

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

**О. В. Русанова,  
О.О.Писарчук**

# **ПЛАНУВАННЯ ОБЧИСЛЕНЬ В ПАРАЛЕЛЬНИХ ТА РОЗПОДІЛЕНИХ КОМП'ЮТЕРНИХ СИСТЕМАХ ЛАБОРАТОРНИЙ ПРАКТИКУМ**

**Навчальний посібник**

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського  
як навчальний посібник для здобувачів ступеня магістра  
за освітньою програмою «Комп'ютерні системи та мережі»  
спеціальності 123 Комп'ютерна інженерія

Електронне мережне навчальне видання

Київ  
КПІ ім. Ігоря Сікорського  
2022

Автори	Русанова Ольга Веніамінівна, доцент, к.т.н., доцент кафедри обчислювальної техніки Писарчук Олексій Олександрович, професор, д.т.н., професор кафедри обчислювальної техніки
Рецензент	Тарасенко-Клятченко О.В., доцент, к.т.н., доцент кафедри системного програмування і спеціалізованих комп'ютерних систем
Відповідальний редактор	Кулаков Ю.О., професор, д.т.н., професор кафедри обчислювальної техніки

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського  
(протокол № 1 від 02.09.2022 р.)  
за поданням Вченої ради факультету інформатики та обчислювальної техніки  
(протокол № 11 від 11.07.2022 р.)*

В учбовому посібнику розглянуті вимоги до виконання лабораторного практикуму, який пов'язаний з дослідженням різних алгоритмів планування обчислень для комп'ютерних та розподілених систем із заданою архітектурою. Отримані результати можуть бути використані для підвищення реальної продуктивності паралельних та розподілених систем за рахунок ефективного планування обчислень. Описані теоретичні основи для виконання робіт. Для кожної роботи викладені мета, вихідні дані, завдання та аналіз отриманих результати. Зазначені правила проведення лабораторних робіт та вимоги до звіту. Викладені вихідні дані, етапи та різні алгоритми планування. Посібник призначений для здобувачів ступеня магістра за спеціальністю 123 Комп'ютерна інженерія, а також буде корисним при виконанні магістерських дисертацій присвячених розробці засобів підвищення продуктивності для сучасних високопродуктивних паралельних комп'ютерних систем.

Реєстр. № НП 21/22-488. Обсяг 1,0 авт. арк.

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
проспект Перемоги, 37, м. Київ, 03056  
<https://kpi.ua>

Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів  
і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

© О. В. Русанова, О.О.Писарчук  
© КПІ ім. Ігоря Сікорського, 2022

## ЗМІСТ

ВСТУП.....	4
ПРАВИЛА ПРОВЕДЕННЯ ЛАБОРАТОРНИХ РОБІТ ТА ВИМОГИ ДО ЗВІТУ ..	5
ЗАГАЛЬНІ ТЕОРЕТИЧНІ ВІДОМОСТІ .....	6
Лабораторна робота № 1.....	13
Лабораторна робота № 2-4 .....	15
Лабораторна робота № 5.....	17
Лабораторна робота № 6-7 .....	18
Лабораторна робота № 8.....	20
СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ .....	22

## ВСТУП

Паралельна обробка інформації створює передумови для суттєвого підвищення продуктивності засобів обчислювальної техніки. Тому останні десятиріччя пов'язані з швидким розвитком паралельних комп'ютерних систем (ПКС). Зараз у світі існує велика кількість архітектурних рішень ПКС і потреба у їх використанні весь час зростає. Сьогодні однією з найскладніших проблем розвитку ПКС є підвищення ефективності їх використання при вирішенні задач користувача. У рамках дисципліни «Планування обчислень в паралельних та розподілених комп'ютерних системах» розглядаються основні напрямки підвищення ефективності ПКС, які пов'язані з досконалим використанням методів та засобів організації паралельних обчислювальних процесів. Теким чином **метою даної дисципліни** є вивчення методів та засобів забезпечення максимальної реальної продуктивності сучасних паралельних комп'ютерних систем з урахуванням специфіки їх архітектур при вирішенні задач користувача.

