

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи



Харківський національний університет
радіоелектроніки

Магістерська кваліфікаційна робота



Методи та засоби моделювання розподілених систем туманних обчислень

Студент гр. СПм-23-2

Керівник

Журавель Д.С.

проф. каф. ЕОМ Волк М.О.

Харків, 2025

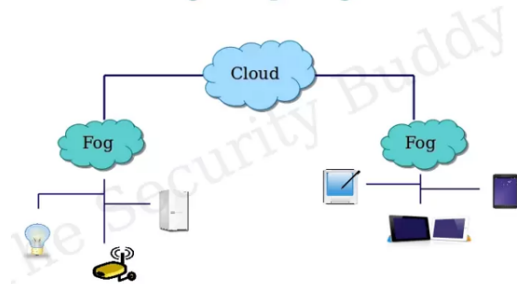
Актуальність дослідження методів засоби моделювання розподілених систем туманних обчислень

Туманні обчислення дозволяють мінімізувати затримки та зменшити навантаження на хмарні дата-центри, забезпечуючи обчислення ближче до кінцевого користувача. Проте відсутність гнучких інструментів моделювання стримує розвиток цієї галузі, що робить такі дослідження актуальними.

Недоліки сучасних рішень.

- Відсутність підтримки мобільності та багатьох мережних функцій у більшості існуючих симуляторів.
- Обмежена гнучкість налаштувань для моделювання реалістичних сценаріїв.
- Неврахування затримок, перевантаження мережі, падіння пакетів у інших симуляторах

Fog Computing



Мета та задачі роботи

Метою роботи є розвиток методів моделювання та розробка симулятора туманних обчислень, що дозволить моделювати сценарії із залученням IoT-пристроїв, оцінювати ефективність алгоритмів управління ресурсами та забезпечувати дослідження у сфері туманних обчислень.

Задачі роботи:

- Аналіз існуючих підходів до моделювання туманних обчислень.
- Розробка архітектури симулятора FogGRASS із підтримкою мережеских функцій та мобільності.
- Моделювання сценаріїв для оцінки продуктивності та масштабованості симулятора.
- Проведення експериментів з різними мережевими параметрами та моделями мобільності.

Об'єктом досліджень є процеси управління ресурсами та оптимізації обчислень у туманних мережах.

Предмет досліджень: Методи та алгоритми планування, розподілу завдань і управління ресурсами в симуляторі FogGRASS.

3

Порівняння існуючих систем моделювання туманних обчислень

Симулятори	Мова програмування	ОС	Мережа	Відкритий код	Мобільні пристрої	Моделі користувача	Планування	Споживання енергії
MobileFog	NA	–	ні	ні	так	ні	ні	ні
Edge-Fog cloud	Python	All	ні	так	обмежено	ні	ні	ні
EdgeCloudSim	Java	All	ні	так	ні	ні	так	ні
EmuFog	Python	All	так	так	ні	ні	ні	ні
RECAP simulator	NA	–	обмежено	NA	ні	ні	ні	NA
FogTorch	Java	All	ні	так	ні	ні	ні	ні
Cooja	C	Linux	ні	так	обмежено	ні	ні	ні
iFogSim	Java	All	ні	ні	так	ні	так	ні
Google IoT Sim	NA	Cloud	ні	ні	так	ні	ні	ні
IBM BlueMax	Java/Python	Cloud	ні	ні	так	ні	ні	ні
SimpleIoTSimulato	Java	Unix	ні	ні	так	ні	ні	ні
MobloTSim	Java	Linux	ні	так	так	ні	ні	ні
FogGRASS	C++	All	так	так	так	так	так	так

4

Основні математичні моделі системи

Інтенсивність прибуття даних $\lambda = \sum_{i=0}^M \lambda_i$

Запит, який може прийняти кожен туманний вузол $\varphi = \begin{cases} 1, & \text{якщо } \phi_f \geq \lambda \\ \frac{\phi_f}{\lambda}, & \text{якщо } \phi_f < \lambda \end{cases}$

Швидкість виконання в вузлах туману $S = \lambda \times \varphi = \begin{cases} \lambda, & \text{якщо } \phi_f \geq \lambda \\ \phi_f, & \text{якщо } \phi_f < \lambda \end{cases}$

