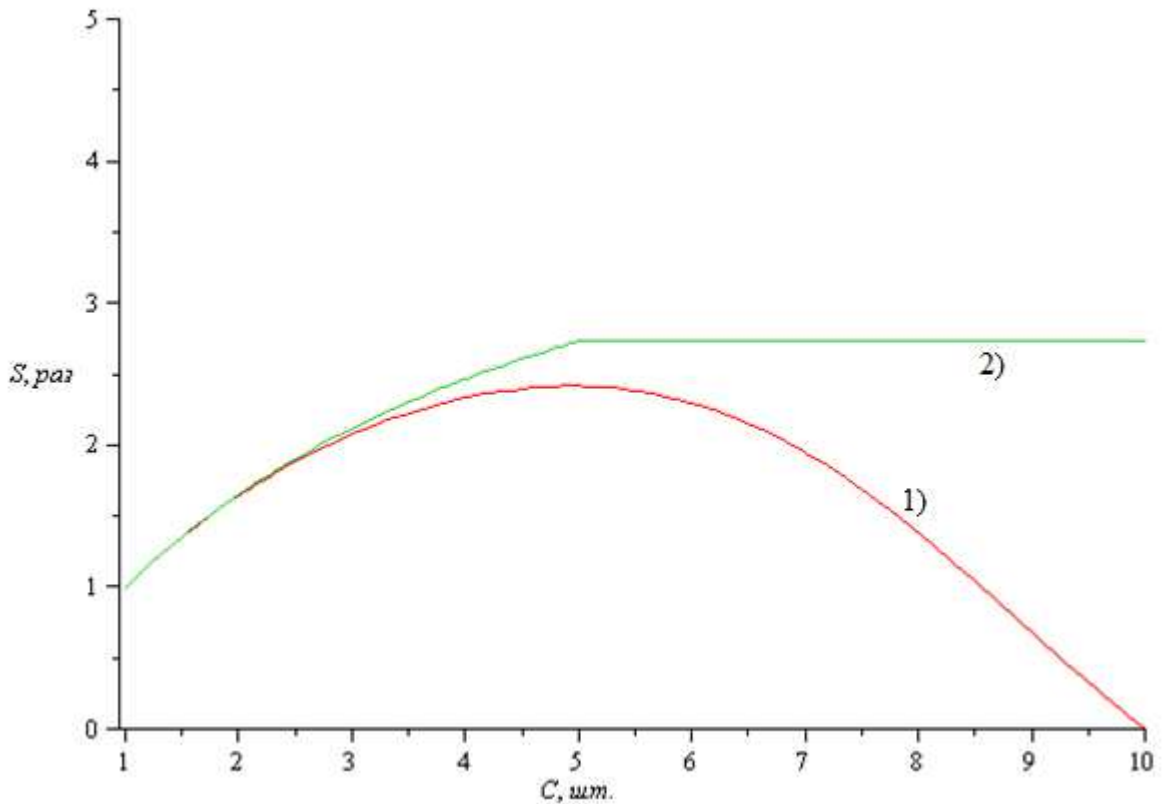


Рисунок 2.5 – Ілюстрація прискорення обчислень під час використання:
 1) всіх обчислювальних вузлів; 2) кількості обчислювальних вузлів згідно
 (2.14) ($a=0.2$, $k=0.5$, $t=100$, $e=3000$, $r=30000$, $n=1$)

2.5 Метод оцінки швидкості паралельної обробки в GRID-системах гетерогенного типу

Для аналізу розрахунків у гетерогенних GRID-системах введемо поняття одиниці обчислювальної потужності вузла "N". Під обчислювальною потужністю вузла N [од./с] будемо розуміти кількість обчислених одним вузлом еталонних пакетів за секунду. Під еталонним пакетом тут розуміється один із пакетів у обчислюваному завданні, щодо якого будуть оцінюватися інші пакети і обчислювальні вузли. Таким чином, кожен вузол представляє собою якусь кількість одиниць обчислювальної потужності в одиницю часу, а кожен пакет вимагає якусь кількість одиниць обчислювальної потужності на рішення. Задача може бути представлена у вигляді набору пакетів $Q(N)$, а GRID-система - у вигляді набору обчислених вузлів $C(N)$. Обидві функції

можуть описувати як вузли/пакети окремо, так і загальна кількість для зазначеної обчислювальної потужності. Для зручності опису надалі припустимо, що функції $Q(N)$ та $C(N)$ описують кожен вузол, а не сумарну кількість вузлів обчислювальної потужності N [47, 53].

Критерієм, що визначає мінімальний час на обчислення всіх пакетів, є умова:

$$|Q_i(N) - C(N)| \rightarrow 0,$$

де i – елемент з багатьох пакетів,

j – елемент з безлічі вільних клієнтів.

Загальний час обчислення етапу становитиме:

$$T_{stage} = t \cdot \sum_{i \in 1, j \in [1, \dots, C_{all-count}]}^{Q_{all-count}} \left(\frac{Q_i}{C_j} \right)$$

де t – час обчислення еталонного пакета (сек) (щодо якого ведуться оцінки обчислювальної потужності вузлів та пакетів);

$Q_{all-count}$ – загальна кількість пакетів (шт.);

$C_{all-count}$ – загальна кількість обчислювальних вузлів (шт.).

Якщо зважити на затримки в ГРІД-системі, то загальний час обчислення етапу складе:

$$T_{stage} = \sum_{i=1}^{Q_{all-count}} \left(t + k + \frac{r}{r} \cdot \left(1 + p \left(z \left(c, \frac{r}{e}, Q_i \cdot t \right), n \right) \right) \right) \cdot \left(\frac{Q_i}{C_j} \right)$$

$$j \in [1, \dots, C_{all-count}],$$

$$|Q_i(N) - C_j(N)| \rightarrow 0,$$

де t - час обчислення еталонного пакету (с) (щодо якого ведуться

оцінки обчислювальної потужності вузлів та пакетів);

$Q_{\text{all-count}}$ – загальна кількість пакетів (шт.);

$C_{\text{all-count}}$ – загальна кількість обчислювальних вузлів (шт.);

k – затримка перед початком обміну (с);

r – обсяг пакета (біт);

e – швидкість передачі з боку сервера (біт у секунду).

Критерій розподілу етапу на C потрібно вузлів такий же, як і в у разі гомогенної GRID-системи:

$$S(0, c, t, e, r, n)^{c:[1, \dots, C_{\text{max}}]} \rightarrow \max$$

Визначемо $q = r / e \cdot t$, і також прийнемо число каналів обробки запитів $n=1$. Тоді:

$$S(0, c, t, e, r, 1) = \frac{1}{\left(\frac{1}{c} + \frac{k}{t} + q \cdot (1 + p(qc, 1) \cdot m(qc, 1))\right) \cdot \left(\frac{Q_i}{C_j}\right)} \quad (2.15)$$

$$m(qc, 1) = \frac{\frac{q^2 c^2}{1 \cdot 1 \cdot c \cdot t \cdot d(1 - qc)^2}}{1 + qc + \frac{q^2 c^2}{\max(1 - qc, 0)}} \quad (2.16)$$

При $qc < 1$ довжина черги до сервера складе:

$$m(qc, 1) = \frac{q^2 c^2}{1 - qc} \quad (2.17)$$

Імовірність виникнення черги становитиме:

$$p(qc, 1) = 1 - \frac{\sum_{h=0}^1 \frac{\left(\frac{(qc)^h}{h!}\right)}{\sum_{k=0}^1 \frac{(qc)^k}{k!} + \frac{(qc)^{(1+1)}}{1! \cdot (1 - qc)}};$$

$$p(qc, 1) = 1 - \left(\frac{1}{1 + qc + \frac{q^2 c^2}{1 - qc}} + \frac{qc}{1 + qc + \frac{q^2 c^2}{1 - qc}} \right);$$

$$p(qc, 1) = q^2 c^2 \quad (2.18)$$

Підставимо значення (17, 18) у формулу (2.6).

$$S_{heterogeneous}(0, c, t, e, r, 1)$$

$$= \frac{1}{\sum_{i=1}^{Q_{all-count}} \left(\frac{1}{c} + \frac{k}{t} + q \cdot \left(1 + q^2 c^2 \frac{c^2 \cdot q^2}{1 - qc} \right) \cdot \left(\frac{Q_i}{C_j} \right) \right)}$$

$$\frac{\partial S_{heterogeneous}}{\partial c} \rightarrow \max: c = \frac{t \cdot e}{2 \cdot r} \cdot \sum_{i=1}^{Q_{all-count}} \left(\frac{C_i}{Q_j} \right)$$

Якщо врахувати, що між етапами змінюються лише значення e та t , то кількість обчислювальних вузлів, що дає максимальний приріст прискорення обчислень, визначатиметься, як:

$$C_{heterogeneous}(t, e, r, C_{max}, C(N), Q(N), Q_{all-count})$$

$$= \left\{ \begin{array}{l} 1, t < \frac{2 \cdot r}{e \cdot \sum_{i=1}^{Q_{all-count}} \frac{C_i}{Q_j}} \\ \frac{e \cdot \sum_{i=1}^{Q_{all-count}} \frac{C_i}{Q_i}}{2r} \cdot t, t \in \left[\frac{2 \cdot r}{e \cdot \sum_{i=1}^{Q_{all-count}} \frac{C_i}{Q_j}} \cdot C_{max}, C_{max} \right]; \\ C_{max}, t > \frac{2 \cdot r}{\sum_{i=1}^{Q_{all-count}} \frac{C_i}{Q_j}} \cdot C_{max} \end{array} \right\}$$

$$j \in [1, \dots, C_{all-count}],$$

$$|Q_i(N) - C_i(N)| \rightarrow 0.$$

2.6 Аналіз продуктивності обчислювальних вузлів з багатоядерними процесорами

Якщо обчислювальний вузол представляє собою систему з багатоядерним процесором, то з'являється додатковий параметр у вигляді кількості ядер в обчислювальному вузлі, який впливає на характеристики обчислювального вузла та GRID-системи в цілому.

Припустимо, що багатоядерний процесор складається з кількох ядер,

кількість яких може змінюватись. Об'єднання кількох ядер дає нелінійний приріст продуктивності не тільки при паралельних обчисленнях, а й при послідовних. Функція продуктивності при послідовних обчисленнях від кількості ядер у процесорі виражається формулою (2.19) [9]:

$$P(c) = \sqrt{c}. \quad (2.19)$$

Можливі варіації формули (2.19), але він не вносять суттєвих змін.

Прискорення обчислень для кількох симетричних багатоядерних процесорів у загальному випадку складе:

$$S = \frac{1}{\frac{a}{P(q)} + \frac{(1-a) \cdot q}{P(q) \cdot c}} \quad (2.20)$$

де a – частка послідовних обчислень,

q – кількість ядер у процесорі,

c – кількість процесорів (обчислювальних вузлів).

Якщо застосувати оцінку (2.20) до GRID-систем, то прискорення обчислень (2.6) набуде вигляду:

$$S(a, c, k, t, e, r, n, q) = \frac{1}{\left(\frac{q}{P(q) \cdot c} (1-a) \cdot \left(\frac{q}{P(q) \cdot ct} + \frac{r}{e \cdot t} \cdot \left(1 + p \left(z \left(c, \frac{r}{e} t \right), n \right) \cdot m \left(z \left(c, \frac{r}{e} t \right), 1 \right) \right) \right) \right)} \quad (2.21)$$

де q – кількість ядер у процесорі обчислювального вузла,

P – функція оцінки продуктивності при послідовних обчисленнях від кількості ядер (19).

Прискорення розрахунків в GRID згідно залежності (2.21) проілюстровано (рисунок 2.6). Як приклад для побудови кривих 1-3 взяті наведені вище параметри (розділ 2.2).