Лабораторні роботи з курсу «Планування обчислень в паралельних та розподілених комп'ютерних системах» пов'язана з дослідженням різних алгоритмів планування обчислень для комп'ютерної системи із заданою архітектурою. Отримані результати можуть бути використані для підвищення реальної продуктивності паралельних систем за рахунок ефективного планування обчислень.

## **ПРАВИЛА ПРОВЕДЕННЯ ЛАБОРАТОРНИХ РОБІТ ТА ВИМОГИ ДО ЗВІТУ**

1. До виконання лабораторних робіт допускаються тільки ті студенти, які пройшли інструктаж з техніки безпеки і неухильно його виконують.
2. Під час проведення лабораторних занять мобільні телефони повинні бути налаштовані на беззвучний режим або вимкнуті.
3. Лабораторні роботи виконуються студентами індивідуально, згідно з завданням, погодженим з викладачем.
4. Необхідними умовами допуску до поточної лабораторної роботи є захист попередньої роботи та підготовка до майбутньої роботи (знання теоретичних відомостей, порядку виконання).
5. Лабораторні роботи супроводжуються складанням звіту і виконуються в лабораторії кафедри обчислювальної техніки. Звіт повинен включати:
  - Титульний лист, на якому має бути вказана тема лабораторної роботи, назва дисципліни, факультет, група, прізвище, ім'я, по батькові студента;
  - Мета роботи;
  - Опис алгоритму виконання роботи;
  - Опис програмної реалізації;
  - Результати виконання;
  - Висновки;
  - Лістинг програми.

## ЗАГАЛЬНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Лабораторні роботи при вивченні курсу «Планування обчислень в паралельних та розподілених комп'ютерних системах» присвячені реалізації різних алгоритмів статичного планування обчислень для комп'ютерної системи із заданою архітектурою.

Відповідно до теорії розкладу, задача планування (побудови розкладу) вважається заданою, якщо визначені:

1. Належні до виконання обчислювальні роботи і їх операції;
2. Кількість і типи машин, що виконують операції;
3. Порядок проходження машин (конвейерний, випадковий, довільний);
4. Критерій оцінки розкладу.

**Належні до виконання обчислювальні роботи і їх операції.** При плануванні обчислень у паралельних системах належні до виконання роботи і їх операції задаються залежно від типу планування (динамічного або статичного). Більшість методів динамічного планування не орієнтована на аналіз усієї сукупності обчислювальних робіт і їх взаємозв'язків, а враховують лише ті обчислювальні роботи, які готові до виконання у момент планування. Тому в даному випадку задається сукупність готових обчислювальних робіт, їх пріоритети, а також такі параметри, як їх трудомісткість і час їх готовності до виконання. Методи статичного планування орієнтовані на аналіз усієї сукупності обчислювальних робіт і їх взаємозв'язків. Тому обчислювальні роботи в даному випадку задаються у вигляді графів задач. Окрім цього, на графове представлення обчислювальних робіт, як правило, накладаються такі обмеження:

- графи розглядають лише ациклічні (усі циклічні ділянки ускладнюють планування і тому передбачається їх розкриття, наприклад, шляхом векторизації або розпаралелювання);
- як правило, розглядаються інформаційні графи, а не інформаційно-логічні.

Таким чином, обчислювальні роботи (паралельний алгоритм обчислювального завдання) може бути представлений орієнтованим ациклічним графом задач (*directed acycaled graph* - *DAG*)  $Gt(T,Et)$ , який представляється множиною

паралельних гілок (процесів). У графі  $Gt$ , множина вершин  $T=\{t_1, \dots, t_m\}$  і множина дуг  $Et$  є, відповідно, процесами і інформаційними залежностями у вигляді зв'язків між ними. При цьому вага вершини  $t_i$  у графі задач визначається, як  $w_i$  і є обчислювальною складністю  $i$ -го процесу. Інформаційна залежність між  $i$ -им і  $j$ -им процесами задається за допомогою дуги між ними у графі задач із вагою  $e_{ij}$ , і означає, що процес  $t_j$  може почати виконуватися лише тоді, коли завершив своє виконання процес  $t_i$ . Окрім цього, перед виконанням процес  $t_j$  повинен одержати необхідний об'єм даних  $e_{ij}$  від свого попередника (процесу  $t_i$ ). Якщо який-небудь процес має декілька попередників з точки зору інформаційної залежності, то у такому разі застосовується операція логічного "І" для визначення готовності процесу до виконання.

