

DOI: <https://doi.org/10.17721/ISTS.2019.1.27-35>

УДК 681.324

ПРАКТИЧНЕ ВПРОВАДЖЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ НА ОСНОВІ ОЦІНЮВАННЯ ЗОВНІШНІХ ВПЛИВІВ

Оксіюк Олександр ¹
orcid.org/0000-0001-9797-6015

Мирутенко Лариса ²
orcid.org/0000-0003-1686-261X

Шестак Яніна
orcid.org/0000-0002-1703-0316 ³

1) Київ, Київський національний університет ім. Тараса Шевченка, 24, oksiuk@ukr.net

2) Київ, Київський національний університет ім. Тараса Шевченка, myrutenko.lara@gmail.com

3) Київ, Київський національний університет ім. Тараса Шевченка, 24, lucenko.y@ukr.net

Історія статті:

Надійшла до редакції 26.07.2019

Прийнято 31.07.2019

Ключові слова:

інформаційна система; граф завдань; алгоритм захисту, кластеризація апаратних ресурсів; кластеризація ядер процесорів; нормалізований розподіл енергії.

Анотація: В статті була відзначена перевага інформаційних систем перед централізованими комплексами з точки зору забезпечення безпеки інфраструктури мережі, даних, що передаються, та процедур, що застосовуються у програмному середовищі комплексу. Було вказано на технічну простоту методів розширення інформаційних систем та, відповідно, їх масштабованість. Розроблено багаторівневу схему балансування енергоспоживання та обчислювальних ресурсів інфраструктури інформаційної системи, що базується на оптимізації графу завдань. Визначено методи оцінки оптимізації графу завдань, що базуються на показниках довжини нормалізованого графу нормалізованому розподілу енергії. Результати математичного моделювання співставлені зі статистичними даними для таких методів роботи з графами завдань як обчислення за ієрархічною структурою, алгоритми розбиття графу, методи на базі алгебраїчної теорії графів, структурування типу «Diamond Dags». Аналіз проводився для таких видів розподілу як рівномірний розподіл, біноміальний розподіл, геометричний розподіл. Результати співставлення вказують на достатньо високу точність прогнозування на рівні математичного моделювання. У більшості випадків значення максимального відхилення, представленого у вигляді відносної похибки, між результатами моделювання та статистичними даними залишаються в межах 10%, що показує адекватність моделювання. Тим не менш, для чотирьох пар функцій довжини нормалізованого графу і нормалізованого розподілу енергії демонструють максимальне відхилення більше 10%, що вказує на необхідність доопрацювання моделі та подальших статистичних досліджень. Запропоновано використовувати при організації захисту ресурсів інформаційної системи методів кластеризації, що дозволяють побудувати прозору схему функціонування комплексу та побудувати алгоритми моніторингу.

Abstract: The article highlighted the advantage of information systems in front of centralized systems in terms of ensuring the security of network infrastructure, data transmitted, and procedures used in the software environment of the complex. It was pointed out the technical simplicity of the methods of expansion of information systems and, accordingly, their scalability. A multilevel energy balance scheme and computing resources of the information system infrastructure, based on the optimization of the task graph, have been developed. The methods of estimating the optimization of the graph of tasks, which are based on the indicators of the normalized graph normalized distribution of energy, are determined. The results of mathematical modeling in comparison with statistical data for such methods of working with task graphs as calculations by hierarchical structure, graph split algorithms, methods based on algebraic theory of graphs, structuring of "Diamond Dags" type. The analysis was conducted for such types of distribution as uniform distribution, binomial distribution, geometric distribution. The results of the comparison point to a sufficiently upstart prediction accuracy at the level of mathematical modeling. In most cases, the value of the maximum deviation, presented as a relative error,

between simulation results and statistical data remains within 10%, which shows the adequacy of the simulation. Nevertheless, for the four pairs of functions of the length of the normalized graph and the normalized energy distribution, the maximum deviation is greater than 10%, indicating the need for a revision of the model and further statistical studies. It is proposed to use in organizing the protection of resources of the information system of clusterization methods, which allow to construct a transparent scheme of functioning of the complex and to construct monitoring algorithms.

1. ВСТУП

Забезпечення високого рівня безпеки інформаційної системи в сучасних умовах стає все більш складним завданням з точки зору постійного, динамічного зростання технологічної складності і розвитку інформаційного середовища.

