

## ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Харківський національний університет  
радіоелектроніки

Кваліфікаційна робота магістранта

МЕТОД ОРГАНІЗАЦІЇ  
ОБЧИСЛЕНЬ У ТУМАННОМУ  
СЕРЕДОВИЩІ

Виконав: студент групи СПм-22-5 Колесник З.В.  
Керівник роботи: професор Кучук Г.А.

2024

РОЗПОДІЛЕНІ  
ОБЧИСЛЕННЯ: ГРІД

2

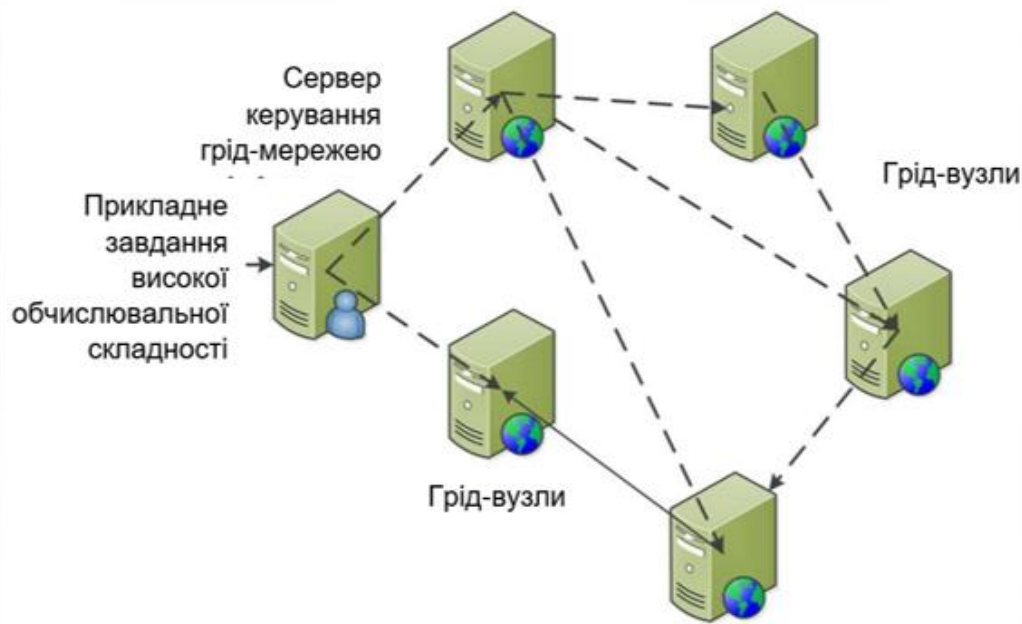
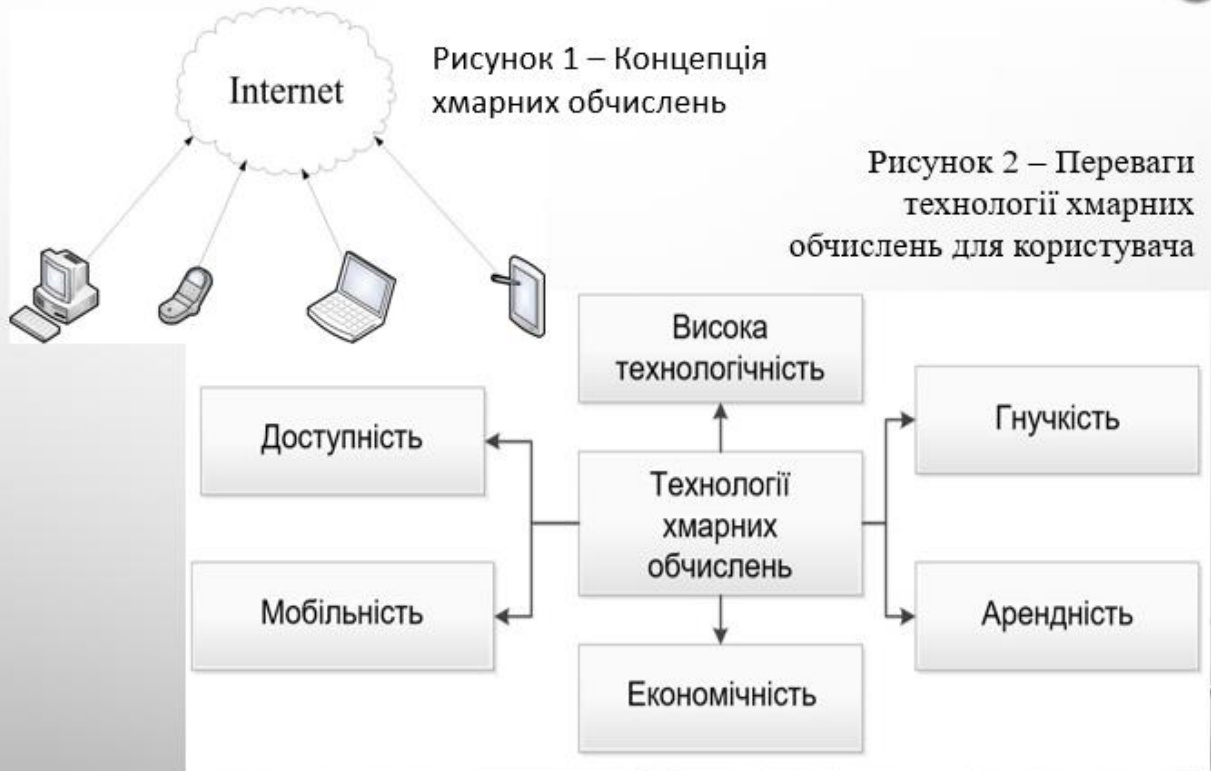