**Кількість і типи машин, що виконують обчислювальні роботи можуть бути задані різним чином. Це залежить від наступних трьох властивостей КС:**

- технічні характеристики ПЕ;
- однорідність ПЕ;
- тип оперативної пам'яті (спільна, розподілена і роздільна).

Більшість КС загального призначення складаються з однорідних процесорних елементів (ПЕ). У цьому випадку параметр тип машини залежить тільки від технічних характеристик ПЕ. Технічними характеристиками ПЕ є:

- продуктивність процесора;
- кількість фізичних каналів для пересилки даних (лінков);
- швидкість передачі даних по лінкам (кількість одиниць інформації в одиницю часу) і можливості пересилки у різних напрямках (дуплексні і напівдуплексні);
- наявність або відсутність у кожному ПЕ процесора введення-виведення, що визначає одночасність або розділення в часі обчислень і пересилок, даних;
- тип комунікаційної моделі.

Для більшості процесорів кількість фізичних каналів даних дорівнює одному. Для трансп'ютерів і деяких спеціалізованих процесорів (наприклад, деяких моделей *ADSP*) число лінков коливається в діапазоні від 4 до 6.

Для неоднорідних КС (тип може включати об'єм оперативної пам'яті, продуктивність, число КВВ і ін.) для кожного ПЕ.

Якщо розглядати КС із загальною (що розділяється) пам'яттю, то вказується лише число ПЕ і їх характеристики, а для неоднородних - число і тип ПЕ.

Як відомо, мультикомп'ютерні MPP системи можуть бути різних типів:

- із статичними (постійними) зв'язками;
- з динамічними зв'язками (реконфігуровані);
- з комбінованими зв'язками (статичними і динамічними).

Топологія MPP системи із статичними зв'язками задається за допомогою неорієнтованого графу системи  $G_p(P, E_p)$ . У графі  $G_p$ , сукупність вершин  $P = \{p_1, \dots, p_n\}$  і сукупність ребер  $E_p$  представляють відповідно процесорні елементи і топологію каналів зв'язку між ними. При цьому вага вершини  $p_i$  в графі системи визначається як  $s_i$  і є продуктивністю  $i$ -го процесора. А вага ребра, що сполучає дві вершини в графі системи, наприклад між  $i$ -ою та  $j$ -ою, визначається як  $u_{ij}$ , і є каналом передачі даних (лінк) між процесорами  $i$  та  $j$ , який має пропускну спроможність, що дорівнює  $u_{ij}$ . Оскільки, як правило, MPP системи відносяться до класу однорідних, значення вагів вершин і вагів ребер у графі системи мають рівні значення.

Реконфігурована КС може бути описана повнозв'язним графом системи.

На рис.1 зображений приклад графів  $G_t$  і  $G_p$  відповідно. Слід зазначити, що швидкість взаємодії між процесами ( $t_i$  і  $t_j$ ) на різних процесорних елементах ( $p_k$  і  $p_l$ ) визначається величиною даних  $e_{ij}$ , що пересилаються і швидкістю передачі даних по каналу  $u_{kl}$ , а саме:  $e_{ij}/u_{kl}$  одиниць часу.

