ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

NURE
Харківський національний університет радіоелектроніки

Магістерська кваліфікаційна робота

Методи управління розподіленим обчислювальним процесом в гетерогенних хмарних системах

Здобувач гр. СПзм-23-1 Бітюкова Є.В.
Керівник ст. викл. каф. ЕОМ Бугрій А.М.

Харків, 2025

Актуальність дослідження методів управління розподіленим обчислювальним процесом в гетерогенних хмарних системах

Причини актуальності напрямку:

Гетерогенність інфраструктури: Сучасні хмарні середовища включають різні типи ресурсів: CPU, GPU, FPGA, різноманітні мережеві технології, технології зберігання даних, що вимагає нових методів ефективного управління обчисленнями.

Розподіленість і мультихмарність: Все більше застосувань розподіляються між кількома хмарами (multi-cloud) або між центрами обробки даних і периферійними обчисленнями (edge computing).

AI/ML робочі навантаження: Системи, що обслуговують штучний інтелект та машинне навчання, потребують складного розподілу задач між неоднорідними обчислювальними вузлами.

Енергетична ефективність: Одним із важливих трендів є оптимізація споживання енергії при розподіленні обчислювальних завдань, що також вимагає нових методів управління.

Безпека та надійність: Управління в умовах динамічних змін і атак вимагає гнучких, адаптивних алгоритмів координації та відновлення.

Основні недоліки сучасних методів

1. **Недостатня адаптивність до динаміки середовища**
2. **Складність оптимізації в умовах гетерогенності**
3. **Низька масштабованість існуючих алгоритмів**
4. **Відсутність глобальної оптимізації**
5. **Обмежена підтримка мультихмарних і гібридних середовищ**
6. **Недостатня енергетична ефективність**
7. **Безпека і стійкість до атак**

3

Мета та задачі роботи

Метою роботи є підвищення ефективності управління ресурсами у хмарних обчисленнях шляхом оптимізації процесу розподілу завдань, що дозволяє скоротити час виконання, покращити балансування навантаження та збільшити пропускну здатність системи

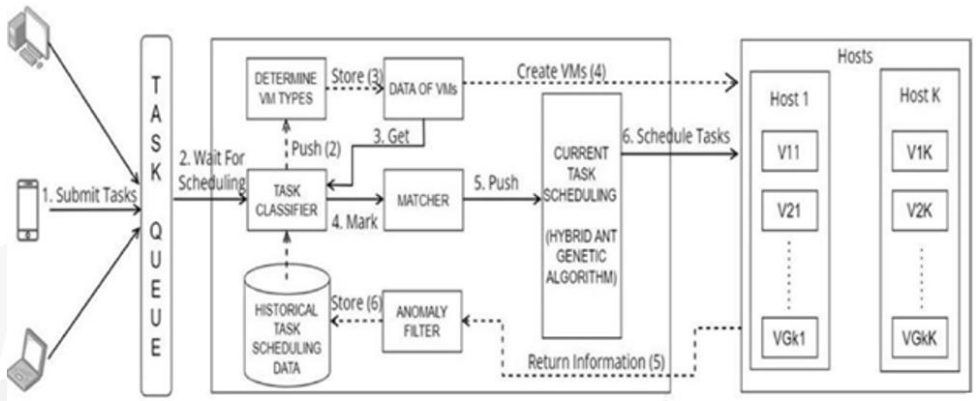
Об'єкт дослідження: процеси управління розподілом завдань у хмарних обчислювальних системах.
Предмет дослідження: методи та алгоритми планування завдань для підвищення ефективності використання ресурсів у хмарних середовищах.

Задачі дослідження:

а) аналіз існуючих методів планування завдань у хмарних системах;
б) розробка двоетапного підходу для оптимізації розподілу завдань;
в) впровадження метаевристичних алгоритмів для покращення ефективності управління ресурсами;
г) оцінка продуктивності запропонованого методу через експериментальне моделювання;
д) порівняння отриманих результатів із традиційними алгоритмами планування.