Рисунок 1 – Узагальнена архітектура технології грід-обчислень

## РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ: ХМАРА

3



## РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ: ТУМАН

4

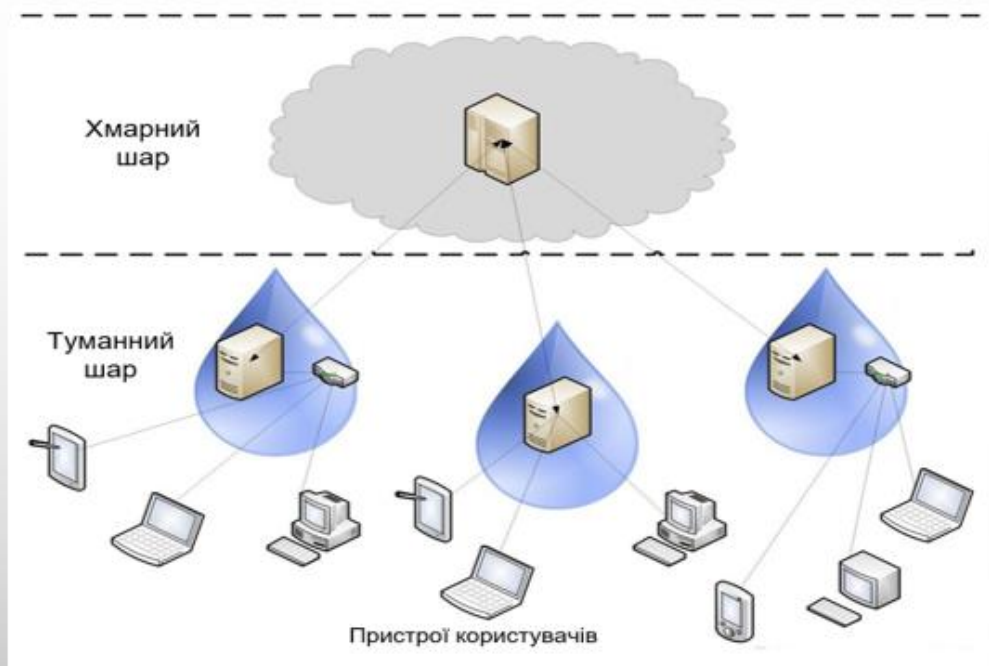


Рисунок 1 – Концепція туманних обчислень

## РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ: ТУМАН

5



Рисунок 1 – Перспективи використання технології туманних обчислень

6

## МЕТА ТА ЗАВДАННЯ РОБОТИ

**Мета дослідження** - підвищення ефективності використання обчислювальних пристроїв туманного середовища Інтернету речей.

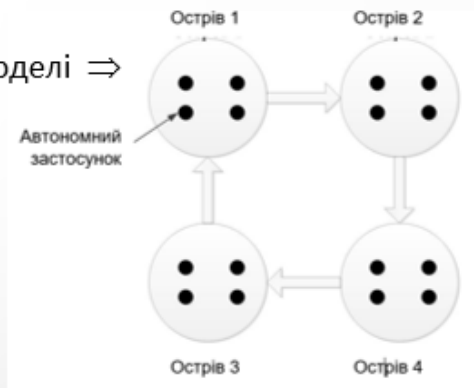
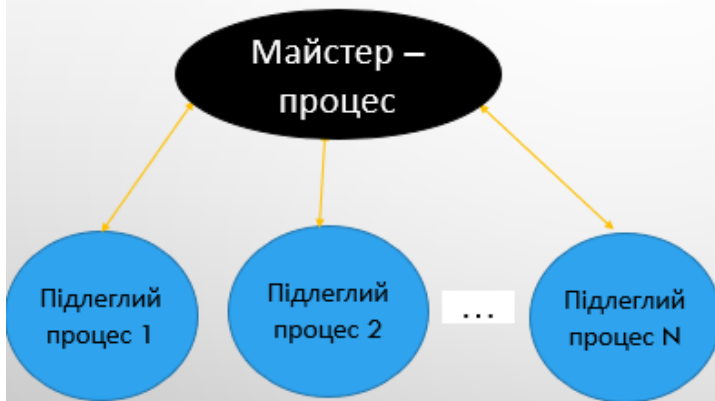
### Основні задачі дослідження:

- 1) розглянути питання перенесення навантаження до туманного шару в системах Інтернету речей;
- 2) розробити метод розподілу навантаження туманного шару Інтернету речей;
- 3) дослідити ефективність методу розподілу навантаження туманного шару Інтернету речей.