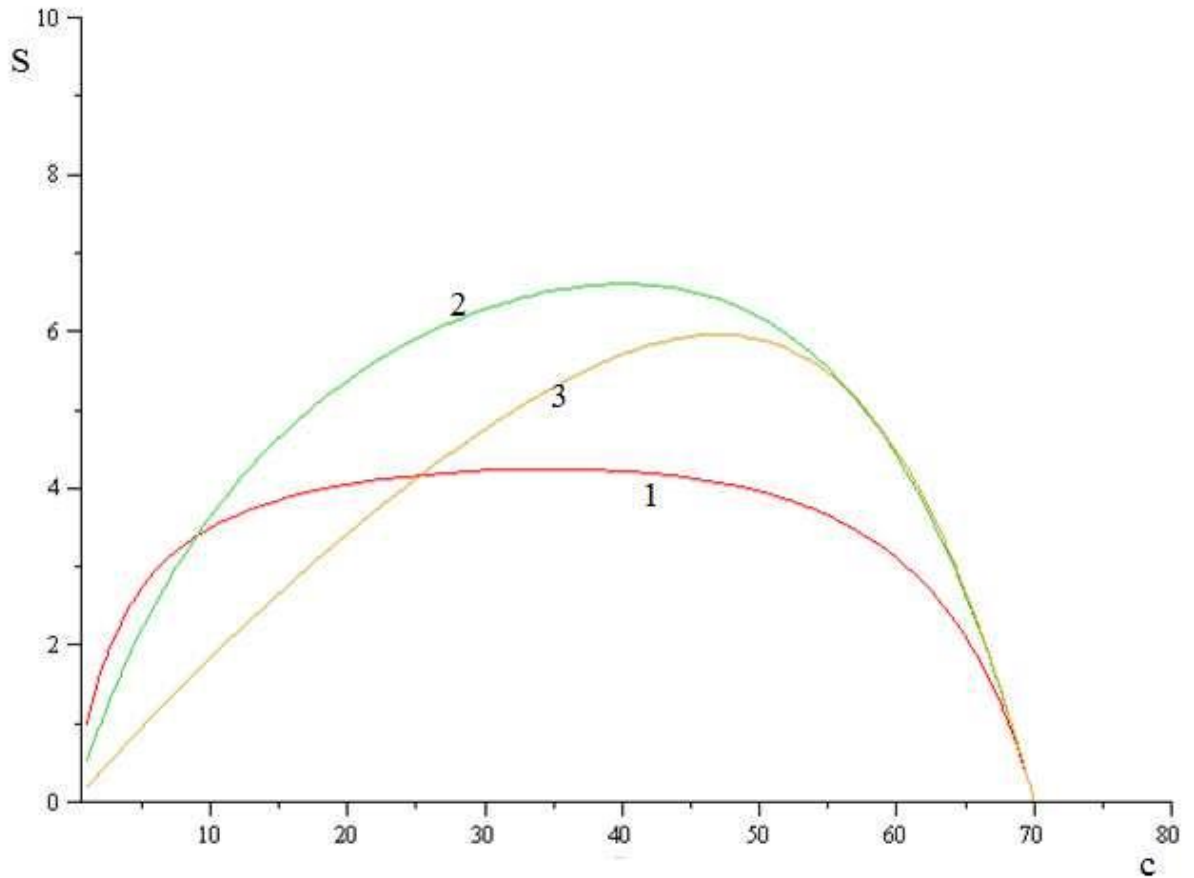


Рисунок. 2.6 – Прискорення розрахунків за (2.21). 1 - $q=1$; 2 - $q=5$; 3 - $q=40$

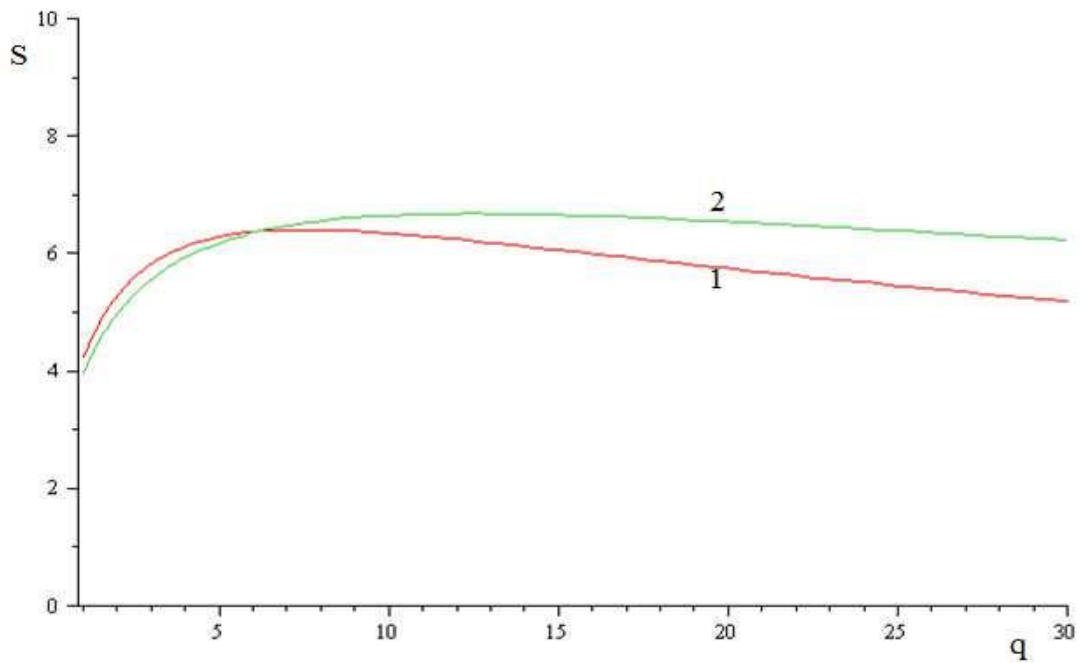


Рисунок 2.7 – Ілюстрація прискорення розрахунків за (2.21). 1 - $c=30$; 2 - $c=50$

У такому разі при визначенні точки максимуму прискорення обчислень

кількістю ядер робиться істотний внесок. Як можна бачити на рисунку 2.7 прискорення обчислень при розподілі одного та тієї ж кількості ядер по 50 обчислювальним вузлам у визначений момент перевищує максимум, досяжний при розподілі по 30 обчислювальним вузлам.

Для прискорення обчислень у GRID-системах можна прийняти увага три загальні рекомендації:

- передавати за одне з'єднання між обчислювальним вузлом та сервером кілька пакетів;
- зменшити обсяг пакетів;
- збільшити кількість паралельних потоків для обробки обчислювальних вузлів.

Передача кількох пакетів за одне з'єднання дозволить уникнути зайвих тимчасових затримок T_1 у стільки разів, скільки буде передаватися за один раз пакетів.

Зменшення обсягу пакетів зменшить тимчасову затримку T_2 та дозволить швидше обмінюватися даними серверу та обчислювальним вузлам.

Кожен потік обробки, що додається, бере на себе частину потоку запитів від обчислювальних вузлів, що дозволяє не допустити падіння прискорення обчислень якнайдовше.

Вказані вище заходи дозволяють лише не допустити раннього падіння прискорення обчислень та наблизити цей показник до значення, одержуваного за теорією Амдала. Для ще більшого наближення до закону Амдала необхідна переробка алгоритму розв'язуваної задачі з метою зменшення частки послідовних обчислень.

2.7 Аналіз навчання нейронних мереж у GRID-системах

Ставиться завдання проаналізувати прискорення навчання нейронних мереж на кінцевому наборі прикладів з наявними відповідями при

застосування генетичного алгоритму, що розпаралелюється на GRID-систему.

На рисунку 2.8 зображено загальну схему пристрою штучного нейрона [18], яка надалі закладається в основу нейронної мережі, можливий приклад якої зображено на рисунку 2.9. На вхід кожного нейрона в процесі обчислень подається вектор $x = (x_1 \dots x_n)$, після чого останній множиться на вектор вагів нейрона (скалярний твір векторів) та над результатом перемноження виконується певна функція $F(S)$, що є вихідним сигналом нейрона. Як показано на рисунку 2.9, безліч нейронів з'єднуються в одну велику мережу, в якій вихід одного нейрона відповідає входам кількох нейронів в іншому шарі. На входи першого, вхідного шару подається приклад для навчання, а результат, що знімається з вихідного шару, порівнюється з еталонним. Після порівняння виконується коригування значень вагів нейронів у мережі.

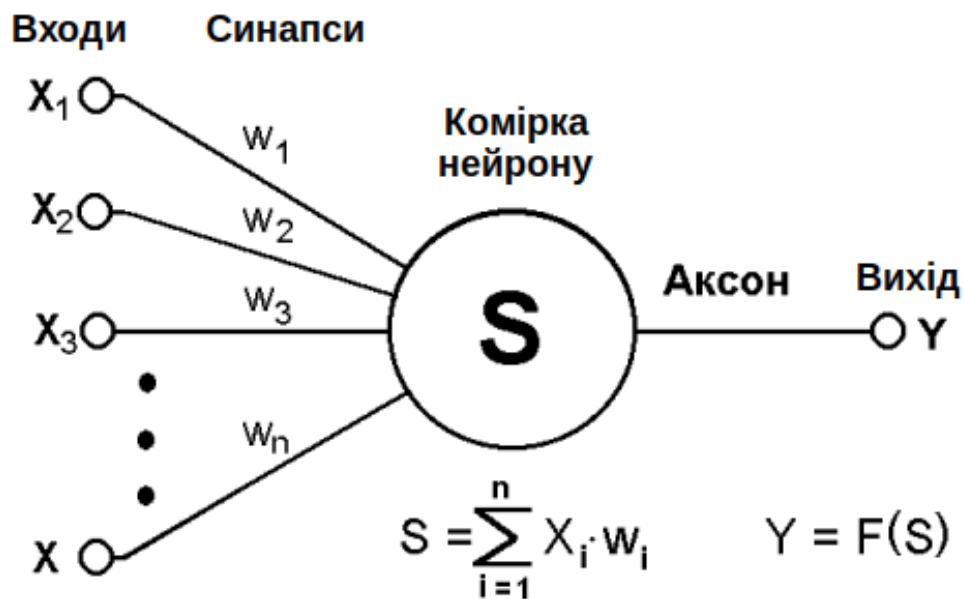


Рисунок 2.8 – Штучний нейрон [18]

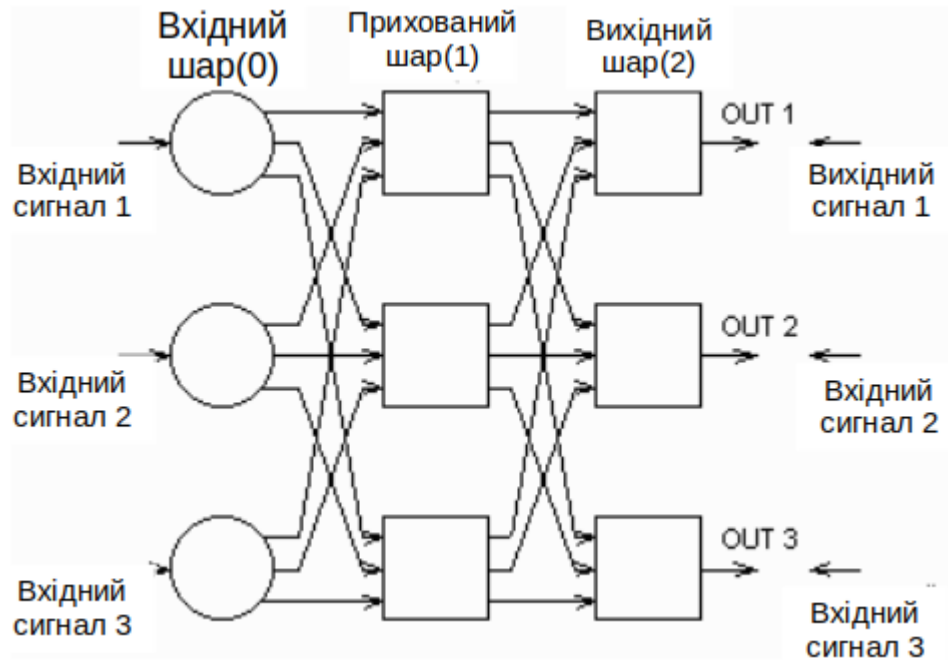


Рисунок 2.9 – Приклад пристрою перцептрону [18]

На вхід подається вектор:

$$\vec{x} = (x_1 \dots x_n),$$

а на виході повинен виходити результат, максимально наближений до:

$$\vec{y} = (y_1 \dots y_m),$$

Позначимо кількість прикладів на навчання, як k . Тоді необхідно знайти найбільш підходящі значення вагів у задачі:

$$\begin{cases} NN(\vec{x}_1) \rightarrow \vec{y}_1 \\ NN(\vec{x}_k) \rightarrow \vec{y}_1 \end{cases} \quad (2.22)$$

де NN – результат обчислень нейронної мережі.

У системі (2.22) потрібно знайти значення всіх проміжних ваг зв'язків нейронів. Завдяки хорошій розпаралелюваності для вирішення задачі одними з найбільш підходящими методами є генетичні алгоритми (ГА), здатні отримувати глобальне оптимальне рішення, необхідне системи (2.22). Для точності та швидкості роботи ГА важлива кількість генерованих особин та кількість оброблюваних поколінь, що потребує достатньо великої кількості

комп'ютерного часу. Великі обчислювальні потужності може надати мережа розподілених обчислень, якщо алгоритм дозволяє виконувати дії незалежно одна від одної дії незалежно одна від одної.

Розпишемо логіку виконання генетичного алгоритму на розподіленій мережі:

- формування задачі та складання рівняння (2.22);
- генерація особин у кількості 1;
- перевірки особин на нейронній мережі із зазначеними значеннями та ошук найбільш підходящих особин для наступного покоління;
- мутація особин;
- повторення циклу 2-4 w-раз або до знаходження рішення.

Сервер перебирає кроки 1,2,4, що дозволяє розділити 3-й крок на деяку кількість окремих обчислювачів в мережі. Скористаємося теорією Амдала [20] для розрахунку приросту мережі розподілених обчислень:

$$S = \frac{1}{a + \frac{1-a}{c}} \quad (2.23)$$

де a – частка послідовної частини завдання,

r – кількість одночасно працюючих обчислювачів.