4

Оптимізована двоетапна задача планування



5

Критерії оцінки ефективності управління розподіленим обчислювальним процесом

1. Час виконання: $T^{VM_i} = \sum_{j=1}^{m} T_i^{task_j}$, де $i \in \overline{VM}$

$$T_{max}^{VM} = max\left(T^{VM_1}, T^{VM_2}, \ldots, T^{VM_n}\right)$$

2. Коефіцієнт середнього використання ресурсів : $U_R = \frac{\sum_{i=1}^{n} T^{VM_i}}{T_{max}^{VM} \times n}$

3. Пропускна здатність: $V = \frac{N}{T_{max}^{VM}}$

4. Час активного використання ресурсів: $T_{avg} = \frac{\sum_{j=1}^{m} T^{task_j}}{m}$

5. Ефективність балансування навантаження: $D = \frac{T_{max} - T_{min}}{T_{avg}}$

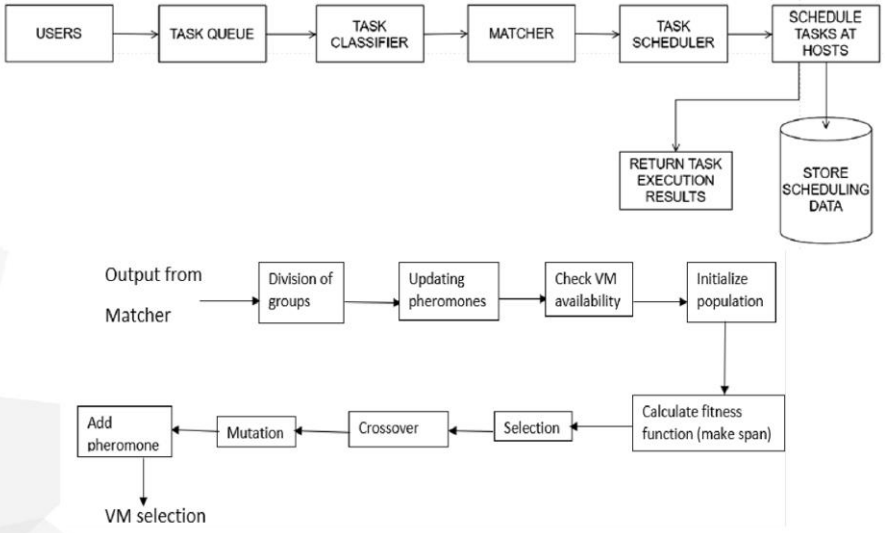
6

Алгоритм пошуку оптимальних параметрів та їх значень

```
INPUT: Range values
OUTPUT: Optimal parameters
1  max_score ← zero
2  Initialize a variable key
3  for each k and v in cluster object items
4      if v [0] >max_score
5          max_score ← v [0]
6          key ← k
7  end for
8  print key //Optimal parameters
9  print max_score //Score
```

7

Покращений генетичний алгоритм та структура модуля планування завдань



8

Особливості розширеного генетичного алгоритму

Функція відбору генетичного алгоритму

```
Input: Task data, VM data
Output: Fitness value of individuals
1  Initialize new list, Execution time
2  for i=0 to task data size do
3      find Execution time and add to list
4  end for
5  Initialize a Map, BusyTime
6  for i=0 to task data size do
7      Put (existing busy time of ith VM + execution time of task i)
8  end for
9  fitness ← 0
10 for all values in BusyTime do
11     fitness ← maximum(values)
12 end for
```

Функція селекції генетичного алгоритму

```
Input: fitness value of individuals
Output: Fittest / Selected individuals
1  maxfit←MAX_VALUE
2  for i← 0 to individuals.length do
3      if maxfit >= ith individual fitness
4          maxfit= ith individual fitness
5          maxfitindex=i
6      end if
7  end for
```

Функція одноточкового кросинговеру

```
Input:  Random Crossover Point
Output: Cross overed Individuals
1  for i←0 to crossoverPoint do
2      swap(fittestGenes[i], secondFittestGenes[i])
3  end for
```