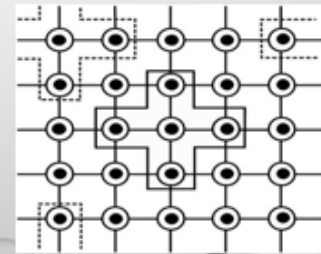
## МОДЕЛІ РОЗПАРАЛЕЛЮВАННЯ ЗАВДАНЬ ІНТЕРНЕТУ РЕЧЕЙ

Концепція острівної моделі ⇒

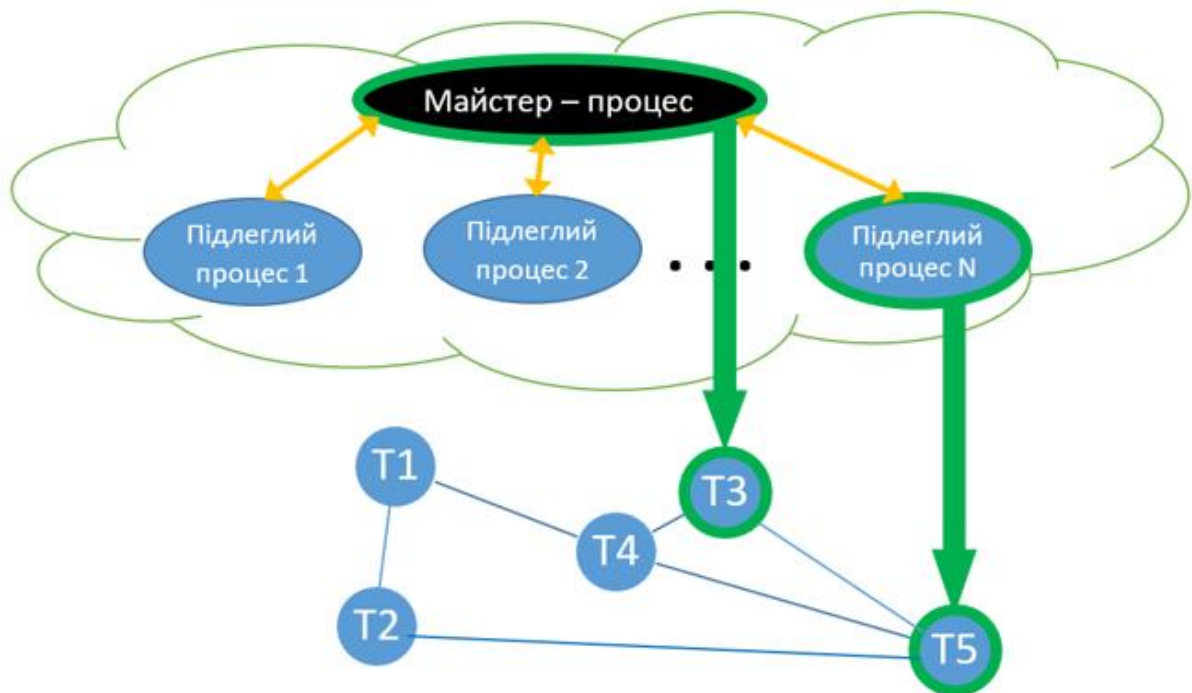
Модель «Майстер-підлеглий»