Введемо позначення:

l – число особин, що генеруються за один цикл (особина - вектор значень, які можуть бути можливим рішенням);

w – число циклів для пошуку рішення;

t_A – час отримання ваги (відстань до відповіді) однієї особини в одному обчислювальному вузлі;

t_S – час на всі операції з однією особою на стороні сервера;

t_q – час на 1 цикл = $r * t_S + r * t_A$ (без розпаралелювання);

T – максимальний час рішення = $q * t_q = qr (t_S + t_A)$;

a – частка послідовної частини задачі = $t_S / (t_S + t_A)$);

$$S = \frac{1}{a + \frac{1-a}{p}} = \frac{t_S + t_A}{pt_S + t_A} \quad (2.24)$$

При використанні завдання GRID-системи необхідно відзначити, що закон Амдала не враховує специфічні затримки, що виникають у середовищі передачі і під час обробки даних клієнтами і сервером.

Проілюструємо прискорення розрахунків системи за законом Амдала та залежно (2.6). Як приклад вкажемо такі параметри (розділ 2.2):

- частка послідовних обчислень – 0.2;
- кількість обчислювальних вузлів варіюється;
- затримка перед початком передачі даних – 0.5 секунд;
- час розрахунку одного пакета на одному обчислювальному вузлі – 70 секунд;
- обсяг пакету – 30 Кбіт;
- швидкість передачі з боку сервера – 30 кілобіт на секунду;
- кількість каналів обробки обчислювальних вузлів – 1.

Ці значення дозволяють наочно показати зміну прискорення при порівняно невеликій кількості обчислювальних вузлів. На рисунку 2.10 представлений графік залежності прискорення від кількості обчислювальних вузлів. На початковому етапі збільшення кількості обчислювальних вузлів (0-20000) спостерігається прискорення обчислень. На середньому етапі значний зріст (20000-50000) приріст прискорення помітно зменшується. На останньому етапі (50000-70000) прискорення починає падати через дедалі більше проявляючого себе третього типу затримки (виникнення черги до сервера). Точка максимуму прискорення в залежності від кількості обчислювальних вузлів приходиться на середній етап і може бути обчислена за формулою (2.7).

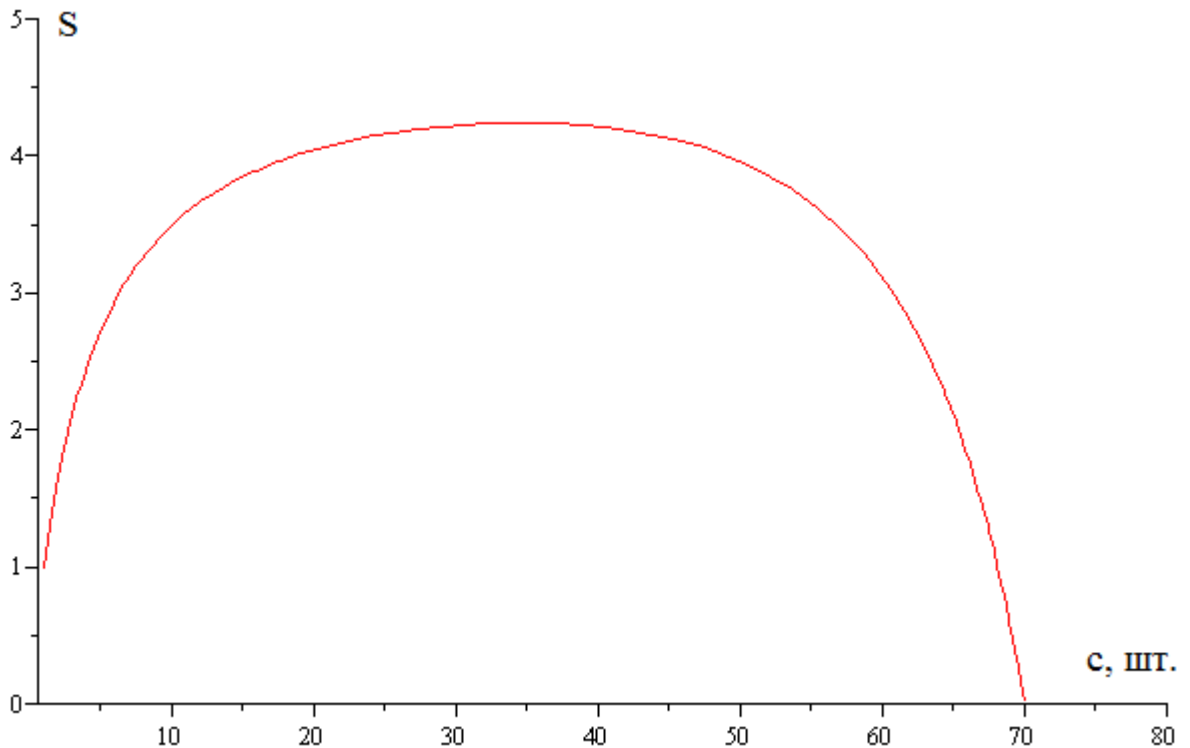


Рисунок 2.10 – Ілюстрація прискорення обчислень в залежності від кількості обчислювачів

Вплив часу обробки однієї особини з генетичного алгоритму на стороні сервера представлено на рисунку 2.11. Час обробки на стороні сервера включено до послідовної частини завдання:

$$a = \frac{t_s}{t_s + t_A}$$

де a – частка послідовної частини задачі,

t_A – час отримання ваги (відстань до відповіді) однієї особини на одному обчислювальному вузлі,

t_s – час на операції з однією особиною на стороні сервера.

При прагненні значення часу обробки на сервері до нуля прискорення обчислень спрямує до нескінченності, при прагненні часу обробки на сервері до нескінченності прискорення обчислень прямує до нуля.

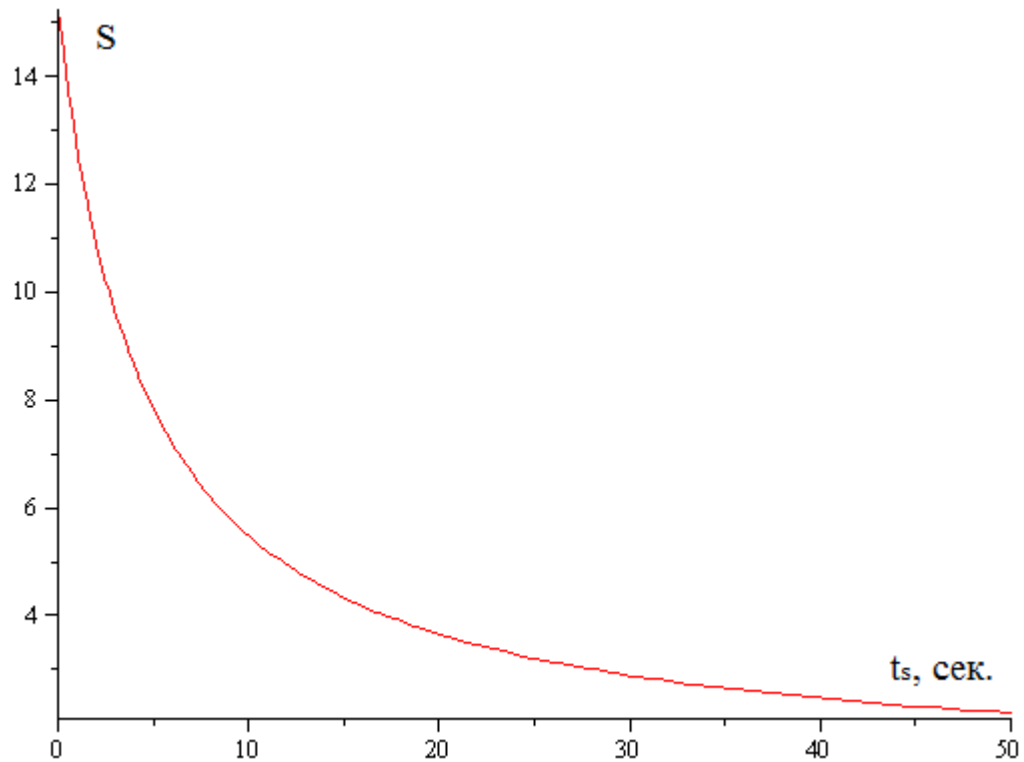


Рисунок 2.11 – Ілюстрація прискорення обчислень в залежності від часу обробки однієї особи з генетичного алгоритму на стороні сервера

На рисунку 2.12 представлений графік залежності прискорення від кількості обчислювальних вузлів та часу обробки однієї особи на стороні сервера (для наочності змінені параметри: час розрахунку одного пакета на одному обчислювальному вузлі - 70 с, швидкість передачі сторони сервера – 30 Кбіт на секунду). Поведінка графіка по осі s пояснюється тими самими явищами, які згадані на рисунку 2.10. При збільшенні часу обробки однієї особи на сервері для кожної особи серверу необхідно приділяти більше часу, що йде як на внутрішньосистемні дії, так і на модифікацію особи для наступного покоління (операції мутації та кросинговера). Зі зменшенням часу обробки на стороні сервера відповідно зменшується і час виконання частки послідовної частини у завданні, що позитивно відбивається на загальному часі розв'язання задачі.

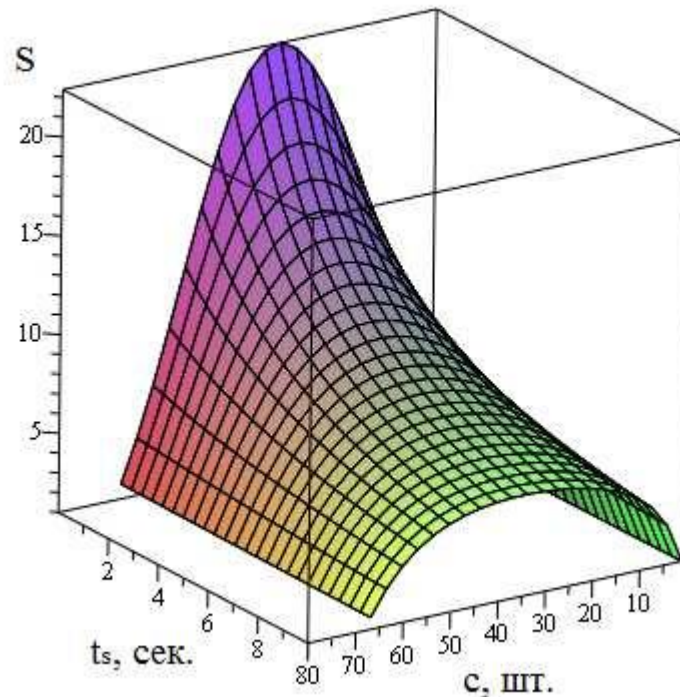


Рисунок 2.12 – Ілюстрація прискорення обчислень (S) залежно від кількості обчислювачів (c) та часу обробки однієї особи на сервері (T_s)

2.8 Взаємозв'язок обсягу послідовних обчислень та часу обробки пакетів у GRID-системах

В рамках методу (2.6) можливий розгляд взаємозв'язків різних параметрів GRID-систем. Інтерес представляють значення, при яких прискорення обчислень $S = 1$, $S > 1$ і $S < 1$ [10].

На рисунках 2.13...2.15 представлена залежність між часткою послідовних обчислень та часом розрахунку одного пакета в рамках методу (2.6).

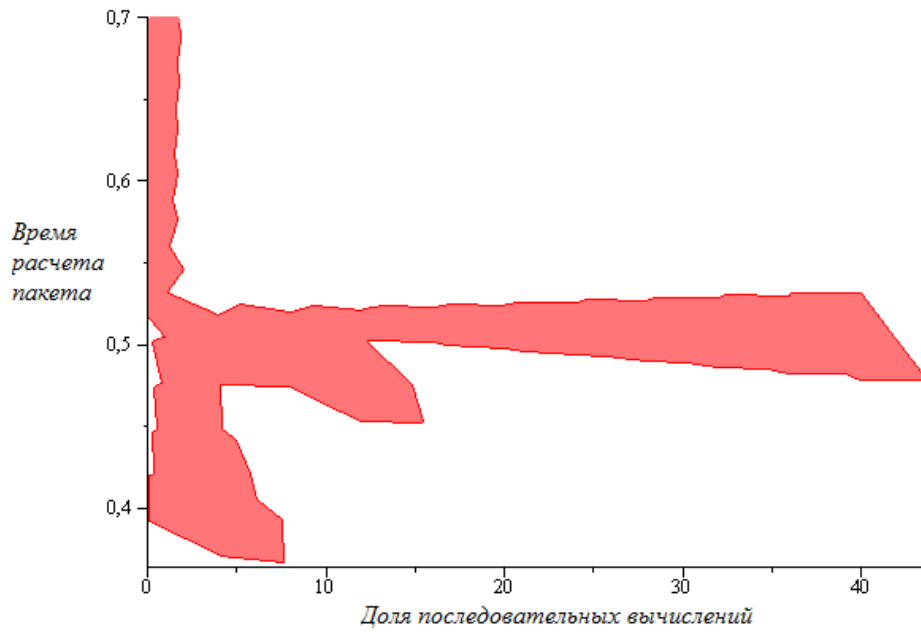


Рисунок 2.13 – Залежність між часткою послідовних обчислень та часом розрахунку одного пакета. $S=1$