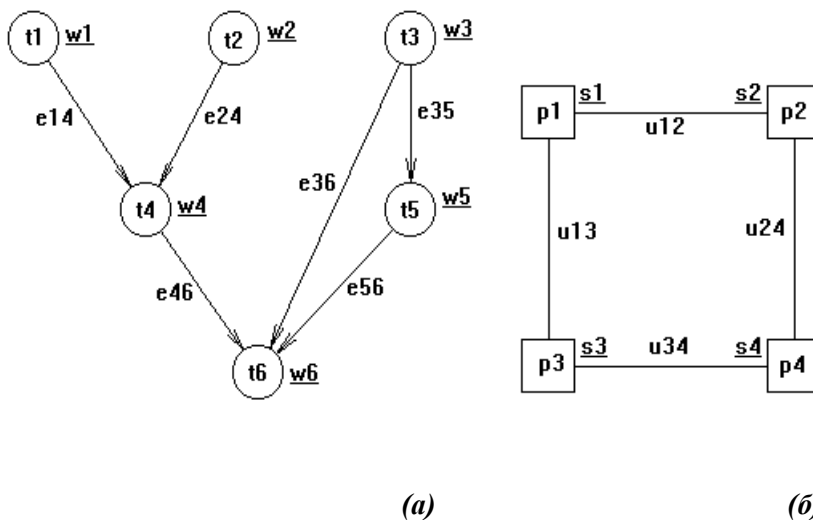


Рис. 1. Приклад графу задачі (а) і графу системи (б)



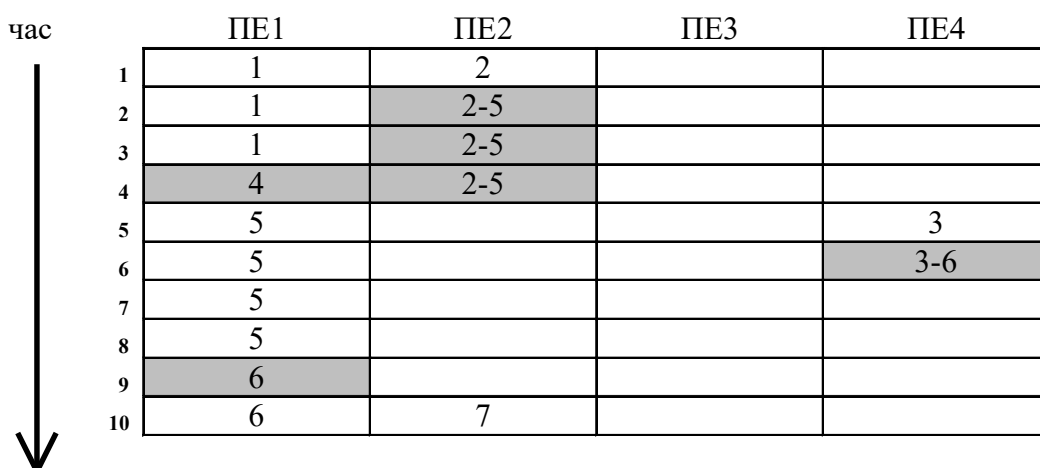
**Порядок проходження машин** може бути конвейєрний, випадковий і довільний. При плануванні найчастіше використовується конвейєрний або довільний порядки проходження. Перший з них використовується для планування робіт в конвейєрних і векторних КС, а другий - для всіх інших. З точки зору теорії розкладу, це дві різні по складності задачі. Конвейєрний порядок проходження передбачає однаковий і визначений порядок для всіх обчислювальних робіт. Конвейєр - це одновходова система. Планування в цьому випадку зводиться до формування черги обчислювальних робіт. Призначення проводиться з цієї черги, і кожна обчислювальна робота виконується в конвейєрі ідентично. Визначається лише порядок призначення на конвейєр. Довільний порядок проходження означає мультипроцесорну обробку. У цьому випадку є багатовходова система, і будь-яка обчислювальна робота може бути (у разі однорідності КС) призначена на будь-який процесор. При цьому планування включає не тільки формування черги обчислювальних робіт, але і алгоритм призначення кожної обчислювальної роботи на процесор. Таким чином, у цьому випадку планування – більш складна процедура. У даних лабораторних роботах при плануванні для *MPP* системи пропонується використовувати довільний порядок проходження машин.