Будуючи та експлуатуючи інформаційної системи потрібно брати до уваги забезпечення їх живучості, тобто, можливі загрози стійкості та захищеності, а особливо ті, що безпосередньо виникають за участі зовнішніх впливів. Процес оцінювання захищеності від зовнішніх впливів інформаційних систем **є актуальним** і потребує дослідження, у зв'язку з постійною модернізацією існуючих та появою нових інформаційних технологій та необхідністю їх функціональної безпеки від зовнішніх і внутрішніх загроз.

Метою роботи є підвищення безпеки інформаційних систем на базі запропонованої технології розроблених методів і моделей оцінювання зовнішніх впливів.

Було розглянуто та вдосконалено методику побудови графу завдань інформаційної системи, що базується на графіку запитів, а також паралелізації планування завдань та багаторівневій схеми розподілу апаратно-програмних ресурсів. Вирішено задачу оптимізації процесу кластеризації апаратних ресурсів шляхом побудови системи віртуальних машин та подальших етапів визначення екстремумів функції розміру графіка завдань та функції повного значення енергії.

При побудові сучасної інформаційної системи (ІС) слід враховувати вимоги по масштабуванню систему, а також збільшення кількості ядер n_j та тактової частоти f центральних процесорів (ЦП) інфраструктури, як на рис.1. Це надає можливість для збільшення обчислювальних потужностей, організації багатопотокової архітектури, віртуалізації апаратних ресурсів та впровадження заходів по економії енергоспоживання.

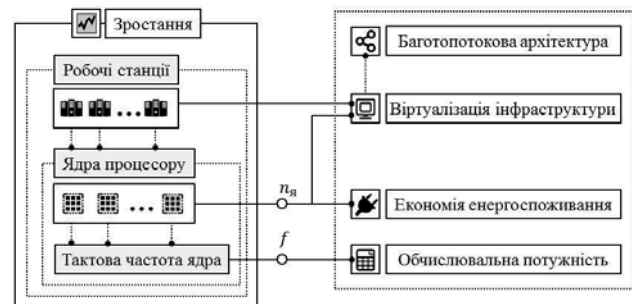


Рис. 1 - Вплив масштабування інфраструктури на можливості ІС

Підвищення продуктивності, що надається інфраструктурі ІС системою серверів на основі багатоядерних процесорів полягає у першу чергу у реалізації паралелізму процесів обробки. При цьому надається можливість запускати алгоритми багатопроекторної обробки та втілювати принципи паралельної архітектури: суперскалярності, багатонитевості та архітектури векторних процесорів. У рамках математичного моделювання можна розглянути процес паралельних обчислень як набір n паралельних завдань з обмеженням черговості, як підмножину множини графу завдань $G(K(i), R(j))$, де $K(i)$ — набір завдань, а $R(j)$ — набір завдань з обмеженням черговості (ребра графу). При цьому i передуює j , тобто $i < j$, завдання i характеризуються такими параметрами як кількість ядер n_j , що використовуються при виконанні завдання розміру $\mu_i \in \{\mu_1, \mu_r\}$ та набір максимальних вимог $\delta_i \in \{\delta_1, \delta_r\}$, що визначається через кількість команд, що мають бути виконані. Відповідно, робота завдання i може бути визначена як добуток $w_i = \delta_i \mu_i$.

Оцінка впливу системи розподілення навантаження на продуктивність роботи ІС.

Багатоядерність процесорів надає широкі можливості по оптимізації роботи обчислювальної системи та координуванню навантаження, як обчислювальних ресурсів, так і електронного вантаження [11-15]. Слід зауважити, що обчислювальна потужність багатоядерного процесору з числом ядер n_j наближається до значення потужності n_j однопоточних процесорів, або потужності одного

однопоточного процесору з n_j разів більшою частотою. Але при цьому рівень електроенергії яка необхідна для його роботи E_n складає менше значення ніж рівень електроенергії, що використовується однопоточним процесором з більшою тактовою частотою:

$$E_n = \frac{E_1}{n_j^\sigma}, \quad (1)$$

де параметр залежить від особливостей побудови багатопоточного процесору і лежить у таких межах:

$$\left\{ \begin{array}{l} \sigma \in N \setminus \{\text{displaystyle } \mathbb{N}\} \\ \sigma > 1 \end{array} \right\} \quad (2)$$

При належній організації роботи ІС на основі багатопоточних серверів з'являється можливість суттєво підвищити продуктивність обробки запитів системою, що базується на розпаралелюванні роботи. При паралельній архітектурі ІС взаємодія між ядрами може здійснюватися через передачу даних і через спільну пам'ять, а ступінь паралелізму, таким чином, збільшується зі збільшенням числа ядер. Це відкриває широкі можливості для створення мультизадачних алгоритмів, а отже обробки значної запитів користувачів на одиницю часу. У разі виконання ІС значної кількості паралельних запитів головним завданням стає виділення ядер для кожного завдання, створення, так званого, графіку завдань, що забезпечує балансування навантаження на апаратні ресурси і оптимальний рівень споживання електроенергії [6-10].