Функція мутації

```
Input: fittest individuals
Output: mutated individual
1  Mutated ← false
2  for i← 0 to geneLength and !mutated do
3      for j ←i+1 fittest geneLength and !mutated do
4          firstSet ← TaskVMdata(i)
5          SecondSet ← TaskVmdata(j)
6          if firstSet contains fittestgenes(j) && secondSet contains fittestgenes(i) then
7              swap fittestgenes of i and j
8              mutated ← true
9          end if
10     end for
11 end for
```

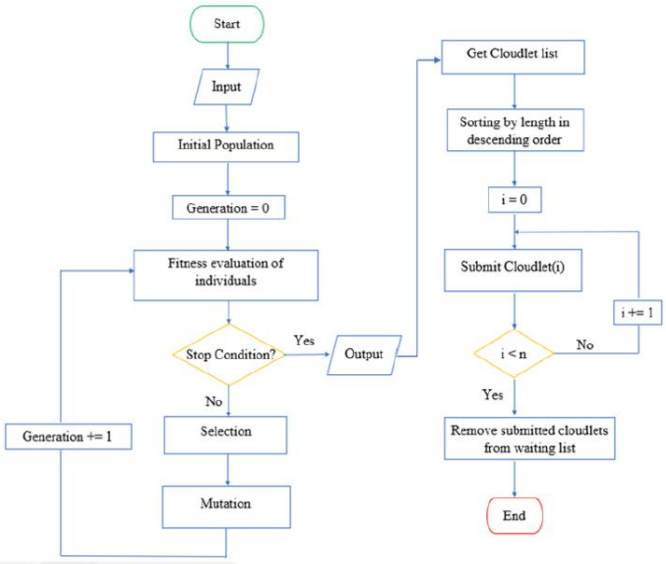
9

Опис завдання

Частота процесора (МГц)	Обсяг пам'яті (Mb)	Кількість циклів	Кількість інструкцій (MIPS)	Номер комп'ютера
1024	900	30	170	1
1024	300	15	255	2
850	150	125	2040	4
256	300	100	320	3
512	900	50	51	2
2048	50	20	20	5
128	412	10	482	5
450	2000	5	3021	6
50	300	20	450	4

10

Блок-схема та псевдокод алгоритму планування завдань

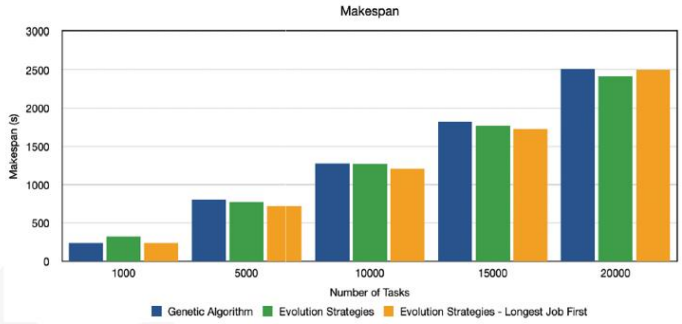


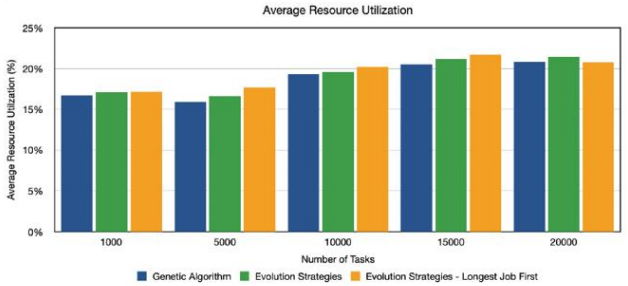
Input: $(\mu, \lambda, \text{GenerationSize}, \text{CloudletList}, \text{VirtualMachineList})$
Output: S_{best} (Mapping of cloudlets to appropriate VMs)

```
begin
Population = InitializePopulation(Random, pairs < Cloudlet, VM >)
EvaluatePopulation(Population)
S_best = FindBest(Fitness)
add S_best to BestSolution
while ¬StopCondition(GenerationSize) do
    for i = 0 to λ do
        Parent = SelectParent(Population, μ)
        Child_i = Mutate
        add Child_i to Children
    end
    EvaluatePopulation(Children)
    S_best = FindBest(Fitness)
    add S_best to BestSolution
    Population = Children
end
return best S_best from BestSolution
```

