

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

**Магістерська кваліфікаційна робота на тему
“Розробка методів і програмного забезпечення для
балансування обчислюваного навантаження на основі
технології WebSocket”**

**Виконав студент гр. 2ПІ-196
Гибало В.В.**

Керівник – доцент, к.т.н. Кательніков Д. І.

Вінниця – 2020

Мета, предмет та об'єкт дослідження

- Мета – збільшення швидкодії додатків за рахунок паралельного виконання частин обчислювального навантаження у розподіленій архітектурі.
- Об'єкт – обчислювальний процес у інформаційних системах з розподіленою архітектурою.
- Предмет – методи та засоби балансування обчислювального навантаження у розподілених архітектурах.

Актуальність розробки

З розвитком комп'ютерних технологій роль цифрових пристроїв в житті людини дедалі зростає. Щодня кожна людина в середньому використовує 5 веб-додатків і за статистикою, якщо час від початку завантаження веб-сторінки до початку її взаємодії з користувачем більше 3 секунд, то половина користувачів залишають сайт. Така статистика наглядно демонструє наскільки важливою є швидкодія для користувача.

І хоча з кожним роком збільшується потужність пристроїв, збільшується і навантаження на них, починають використовуватись все складніші алгоритми і, як наслідок, збільшується час виконання програми. Для зменшення навантаження на комп'ютери, збільшення обчислювальної потужності та пришвидшення виконання алгоритмів доцільно розробити систему, яка дозволить пришвидшити роботу всього додатку, дозволить збільшити користувацьку базу, а отже і прибуток власників. Подібні системи користуються попитом серед розробників програмного забезпечення, в них інвестують значні кошти, адже в результаті отримують більшу швидкість роботи програмного додатку. Тому актуальним є питання підвищення продуктивності інформаційних систем шляхом балансування обчислювального навантаження у розподілених архітектурах.

Наукова новизна – проведення обчислень

1. Запропоновано метод об'єднання обчислювальних ресурсів у кластер у комп'ютерній мережі, який на відміну від існуючих методів використовує не спеціалізовані бібліотеки паралельних обчислень, а програмне забезпечення мережевого обміну загального призначення, що значно розширює можливості конфігурації обчислювального процесу з боку користувача та дозволяє досягти оптимальної продуктивності шляхом балансування обчислювального навантаження у відповідності до фактичної продуктивності об'єднаних у кластер ресурсів.