**Критерії оцінки планування.** Існує три основні критерії оптимізації при рішенні задач планування:

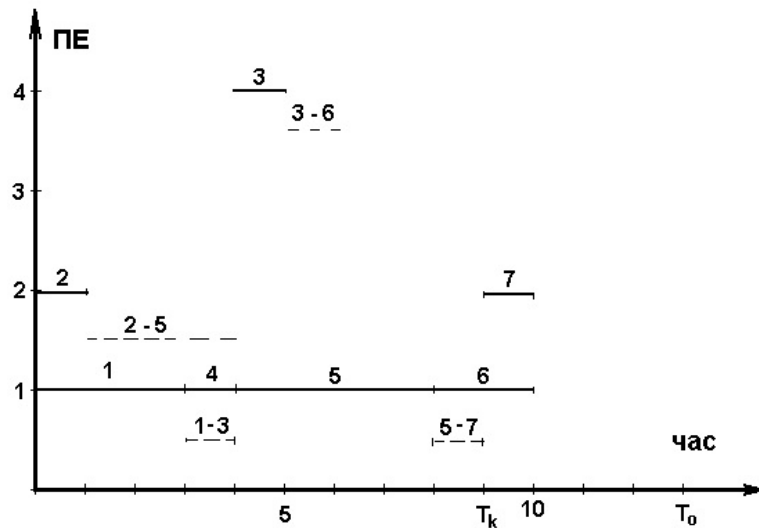
- мінімальний час виконання обчислювальних робіт у КС при заданій вартості (тобто із заданими параметрами, головними з яких є число процесорів і структура взаємозв'язків);
- мінімальна вартість КС, що забезпечує виконання обчислювальних робіт за заданий час. У цьому випадку найбільш поширеним варіантом є ототожнення мінімальної вартості з мінімальним числом процесорів;
- мінімальний час виконання обчислювальних робіт при мінімальній вартості КС. Не завжди при великій кількості ПЕ можна отримати менший час виконання завдання. Цей критерій часто використовується для кластерних систем та систем із масовим паралелізмом.

Перший і третій критерії використовуються для вирішення завдання оптимального використання ресурсів КС (третій має особливу актуальність для систем з масовим паралелізмом, що складаються з величезного числа процесорів), а другий - для оптимального вирішення завдання синтезу КС. У даних лабораторних роботах пропонується використовувати перший критерій оптимізації.

Результат планування може бути представлений за допомогою діаграми Ганта. Діаграма Ганта є розкладом роботи всіх процесорів системи і для кожного з них потактовий порядок виконання всіх завдань, призначених на даний процесор. Проте традиційна діаграма Ганта містить не всю інформацію про дані, що пересилаються, з одного процесора на іншій. Пропонується в лабораторних роботах використовувати модифіковану діаграму Ганта, в якій розклад роботи кожного процесора включає не тільки чергу виконуваних завдань, але і потактовий стан усіх його каналів, при пересилці даних. Приклади діаграм Ганта представлені на рис.2.



(a)



(б)

Рис.2. Приклади результатів планування: традиційна діаграма Ганта (а) і модифікована діаграма Ганта (б)

Згідно вибраному критерію оптимізації алгоритм планування повинен забезпечувати мінімальний час виконання вказаного обчислювального завдання на *MPP* системі заданої топології.

Критичний час  $T_{критич}$  - мінімальний час виконання завдання, яке не враховує витрат при пересилках даних і таким чином є ідеальним, але в більшості випадків недосяжним через те, що в реальній ситуації не вдається уникнути пересилок даних. Виключеннями можуть бути ті випадки, коли вдається частково уникнути пересилок за допомогою призначення зв'язаних процесів, що знаходяться на критичному шляху, на один і той же процесор, а решту пересилок виконувати без втрати часу, зарезервованого за рахунок паралельного виконання інших процесів. З іншого боку, максимальний час виконання алгоритму в будь-якій системі - це час її виконання на одному процесорі  $T_{послед}$ . У цьому випадку пересилки даних відсутні і цей час визначається як сума обчислювальних трудомісток (ваг) кожного процесу обчислювального завдання. Тому, величина реального мінімального часу виконання алгоритму  $T_{реальн}$  завжди знаходиться в наступному діапазоні:  $T_{критич} < T_{реальн} < T_{послед}$ .