11

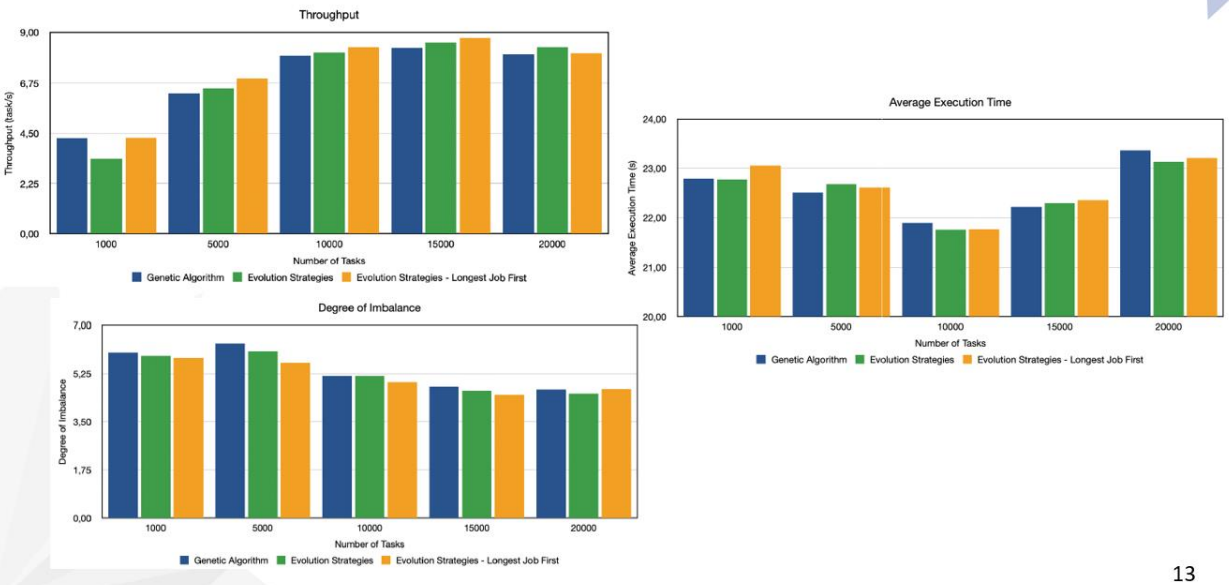
Час виконання завдань та Середнє використання ресурсів





12

Пропускна здатність системи, Середній час виконання завдань та Ступінь дисбалансу



13

ВИСНОВКИ

В процесі роботи реалізовано підвищення ефективності управління ресурсами у хмарних обчисленнях шляхом оптимізації процесу розподілу завдань, що дозволяє скоротити час виконання, покращити балансування навантаження та збільшити пропускну здатність системи

Вирішені наступні задачі дослідження:

а) проведено аналіз існуючих методів планування завдань у хмарних системах;

б) розроблено двоетапний підхід до оптимізації розподілу завдань;

в) впроваджено генетичний алгоритм для покращення ефективності управління ресурсами;

г) виконана оцінка продуктивності запропонованого методу через експериментальне моделювання;

д) проведено порівняння отриманих результатів із традиційними алгоритмами планування.

Публікація:

1. Волк М.О., Бугрій А.М., Бітюкова Є.В. та інші. Розподілене моделювання гетерогенних систем інтернету речей. Вісник Херсонського національного технічного університету. Том 2 No 1(92). 2025. - С. 45-50.
DOI: https://doi.org/10.35546/kntu2078-4481.2025.1.2.6

14