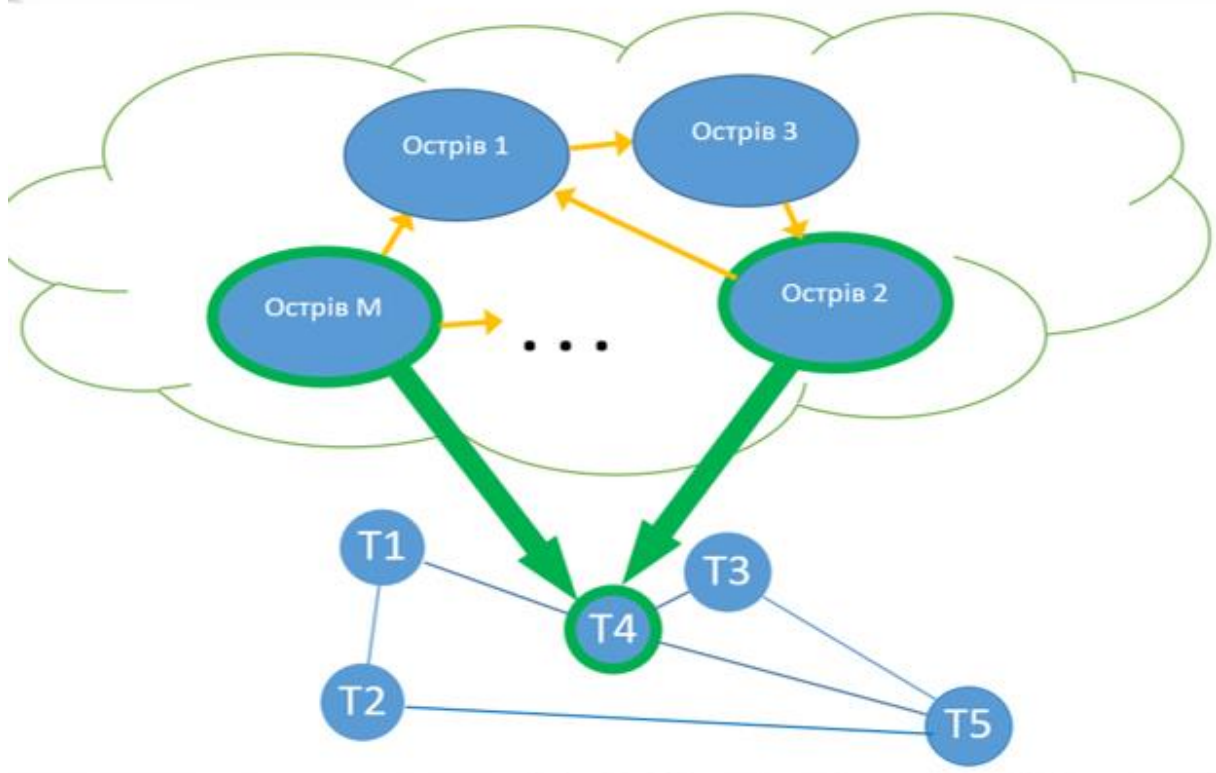
Дифузійна модель ⇒



## ПЕРЕНОСЕННЯ НАВАНТАЖЕННЯ ДО ТУМАННОГО ШАРУ

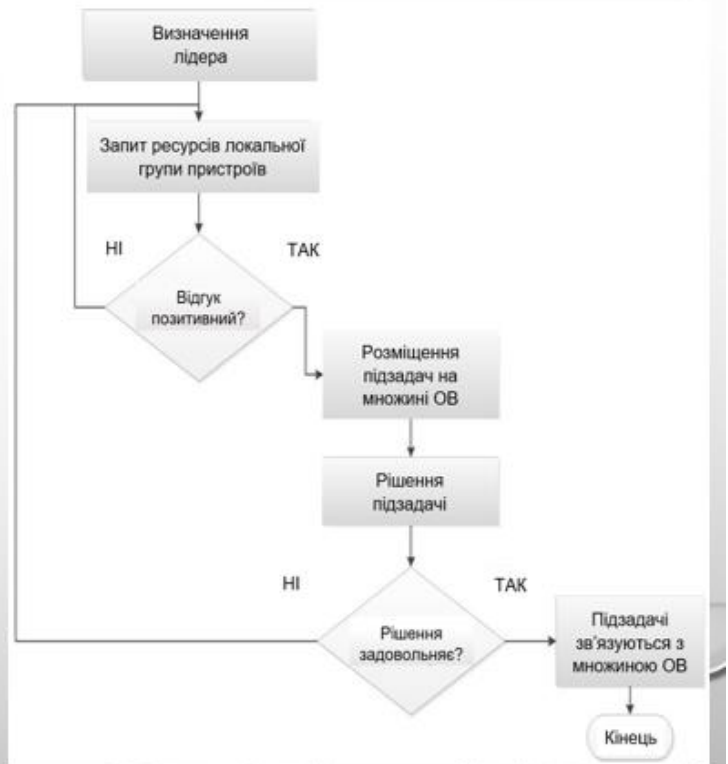
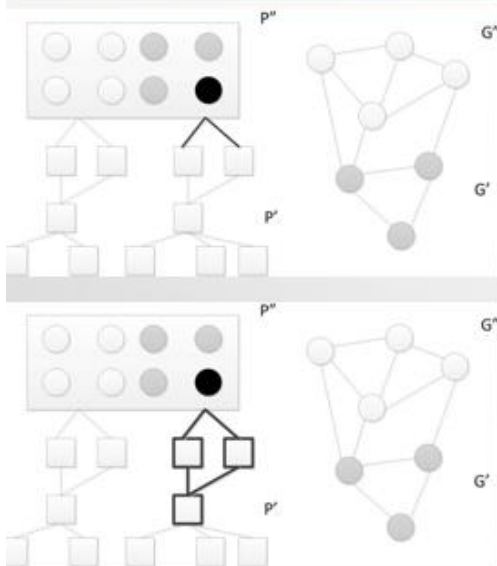


## ПЕРЕНЕСЕННЯ НАВАНТАЖЕННЯ ДО ТУМАННОГО ШАРУ



## МЕТОД ВИБОРУ ЛОКАЛЬНОЇ ГРУПИ ПРИБРОЇВ

**Завдання.** Розмістити підграф  $G'$  на обчислювальних пристроях туманного шару  $P'$ , й одночасно обчислювальні завдання підграфа  $G''$  повинні продовжувати виконуватися на обчислювальних вузлах хмарного шару  $P''$



## ОЦІНКА ЧАСУ ПОШУКУ ВАРІАНТА РОЗМІЩЕННЯ ПІДЗАВДАННЯ МЕТОДОМ ЛГП

11

$$T_{LDG} = \alpha N + \left[ \sum_{i=1}^s (vL_i) + N_t L_s \gamma \right] I_g,$$

$N$  – кількість вузлів, на якому від самого початку виконується обчислювальне завдання;

$N_t$  – кількість завдань, що підлягають розміщенню;

$L_i$  – розмір ЛГП для  $i$ -го запиту;

$I_g$  – загальна кількість ітерацій;

$S$  – кількість ітерацій, необхідна для формування ЛГП з можливістю вирішити завдання