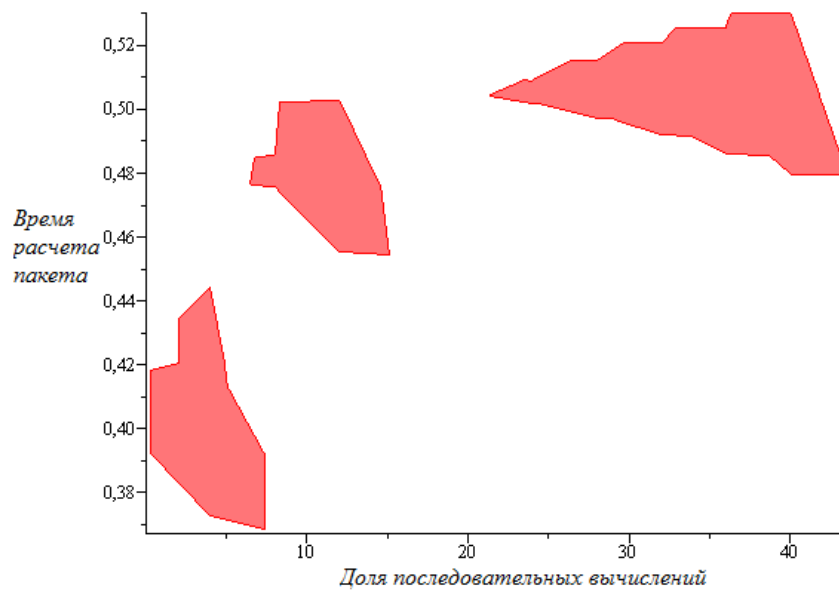


Рисунок 2.14 – Залежність між часткою послідовних обчислень та часом розрахунку одного пакета. $S = 2$

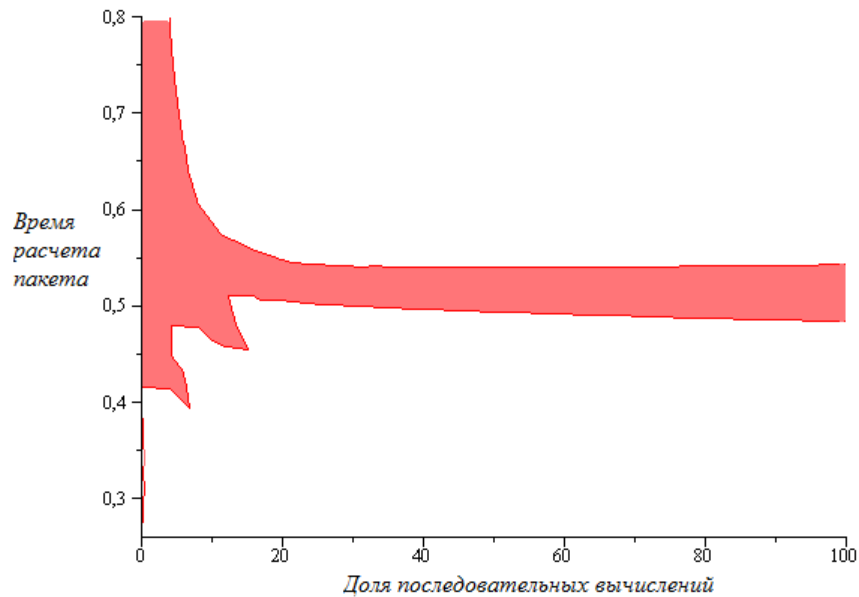


Рисунок 2.15 – Залежність між часткою послідовних обчислень та часом розрахунку одного пакета. $S = 0.5$ (уповільнення обчислень)

Виділені темним кольором області (рисунки 2.13-2.15) представляють собою зрізи поверхні, що утворюється при варіюванні частки послідовних обчислень, часу розрахунку пакету та прискорення обчислень. Внутрішні області зрізів відповідають більш високому прискоренню обчислень, якого можна досягти за відповідних параметрів GRID-системи та розв'язуваних завдань. Чим більше прискорення, тим сильніші обмеження накладаються на зазначене співвідношення параметрів. На рисунку 2.14 видно, що для виконання необхідних умов (прискорення обчислень $S = 2$) зрізи локалізуються у три окремих області. Подальший аналіз поверхні може дозволити встановити інші глибші співвідношення в параметрах GRID-систем.

2.9 Динаміка характеристик завантаженості GRID-систем у процесі обробки даних

Під час розгляду ступеня завантаженості системи проявляється певна залежність між завантаженістю сервера і обчислювальних вузлів [10]. На

рисунках 2.16...2.19 наведено результати аналізу імітаційної моделі за різного часу розрахунку пакетів на стороні обчислювальних вузлів. Під імітаційним моделюванням розуміється окремий випадок математичного моделювання, при якому система, що вивчається, замінюється моделлю, з достатньою точністю описує реальну систему. Імітуються елементарні явища, складові процес, зі збереженням їх логічної структури та послідовності перебігу. По вертикальній осі відзначено чергу пакетів до сервера щодо загальної кількості обчислювальних вузлів, по горизонтальній осі відзначено частку обчислювальних вузлів, зайнята розрахунками.

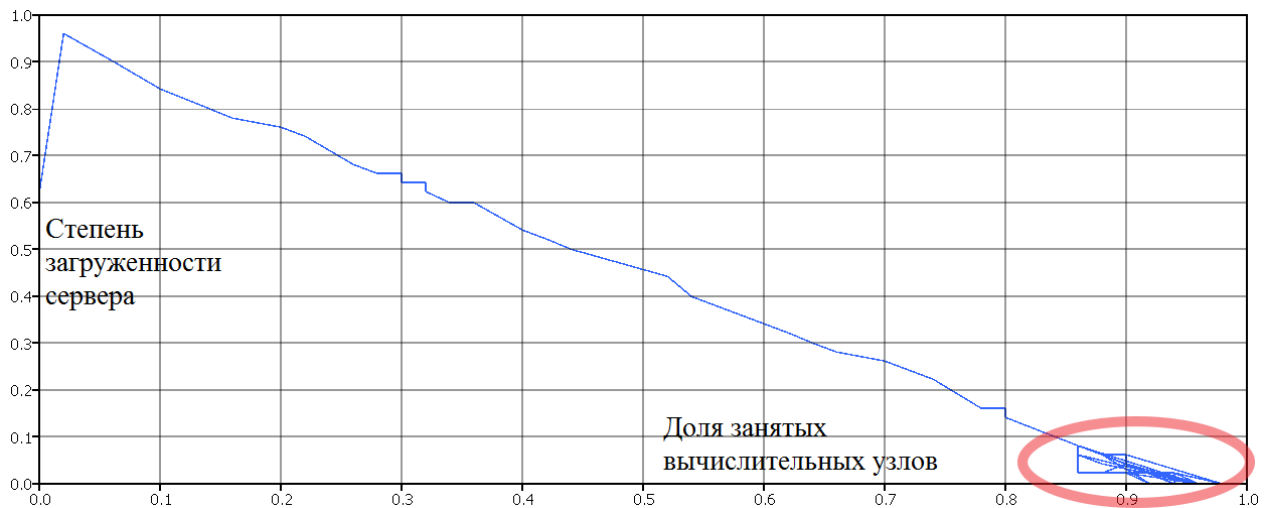


Рисунок 2.16 – Завантаженість сервера та обчислювальних вузлів у ході паралельної обробки інформації

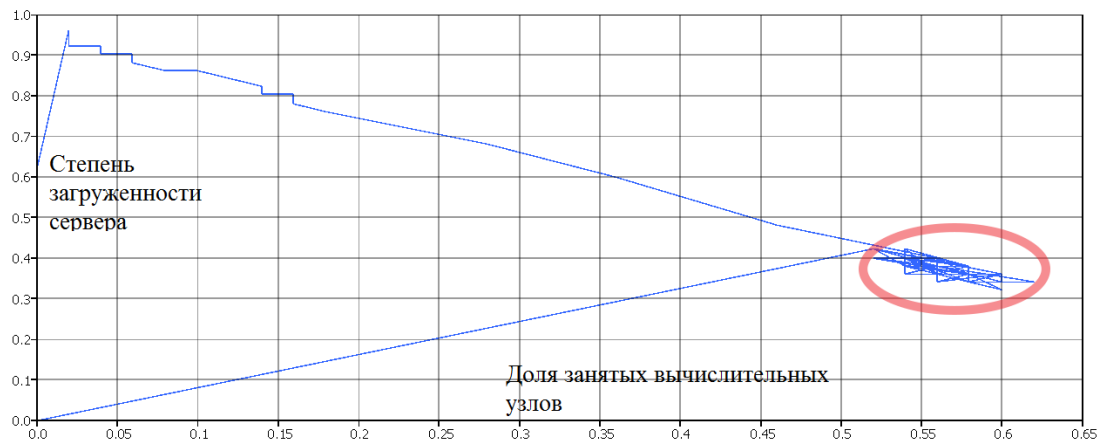


Рисунок 2.17 – Завантаженість сервера та обчислювальних вузлів у ході паралельної обробки інформації

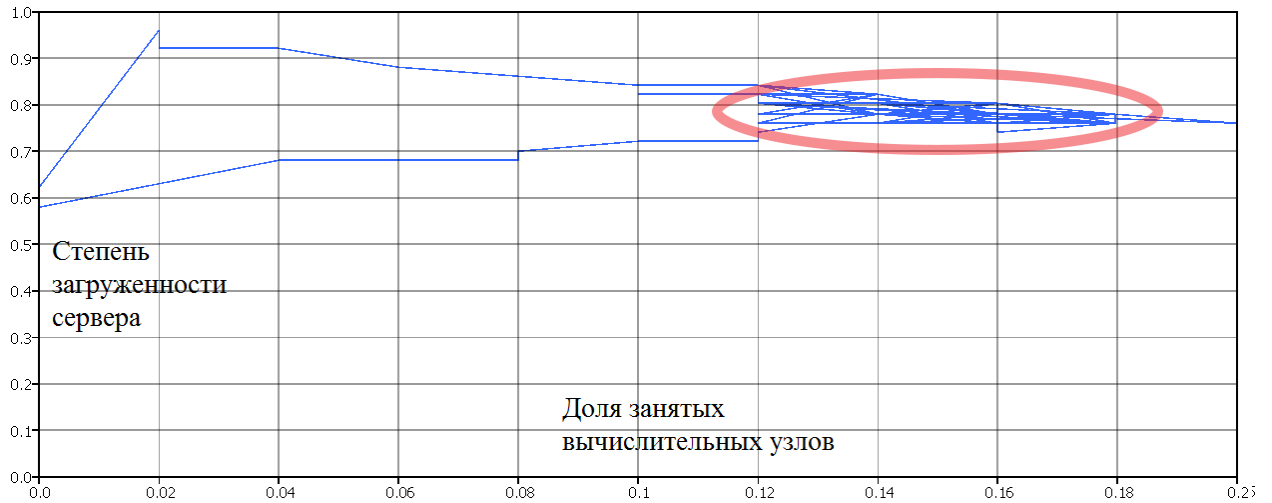


Рисунок 2.18 – Завантаженість сервера та обчислювальних вузлів у ході паралельної обробки інформації

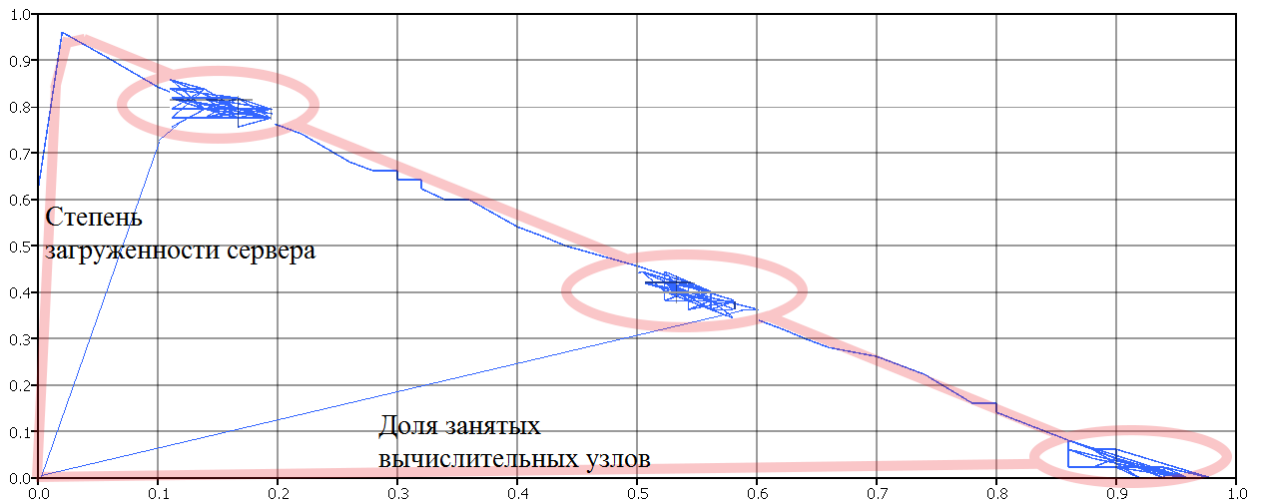


Рисунок 2.19 – Области стабільності GRID-системи в ході паралельної обробки інформації

На графіку рисунка 2.17 час обробки пакета в 3,6 рази більший, ніж на графіку рисунка 2.16, але на рисунку 2.18 – у 6 разів. Як видно з побудов рисунка 2.19, GRID-система під час вирішення задачі знаходиться у визначеній області станів, розташованої на прямій в інтервалах від повної завантаженості сервера та вільних клієнтів до протилежного стану (вільний сервер та повністю завантажені клієнти).

