Теоретико-ігровий аналіз планувальників у гетерогенному багатопроцесорному середовищі

студент 6-го курсу
КА-61м, Одобеску Владислав

Інститут прикладного системного аналізу
керівник: доц. Ігнатенко Олексій Петрович

Актуальність роботи

- На даний момент використання зовнішніх ресурсів стає більш популярним ніж залучання власних до обчислень.
- Питання про ефективність обчислень відкрите - часто для більш швидкого виконання задач просто збільшують кількість обчислювальних вузлів у середовищі.

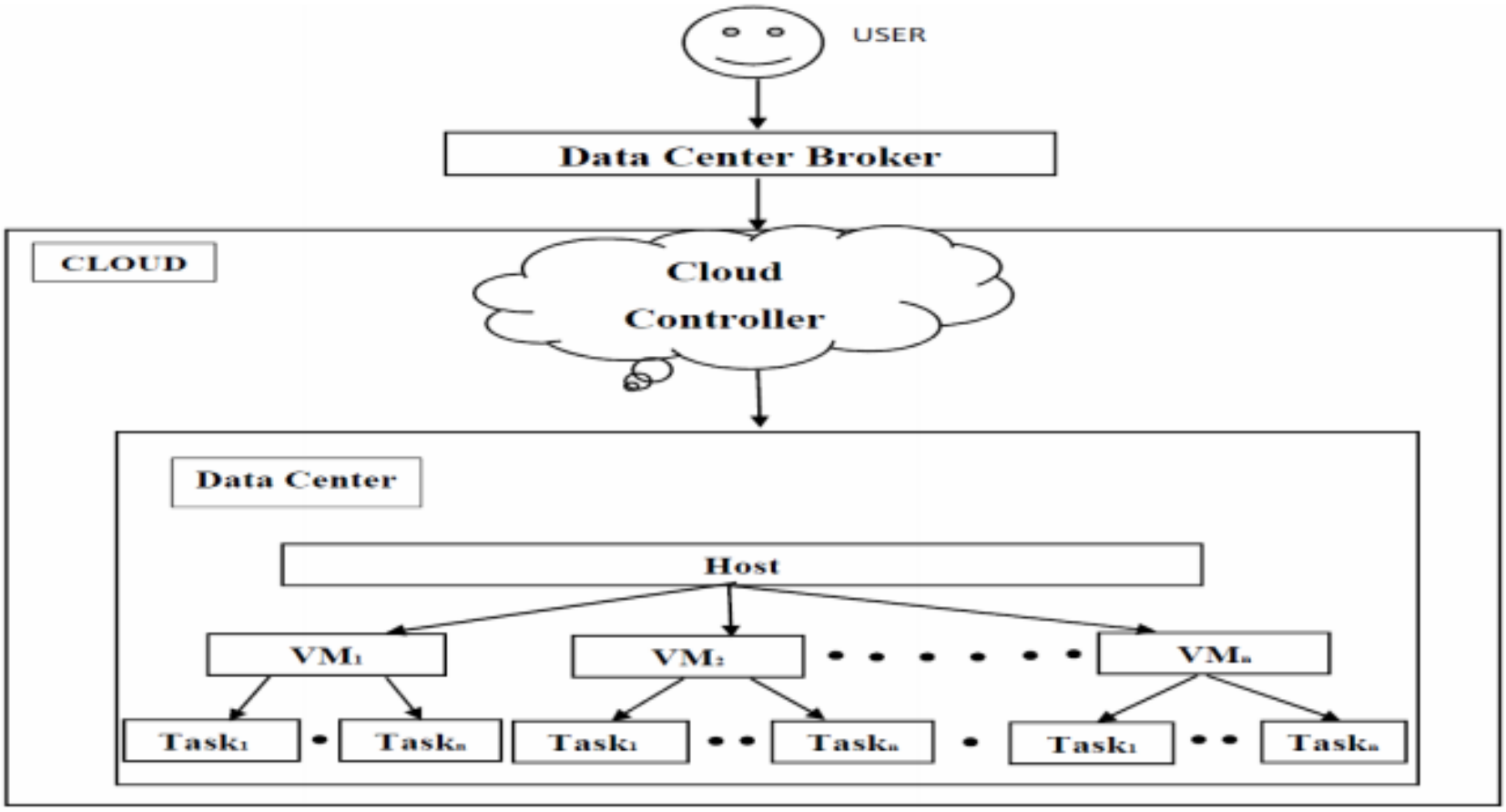
Актуальність роботи

Цікавою роботою у цій сфері є "Coded Computation over Heterogeneous Clusters" авторів Amirhossein Reisizadeh, Saurav Prakash, Ramtin Pedarsani, Amir Salman Avestimehr.
У ній побудована система прийняття рішень по динамічному вибору плану на платформі Amazon AWS з мінімізацією витрат.