Графік завдань є сучасним засобом побудови алгоритмів роботи ІС. Згідно даної моделі запити, що знаходяться на різних рівнях системи вважаються незалежними, і тому для їх вирішення можуть бути використані типові алгоритми. Це певною мірою обмежує інструментарій розробки ІС, але при цьому допомагає проблемі обмеження черговості при плануванні завдань і спрощує аналіз системи [7, 8]. Іншим засобом є кластеризація апаратних ресурсів ІС. У рамках даної архітектури кожне завдання надає запит до кількох ядер, тому виникає необхідність розділення ядер процесора на кластери. Крім того схема враховує механізм організації розподілу ресурсів, тобто визначення оптимальних пропорцій при наданні для окремого завдання апаратних ресурсів з метою забезпечення надійної роботи при обробці

запитів, що було розглянуто у попередньому підрозділі.

Проектування ІС починається з розподілу для ЦП серверів електроенергії, рівень якої визначається через величину:

$$P = \eta (f \times CU^2) \quad (3)$$

де η – коефіцієнт використання ЦП, f – тактова частота, C – ємність навантаження, U – напруга живлення.

Слід зауважити, що дані параметри не є незалежними, так, наприклад напруга живлення є функцією від частоти:

$$U = b f^{(\sigma+1)}, \quad (4)$$

де b є константою, що визначається окремо для кожного типу багатопоточного процесору.

Наступним кроком при розробці ІС є визначення швидкодії роботи ЦП, що також залежить від тактової частоти f та параметру, що характеризує процесор d :

$$S = df \quad (5)$$

Отже, показник споживання електроенергії ЦП серверів ІС може бути визначено як:

$$P = \eta b^2 C f^{2\sigma+3} \quad (6)$$

що можна спростити до вигляду:

$$P = \beta s^\alpha \quad (7)$$

$$\text{де } \begin{cases} \alpha = 2\sigma + 3 \\ \beta = \eta b^2 C / d \alpha \end{cases} \quad (8)$$

При обробці r паралельних запитів, завдання з обмеженням черговості є підмножиною множини графу завдань. За схемою, що буде розглянута у даній роботі виконання завдання j слідує за виконанням завдання i . Всі завдання i при цьому виконуються паралельно і характеризуються наступним набором параметрів [6-10]:

- кількість ядер ЦП серверів, що використовуються при виконанні завдання μ_i ;
- максимальний рівень вимог до обчислювальних потужностей ядер δ_i , що

визначається через кількість команд, що мають бути виконані при обробці запиту;

- робота завдання $w_i = \mu_i \delta_i$.

Швидкодія обробки запиту і потужність, що надається для його виконання можуть бути пов'язані через параметр α :

$$P_i = S_i^\alpha \quad (9)$$

а час обробки запиту через параметр:

$$t_i = \frac{\delta_i}{S_i} \Rightarrow t_i = \frac{\delta_i}{\sqrt[\alpha]{P_i}} \quad (10)$$

У випадку коли система складається з однакових ЦП, що при балансуванні навантаження працюють з однією швидкістю S_i протягом одного t_i , електроенергія, яка витрачається на запит і множини $[1, r]$ визначається наступним чином:

$$e_i = \mu_i P_i t_i \Rightarrow e_i = w_i S_i^{\alpha-1} \quad (11)$$

Таким чином, процес оптимізації ІС при цьому складається з двох базових етапів [17, 18]:

- етап побудови графіка завдань, що характеризується найменшим розміром, з урахування обмеження на величину повної енергії E_Σ ;
- визначення мінімуму функції повної енергії E_Σ , з урахуванням обмежень на розмір графіка завдань.

При цьому, на першому етапі за основу береться множина розміру завдань $\mu_i \in \{\mu_1, \mu_r\}$ та множина вимог $\sigma_i \in \{\sigma_1, \sigma_r\}$, а для другого множина значень потужностей $p_i \in \{p_1, p_r\}$, як показано на Рис. 2.