3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ВИКОРИСТАНИХ МЕТОДИК ОЦІНКИ ПРОДУКТИВНОСТІ ГРІД-СИСТЕМ. ПИТАННЯ ПРАКТИЧНОГО ЗАСТОСУВАННЯ

3.1 Розробка архітектури та структурної схеми моделюваної GRID-системи. Імітація вирішення завдань у GRID-системі

У цьому розділі досліджується, в умовах, наближених до реальних, прискорення обчислень, що досягається при використанні GRID-системи та вплив тимчасових затримок, описаних у другому розділі справжньої роботи. На основі зазначених у розділах 1.3, 2.1 та 2.2 характеристик, а також проаналізованої архітектури GRID-системи визначено топологію мережі, представлену на рисунку 3.1.

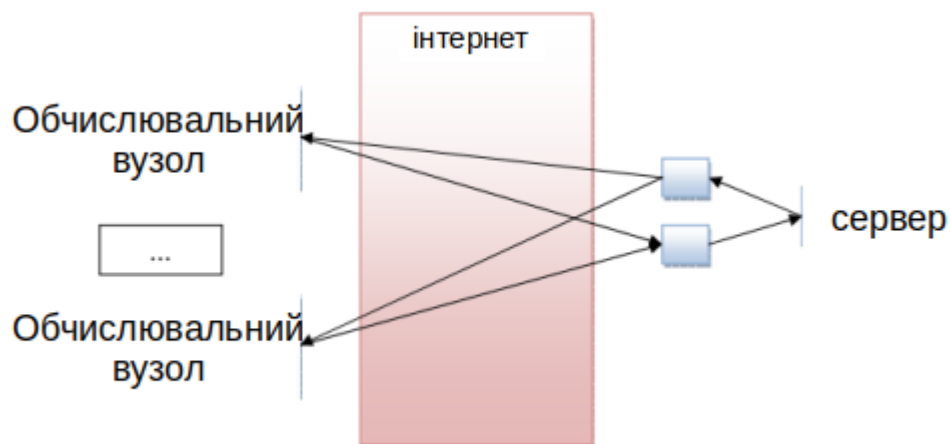


Рисунок 3.1 – Топологія мережі GRID-системи

З метою забезпечення максимальної наближеності умов проведення експериментальних досліджень до реальних, при проведенні експериментів використовувалися наступні вхідні умови та обмеження [10]:

- сервер обробляє пакети від обчислювальних вузлів у режимі однієї спільної черги, по черзі розбираючись із запитом кожного вузла;
- тимчасові затримки на дії сервера, обчислювального вузла та

затримок середовища обміну даними представлені у вигляді нормального розподілу;

- кожен обчислювальний вузол з ймовірністю 0.01 може на невизначений час вийти з ладу, тобто. затримати розрахунки прийнятого пакета на випадковий проміжок часу за внутрішнім причин (перезавантаження, збої тощо).

Подібні умови та обмеження під час проведення експериментів обрані відповідно до найпоширеніших ситуацій, а також у зв'язку з необхідністю забезпечити наочність результатів.

Основною метою проведення експерименту ставиться завдання визначення точки максимуму прискорення обчислень та, відповідно, точки мінімуму затрачуваного на розрахунки часу. Точка максимуму прискорення обчислень одна із найважливіших точок, так як саме вона показує, до якої межі має сенс нарощувати кількість обчислювальних вузлів і у скільки разів може збільшитись прискорення розрахунків. Метод її визначення для GRID-систем наводиться в розділах 2.2 та 2.3.

Експерименти проводилися в середовищі імітаційного моделювання AnyLogic. AnyLogic включає в себе графічну мову моделювання, а також дає можливість користувачеві розширювати створені моделі за допомогою мови Java. Інтеграція компілятора Java до AnyLogic надає ширші можливості створення моделей. Оскільки імітаційна модель дозволяє провести велику кількість експериментів з різними початковими значеннями та станами моделі, такий підхід було взято як основний. Як початкові значення використовуються кількість пакетів, що обраховуються, і максимальна кількість клієнтів, які одночасно взаємодіють із сервером. Результати функціонування моделі – статистична інформація – обробляється з використанням математичних методів, з яких виводиться залежність прискорення моделі від кількості обчислювачів.

Як уже зазначалося у 2 розділі, під імітаційним моделюванням розуміється приватний випадок математичного моделювання, при якому

система, що вивчається, замінюється моделлю, з достатньою точністю описуючою реальну систему. Імітуються елементарні явища, що становлять процес, із збереженням їх логічної структури та послідовності протікання. Моделювана GRID-система складається з 3-х елементів: мережа, сервер і змінне число клієнтів. Час розрахунку пакетів має ймовірнісний характер і залежить від багатьох факторів, які приносять додаткові труднощі в опис системи загалом. Найбільш продуктивним і найменш витратним як за засобам, і за часом у разі є імітаційне моделювання.

Надалі під елементом імітаційної моделі розуміється обробник імітованих у системі заявок, що впливає на їх подальше переміщення по складеній структурою моделі. Елемент може затримувати заявку на якийсь час, визначати її подальше переміщення, а також змінювати її властивості.

Під блоком в системі, що моделюється, розуміється умовне об'єднання кількох елементів, які в реальній системі можуть бути логічно згруповані та віднесені до будь-якої частини системи.

Оскільки статистичні дані вимагають формалізації, в якості апарату моделювання використані принципи апроксимації функцій. Апроксимація функцій має деяку похибку, що найбільше буде помітно в кінці очікуваної функції приросту.

Модель досліджуваної GRID-системи складається із трьох основних блоків: сервер, середовище обміну інформацією та обчислювальні вузли. Базова ділянка мережі зображена на рисунку 3.2.

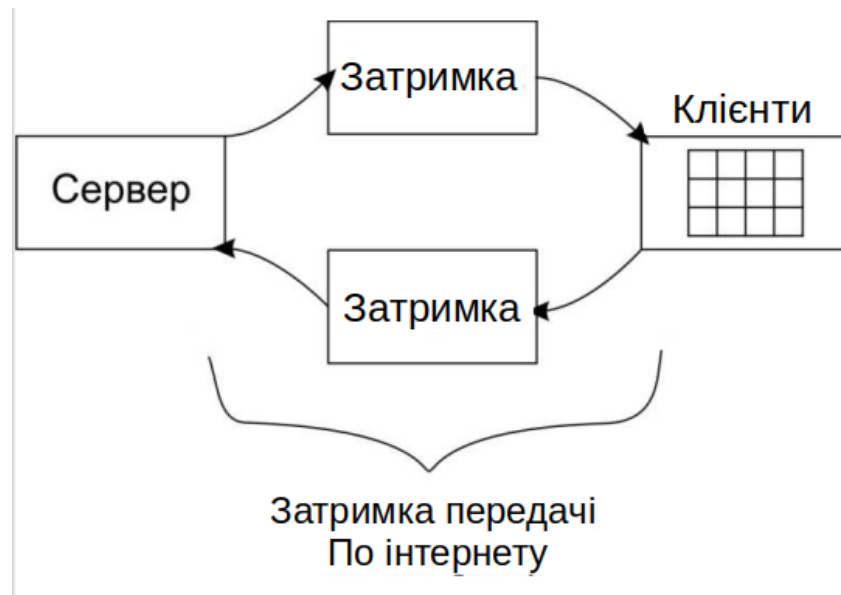


Рисунок 3.2 – Узагальнена схема ГРІД-системи, що моделюється

Середовище обміну інформацією (рисунок 3.2) представлено у вигляді двох затримок, об'єднаних у загальний клас «Затримок передачі по інтернету»: від сервера до обчислювальних вузлів та від обчислювальних вузлів до сервера. Елементи сервера та його топологія мережі узагальнені у блок "Сервер". Безліч обчислювальних вузлів, що беруть участь у розрахунках задач, об'єднаних у блок «Клієнти». Це об'єднання узгоджується з описаними в розділах 2.1...2.7 аспектами GRID-систем і носить якісний характер, так як всі обчислювальні вузли виконують одну і ту ж роль, будучи при цьому географічно розподіленими та незалежними один від одного.

Структура обчислювального вузла, що представлена на рисунку 3.3 на першому перемикачі перевіряється, чи вже зайнятий обчислювальний вузол розрахунками пакету або ж вільний. Якщо обчислювальний вузол зайнятий, триває перевірка наступного вузла. Ця перевірка не займає часу при моделюванні, тому вона еквівалентна відправленню пакета відразу ж цільовому вільному вузлу для розрахунків. Якщо обчислювальний вузол вільний, він приймає пакет для розрахунків та встановлює перемикач у позицію "зайнятий розрахунками". Далі встановлено затримку, еквівалентну часу розрахунку пакету. Для кожного вузла затримка може бути налаштована окремо. Після розрахунку перший перемикач встановлюється в положення

"вільний", а результати розрахунків надсилаються на сервер.

Обчислювальний вузол реалізований на основі базових елементів середовища моделювання та зібраний у єдиний елемент. Для спрощення процесу моделювання та більш зручного масштабування GRID-системи обчислювальні вузли були зібрані в групи по кілька десятків.

Рисунок 3.3 ілюструє структуру детермінованого обчислювального вузла, в якому не реалізовані ймовірність відключення від процесу обчислень (перезавантаження, вимикання, поломка чи інші події, що призводять до неможливості розрахунків). У моделі GRID-системи до структури обчислювальних вузлів внесено відповідні зміни, що враховують ймовірність відключення, що відображено на рисунку 3.4.

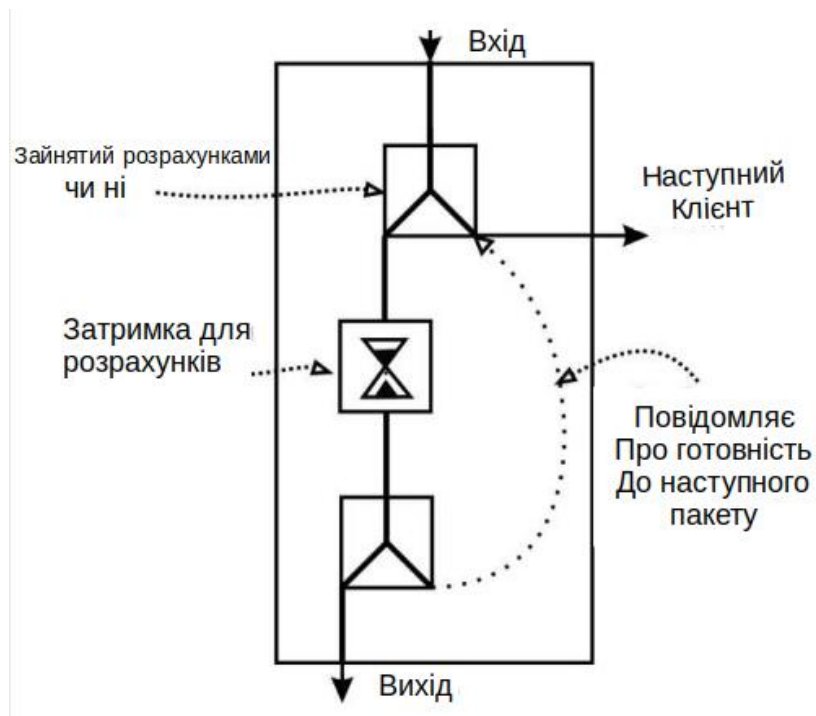


Рисунок 3.3 – Структура детермінованого обчислювального вузла

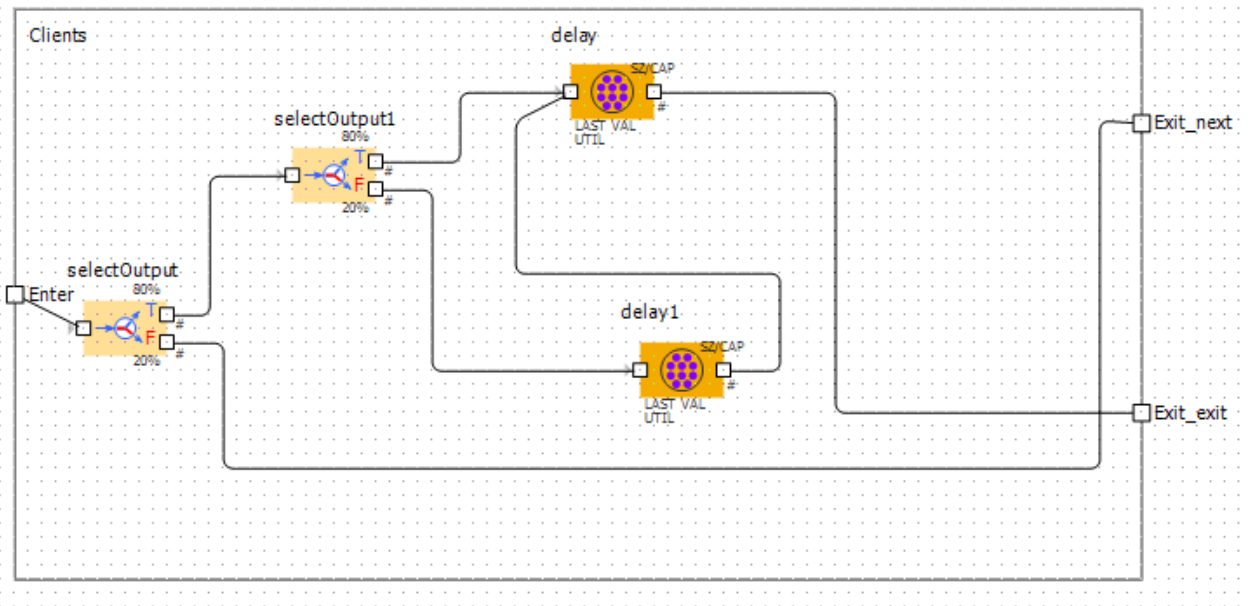


Рисунок 3.4 – Структура обчислювального вузла з ймовірністю збільшення стандартної затримки (delay1)

У структурі, що зображена на рисунку 3.4 елемент "delay1" відповідає за імітацію відключення від процесу обчислень.

