

# Прикладні високопродуктивні кластерні обчислення в HADOOP (“Applied high-performance cluster computing on HADOOP”)

Автор: студент 4-го курсу, групи ДА-12  
ННК «ІПСА» НТУУ «КПІ»  
Качко Микита Андрійович  
Науковий керівник:  
Харченко Костянтин Васильович

# Ціль

- ✓ Дослідити пришвидшення обчислення об'ємних задач за допомогою HADOOP.

# Предмет дослідження

- ✓ Використання програмного забезпечення HADOOP для прикладних високопродуктивних кластерних обчислень.

# Актуальність задачі

- ✓ Розподіл обчислень на декілька комп'ютерів зменшить час та зусилля, що необхідні для цього завдання.

# Які методи досліджуються

- ✓ Генерація таблиці простих чисел.
- ✓ Визначення числа на належність до чисел Серпінського.
- ✓ Розв'язування задачі комівояжера.

# Чому було обрано розподілення обчислень?

- ✓ В порівнянні з паралелізацією обчислень є достатньо гнучкими та в певній мірі вузло-незалежними.
- ✓ Нема необхідності розташовувати всі вузли в одному місті (без прив'язки до простору).
- ✓ Дозволяють працювати із величезними об'ємами даних.
- ✓ Додавання нових вузлів до системи не є складною задачею.

# Результати

- ✓ Було розглянуто три обчислювально складних задачі на прикладі яких можна виявити переваги та недоліки систем розподілених обчислень
- ✓ Було отриманого програмний продукт, за допомогою якого можна дослідити роботу системи розподілених обчислень HADOOP
- ✓ Отримані результати для подальших досліджень

# В чому полягає відмінність підходів?

- ✓ У випадку генерації певна кількість часу витрачається на розсилання задачі всім вузлам системи та збір результатів, тоді як прямий спосіб увесь час працює із одним списком.
- ✓ У випадку пошуку числа Серпінського файл таблиці простих чисел виявився настільки великим, а час пошуку настільки малим, що використання системи виявилось недоцільним.
- ✓ У випадку задачі комівояжера кожен вузол розв'язав задачу для певного підшляху, а для збору результатів видав найкращий, таким чином зменшивши розмір задачі щонайменше на один порядок

# Інноваційність запропонованого підходу

Для визначення належності числа до складу Серпінського треба пришвидшити тест на простоту

- ✓ Будемо використовувати вже готову таблицю простих чисел
- ✓ Щоб таблицю використовувати – її треба спочатку згенерувати

# Інноваційність запропонованого підходу

Для генерації таблиці простих чисел треба пришвидшити тест на простоту

- ✓ Будемо використовувати ймовірнісні методи визначення простоти
- ✓ Будемо звіряти останнє просте число з його порядковим номером
- ✓ Для розділення задачі на вузли будемо використовувати метод описаний Качко М. в статті «Prime table generation»



# Висновки

- ✓ На прикладі генерації простих чисел як і очікувалося, система добре себе проявила на об'ємних даних
- ✓ Однак на прикладі пошуку числа Серпінського прямий метод виявився швидшим на всьому проміжку даних
- ✓ Найкраще всього Nadoor проявив у задачі пошуку шляху комівояжера