Постановка задачі

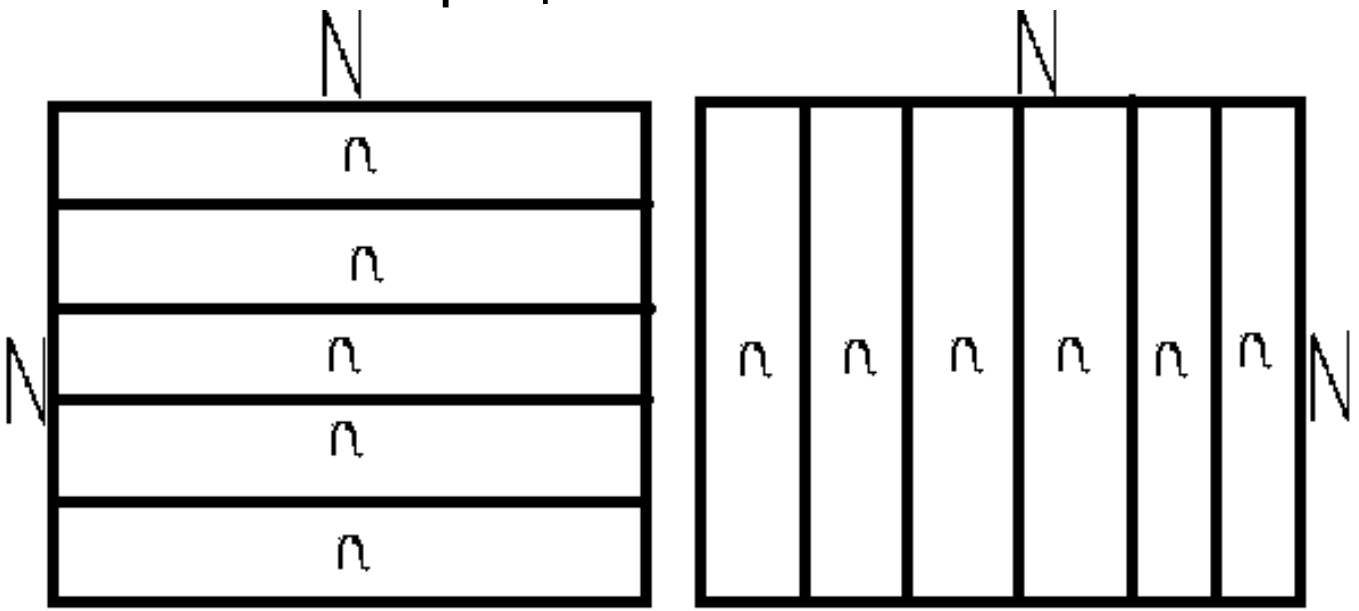
Користувачі мають 2 матриці розмірів $N \times N$ та хочуть обрахувати їх добуток у розподіленому середовищі. Між користувачами виникає конфлікт, оскільки у них спільний, рівноправний та конкурентний доступ до розподіленого середовища.