Наукова новизна – асинхронне тестування

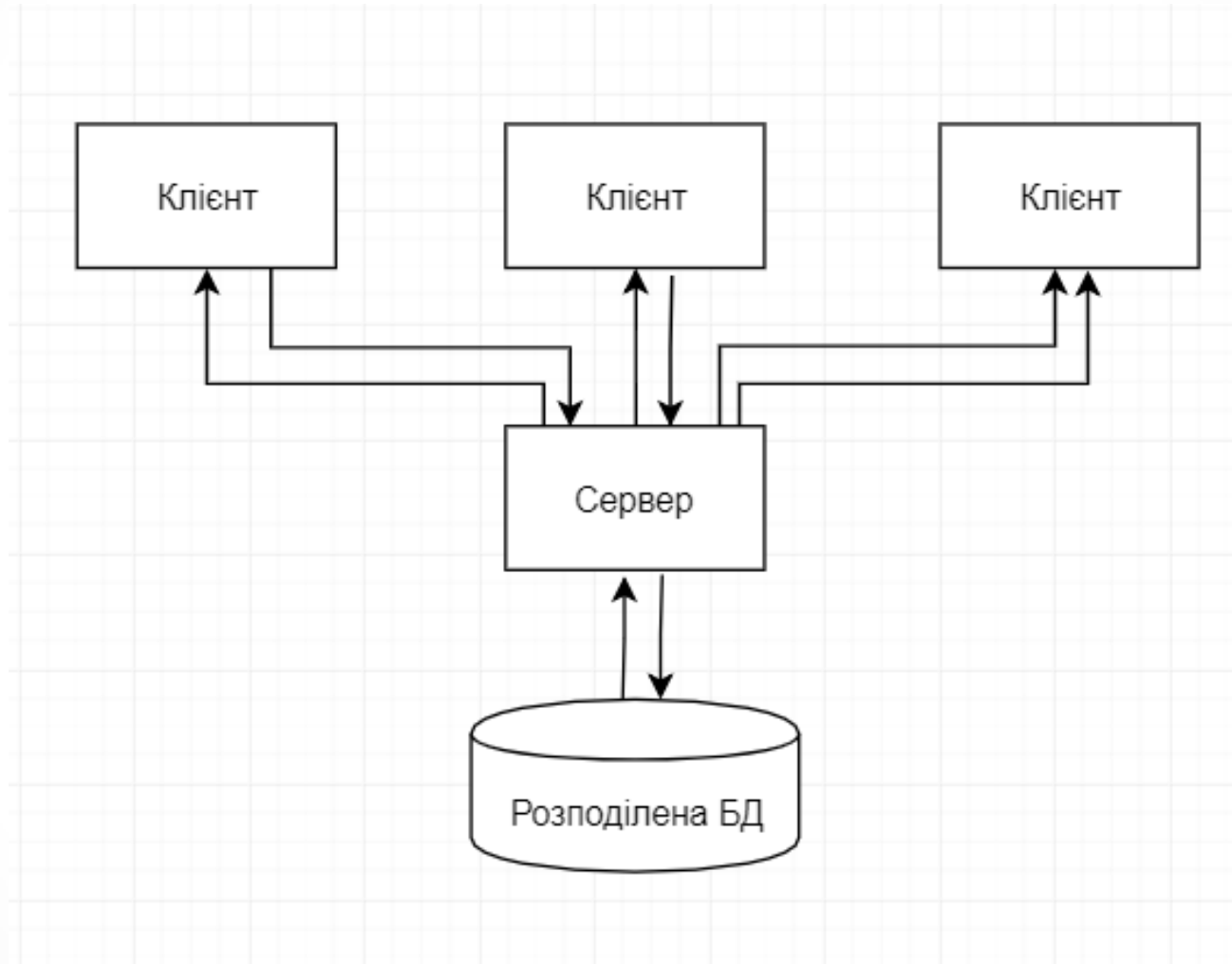
Запропоновано метод тестування продуктивності функціонування розподіленої обчислювальної системи, який на відміну від існуючих методів тестування використовує асинхронні запити до складових частин кластеру, що дозволяє підвищити точність оцінок.

Даний метод за допомогою запитів дозволяє визначити точність розподілення задач на основі оціненої продуктивності клієнта, порівняти очікувану продуктивність кожного клієнта з отриманою та оцінити точність отриманих коефіцієнтів розподілення.

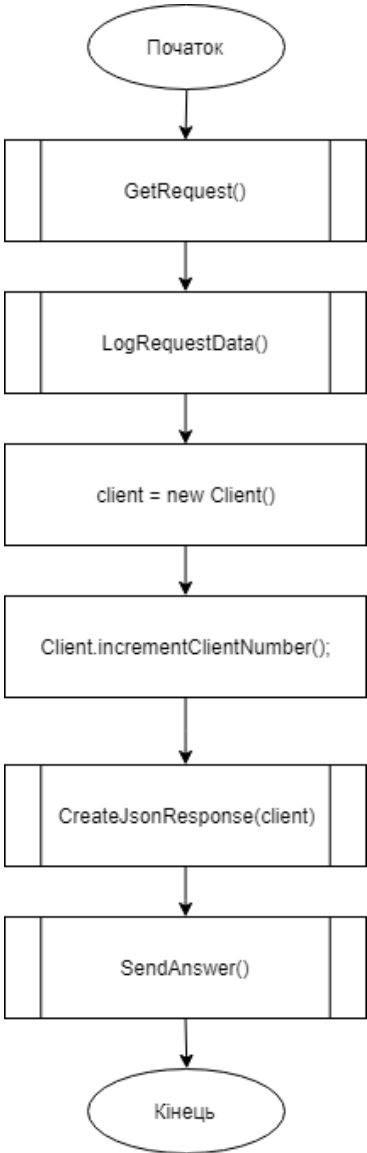
Порівняльний аналіз аналогів

Критерій	Smart Calculation	Apache	Nginx	Elastic
Маштабування при мінімальних вимогах до апаратної частини	+	-	+	+
Ефективне використання ресурсів	+	-	+	+
Можливість самостійно опрацювати динамічний контент	+	+	-	+
Повністю безкоштовний функціонал	+	+	-	-
Сумарний коефіцієнт	4	2	2	3