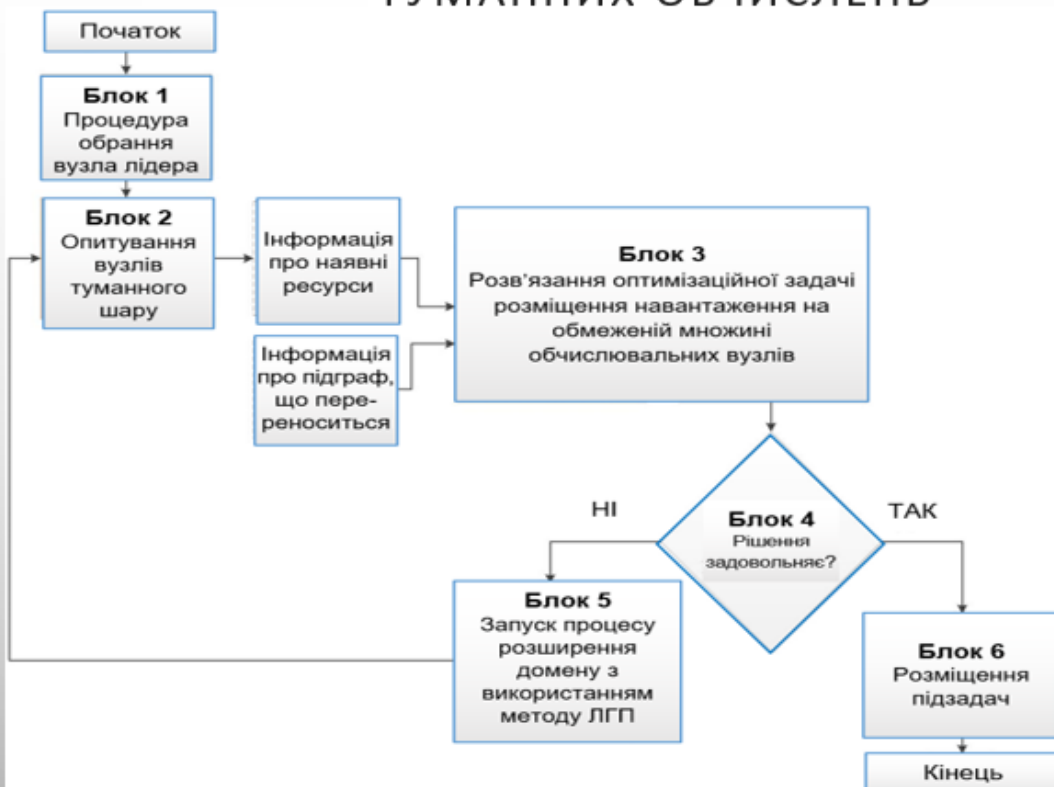
$\alpha$  – коефіцієнт, що зв'язує час вибору вузла-лідера і кількість вузлів;

$v$  – коефіцієнт, що зв'язує час обробки запитів та кількість вузлів, які мають бути опитані;

$\gamma$  – коефіцієнт, що зв'язує розмір завдання розподілу навантаження туманного шару і час його виконання

## МЕТОДИКА РОЗМІЩЕННЯ НАВАНТАЖЕННЯ В СЕРЕДОВИЩІ ТУМАННИХ ОБЧИСЛЕНЬ

12



# ВИСНОВКИ

Сукупність отриманих у кваліфікаційній роботі результатів дозволило вирішити актуальне науково-технічне завдання розробки методу розподілу навантаження туманного шару Інтернету речей.

В результаті проведених досліджень отримані такі результати:

1. Проведений аналіз сучасних тенденцій в області організації розподілених обчислень, зроблено висновок, що поставлене в роботі завдання щодо методу розподілу навантаження туманного шару Інтернету речей, є актуальним
2. Розглянуто питання перенесення навантаження до туманного шару в системах Інтернету речей; Сформульована постановка завдання перенесення навантаження до туманного шару в системах Інтернету речей та розроблена математична модель розподілу відповідного навантаження.
3. Розроблено метод розподілу навантаження туманного шару Інтернету речей, він дозволяє суттєво скоротити час опитування вузлів туманного шару в залежності від топології мережі за рахунок підбору найбільш ефективних алгоритмів, необхідних для його реалізації.
4. Досліджена ефективність методу розподілу навантаження туманного шару Інтернету речей, доведені переваги запропонованого методу над стандартним.

# ПУБЛІКАЦІЇ

ISSN 2673-7394

Национальний університет  
"Полтавська політехніка імені Юрія Кондратюка"  
National University  
"Yuri Kondratyuk Poltava Polytechnic"

**Системи управління, навігації та зв'язку**      **Control, navigation and communication systems**

Випуск 1 (75)      Issue 1 (75)

**Щоквартальне видання**      **Quarterly**

Засноване у 2007 році  
У журналі надруковано результати наукових досліджень, з метою їх широкого вжитку у науковій, освітній та інженерній сферах.

Засновник і видавець:  
Національний університет "Полтавська політехніка імені Юрія Кондратюка"

Телефон:  
+38 (061) 302-20-71

E-mail редакції:  
kntn@upm.edu.ua

Інформаційний сайт:  
http://journals.upm.edu.ua/kntn

Quarterly  
Founded in 2007  
Journal presents the research results on the development and improvement of control, navigation and communication systems in various areas.