Структура Cloud середовища

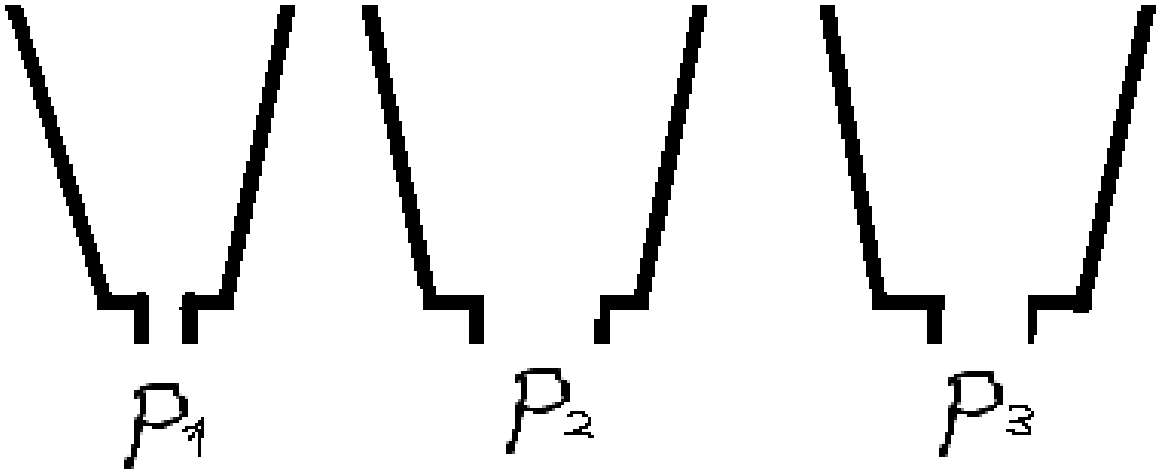


Множення матриць блочно

Для матриць розмірів $N \times N$ вибирається розмір розрізання $n : n \mid N, k = \frac{N}{n}$. Отримуємо k^2 задач множення матриць $n \times N$ та $N \times n$.



Потокова модель



V - загальний об'єм рідини.

$P = \sum_{i=1}^{m} p_i$ - загальна пропускна здатність посудин.

$x_i = \frac{p_i}{P}$.

$v_i = V * x_i$ - об'єм води для посудини i.

Потокова модель

$$T_s(x, X(n)) = \max_{i=1,\dots,m} \left\{ d_i \frac{N^3}{p_i} + d_i \frac{N^2(1 + 2 * \frac{N}{n})}{q} + d_i \frac{N^2}{n^2} l \right\}$$

N - розмір матриць

n - розмір розбиття

p_i - потужність вузла i

d_i - доля обчислень вузла i

q - ширина каналу передачі даних i

l - час на створення з'єднання

Потокова модель (дискретний варіант)

$K = \frac{N^2}{n^2}$ - загальна кількість задач однакової складності множення матриць $n * N$ та $N * n$

$$\hat{k_i} = d_i * K$$

$k_i : \sum_{i=1}^{m} |k_i - \hat{k_i}| \Rightarrow min$ - кількість задач, обчислених на вузлі i

Пояснення штрафів

У випадку коли матриця $N * N$ не розрізається, то передаються 2 матриці A, B та результат $C = AB$. Тобто передається $3 * N * N$ елементів.

При розбитті n формується $K = \frac{N^2}{n^2}$ задач для кожної задачі передається $n * N + N * n + n * n$ елементів.

Тобто у загальному це $\frac{N^2 * (n*N + N*n + n*n)}{n^2}$.Або $N^2 * (2\frac{N}{n} + 1)$.

Порівнюючи це з випадком без розрізання маємо: $\frac{2\frac{N}{n}+1}{3}$.

Ігрова постановка задачі

Гра двох користувачів:

1. Користувачі вибирають розбиття n_1, n_2.
2. Користувачі розрізають матриці, формують задачі та надсилають їх до хмари.
3. Користувачі отримують результати.

Часом для користувача вважається час отримання усіх результатів надісланих задач.

Ігрова постановка задачі

Таким чином отримано біматричну гру з матрицями програшів (A, B).
Стратегіями користувачів є саме їх вибране розбиття.
$A^T = B$ оскільки час для гравця 1 у випадку профілю стратегій (n_1, n_2) дорівнює часу гравця 2 з профілем стратегій (n_2, n_1).