Загальна модель системи



Алгоритм роботи клієнтського додатку



Дизайн клієнтської та серверної частин

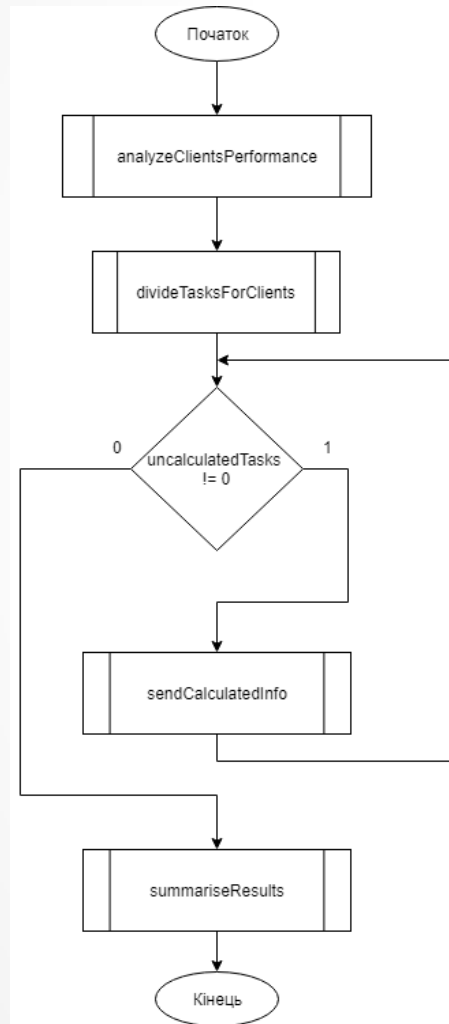
Кількість підключених клієнтів: 1

Результати тесту: 10000 мс

Start calculation

```
$ nodemon app
[nodemon] 1.18.10
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching: *.*
[nodemon] starting 'node app.js'
```

Тестова задача з використанням генетичних алгоритмів



```
[nodemon] starting `node app.js`  
Init client 1  
Start calculation  
Total time 23403  
█
```

```
[nodemon] starting `node app.js`  
Init client 1  
Init client 2  
Init client 3  
Start calculation  
Total time 9816  
█
```

Результати тестування веб-сайту

Модульне тестування серверної частини

```
$ npm test  
  
> smart_calculations@1.0.0 test C:\Users\Владислав\Desktop\smart_calculations\server  
> mocha --watch tests/*.test.js  
  
  ✓ should multiply two numbers  
  ✓ should identify client  
  ✓ should correctly separate algorithm  
  ✓ should correctly join test results  
  ✓ should make algorithm summary  
  
5 passing (28ms)
```

Економічний потенціал розробки

- Аналіз комерційного потенціалу розробки показав, що програмний продукт за своїми характеристиками випереджає аналогічні програмні продукти і є перспективною розробкою. Він має кращі функціональні показники, а тому є конкурентоспроможним товаром на ринку. Існуючі переваги нової розробки дозволять швидко її поширити та популяризувати.
- Згідно із розрахунками всіх статей витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи загальні витрати на розробку складають 146882,94 грн.
- Розрахована абсолютна ефективність вкладених інвестицій в сумі 620819,38 грн свідчить про отримання прибутку інвестором від комерціалізації програмного продукту.
- Ефективність вкладених в наукову розробку інвестицій складає 73 %, що вище за мінімальну бар'єрну ставку дисконтування, яка складає 25%. Це означає потенційну зацікавленість інвесторів у фінансуванні розробки.
- Термін окупності вкладених у реалізацію проекту інвестицій становить 1,36 року, що також свідчить про доцільність фінансування нової розробки.

Дякую за увагу !