Рис. 2 - Оптимізація роботи ІС через створення графіку завдань

Оптимальне значення розміру графіка завдань G_{min} може бути визначено через значення повної роботи W :

$$T^{min} \geq n_y (W^\alpha / E)^{\frac{1}{\alpha-1}}, \quad (12)$$

а оптимальне значення функції повної енергії у такому разі, відповідно визначається як:

$$E_\Sigma^{min} \geq W^\alpha (n_y T)^{1-\alpha}, \quad (13)$$

де повний об'єм роботи для r паралельних запитів, обчислюється як:

$$W = \sum_{i=1}^r w_i \Rightarrow W = \sum_{i=1}^r \mu_i \delta_i \quad (14)$$

Узагальнення математичної моделі розподілення навантаження на апаратні ресурси ІС включає у себе побудову системи багатоядерних процесорів з n_{y-i} ідентичними ядрами у кожному. При цьому можна побудувати графік роботи x алгоритмів, що виконуються паралельно, яким відповідає набір графіків завдань $G_i \in \{G_1, G_x\}$ та багатоядерні процесори $P_i \in \{P_1, P_y\}$ з кількістю ядер у кожному $n_{y-i} \in \{n_{y-1}, n_{y-y}\}$, відповідно [11-16]. Оптимізація полягає у тому, щоб для r паралельних завдань розміру $\mu_i \in \{\mu_1, \mu_r\}$, що характеризуються вимогами $\delta_r \in \{\delta_1, \delta_r\}$ отримати мінімальний розмір графіка завдань T^{min} (рівняння 12) через обрахування множини работ завдань $\{w_1 \dots w_r\}$. При цьому слід врахувати обмеження на повне значення енергії системи та мінімум функції енергії E_Σ^{min} (рівняння 12). Також для множини $\{p_1 \dots p_r\}$ необхідно визначити мінімум функції енергії E_Σ^{min} , з урахуванням обмеження на розмір графіка завдань T ; оптимальні параметри роботи ІС отримуються при співставленні цих значень, що можуть бути обраховані через наступну систему рівнянь:

$$\begin{cases} T^* \geq \alpha^{-1} \sqrt{\frac{n_j}{E} \left(\frac{\sum_{i=1}^r w_i}{n_j} \right)^\alpha} \\ E^* \geq n_j \frac{\left(\sum_{i=1}^r w_i / n_j \right)^\alpha}{T^{\alpha-1}} \end{cases} \quad (15)$$

Як було вказано ребра графу завдань мають будуватися таким чином, що для вузлів i та j — $i < j$, де i попередник j . Додатковою умовою є те, що якщо існує вузол k , для якого можна знайти ребра (i, k) та (k, j) , то ребро (i, j) виключається з графу, тобто графу завдань є направленим ациклічним графом, який може бути розкладено на y рівнів $l \in \{1 \dots y\}$.

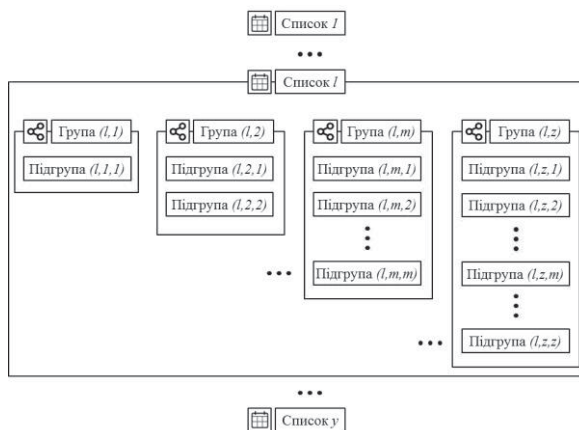


Рис.3 - Структурування графіку завдань на рівні списків, груп і підгруп

Завдання i знаходиться на рівні l^+ , якщо кількість вузлів від даного завдання до початкового завдання по найдовшому шляху складає величину l^+ . При цьому всі завдання, які знаходяться на одному рівні виконуються незалежно один від одного. Список незалежних паралельних завдань (рис. 3) на рівні l поділяється на z груп $(l, 1) \dots (l, m) \dots (l, z)$. Таким чином, для величини $m \in [1, z - 1]$, можна отримати групу (l, m) , що характеризується розміром n_{j-i} (по кількості ядер), який лежить у межах:

$$\frac{n_j}{m+1} < n_{j-i} \leq \frac{n_j}{m} \quad (16)$$

Така система ефективно працює з великим набором завдань мінімального розміру, що є типовим для сучасних ІС. На її основі можна

побудувати алгоритми, що поділяють n_j ядер на m кластерів, кожен з яких містить n_j/m ядер. Кожен окремий кластер обслуговує одне завдання групи (l, m) , де не паралельні завдання кількістю $n \leq m$ виконуються одночасно.