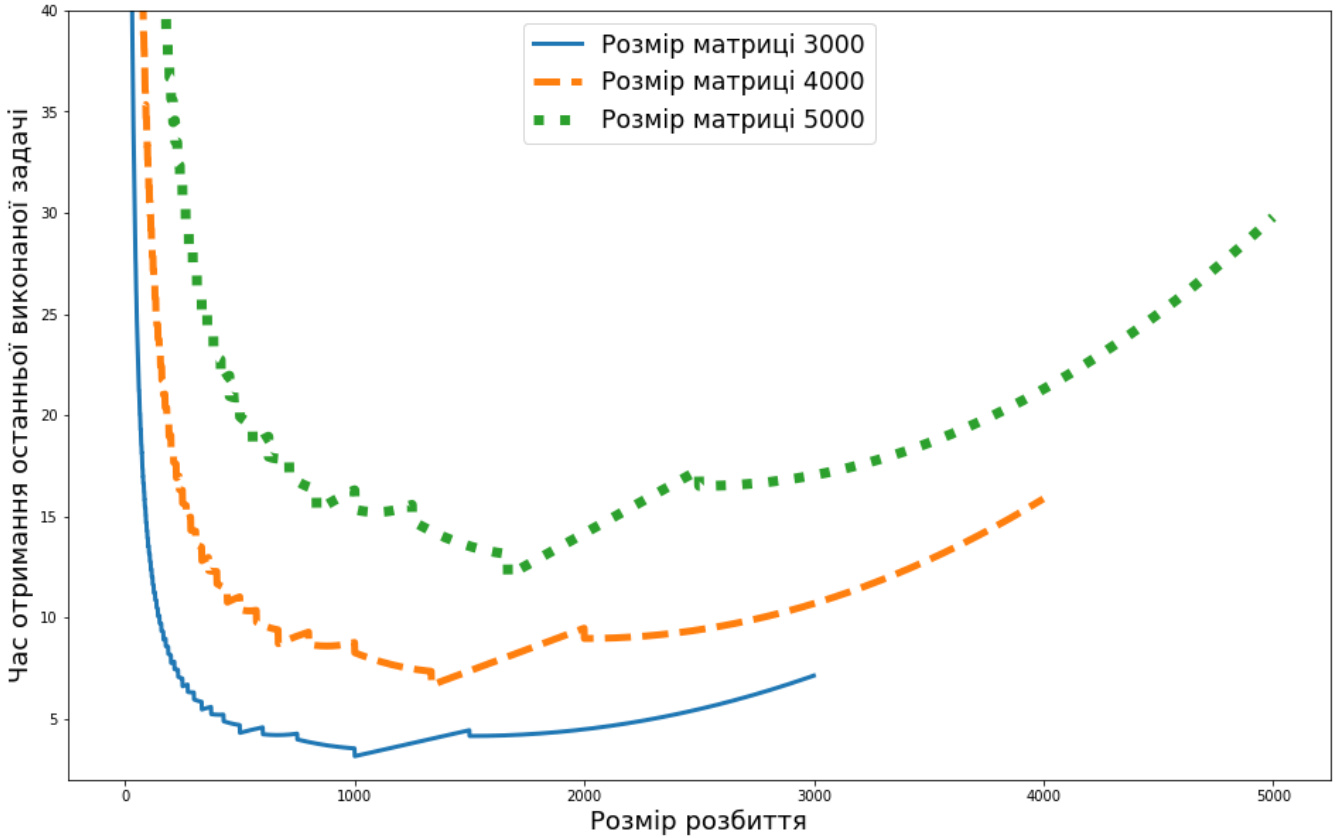
Ігрова постановка задачі

Досліджувані планувальники - статичні планувальники типу extr-extr.
min-min - посилає задачу із черги з найменшою складністю на обчислювальний вузол с найменшою потужністю.