Далі на рисунку 3.5 представлена остаточна схема реалізації моделі. Вся модель поділена на кілька блоків трьох можливих класів: «Обчислювальний вузол», «Інтернет» і «Сервер». Подана схема побудована для трьох обчислювальних вузлів, але передбачено можливість її розширення на будь-яку кількість клієнтів. Три обчислювальні вузли відзначені відповідно трьома блоками, що належать до «Групи клієнтів». У центрі знаходиться блок «Інтернет», що імітує канали передачі даних між сервером та обчислювальними вузлами. Праворуч розташований блок «Сервер», що імітує обробку пакетів з даними на стороні сервера.

Блок сервера умовно поділяється на групу з трьох верхніх та чотирьох нижніх елементів. Зверху знаходяться елементи, які вибирають із бази даних не розрахований пакет і надсилають його клієнтам. Нижні елементи приймають розраховані пакети від клієнтів, заносять результати в базу даних і змінюють значення лічильника, що показує, скільки пакетів залишилося обробити.

На рисунку 3.6 представлена структура моделюється GRID-системи, реалізована серед імітаційного моделювання AnyLogic.

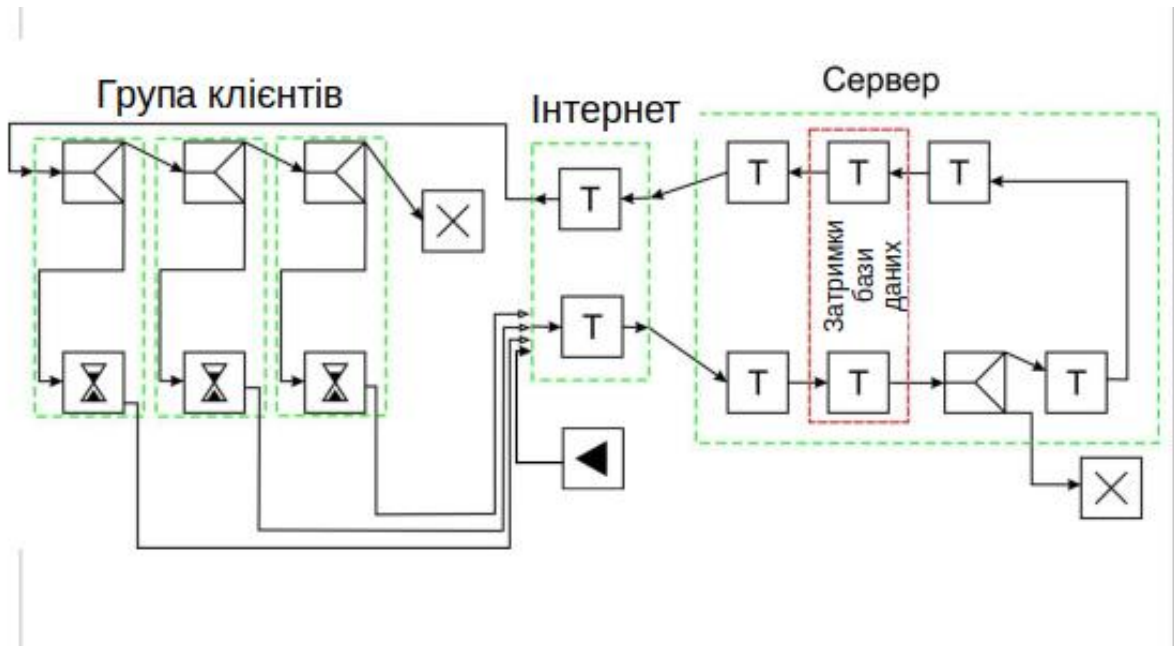


Рисунок 3.5 – Структура GRID-системи

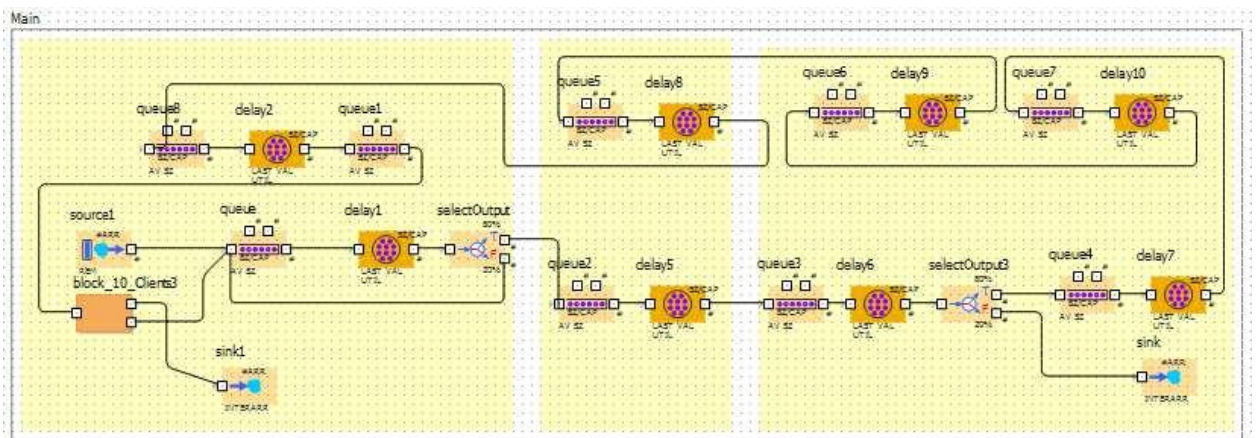


Рисунок 3.6 – Структура GRID-системи в AnyLogic

На рисунку 3.7 представлена докладніша структура сервера. Розглянемо зазначені елементи «delay» і кожна з них відповідає:

- «delay10» - черга запитів від вільних обчислювальних вузлів отримання пакета для розрахунків;
- "delay9" - запит до БД для отримання пакета;
- «delay8» - черга на постобробку пакетів і відправку обчислювальним

вузлам;

- «delay5» - черга з прийнятих пакетів, передобробка, перевірка та т.п. дії перед додаванням результатів БД;
- «delay6» – додавання результатів у БД;
- delay7 - затримка на внутрішні розрахунки сервера, пов'язані з обробленими пакетами.

Елемент "selectOutput3" на рисунку 3.7 визначає, чи є ще не оброблені обчислювальними вузлами пакети і, відповідно, чи потрібно подавати запит від вільного клієнта на отримання нового пакету.

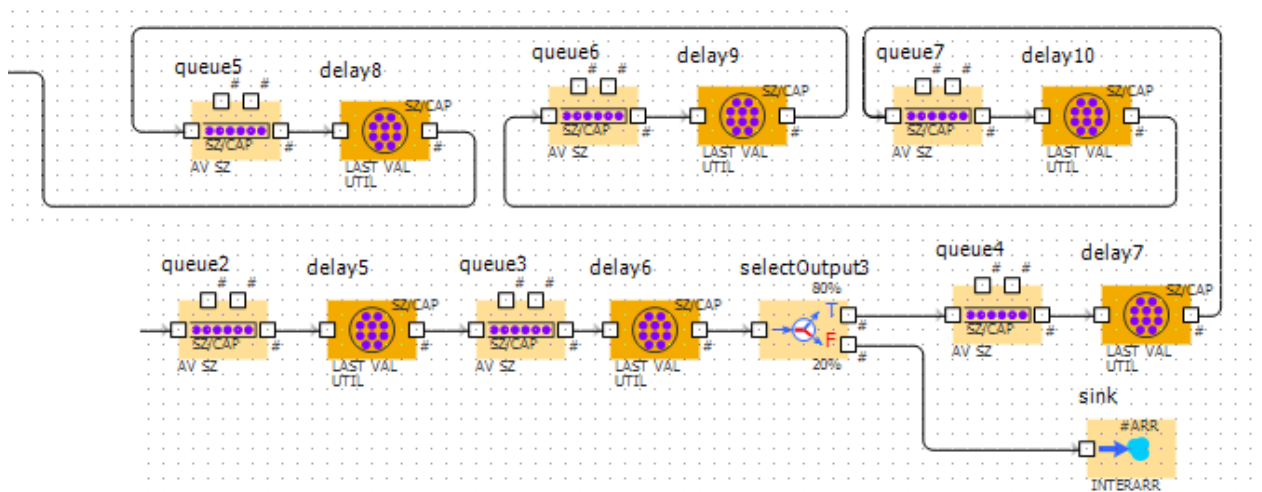


Рисунок 3.7 – Блоки затримки сервера на обробку результатів та запис у БД

Проведено низку експериментів, у процесі яких змінювалося кількість пакетів для розрахунків та кількість обчислювальних вузлів у моделі. Реалізована модель мережі описується за допомогою апарату мереж Петрі (рисунки 3.8, 3.9, 3.10) [10]. Мережі Петрі розроблялися для моделювання систем з паралельними взаємодіючими компонентами і добре підходять для опису GRID-систем. Маркування мережі відповідає пакетам даних, що вирішується в GRID-системі задачі. Вершини графа відповідають різним станам пакетів в системі. Структура графа відповідає структурі, представлений на рисунку 3.5 а варіації структури клієнта відповідають його різним структурам: з обліком збою (згідно структури, зображеній на рисунку 3.4), з урахуванням збоїв в цілому та кожного збою в окремо (згідно з

структури, зображеної на рисунку 3.4), без урахування збою (згідно з структурою, зображеної на рисунку 3.3).

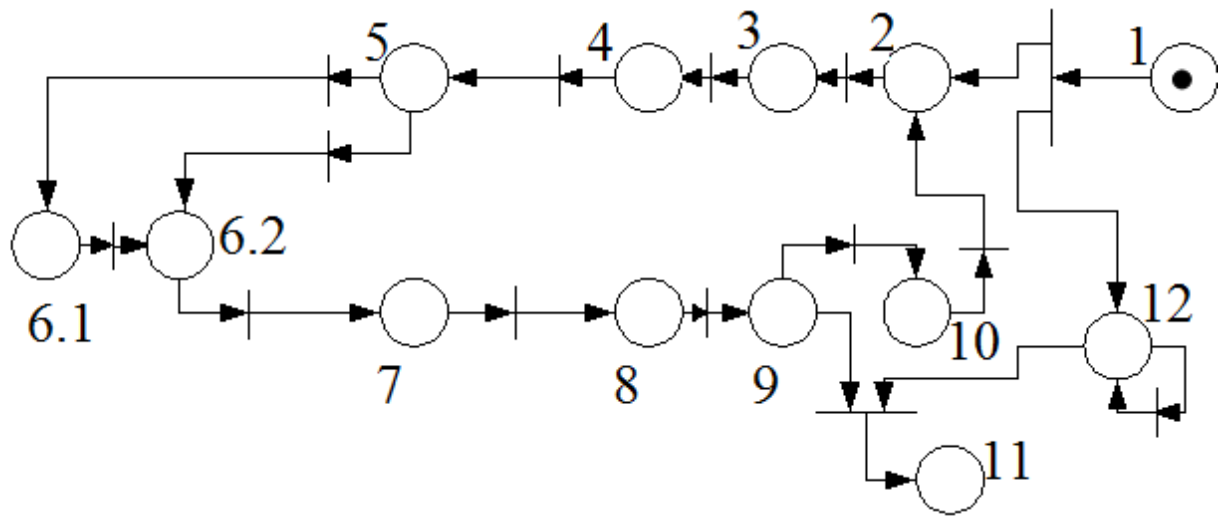


Рисунок 3.8 – Мережа Петрі з урахуванням збою у обчислювальних вузлах

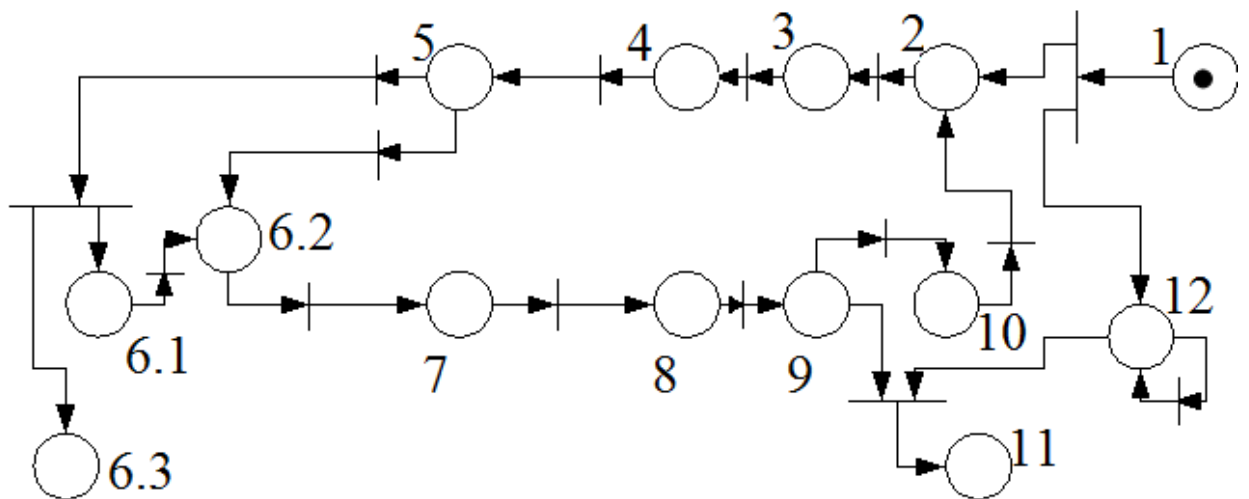


Рисунок 3.9 – Мережа Петрі з урахуванням збою у обчислювальних вузлах та обліком кожного збою

- черга із прийнятих пакетів, передобробка, перевірка та інші дії перед додаванням результатів БД;
- додавання результатів розрахунків у БД;
- затримка на внутрішні розрахунки сервера та аналіз необроблених пакетів;
- лічильник пакетів після закінчення розрахунків;
- лічильник пакетів, що у обробці.