Планування паралельних завдань групи (l, m) на m кластерах, кожне завдання якого характеризується множиною розміру завдань $\mu_i \in \{\mu_1, \mu_r\}$ та набором максимальних вимог $\delta_i \in \{\delta_1, \delta_r\}$ може бути представлено як побудова списку послідовних завдань на m процесорах, де кожне завдання характеризується максимальними вимогами до процесору (без врахування енерговитрат, див. рівняння 1-2). Якщо багатоядерний процесор з n_j ядрами розбито на m кластерів, у рамках виконання завдань з групи (l, m) дана група поділяється на підгрупи завдань $\{(l, m, 1) \dots (l, m, m)\}$, причому кожна з зазначених підгруп виконується на одному кластері.

Зазначений етап розрахунку оптимальних значень енергії та розміру графіка завдань виконується після етапу кластеризації ядер процесорів. Таким чином, процес графу завдань здійснюється шляхом визначення часу виконання завдань i за допомогою параметрів з набору максимальних вимог δ_i . Алгоритм планування списку у такому разі складатиметься з наступних етапів [11-16]:

1. Окрема задача k має бути розподілена по відношенню до кластеру $k \in [1, m]$;

2. Перше позапланове завдання списку $(m+1)$ видаляється зі списку і включається для виконання до кластеру;

3. Перехід до пункту (1) з повторенням процесу до завершення повного набору процесів списку.

У таблиці вказано повний набір алгоритмів, що базуються на значеннях розміру завдання $\mu_i \in \{\mu_1, \mu_r\}$, набору максимальних вимог $\delta_i \in \{\delta_1, \delta_r\}$ та роботи завдання w_i , що можуть бути використані при упорядкуванні складових графу завдань.

Розділення потужностей ІС за напрямками, специфікою роботи, та оцінка підвищення ефективності виконання завдань ІС внаслідок здійснення такого розподілення.

Поділення складових графу завдань ІС на списки, групи та підгрупи дозволяє розробити оптимальну схему розподілу енергії та

обчислювальної потужності апаратних ресурсів інфраструктури.

завдань підгрупи мінімізується при відповідних значеннях p_i і T [11-16]:

$$\left[\begin{aligned} p_i &= \frac{\alpha^{-1} \sqrt{\sum_i E_i^\alpha \div \sum_i \mu_i \delta_i^\alpha}}{\mu_i} \\ T &= \alpha^{-1} \sqrt{\frac{\sum_i \mu_i \delta_i^\alpha}{E_i}} \end{aligned} \right. \quad (16)$$

Відповідно на другому рівні розглядається взаємодія між підгрупами завдань. При цьому n_a ядер розділяється на m кластерів, причому кожен кластер утримує n_a / m потужності.

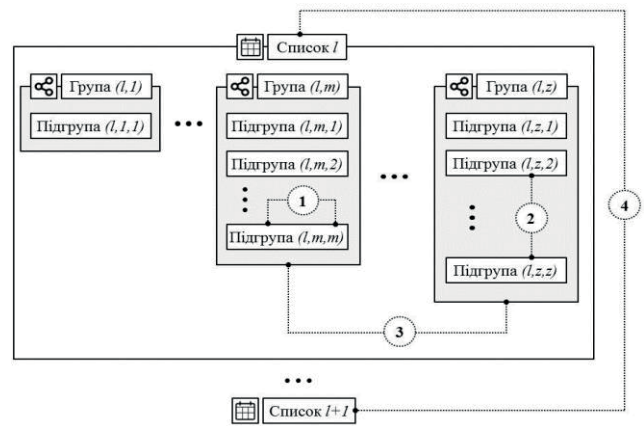


Рис. 4 - Багаторівнева схема оптимізації розподілу апаратних ресурсів ІС

Кожен кластер розглядається як окремий елемент призначений для обробки окремої задачі, відповідно повний набір задач поділяється на m підгруп, отже кожне завдання підгрупи k виконується кластером k . Оптимальні значення p_i , E і T можуть бути визначені аналогічно до рівняння (17) як:

$$\left[\begin{aligned} p_i &= \frac{\alpha^{-1} \sqrt{\sum_i E_{k|i}^\alpha / \sum_i \mu_{k|i} \delta_{k|i}^\alpha}}{\mu_i} \\ E_k &= \frac{E \sum_i \mu_{k|i} \delta_{k|i}^\alpha}{\sum_j (\sum_i \mu_{j|i} \delta_{j|i}^\alpha)} \\ T &= \alpha^{-1} \sqrt{\frac{\sum_i \mu_{k|i} \delta_{k|i}^\alpha}{E}} \end{aligned} \right. \quad (18)$$

На третьому рівні оптимізація завдань здійснюється для всіх z груп окремого списку, причому у відповідності до правил розбиття

Таблиця 1. Схеми організації списку планування за ключовими параметрами

Параметр	Назва	Принцип
Розмір завдання	SRF: Smallest Requirement First	$\delta_1 \leq \delta_2 \leq \dots \leq \delta_n$
	LRF: Largest Requirement First	$\delta_1 \geq \delta_2 \geq \dots \geq \delta_n$
Максимальні вимоги	SSF: Smallest Size First	$\mu_1 \leq \mu_2 \leq \dots \leq \mu_n$
	LSF: Largest Size First	$\mu_1 \geq \mu_2 \geq \dots \geq \mu_n$
Робота по виконанню завдання	SWF: Smallest Work First	$w_1 \leq w_2 \leq \dots \leq w_n$
	LWF: Largest Work First	$w_1 \geq w_2 \geq \dots \geq w_n$

Паралельні задачі розкладаються на u рівнів, завдання на кожному рівні l поділяються z груп $\{(l,1) \dots (l,z)\}$, а завдання у кожній групі m поділяються на z підгруп $\{(l,m,1) \dots (l,m,m)\}$. Таким чином, оптимальна схема розподілу потужності може бути визначена через обмеження довжини графу або енергоспоживання в межах, як груп так і підгруп, що дозволяє побудувати чотири рівні [11-16] аналізу (рис. 4):

1. Оптимізація розподілу апаратних ресурсів РІТС у межах однієї підгрупи завдань;
2. Оптимізація розподілу апаратних ресурсів РІТС на рівні взаємодії між підгрупами завдань;
3. Оптимізація розподілу апаратних ресурсів РІТС на рівні взаємодії між групами завдань;
4. Оптимізація розподілу апаратних ресурсів РІТС на рівні взаємодії між списками завдань.

Реалізація схеми оптимізації для першого рівня є найпростішим у реалізації етапом.

Виконання завдань у межах підгрупи здійснюється послідовно, довжина списку

кожна j група утримує i підгруп, причому $i = j$.
Значення T у відповідності до (рівняння 18) можна записати як [11-16]:

$$T = \alpha^{-1} \sqrt[\alpha]{\frac{\sum_i \mu_{ji} \delta_{ji}^\alpha}{E_j}} \quad (19)$$

Зменшення T відбувається через оптимальний розподіл E_j

$$\begin{cases} E_j = \left(\frac{\sqrt[\alpha]{\sum_j \mu_{ji} \delta_{ji}^\alpha}}{\sum_i \sqrt[\alpha]{\sum_j \mu_{ji} \delta_{ji}^\alpha}} \right) E \\ T = \alpha^{-1} \sqrt[\alpha]{\frac{\sum_i \sum_j \mu_{ji} \delta_{ji}^\alpha}{E}} \end{cases} \quad (20)$$

Відповідно, при оптимізації розподілу апаратних ресурсів ІС на рівні взаємодії між списками завдань (четвертий рівень) ми переходимо до $l \in [1, y]$ списків завдань. Значення T аналогічно до виразу (20) розраховується як:

$$T = \sum_l \frac{\sum_i \sqrt[\alpha]{\mu_{li} \delta_{li}^\alpha}}{\alpha^{-1} \sqrt[\alpha]{E_l}} \quad (21)$$

На четвертому рівні зменшення T аналогічно до виразу (21) відбувається через оптимальний розподіл E_l :

$$\begin{cases} E_l = \left(\frac{\sum_j \sqrt[\alpha]{\sum_i \mu_{li} \delta_{li}^\alpha}}{\sum_i \sum_j \sqrt[\alpha]{\sum_i \mu_{li} \delta_{li}^\alpha}} \right) E \\ T = \alpha^{-1} \sqrt[\alpha]{\frac{\sum_l \sum_j \sum_i \mu_{li} \delta_{li}^\alpha}{E_l}} \end{cases} \quad (22)$$