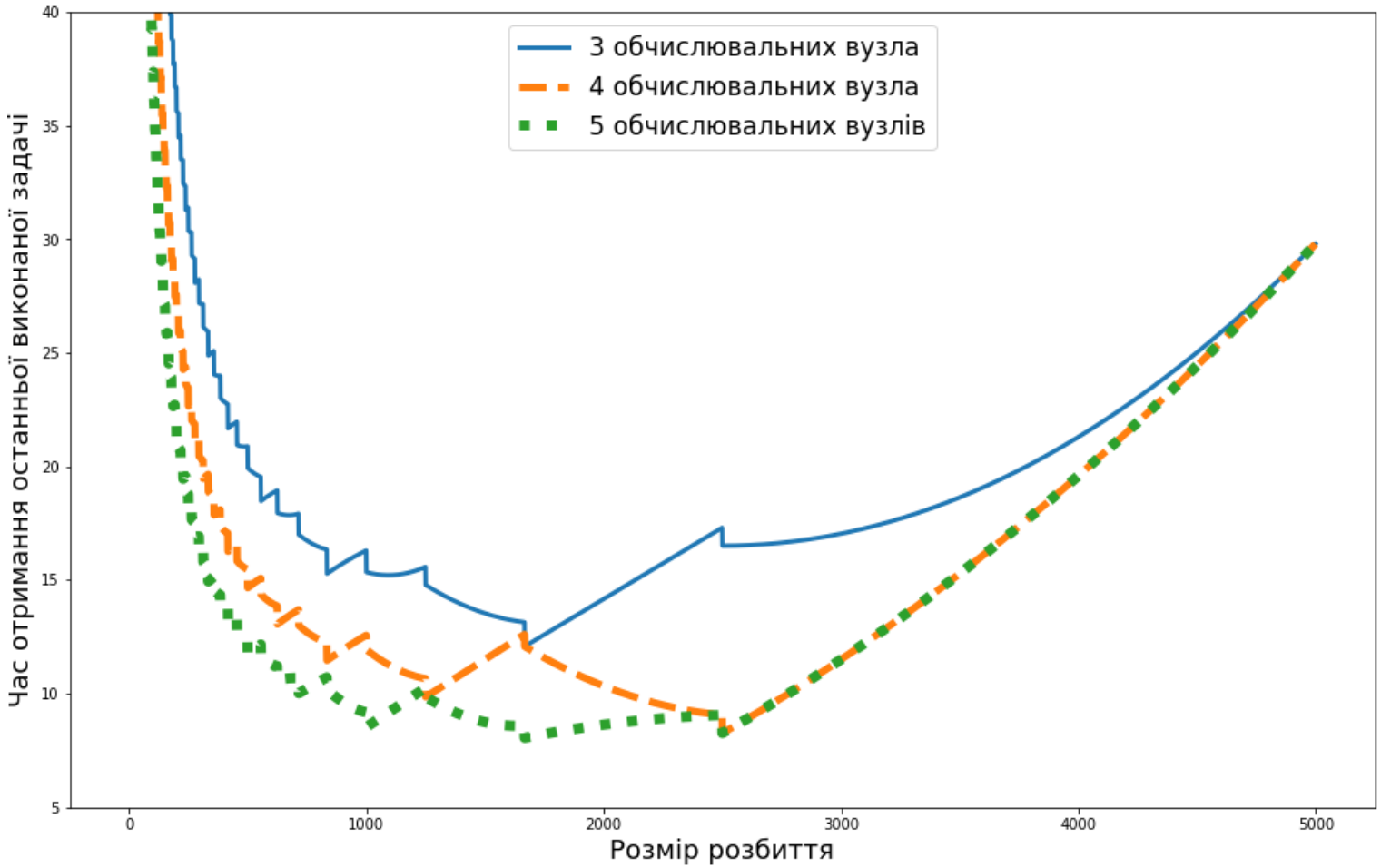
Проведення експериментів

Для проведення експериментів існують спеціальні пакети на мові Java - CloudSim, GridSim, DARTCSIM та інші.
Проблема їх усіх в тому, що вони базуються на мові Java та працюють дуже повільно у випадку великої кількості симуляцій.
Тому була розроблена власне симуляційна програма.