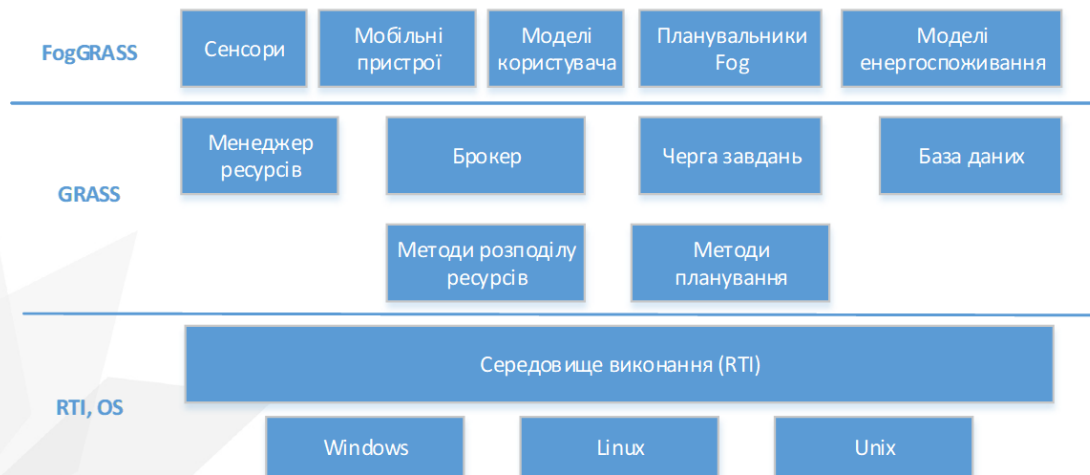
Середній час очікування кожного запиту у вузлу туману $T = \frac{\lambda}{k\theta_f} + \frac{1}{\theta_f}$

Модель енергоспоживання при обчисленні однієї задачі $E_t^f = \frac{Size_t}{N} P_w + \frac{Size_t}{\rho_f} P_{w, idle}$

Загальне енергоспоживання для C завдань буде оцінюватися як $E = \sum_{t=0}^C E_t^f$