У ході проведення експерименту були отримані чисельні дані, подані в таблиці 3.1 та рисунках 3.11, 3.12. Для кожних параметрів мережі проводили 1000 запусків моделі. У таблиці 3.1 представлені тільки мінімальний, максимальний і середній час по всім запускам моделі [10].

Таблиця 3.1 – Результати експерименту

Кількість обчислювальних вузлів	Мінімальний час	Максимальний час	Середній час	Прискорення
1	687	697	737	1,00
5	144	184	163	4,53
7	144	197	130	4,23
10	100	111	109	6,78
20	83	101	94	7,87
30	86	115	101	7,27

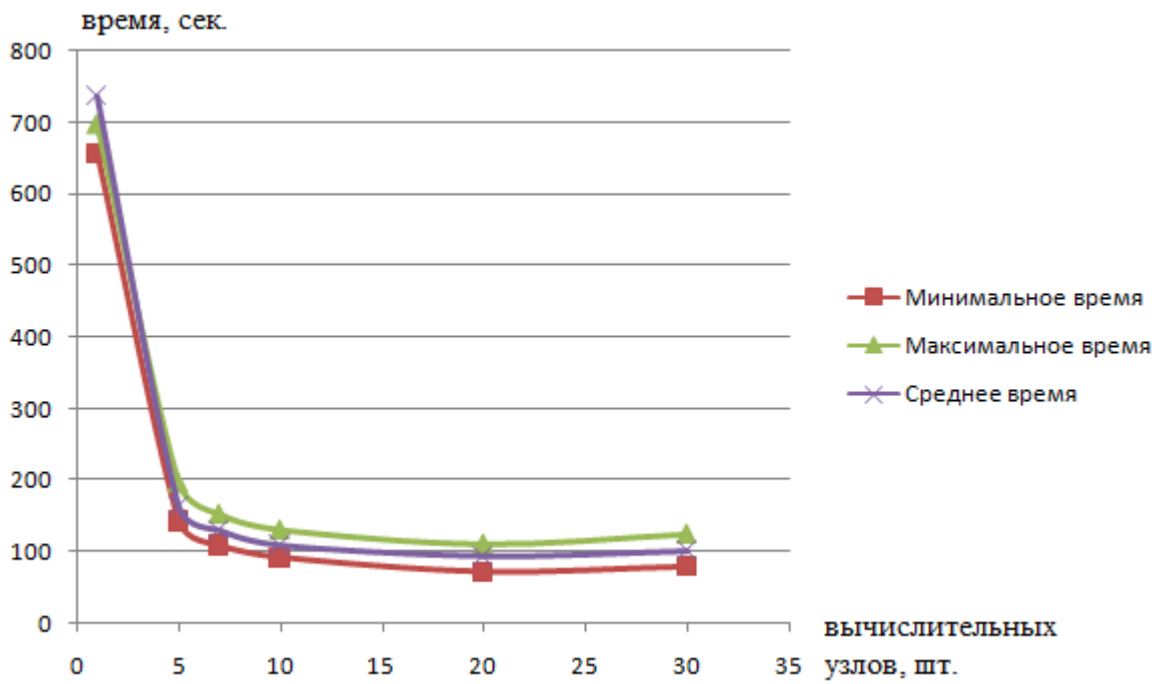


Рисунок 3.11 – Результати експерименту, час розрахунків

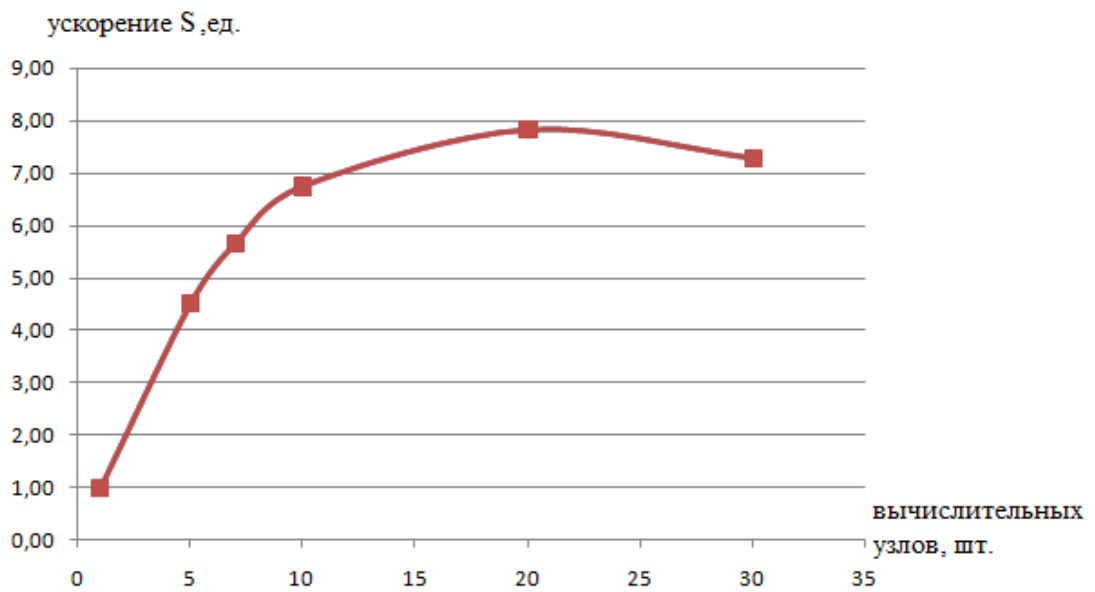


Рисунок 3.12 – Досягнуте прискорення під час кожного із запусків експериментальної моделі

Таблиця 3.2 – Значення прискорення обчислень за результатами експерименту та за формулою (2.6)

Кількість обчислювальних вузлів	Прискорення. [од.]	math_S/ [од.]
1	1,00	0,97
5	4,52	3,83
7	5,67	4,83
10	6,76	5,98
20	7,84	7,91
30	7,30	7,22

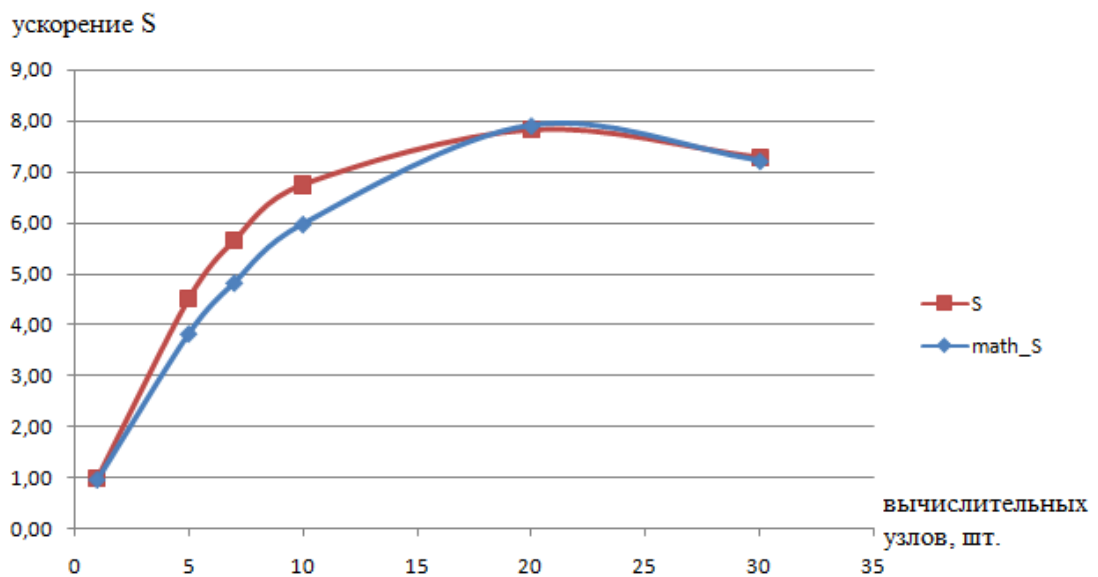


Рисунок 3.13 – Значення прискорення обчислень за результатами експерименту (S) та за формулою (2.6) (math_S)

На рисунку 3.13 та в таблиці 3.2 наведено значення прискорення обчислень за результатами експерименту та за формулою (2.6), що досягаються за рахунок розпаралелювання задачі на безліч обчислювальних вузлів. У в ході експерименту були отримані значення прискорення

обчислень при різну кількість обчислювальних вузлів. (рисунк 3.13) наведено значення експерименту S , а також отримані за формулою (2.6) значення для середнього часу обробки одного пакета обчислювальним вузлом усереднення ($math_S$). пов'язано з тим, що час обробки в експериментальній моделі встановлено нормальним розподілом.

Таблиця 3.3 – Відхилення експериментальних значень прискорення обчислень від теоретичних (d_math_S)

Кількість обчислювальних вузлів	Прискорення [од.]	d_math_S	Відхилення значень у %
1	1,00	0,03	3,00
5	4,52	0,69	15,29
7	5,67	0,84	14,80
10	6,76	0,78	11,56
20	7,84	0,07	0,89
30	7,30	0,08	1,06

У таблиці 3.3 наведено відхилення експериментальних значень прискорення обчислень від теоретичних. Середнє відхилення експериментальних значень від теоретичних 0.41 одиниць прискорення обчислень або 7.77%. Таке відхилення пояснюється як типом розподілу часу затримок на боці сервера, обчислювальних вузлів (досить сильний вплив на значення дисперсії) та середовища обміну даними, так і з урахуванням ймовірності виходу обчислювального вузла з ладу на невизначений час.

Як видно з топології GRID-системи з урахуванням ймовірності збою обчислювальних вузлів, система являє собою набір паралельно з'єднаних елементів – обчислювальних вузлів, а також паралельних з'єднань цих елементів для обліку кількості пакетів з даними для розрахунків.

Таблиця 3.4 – Імовірність виникнення збою в GRID-системі

Кількість пакетів	Кількість Обчислювальних вузлів	Вірогідність збою системи	Кількість збоїв обчислювальних вузлів
10	1	0,1	0,1
10	5	0,02	0,5
10	7	0,014	0,7
10	10	0,01	1
10	20	0,005	2
10	30	0,003	3

У таблиці 3.4 та на рисунках 3.14, 3.15 представлені значення ймовірності збою як у системі загалом, і у якомусь із обчислювальних вузлів з розрахунку. Кількість пакетів із даними становить 10 пакетів. З таблиці 3.4 видно, що зі збільшенням кількості пакетів збільшується ймовірність збою у системі, так як збільшується і тривалість експерименту, а зі збільшенням кількості обчислювальних вузлів ймовірність збою у системі зменшується, так як вузли з'єднані паралельно і одночасний вихід із ладу всіх вузлів відразу мало ймовірний [10].