У такому разі довжина нормалізованого графа завдань (NSL: Normalized Schedule Length) може бути визначена відповідно до [15] як:

$$NSL = n_\alpha \left(\frac{\left(\sum_i \sqrt[\alpha]{\sum_j \sum_i \mu_{li} \delta_{li}^\alpha} \right)^{\frac{\alpha}{\alpha-1}}}{W} \right)^{\frac{\alpha}{\alpha-1}} \quad (23)$$

Відповідно нормалізований розподіл енергії (NEC: Normalized Energy Consumption) обраховується згідно [14] як:

$$NEC = n_\alpha^{\alpha-1} \frac{\left(\sum_i \sqrt[\alpha]{\sum_j \sum_i \mu_{li} \delta_{li}^\alpha} \right)^{\frac{\alpha}{\alpha-1}}}{W} \quad (24)$$

Для верифікації даних підходів було проведено моделювання зазначеної схеми для типових методів роботи з графами завдань [15]:

- обчислення за ієрархічною структурою;
 - алгоритми розбиття графу (Partitioning Algorithms);
 - методи на базі алгебраїчної теорії графів (Linear Algebra Task Graphs);
 - структурування типу «Diamond Dags».
- Отримані результати математичного моделювання були надалі співставленні зі статистичними даними [14], що були визначені для наступних видів розподілу:
- рівномірний розподіл;
 - біноміальний розподіл;
 - геометричний розподіл.

Результати співставлення представлені на рис. 5 — 6. Вони вказують на достатньо високу точність прогнозування на рівні математичного моделювання.

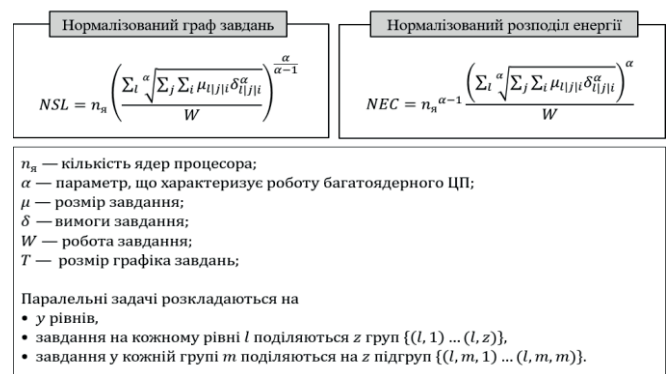


Рис. 5 — Результати математичного моделювання

Наведені графіки залежностей вказують подібність залежностей для різних методів роботи з графами при роботі з одним типом розподілу при різних абсолютних значеннях **NEC** та **NSL**, що доводить ефективність розробленого підходу для широкої групи методів.

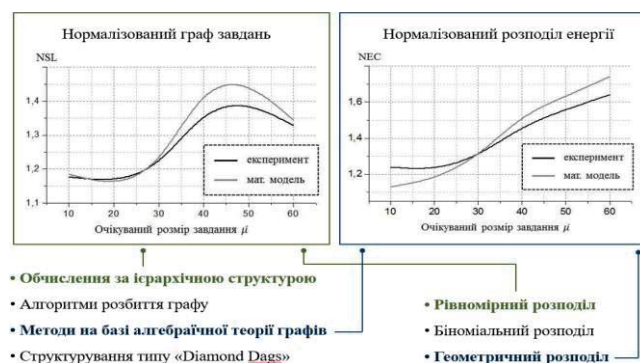


Рис. 6 – Результати графіків залежностей

6. ВИСНОВОК

Розглянуто та вдосконалено методику побудови графу завдань інформаційної системи, що базується на графіку запитів, а також паралелізації планування завдань та багаторівневій схемі розподілу апаратно-програмних ресурсів. Вирішено задачу оптимізації процесу кластеризації апаратних ресурсів шляхом побудови системи віртуальних машин та подальших етапів визначення екстремумів функції розміру графіка завдань та функції повного значення енергії.

7. СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] Gupta, H. Load “Balancing In Cloud Computing,” *International Journal of Recent Trends in Engineering and Research*, Vol. 3(3), pp. 260-267, 2017.
- [2] S. Kaur, T. Sharma, “Efficient load balancing using improved central load balancing technique,” in *Proceedings of the 2nd International Conference on Inventive Systems and Control (ICISC)* 2018.
- [3] S. Pandey, “Cloud Load Balancing: A Perspective Study,” *International Journal Of Engineering And Computer Science*, Vol. 6, Issue 6, pp. 21602-21611, 2017
- [4] R. Podaralla, “Balancing Load in Smartphone’s,” *International Journal Of Engineering And Computer Science*,” Vol. 6, Issue 1, pp. 19987-19983, 2017.
- [5] P. Membrey, D. Hows, E. Plugge, *Load Balancing Basics*, in: Practical Load Balancing, Apress, 2012, pp. 109-116.
- [6] W. N. Ariffin, S. Salleh, “The matching technique of directed cyclic graph for task assignment problem,” in *Proceedings of the International Conference on Quantitative Sciences and Its Application*, At Langkawi, 2014, pp. 387-394.
- [7] Ariffin, W. N., & Salleh, S. “Bi-partition approach of directed cyclic task graph onto multicolumn processors for total completion time minimization task assignment problem,” in *Proceedings of the 2nd International Conference on Mathematics, Engineering and Industrial Applications* 2016, Songkhla; Thailand, 2016.
- [8] C. Yang, C. Lin, “Time-Varying Network Measures in Resting and Task States Using Graph Theoretical Analysis,” *Brain Topography*, Vol. 28, Issue 4, pp. 529-540, 2015.
- [9] Y. Masoudi, S. Lotfi, D. Karimzadgan, F. Fathy, K. Abdi, “Static Task Graph Scheduling in Real Time Homogenous Multiprocessor Systems Using Learning Automata,” in *Proceedings of the 2011 International Conference on Communication Systems and Network Technologies*, Katra, Jammu, 2011, pp. 423-429.
- [10] Yang, C., & Lin, C. (2015). Time-Varying Network Measures in Resting and Task States Using Graph Theoretical Analysis. *Brain Topography*, 28(4), 529-540.
- [11] H. Takabi, J. B. D. Joshi, and G.-J. Ahn, “Security and privacy challenges in cloud computing environments,” *IEEE Security & Privacy*, vol. 8, no. 6, pp. 24–31, 2010.
- [12] K. Li, “Performance analysis of power-aware task scheduling algorithms on multiprocessor computers with dynamic voltage and speed,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 11, pp. 1484–1497, 2008.
- [13] K. Li, “Energy efficient scheduling of parallel tasks on multiprocessor computers,” *Journal of Supercomputing*, vol. 60, no. 2, pp. 223–247, 2012.
- [14] K. Li, “Power allocation and task scheduling on multiprocessor computers with energy and time constraints,” in: Y. Zomaya and Y. C. Lee (Eds.), *Energy-Efficient Distributed Computing Systems*, A., Chapter 1, John Wiley & Sons, 2012, pp. 1-37.
- [15] S. U. Khan, *Handbook on Data Centers*, New York, NY: Springer, 2015, 369 p.
- [16] K. Li, “Algorithms and analysis of energy-efficient scheduling of parallel tasks,” in: I. Ahmad and S. Ranka, (Eds.), *Handbook of Energy-Aware and Green Computing*, Vol. 1 (Chapter 15), CRC Press/Taylor & Francis Group, 2012, pp. 331-360.



Оксіюк Олександр Глібович, д.т.н., професор, завідувач кафедри кібербезпеки та захисту інформації Київського національного університету імені Тараса Шевченка. Основними напрямками наукових досліджень є фундаментальні та прикладні дослідження, пов'язані з розвитком інтелектуальних систем, логіко-лінгвістичного моделювання в кіберпросторі, теорії та практики розробки автоматизованих систем та проведення експертних оцінок, моделей та методів побудови інтегрованої інформаційної безпеки.



Шестак Яніна Володимирівна, асистент кафедри кібербезпеки та захисту інформації Київського національного університету імені Тараса Шевченка. Основними напрямками наукових досліджень є процес оцінювання захищеності від зовнішніх впливів інформаційних систем.



Мирутенко Лариса Вікторівна, к.т.н., асистент кафедри кібербезпеки та захисту інформації Київського національного університету імені Тараса Шевченка. У 2017 році отримала науковий ступінь кандидата технічних наук за спеціальністю "Інформаційні технології". Основними напрямками наукових досліджень є системи технічної захисту інформації, кібербезпека.