Founder and publisher:  
National University "Yuri Kondratyuk Poltava Polytechnic"

Phone:  
+38 (061) 302-20-71

E-mail of the editorial board:  
kntn\_upm@un.net

Information site:  
http://journals.upm.edu.ua/kntn

Відповідальні за зміст: доктор технічних наук, професор кафедри систем управління, навігації та зв'язку Полтавського національного університету імені Юрія Кондратюка, доктор технічних наук, професор кафедри систем управління, навігації та зв'язку Полтавського національного університету імені Юрія Кондратюка, доктор технічних наук, професор кафедри систем управління, навігації та зв'язку Полтавського національного університету імені Юрія Кондратюка

Заснований 26 лютого 2007 року Полтавським національним університетом імені Юрія Кондратюка  
Полтавська політехніка імені Юрія Кондратюка, Лермонтова вул. 28, 37010, м. Полтава, Україна  
Сайт: http://journals.upm.edu.ua/kntn

Випуск 1 (75) за 2024 рік  
Відповідальні за зміст: доктор технічних наук, професор кафедри систем управління, навігації та зв'язку Полтавського національного університету імені Юрія Кондратюка, доктор технічних наук, професор кафедри систем управління, навігації та зв'язку Полтавського національного університету імені Юрія Кондратюка, доктор технічних наук, професор кафедри систем управління, навігації та зв'язку Полтавського національного університету імені Юрія Кондратюка

Полтава • 2024  
© "Національний університет "Полтавська політехніка імені Юрія Кондратюка"

Control, Navigation and Communication Systems, 2024, No. 1      ISSN 2673-7394  
UDC 621.372.2      doi: 10.26907/2673-7394.2024.1.004

Zakhar Klymenko<sup>1</sup>, Oleksandr Medvedev<sup>2</sup>, Oleksandr Derjabin<sup>3</sup>, Boris Kuchuk<sup>4</sup>  
<sup>1</sup>Kharkiv National University of Radio Electronics, Kharkiv, Ukraine  
<sup>2</sup>National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine

**FOG COMPUTING TECHNOLOGY IN DISTRIBUTED SYSTEMS**

**Abstract.** The concept of fog computing is an evolutionary step in the development of the cloud concept. It integrates existing problems among the present needs in the development of information technology. The emergence of this concept is closely related to the origin and development of the concept of the Internet of Things. The result: The subject area was analyzed. It includes an analysis of current trends in the field of expanding distributed computing, an analysis of the use of predictive algorithms and security models for solving optimization problems in distributed systems, an analysis of control, analysis and algorithms for solving the problem of transferring the computational load to distributed systems represented as the base of fog computing. Conclusions: It has been revealed that the concept of fog computing makes it possible to solve some of the problems associated with the need for communication infrastructure and the density of information exchange. But the data are not made more related to the high dimensions of the fog environment and the increased demand for efficiency of the distributed system.

**Keywords:** compute cross, distributed system, fog computing, cloud computing, Internet of Things.

**Introduction**  
Concepts, the core paradigm of fog computing, which can be considered as an extension of the cloud concept, has found wide application in many fields. This is due to the fact that today clouds do not meet the high requirements of mobility, low latency and local processing [1, 2]. A promising solution in this case seems to be a sensitive shift of computing, communication, control and decision-making to the places where data is generated, which corresponds to the basic principles of the concept of fog computing [3, 4].

The concept of fog computing is an evolutionary step in the development of the cloud concept. It integrates existing problems among the present needs in the development of information technology. The emergence of this concept is closely related to the origin and development of the concept of the "Internet of Things" (IoT) [1, 4]. Innovations in using the concept of fog computing are very diverse. They are determined by the development of related technologies. This concept has been successfully used in the analysis of various tasks in smart farms, smart transport, a health, a government, trade and financial services, industrial production, technological and business process management and much more [5-12].

The definition of "fog computing" was first introduced by Cisco in 2011. "Fog" and is currently actively developing due to the following factors:  
– the computing power of communication equipment and devices located at the fog level allows the additional calculations;  
– the number of end devices is growing very quickly; and this trend will continue in the foreseeable future;  
– it is all devices release the data processing center (including cloud) from performing complex computing processes [13, 14].

Fog computing is a hybrid model that provides distributed services to a distributed pool of mobile computing resources. Fog computing minimizes the network response time of supported applications and also provides end-devices with local computing resources and, if necessary, network connectivity to centralized services [15]. The fog computing architecture can be considered as a "layer" between the cloud and end (local) devices (Fig. 1).

Fig. 1. Fog computing concept

© Klymenko Z., Medvedev O., Derjabin O., Kuchuk B., 2024

Дякую за увагу!

ДОДАТОК Б

Публікація

ISSN 2073-7394

Національний університет  
"Полтавська політехніка імені Юрія Кондратюка"

National University  
"Yuri Kondratyuk Poltava Polytechnic"

**Системи  
управління,  
навігації  
та зв'язку**

Випуск 1 (75)

Щоквартальне видання

Засноване у 2007 році

У журналі відображені результати наукових досліджень з розробки та удосконалення систем управління, навігації та зв'язку у різних проблемних галузях.