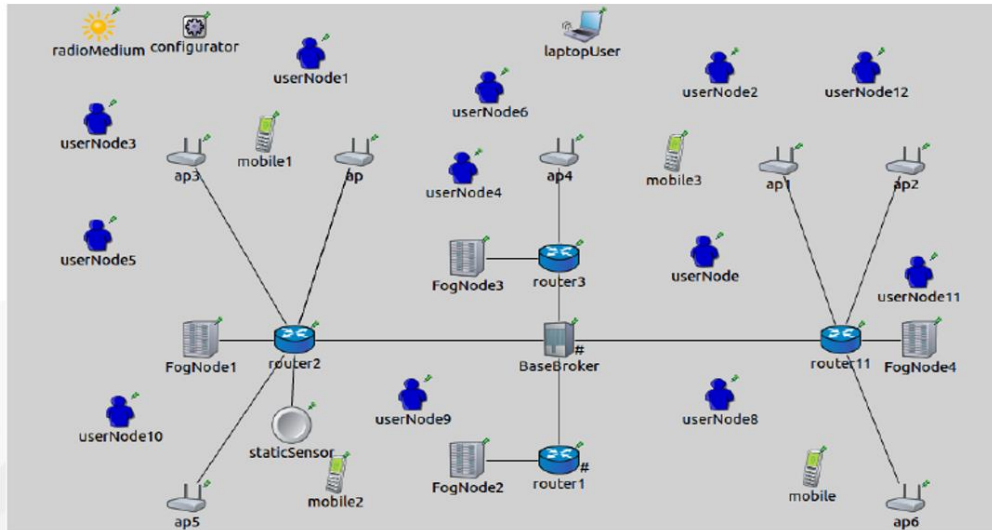
5

Структура запропонованого фрейворка FogGRASS



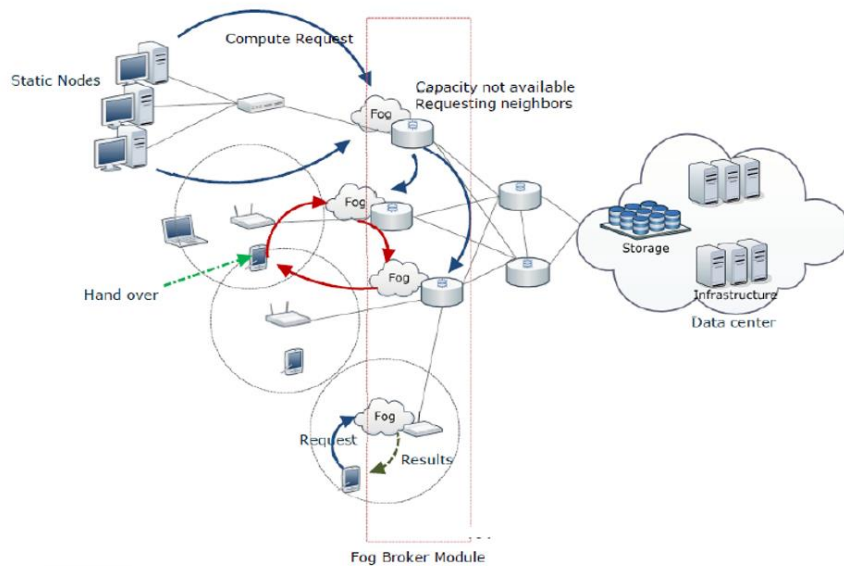
6

Графічний інтерфейс користувача FogGRASS



7

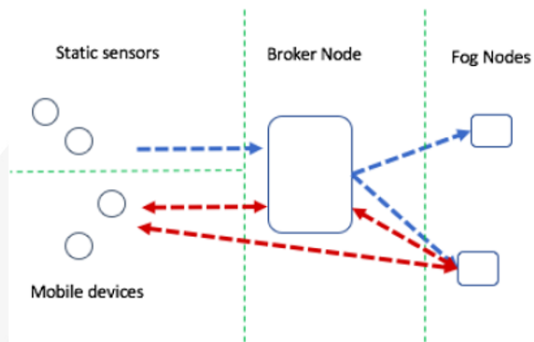
Архітектурна модель FogGRASS



8

Опис експериментів

Приклад змодельованої топології



Конфігурація експериментальної системи

Параметр системи	Значення
Кількість центральних процесорів	8
Кількість ядер	4
Кількість потоків на ядро	2
Частота процесора	1300Мгц
Пам'ять	16Гб

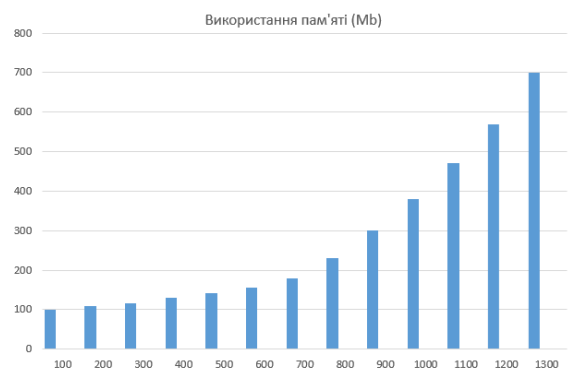
Параметри моделювання

Параметр системи	Значення
Брокери	До 500
Бездротові датчики	100-1000
Точки доступу	100-500
Пристрої	100-1500
Хмарні датацентри	1-8
Мобільні пристрої	100-1500
Туманні вузли	10-800
Швидкість мережі	10 Гбіт/с

Результати експериментів



Використання ЦП FogGRASS щодо кількості вузлів



Використання пам'яті FogGRASS від кількості вузлів

Апробація роботи та напрямки подальших досліджень



Публікації:

1. Volk Maksym, Buhrii Andrii, Kovtun Evgenii, Zhuravel Denys, Zborovskyi Mykhailo. Simulation and management of fog computing for IoT. 7-th International Scientific and Technical Conference "COMPUTER AND INFORMATION SYSTEMS AND TECHNOLOGIES". Kharkiv: NURE. – 2024. – P. 11-12
2. Волк М.О., Бугрій А.М., Брестовицький Р.М., Соробей Б.В., Журавель Д.С. Моделі та методи ефективного управління ресурсами в системах хмарних обчислень. Проблеми інформатизації: Матеріали дванадцятої міжнародної науково-технічної конференції. –Баку – Харків – Бельсько-Бяла , 21 – 22 листопада 2024 року. С.49. doi: <https://doi.org/10.32620/PI.24.t2>

Напрямки подальших досліджень

- Додаткове впровадження міграції віртуальних машин між туманними вузлами.
- Розробка методів інтеграції туманних федерацій для забезпечення сумісності між різними системами.
- Удосконалення алгоритмів балансування навантаження для туманних обчислень із врахуванням енергоефективності.