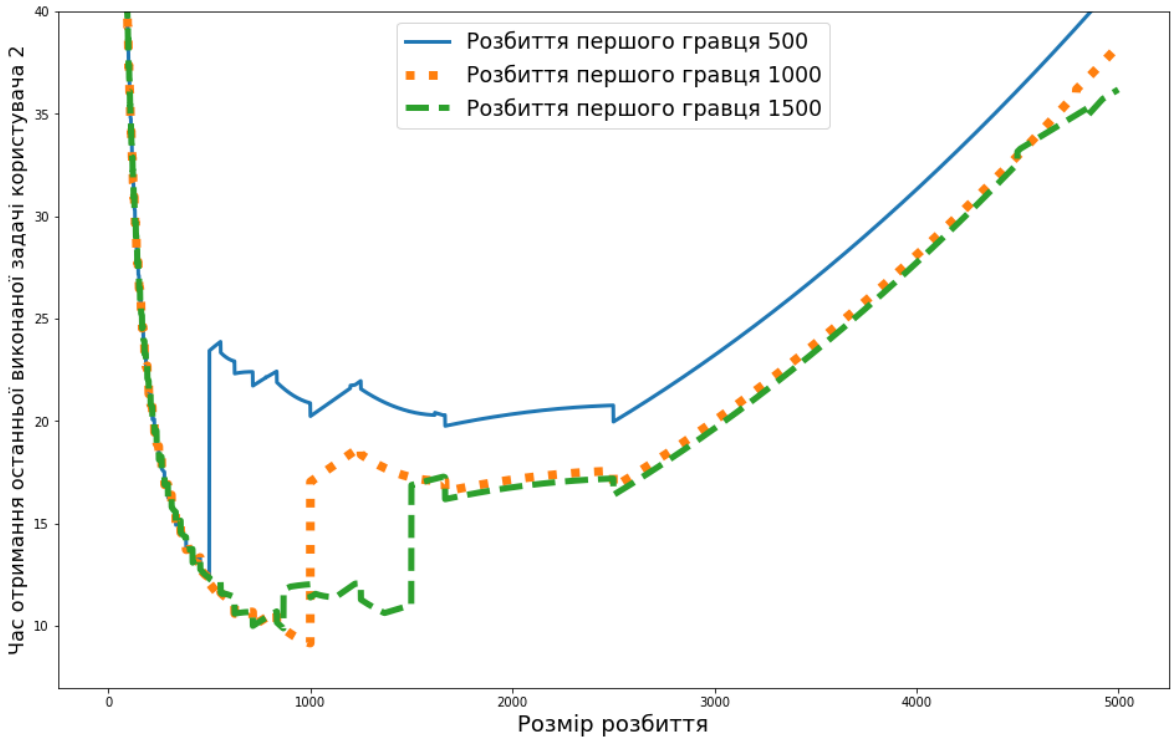
Графіки часів для різних розмірів матриць (один користувач)



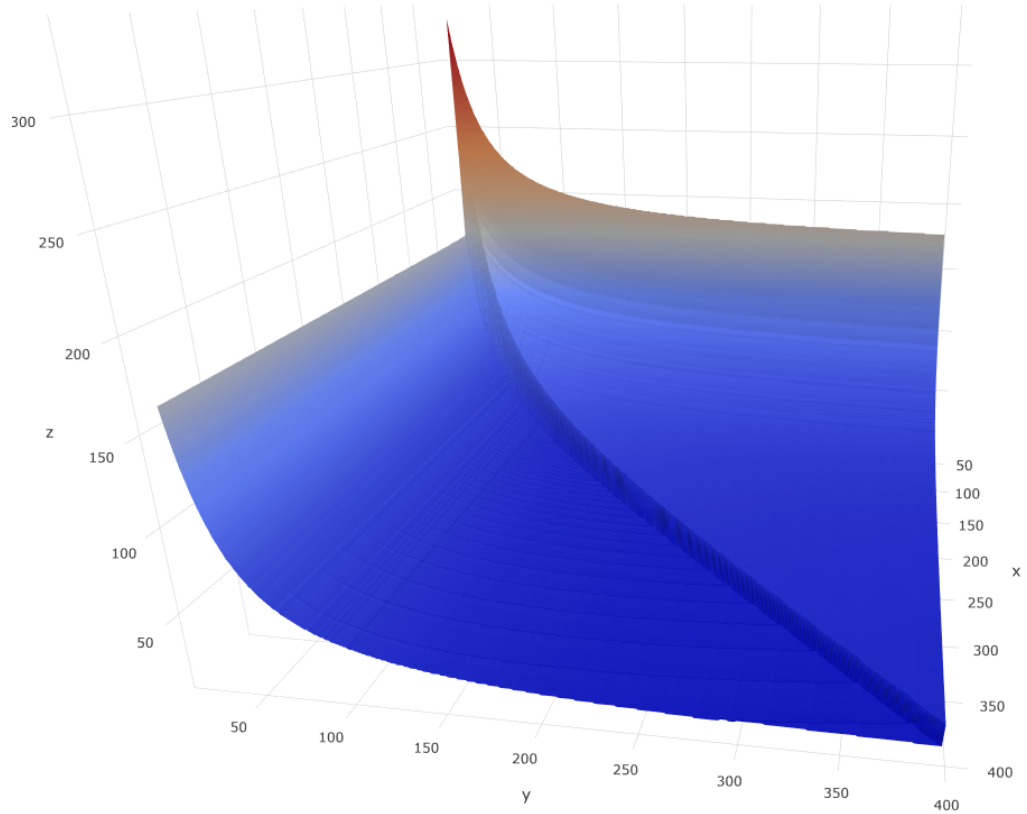
Графіки часів для різних розмірів матриць