**Засновник і видавець:**  
Національний університет  
"Полтавська політехніка імені Юрія Кондратюка"

**Телефон:**  
+38 (050) 302-20-71

**E-mail редколегії:**  
kuchuk\_nina@ukr.net

**Інформаційний сайт:**  
<http://journals.nupp.edu.ua/sunz>

**Control,  
navigation and  
communication  
systems**

Issue 1 (75)

Quarterly

Founded in 2007

Journal represent the research results on the development and improvement of control, navigation and communication systems in various areas

**Founder and publisher:**  
National University  
"Yuri Kondratyuk Poltava Polytechnic"

**Phone:**  
+38 (050) 302-20-71

**E-mail of the editorial board:**  
kuchuk\_nina@ukr.net

**Information site:**  
<http://journals.nupp.edu.ua/sunz>

*За достовірність викладених фактів, цитат та інших відомостей відповідальність несе автор*

*Журнал індексується міжнародними наукометричними базами: Index Copernicus (ICV = 82.05), General Impact Factor, Google Scholar, Academic Resource Index, Scientific Indexed Service*

*Затверджений до друку Вченою Радою Національного університету "Полтавська політехніка імені Юрія Кондратюка" (протокол від 09 лютого 2024 року № 2).*

*Свідоцтво про державну реєстрацію КВ № 24464-14404 ПР від 27.03.2020 р.*

*Включений до "Переліку наукових фахових видань України, в яких можуть публікуватися результати дисертаційних робіт на здобуття наукових ступенів доктора наук, кандидата наук та ступеня доктора філософії" до категорії Б – наказами МОН України від 17.03.2020 № 409 та від 09.02.2021 № 157*

Полтава • 2024

© Національний університет "Полтавська політехніка імені Юрія Кондратюка"

Zakhar Kolesnyk<sup>1</sup>, Oleksandr Mezhenyskyi<sup>1</sup>, Oleksandr Davykoza<sup>1</sup>, Heorhii Kuchuk<sup>2</sup>

<sup>1</sup>Kharkiv National University of Radio Electronics, Kharkiv, Ukraine

<sup>2</sup>National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine

## FOG COMPUTING TECHNOLOGY IN DISTRIBUTED SYSTEMS

**Abstract. Topicality.** The concept of fog computing is an evolutionary stage in the development of the cloud concept. It occupies a leading position among the general trends in the development of information technology. The emergence of this concept is closely related to the origin and development of the concept of the Internet of Things. **The results.** The subject area was analyzed. It includes an analysis of current trends in the field of organizing distributed computing, an analysis of the use of population algorithms and ontology models for solving optimization problems in distributed systems, an analysis of models, methods and algorithms for solving the problem of transferring the computational load in distributed systems implemented on the basis of fog computing. **Conclusion.** It has been revealed that the concept of fog computing makes it possible to solve most of the problems associated with the load on the communication infrastructure and the latency of information exchange. But they do not resolve issues related to the high dynamism of the foggy environment and the concomitant decrease in the efficiency of the distributed system.

**Keywords:** computer system, distributed system, fog computing, cloud computing, Internet of Things.

### Introduction

Currently, the new paradigm of fog computing, which can be considered as an extension of the cloud concept, has found wide application in many fields. This is due to the fact that today clouds do not meet the high requirements of mobility, low latency and local awareness [1, 2]. A promising solution in this case seems to be a selective shift of computing, communication, control and decision making to the places where data is generated, which corresponds to the basic principles of the concept of fog computing [3, 4].

The concept of fog computing is an evolutionary stage in the development of the cloud concept. It occupies a leading position among the general trends in the development of information technology. The emergence of this concept is closely related to the origin and development of the concept of the "Internet of Things" (IoT) [5, 6]. Scenarios for using the concept of fog computing are very diverse. They are determined by the development of related technologies. This concept has been successfully used in the creation of systems such as smart home, smart transport, e-health, e-government, trade and financial services, industrial production,

technological and business process management, and much more [7–12].

The definition of "fog computing" was first introduced by Cisco in 2011.

IIImE and is currently actively developing due to the following factors:

- the computing power of communication equipment and devices located in the fog layer allows for additional calculations;

- the number of end devices is growing very quickly, and this trend will continue in the foreseeable future.

- it is advisable to relieve the data processing center (including cloud ones) from performing complex computing processes [13, 14].

Fog computing is a tiered model that provides ubiquitous access to a shared pool of scalable computing resources. Fog computing minimizes the network response time of supported applications and also provides end devices with local computing resources and, if necessary, network connectivity to centralized services" [15]. The fog computing architecture can be considered as a "layer" between the cloud and end (user) devices (Fig. 1).