Процес планування для *MPP* систем на основі спискового підходу складається з наступних двох етапів:

- 1 Формування черги готових обчислювальних робіт;
- 2 Призначення обчислювальних робіт на процесори.

У різних алгоритмах планування реалізація цих етапів відрізняється один від одного. Окрім цього, існує два основні підходи виконання цих етапів: послідовний і комплексний. Перший підхід простіше в реалізації, але приводить до менш ефективного результату, а другий – більш складний, але може забезпечити результати ближчі до оптимальних.

## Лабораторна робота № 1

**Мета роботи** – навчитися створювати інтерфейс програмної моделі для дослідження алгоритмів планування обчислень.

**Вхідні дані** – тип КС ( SMP, MPP, кластерна, неоднорідна розподілена, Grid).  
Варіант КС вибирається по узгодженню з викладачем.

**Завдання роботи** – створити інтерфейс програмної моделі для дослідження алгоритмів планування обчислень. У даній роботі необхідно створити меню програми, а також розробити редактор графа задачі і системи (крім SMP) з необхідними перевітками на будь-якій мові програмування. Спосіб зображення отриманих графів визначається студентом.

Головне меню повинно включати:

- граф задачі;
- граф КС (а для SMP - кількість процесорів та число банків оперативної пам'яті);
- моделювання;
- статистика;
- допомога;
- вихід.

Граф задачі:

- новий;
- файл (завантаження із файлу);
- запис (у файл).

Новий граф задачі:

- редактор графу;
- генерування графу.

Граф КС:

- нова;
- файл (завантаження із файлу);
- запис (у файл).

Моделювання:

- параметри процесорів;

- діаграма Ганта;
- статистика.

Параметри процесорів включають:

- число фізичних лінків
- наявність процесора вводу-виводу;
- дуплексність зв'язків;
- спосіб пересилки даних: пересилка повідомлень; конвейеризація пересилок пакетами (в цьому випадку необхідно задавати довжину пакету).

Статистика включає:

- параметри статистики;
- результати статистики.

## Лабораторні роботи № 2-4

**Мета роботи** – реалізувати перший етап планування шляхом формування черг обчислювальних робіт.

**Вхідні дані** – коректний граф задачі (результат виконання лабораторної роботи №1), три задані алгоритми формування черг із таблиці 1.

Таблиця 1

Номер	Складність	Метод формування черг	Використанні характеристики графа
1	4	У порядку спадання проноормованої суми критичних по часу і по кількості вершин шляхів до кінця графа задачі.	Критичний шлях графа та вершин по часу.
2	4	У порядку зростання різниці між пізнім та раннім строками виконання вершин графу задачі	Критичний шлях по часу і кількості вершин.
3	3	У порядку спадання критичного по часу шляхів до кінця графа задачі.	Критичний шлях по часу для всіх вершин
4	4	У порядку спадання критичного по кількості вершин шляхів до кінця графа задачі, а при рівних значеннях – в порядку спадання зв'язності вершин.	Критичний шлях по кількості вершин та їх зв'язність
5	4	У першу чергу вершини, що знаходяться на критичному шляху графа (по кількості вершин), а інші – в порядку спадання критичного по числу вершин шляхів до	Критичний шлях графа, критичні шляхи вершин