11

ВИСНОВКИ



В процесі роботи отримали розвиток методи моделювання та розроблено симулятор туманних обчислень, що дозволяють моделювати сценарії із залученням IoT-пристроїв, оцінювати ефективність алгоритмів управління ресурсами та забезпечувати дослідження у сфері туманних обчислень.

Вирішено наступні задачі:

- Проаналізовано існуючі підходи до моделювання туманних обчислень та виявлено їх недоліки.
- Розроблена архітектура симулятора FogGRASS із підтримкою мережових функцій та мобільності.
- Організовано моделювання сценаріїв для оцінки продуктивності та масштабованості симулятора.
- Проведено експерименти з різними мережевими параметрами та моделями мобільності, що довели ефективність запропонованих рішень.

12

ДОДАТОК Б

Алгоритми управління засобами моделювання

Algorithm 1 Algorithm – Broker Node (B)

```

1: List fogNodes[]  $\leftarrow$   $FD_M$ 
2: List devices[]  $\leftarrow$   $d_P$ 
3: Queue taskQueue[]  $\leftarrow$  null
4: Timer timer  $\leftarrow$  0
5:
6: while ( true ) do
7:   if MessageRecInWaiting then
8:      $Msg \leftarrow Message.Received$ 
9:     if  $Msg.Type == Result$  then
10:      Forward  $Msg$  to  $d_i$ 
11:     else if  $Msg.Type == FN.\Phi_i$  then
12:       Update  $FN_i.\Phi_i \leftarrow \Phi_i$ 
13:     else if  $Msg.Type == d_i.position$  then
14:       Update All  $d_i.Position$  for Handoff
15:     else if  $Msg.Type == Service_{req}$  then
16:        $taskQueue \leftarrow Msg$ 
17:     end if
18:   end if
19:
20:   if  $taskQueue \neq Empty$  then
21:      $Msg_{req} \leftarrow taskQueue_{pop}$ 
22:     if  $FN_i.workload < FN_i.\Phi_i$  then
23:       Forward  $Msg_{req}$  to  $FN_i$ 
24:     else
25:        $boolflag \leftarrow true$ 
26:       for  $i=0 \dots M$  do
27:         if  $FN_i.workload > FN_i.\Phi_i$  then
28:            $Msg_{req} \leftarrow taskQueue_{pop}$ 
29:           Forward  $Msg_{req}$  to  $i$ 
30:            $flag \leftarrow true$ 
31:         end if
32:       end for
33:       if  $flag$  then
34:          $Starttimer \leftarrow \Delta_T$ 
35:       end if
36:     end if
37:   end if
38:
39:   if timereexpire then
40:     for  $i=0 \dots M$  do
41:       if  $FN_i.workload > FN_i.\Phi_i$  then
42:          $Msg_{req} \leftarrow taskQueue_{pop}$ 
43:         Forward  $Msg_{req}$  to  $i$ 
44:       end if
45:     end for
46:   end if
47: end while

```

Algorithm 2 Algorithm – Fog Node (FN)

```

1: Timer timer  $\leftarrow$  0
2: Queue taskQueue[]  $\leftarrow$  null
3: while ( true ) do
4:   if MessageRecInWaiting then
5:     Msg  $\leftarrow$  Message.Received
6:     taskQueue  $\leftarrow$  Msg
7:   end if
8:   if taskQueue  $\neq$  Empty then
9:     Msgreq  $\leftarrow$  taskQueuepop
10:    Outcome  $\leftarrow$  ExecuteMsgreq
11:    Send(Outcome, B)
12:  end if
13:  if timer expired then
14:    Send( $\phi_i$ , B)
15:    timer  $\leftarrow$  reset
16:  end if
17: end while

```