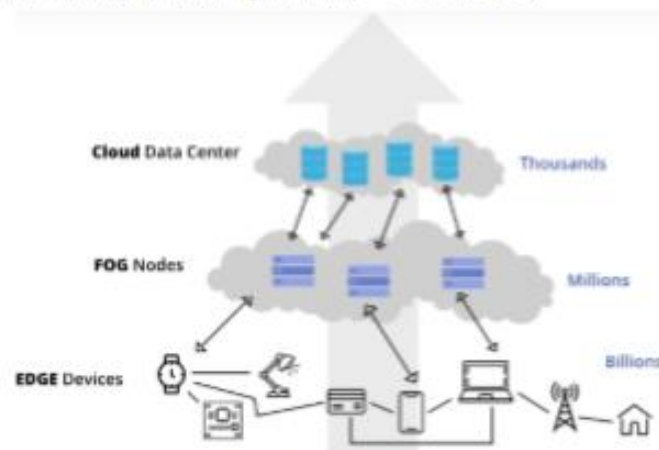


Fig. 1. Fog computing concept

## ДОДАТОК В

## Тексти окремих модулів програмного комплексу

```

// хвильовий алгоритм
onmessage = function(e) {
    var obj = e.data;
    var y =obj.i;
    var x =obj.j;
    mas = obj.m;
    if(y<0 || x< 0 || y>=mas.length ||x>=mas[0].length)
    postMessage('lim');

    var res = Array();
    if(checkPoint(x+1,y,mas))
    if(mas[y][x+1] ==0 )
        res[res.length] = {i :y, j:x+1, v: mas[y][x]+1};

    if(checkPoint(x-1,y,mas))
    if(mas[y][x-1] ==0 )
    res[res.length] = {i :y, j:x-1, v: mas[y][x]+1};

    if(checkPoint(x,y+1,mas))
    if(mas[y+1][x] ==0 )
    res[res.length] = {i :y+1, j:x, v: mas[y][x]+1};

    if(checkPoint(x,y-1,mas))
    if(mas[y-1][x] ==0 )
    res[res.length] = {i :y-1, j:x, v: mas[y][x]+1};

    postMessage(res);
};

this.checkPoint = function(j,i,mas) {
    return(i<0 || j< 0 || i>=mas.length ||j>=mas[0].length) ?
false : true;

//перевірка
for(i=0;i<pCount;++i)
{
    var obj = {m: mas, i: points[i].i, j:points[i].j };
    if(i<workers.length)
        workers[i].w.postMessage(obj);
    else
        self.createWorker(obj);

// формування тестової матриці
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

namespace AStarInMatrix
{
    class AStar
    {
        int[,] Map;
        int MapWidht;
        int MapHeight;

        int[,] WayMap;
        /// <summary>
        /// Конструктор
        /// </summary>
        public void ReadMap()
        {
            MapWidht = 16;
            MapHeight = 9;
            Map = new int[,] {
                {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1},
                {1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1},
                {1,0,1,0,1,1,1,1,1,1,1,1,1,1,1,1},
                {1,0,1,0,1,1,1,1,1,1,1,1,1,1,1,1},
                {1,0,1,0,1,1,1,1,1,1,1,1,1,1,1,1},
                {1,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1},
                {1,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1},
                {1,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1},
                {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}};
            WayMap = new int[10, 10];
        }
        /// <summary>
        public void DrawMap()
        {
            for (int y = 0; y < MapHeight; y++)
            {
                Console.WriteLine();
                for (int x = 0; x < MapWidht; x++)
                    if (Map[y,x]==1)
                        Console.Write("+");
                    else
                        Console.Write(" ");
            }
            Console.ReadKey();
            FindWave(1, 1, 3, 4);
        }
        /// <summary>
        /// пошук шляху
        /// </summary>
        /// <param name="startX">Координата старта X</param>
        /// <param name="startY">Координата старта Y</param>
        /// <param name="targetX">Координата фініша X</param>
        /// <param name="targetY">Координата фініша Y</param>
        public void FindWave(int startX, int startY, int
targetX, int targetY)

```

```

{
    bool add=true;
    int[,] cMap = new int[MapHeight,MapWidht];
    int x, y,step=0;
    for (y = 0; y < MapHeight; y++)
        for (x = 0; x < MapWidht; x++)
            {
                if (Map[y, x] == 1)
                    cMap[y, x] = -2;//індикатор1
                else
                    cMap[y, x] = -1;//індикатор2
            }
    cMap[targetY,targetX]=0;//обнулення
    while (add==true)
    {
        add = false;
        for (y = 0; y < MapWidht; y++)
            for (x = 0; x < MapHeight; x++)
                {
                    if (cMap[x, y] == step)
                        {
                            if (y - 1 >= 0 && cMap[x - 1, y] != -2 && cMap[x
- 1, y] = -1)
                                cMap[x - 1, y] = step + 1;
                            if (x - 1 >= 0 && cMap[x, y - 1] != -2 && cMap[x,
y - 1] = -1)
                                cMap[x, y - 1] = step + 1;
                            if (y + 1 < MapWidht && cMap[x + 1, y] != -2 &&
cMap[x + 1, y] = -1)
                                cMap[x + 1, y] = step + 1;
                            if (x + 1 < MapHeight && cMap[x, y + 1] != -2 &&
cMap[x, y + 1] = -1)
                                cMap[x, y + 1] = step + 1;
                        }
                }
        step++;
        add = true;
        if (cMap[startY,startX] != -1)//рішення є
            add = false;
        if (step > MapWidht * MapHeight)//рішення немає
            add = false;
    }
    for (y = 0; y < MapHeight; y++)
    {
        Console.WriteLine();
        for (x = 0; x < MapWidht; x++)
            if (cMap[y, x] == -1)
                Console.Write(" ");
            else
                if (cMap[y, x] == -2)
                    Console.Write("#");
                else
                    if (y == startY && x == startX)

```

```
                Console.WriteLine("S");
            else
                if (y == targetY && x == targetX)
                    Console.WriteLine("F");
                else
                    if (cMap[y, x] > -1)
                        Console.WriteLine("{0}", cMap[y, x]);
            }
        Console.ReadKey();
    }
}
```