		кінця графа задач.	
6	3	У порядку спадання критичних по кількості вершин шляхів до кінця графа задачі.	Критичний шлях по кількості вершин.
7	4	В порядку зростання критичного по кількості вершин шляхів від початку графа, а при рівних значеннях у порядку спадання зв'язності.	Критичний шлях по кількості вершин та їх зв'язність
8	4	У порядку зростання критичного шляху по кількості вершин від початку графа, а при рівних значеннях – в порядку спадання ваги вершин	Критичний шлях по кількості вершин та їх вага
9	3	У порядку зростання по кількості вершин шляхів від початку графа задач.	Критичний шлях по кількості вершин
10	4	У порядку спадання зв'язності, а при рівних значеннях – в порядку спадання критичного по кількості вершин шляхів до кінця графа.	Критичний шлях по кількості вершин та їх зв'язність
11	4	У порядку спадання зв'язності вершин, а при рівних значеннях – в порядку зростання критичного по кількості вершин шляхів від початку графа задачі.	Критичний шлях по кількості вершин та їх зв'язність
12	2	У порядку спадання кількості вихідних дуг вершин	Кількість вихідних дуг вершин
13	1	Випадковим чином.	
14	2	У порядку спадання ваги вершин	Вага вершин



15	2	У порядку зростання ваги вершин	Вага вершин
16	3	У порядку зростання критичного по часу шляхів вершин від початку графа задачі.	Критичні по часу шляхи вершин від початку графа задачі.
17	2	У порядку зростання часу готовності.	

**Завдання роботи** – для побудованого графа задачі реалізуються три заданих варіанти формування черг обчислювальних робіт.

### Лабораторна робота № 5

**Мета роботи** – навчитись генерувати випадкові графи задачі із заданими параметрами.

**Вхідні дані** – для генерування графу задачі задаються наступні його параметри:

- мінімальна вага вершини графу;
- максимальна вага вершини;
- кількість вершин графу;
- зв'язність графу (співвідношення часу виконання до часу пересилок);
- мінімальна вага дуг графа задачі (необов'язково);
- максимальна вага дуг графа задачі (необов'язково).

Зв'язність графу визначається за допомогою формули:

$$correlation = \frac{\sum_{i=1}^n W_i}{\sum_{i=1}^n W_i + \sum_{i=1}^n L_i}$$

де  $\sum_{i=1}^n W_i$  - сумарна вага вершин графу задачі;  $\sum_{i=1}^n L_i$  - сумарна вага дуг графу задачі;  $n$  - кількість вершин у графі задачі.

**Завдання роботи** – виконати програмну генерацію випадкового графа задачі з наведеними вище вхідними даними. Результати цієї роботи потрібні для більш об'єктивного аналізу алгоритмів планування. Відомо, що для виконання порівняльного аналізу алгоритмів планування, може бути використаний тільки статистичний підхід, оскільки задачі планування відносяться до класу NP-повних. Для цього необхідно виконати порівняння на великій множині графів задач із різними параметрами.

### **Лабораторні роботи № 6-7**

**Мета роботи** – реалізувати другий етап планування шляхом призначення обчислювальних робіт по процесорах заданої КС.

**Вхідні дані** – граф задачі та побудована для нього черга обчислювальних робіт; граф КС (а для SMP - кількість процесорів та число банків оперативної пам'яті); два алгоритми призначення; параметри процесорів КС, які включають:

- число фізичних лінків;
- наявність процесорів вводу-виводу;
- дуплексність або напівдуплексність зв'язків;
- спосіб пересилки даних: пересилка повідомлень; конвейеризація пересилок пакетами (в цьому випадку необхідно задавати довжину пакету).

Алгоритми призначення обчислювальних робіт на процесори вибираються за погодженням з викладачем.

**Для SMP систем розглядаються наступні варіанти алгоритмів призначення:**

1. Алгоритм випадкового призначення (на будь-який вільний процесор).
2. Призначення на перший процесор, що звільнився. У цьому випадку, при призначенні, з вільних процесорів вибирається той, який звільнився і простоює довше за інших. Такий підхід дозволяє збалансувати завантаження всіх процесорів системи.

