

# 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.

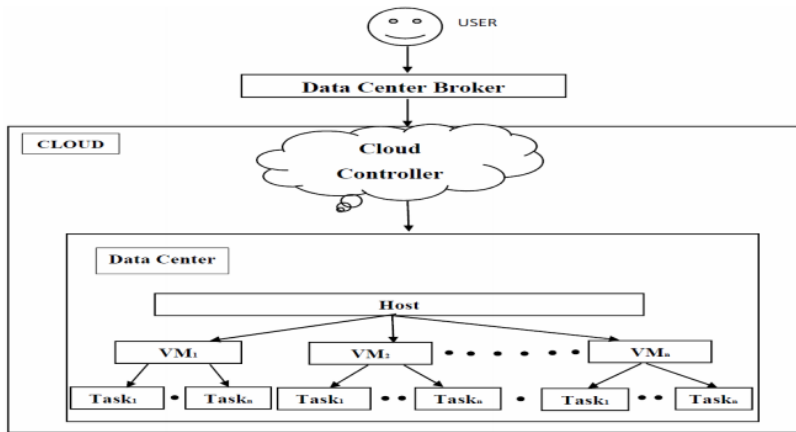


# Постановка задачі

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.



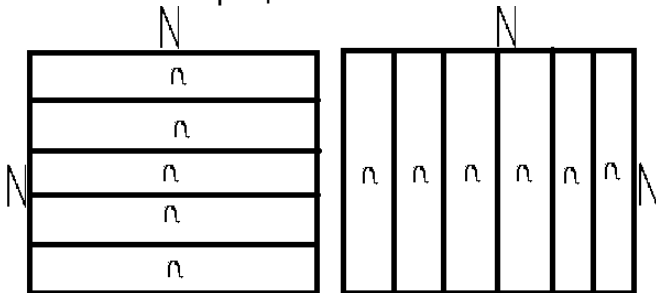
# Структура Cloud середовища





# 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  $N \times n$ .





# 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 програвів  $(A, B)$ .

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

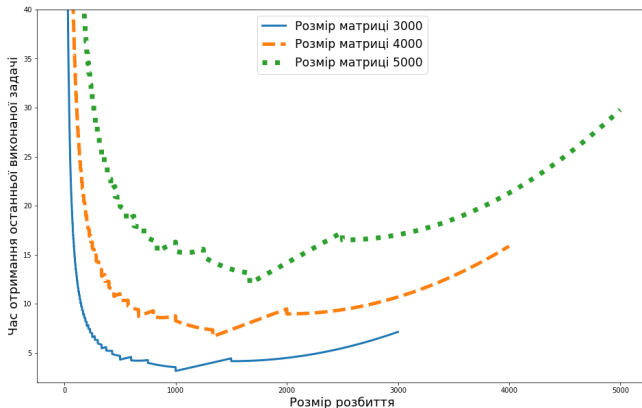
Для проведення експериментів існують спеціальні пакети на мові Java - CloudSim, GridSim, DARTCSIM та інші.

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.

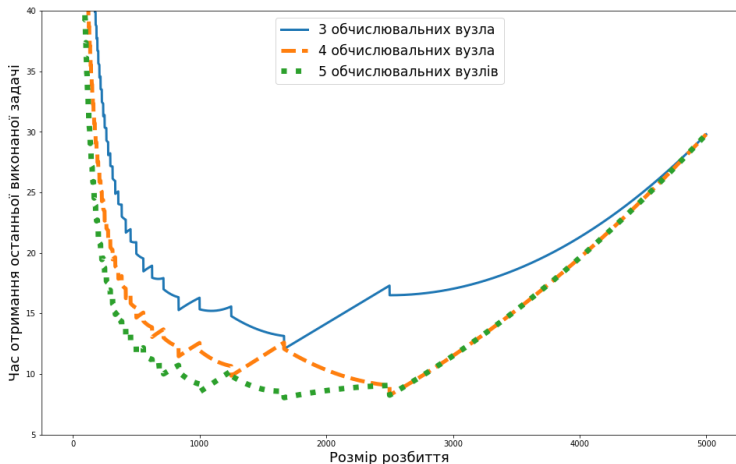


# Графіки часів для різних розмірів матриць (один користувач)



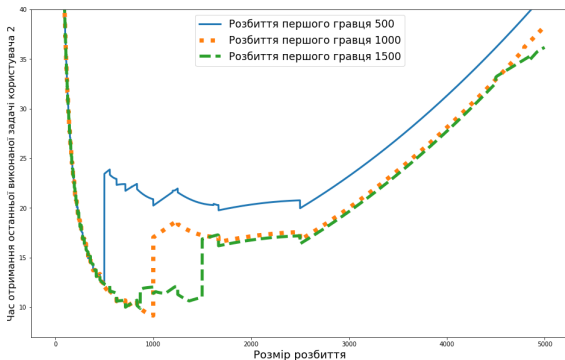


# Графіки часів для різних розмірів матриць



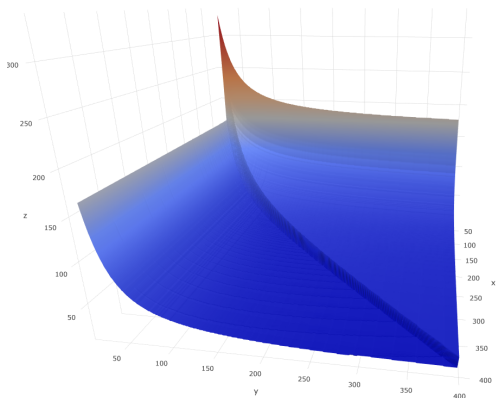


# Графіки часів для різної кількості обчислювальних вузлів (один користувач)





# Графіки часів для двох користувачів при фіксованій стратегії першого користувача





# 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.