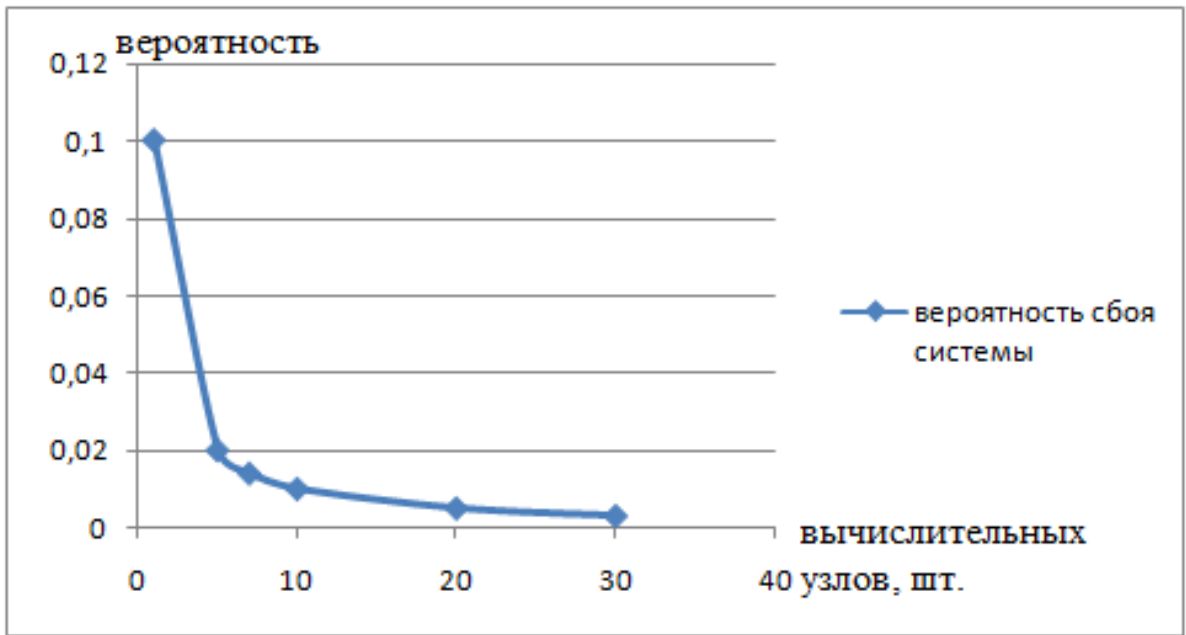


Рисунок 3.14 – Можливість збою системи в залежності від кількості обчислювальних вузлів (з розрахунку на 10 пакетів із даними)

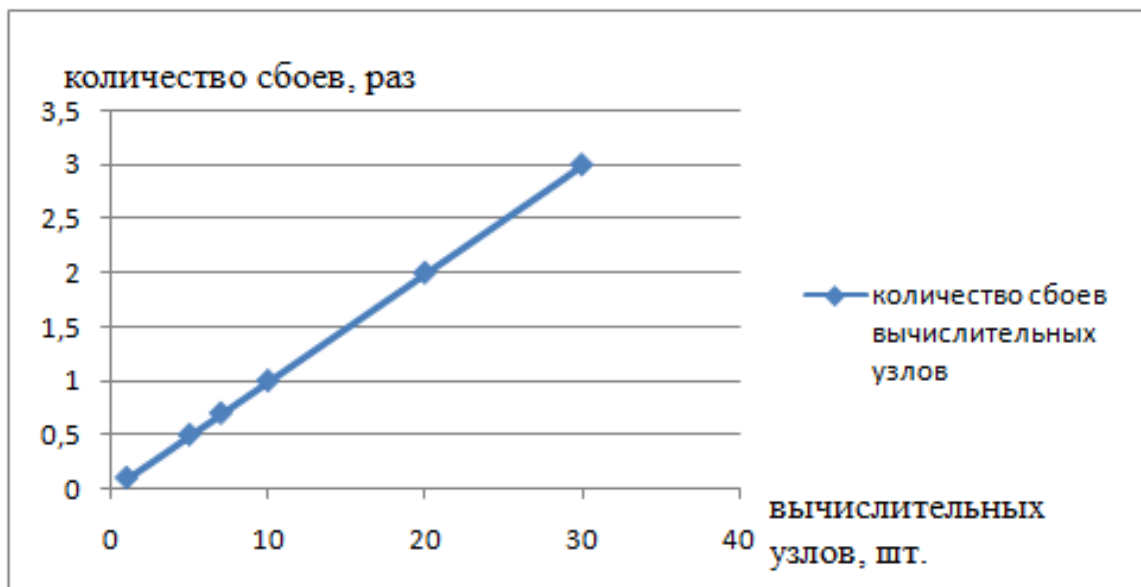


Рисунок 3.15 – Кількість збоїв обчислювальних вузлів у ході експерименту (з розрахунку на 10 пакетів)

3.2 Розгортання високопродуктивного обчислювального кластера з урахуванням дистрибутива PelicanHPC

Дистрибутив PelicanHPC, раніше відомий як ParallelKnorrіx, надає всі

технології для розгортання обчислювальних кластерів з додатковими обчислювальними вузлами, підтримують бездискове завантаження по мережі. З недоліків можна вказати надання у вигляді LiveCD без можливості встановити на внутрішній накопичувач сервера.

Материнські плати підтримують специфікації PXE 2.1 (Preboot execution Environment) – середовища для завантаження комп'ютерів за допомогою мережевої картки без використання жорстких дисків, компакт-дисків та інших пристроїв. Для організації завантаження системи в PXE використовуються протоколи IP, UDP, BOOTP та TFTP. Це дозволяє заощадити на жорстких дисках (вони будуть не потрібні) і прискорює внутрішньосистемну роботу (бо вся дискова підсистема буде утримуватися в оперативній пам'яті з відповідною швидкістю роботи). Об'єм початкової пам'яті в 128 МБ на вузол достатньо його завантаження, так як в оперативну пам'ять завантажуються ядро та оточення порівняно невеликого розміру, а файлова система підключається з файлу на сервері та протоколу NFS; загальний обсяг пам'яті буде залежати від конкретної задачі. Так ж, при завантаженні по PXE всі вузли конфігуруються з однієї локалізації. У кожному вузлі розгорнуто свою копію ОС, на сервері встановлена повна версія системи і сервіси забезпечення працездатності, на вузлах встановлено версію ОС, яка вимагає оновлення в ручному режимі щоразу при внесенні змін до серверу. Завдяки функціям ядра забезпечується унікальність ідентифікаторів процесів у межах всього кластера, а не окремих вузлів, а також "віддалена доставка" сигналів ОС Linux.

3.3 Розробка GRID-системи на основі інфраструктури BOINC

BOINC – відкрита інфраструктура для розподілених обчислень університету Берклі (Berkeley Open Infrastructure for Network Computing) - програмна платформа для організації розподілених обчислень використовуючих добровільно надавані обчислювальні ресурси. Програма

розробляється Каліфорнійським університетом у Берклі (University of California, Berkeley). Усі вихідні тексти BOINC доступні у рамках LGPL ліцензії. Підтримуються ОС Windows, Linux, Mac OS X, Solaris. BOINC підтримується National Science Foundation. BOINC складається із серверної та клієнтської частин.

Серверна частина складається з HTTP-сервера із сайтами проектів, бази даних MySQL та набору допоміжних утиліт (генератор завдань, планувальник, валідатор, асимілятор результатів). HTTP сервер необхідний для загального управління проектом: реєстрація учасників, розподіл завдань для обробки, отримання результатів і управління базами даних проекту. У базі даних зберігаються користувачі, паролі, записи завдань, результатів, інформація про хостів, програми проекту та інше.

BOINC-клієнт – універсальний клієнт для роботи з різними (BOINC-сумісними) проектами розподілених обчислень. Етапи завантаження завдань, обробки та надсилання результатів повторюються в циклі для кожного пакета даних, які необхідно обробити в рамках розв'язуваного завдання.

ВИСНОВКИ

На підставі виконаних досліджень проведено аналіз проблематики в галузі організації обчислень у паралельних системах та стану досліджень та розробок в області GRID-систем. Позначено шляхи розвитку GRID-систем, виділено особливості паралельних обчислень у GRID-системах, наведена класифікація архітектур та стек протоколів цього класу систем. Досліджено найбільш вживані методи аналізу часу обчислень та прискорення обчислень у GRID-системах. Доведено наявність залежностей між параметрами GRID-систем та часом вирішення завдань у системах, що розглядаються, а також досягається прискоренням обчислень. Щодо проблематики, у кваліфікаційної роботи використано комплекс існуючих базових методів дослідження, в тому числі чисельних методів, експериментальних методик та методів імітаційного моделювання. Вивчено зв'язки параметрів GRID-систем та їх компонентів та їх вплив, як на прискорення, що досягається, так і на розв'язувані у яких завдання. Проведено огляд існуючих математичних моделей з метою детальнішого дослідження GRID-систем та отримання нових результатів по темі роботи.

Теоретична значущість дослідження обґрунтована тим, що в роботі на основі специфічних для досліджуваного класу систем характеристик розроблено та реалізовано комплексні підходи до оцінки продуктивності обчислювальних GRID-систем. Виділено фактори, що уповільнюють процес паралельної обробки інформації. Використано узагальнену формулу обмеження зростання продуктивності обчислювальної GRID-системи. Описана мінімізація кількості обчислювальних вузлів у GRID-системах. Розроблено методи визначення швидкості паралельної обробки в GRID-системах гомогенного та гетерогенного типу. Виділено характеристики обчислювальних вузлів з багатоядерними процесорами та їх вплив на обмеження зростання продуктивності системи. Розроблено рекомендації для прискорення паралельної обробки даних у GRID-системах. Проведено аналіз

прискорення навчання нейронних мереж за допомогою застосування GRID-систем. Вказані взаємозв'язки обсягу послідовних обчислень та часу розрахунку у GRID-системах. Показано динаміку характеристик завантаженості GRID-систем у ході обробки даних. На основі використаних математичних моделей та алгоритмів створено програмний продукт аналізу прискорення розподілених обчислень на алгоритми кластеризації.

Визначено залежності швидкості обчислень у GRID-системах від кількості обчислювальних вузлів, що дозволяє більш точно оцінювати тимчасові та ресурсні витрати для вирішення завдань у GRID-системах. Розроблено методи оцінки оптимальних у сенсі прискорення обчислень кількості обчислювальних вузлів у GRID -системах, враховуючі як параметри GRID-системи, так і параметри розв'язуваних завдань. Розроблено методи, що дозволяють проводити аналіз часу обчислень та прискорення обчислень у GRID-системах, завдяки чому можливе більш точне визначення необхідного рішення поставленої задачі часу за відомих параметрів системи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Worldwide LHC Computing Grid [Електронний ресурс]. – Режим доступу: <http://wlcg.web.cern.ch/>
2. Amazon Elastic Compute Cloud (Amazon EC2) [Електронний ресурс] Режим доступу: <http://aws.amazon.com/ec2/>
3. Ubuntu Cloud Infrastructure [Електронний ресурс]. – Режим доступу: <https://help.ubuntu.com/community/UbuntuCloudInfrastructure>
4. [Einstein@Home](http://einstein.phys.uwm.edu/) [Електронний ресурс] – Режим доступу: <http://einstein.phys.uwm.edu/>
5. SETI@home [Електронний ресурс] — Режим доступу: <http://setiathome.berkeley.edu/>
6. [Cosmology@home](http://www.cosmologyathome.org/) [Електронний ресурс] — Режим доступу: <http://www.cosmologyathome.org/>
7. [MilkyWay@Home](http://milkyway.cs.rpi.edu/milkyway/) [Електронний ресурс] — Режим доступу: <http://milkyway.cs.rpi.edu/milkyway/>
8. Вентцель Е.С. Теория вероятностей // Москва: Высшая школа, 1998 - 576 с.
9. Hill M.D., Marty M.R., Amdahl's Law in the Multicore Era. // IEEE Computer, 07.2008, P. 33-38.
10. Дробнов С.Е. Анализ ускорения распределенной обработки информации в GRID-системах // Lambert Academic Publishing 27.12.2012, Германия, 67 с.
11. Европейський проект EGEE. [Електронний ресурс]. – Режим доступу: <http://www.eu-egee.org>
12. Foster I. What is the Grid? A Three Point Checklist // Argonne National Laboratory & University of Chicago, 20.07.2002.
13. Проект DEISA. 2011 г. [Електронний ресурс]. – Режим доступу: <http://www.deisa.org>

14. Foster I., Kesselman C. The Grid: Blueprint for a New Computing Infrastructure. // Calif.: Morgan Kaufmann.- San Francisco, 1998.
15. Kurdi H., Li M., Al-Raweshidy H. A Classification of Emerging and Traditional Grid Systems. // Brunel University Research Archive, 2008.
16. Friese T., Smith M., Freisleben B. Hot Service Deployment in an Ad Hoc Grid Environment. // Proc. 2nd Int'l Conf. Service Oriented Computing (ICSOC 04), ACM Press, 2004.
17. McKnight L.W., Howison J., Bradner S. Wireless Grids: Distributed Resource Sharing by Mobile, Nomadic, and Fixed Devices. // IEEE Internet Computing, vol. 8, no. 4, 2004.
18. Корнеев В.В. Параллельные вычислительные системы // М.: «Нолидж», 1999.- 312 с.
19. Дробнов С.Е. Оценивание ускорения вычислений в распределенных системах. // Научно-практический журнал «Прикладная информатика», Москва, «Синергия», №1(43) 2013, С. 40-48.
20. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления // Спб.: БХВ-Петербург, 2004. - 599 с.
21. Костенко О.С. Методи аналізу швидкості паралельної обробки інформації у GRID системах. Проблеми інформатизації : тези доп. X міжн. наук.-техн. конф., 24 – 25 лист. 2022 р. / ЧДТУ, ВА ЗС АР, УТiГН, НТУ “ХПІ”, ХНУРЕ, "ПiД ПКНДi АП", 2022. С. 94.