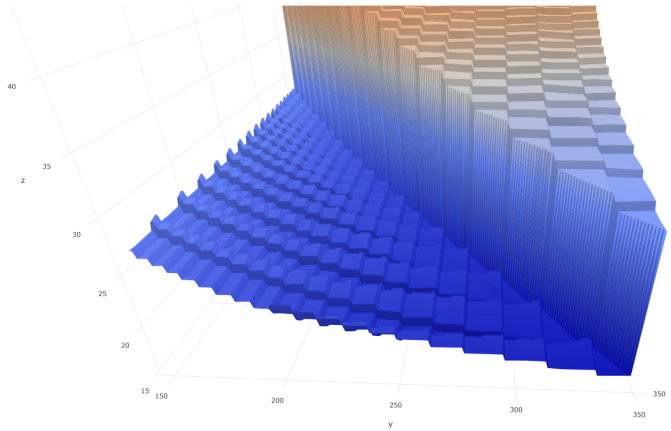
Графіки часів для різної кількості обчислювальних вузлів (один користувач)

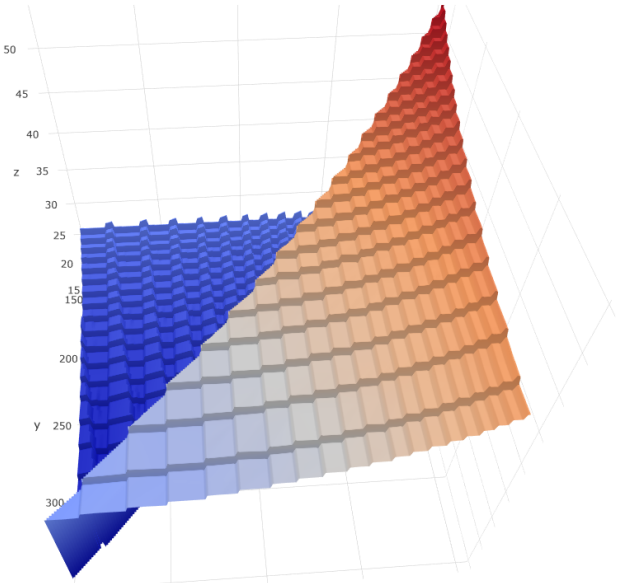


Графіки часів для двох користувачів

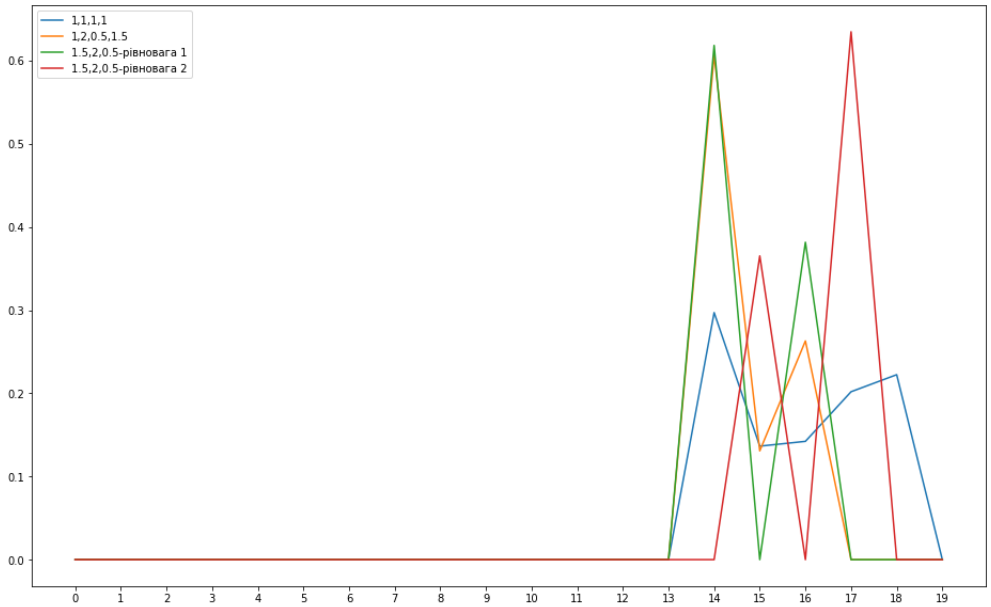


Графіки часів для двох користувачів

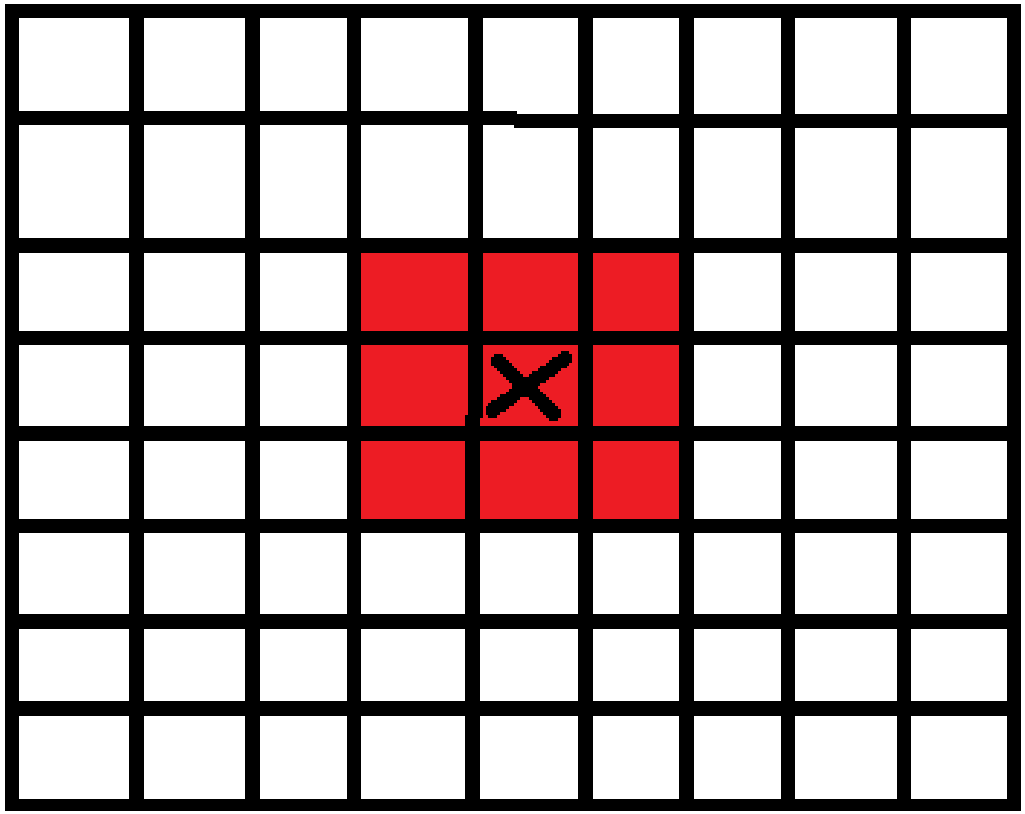




Графіки часів для двох користувачів



Застосування методу оптимізації для спуску до оптимальної точки



Застосування методу оптимізації для спуску до оптимальної точки

№	Початкова стратегія	Початковий час	Кінцева стратегія	Кінцевий час
1.	(2,2)	6670.72055426	(1250, 2000)	11.0248749
2.	(10000,10000)	14.65571875	(5000,10000)	13.7184062
3.	(5,25)	1335.36618371	(1000,1250)	11.0248749
4.	(25,5)	1604.09384557	(1000,1250)	11.0248749
5.	(25,5)	1604.09384557	(1000,1250)	11.0248749
6.	(625,40)	184.069188072	(1000,1250)	11.0248749

Висновки

- У роботі побудована модель задачі множення матриць блочно у розподіленому середовищі. Проаналізовані штрафи за дрібність розбиття.
- Проведено пошук рівноваг Неша.
- Розглянуло альтернативний підхід до пошуку оптимальної точки.

Шляхи подальшого розвитку

У подальшому можна розглянути інші стратегії розбиття задачі множення матриць на підзадачі та більш складні структури Cloud середовищ.
Також слід розглянути інші програми із стандарту BLAS, оскільки вони є основою усіх наукових проектів.

Дякую за увагу.