3. «Сусіднє» призначення для систем без автономних контролерів вводу-виводу з застосуванням варіанту комунікаційної моделі, коли пересилання даних для обчислювальної роботи починає здійснюватися тільки після формування всіх її вихідних даних (пересилання без «попередження»).
4. «Сусіднє» призначення для систем з автономними контролерами вводу-виводу із застосуванням варіанту комунікаційної моделі, коли пересилання даних для обчислювальної роботи починає здійснюватися тільки після формування всіх її вихідних даних.
5. «Сусіднє» призначення для систем з автономними контролерами вводу-виводу із застосуванням варіанту комунікаційної моделі, коли пересилка даних для обчислювальної роботи здійснюється по мірі готовності її вихідних даних асинхронно «з попередженням».
6. Алгоритм «сусіднього» призначення з урахуванням зайнятих процесорів. Даний алгоритм відрізняється від попереднього тим, що при призначенні обирається той процесор, який забезпечує мінімальний стартовий час обчислювальної роботи незалежно від того, зайнятий він чи вільний у момент призначення.

**Для MPP систем розглядаються наступні алгоритми призначення:**

1. Алгоритм випадкового призначення (на будь-який вільний процесор).
2. Призначення на перший процесор, що звільнився. У цьому випадку, при призначенні, з вільних процесорів вибирається той, який звільнився і простоює довше за інших. Такий підхід дозволяє збалансувати завантаження всіх процесорів системи.
3. Алгоритм призначення з урахуванням пріоритетів процесорів (по зв'язності). Даний алгоритм може використовуватись як для всіх обчислювальних робіт графа задачі, так і для призначення обчислювальних робіт першого ярусу графа задачі у поєднанні з алгоритмами «сусіднього» призначення
4. Алгоритми «сусіднього» призначення без пересилань «з попередженням». У даному випадку використовується комунікаційна модель, коли тільки після формування всіх даних починається пересилка для будь-якої обчислювальної роботи.

5. Алгоритм «сусіднього» призначення із пересилками «з попередженням». У даному випадку використовується, як і в аналогічному алгоритмі для SMP систем, комунікаційна модель, коли дані передаються асинхронно відразу після їх формування.
6. Алгоритм «сусіднього» призначення з використанням моделювання для визначення початкового часу виконання обчислювальних робіт.
7. Алгоритм оптимізованого «сусіднього» призначення, у якому враховуються всі процесори незалежно від їх стану в момент призначення (зайнятий або вільний).

**Завдання роботи** – реалізувати два заданих алгоритми призначення обчислювальних робіт графу задачі по процесорах КС. Навести результати планування у вигляді модифікованої діаграми Ганта, в якій відображається в часі зайнятість, як процесорів системи, так і каналів при пересиланні необхідних даних.

## Лабораторна робота № 8

**Мета роботи** – вибір оптимального способу планування обчислень на заданій ПКС з урахуванням її параметрів.

**Вхідні дані** – результати шести алгоритмів планування, отриманих у результаті виконання лабораторних робіт №2-7.

**Завдання роботи** - порівняти шість реалізованих алгоритмів планування з точки зору їх ефективності. Порівняння необхідно виконувати на множині згенерованих випадкових графів задач. Для аналізу роботи алгоритмів використовувати наступні показники ефективності:

1. **Коефіцієнт прискорення**  $K_{np} = \frac{T_1}{T_n}$ , де  $T_1$  – час виконання задачі на одному процесорі,  $T_n$  - час виконання задачі на заданій КС.

2. **Коефіцієнт ефективності системи**  $K_e = \frac{K_{np}}{N}$ ,

де  $K_{np}$ - коефіцієнт прискорення,  $N$  – кількість процесорів КС.

3. **Коефіцієнт ефективності алгоритму планування**  $K_{e.алг.} = \frac{T_{кр}}{T_n}$ ,

де  $T_{кр}$  – критичний час (теоретичний мінімум) виконання задачі на заданій КС.

Для виконання даної роботи необхідно задати наступні параметри в меню «Статистика»:

- мінімальну кількість вершин в графі задач;
- максимальну кількість вершин в графі задач;
- крок нарощування кількості вершин;
- початкова зв'язність згенерованої задачі;
- кінцева зв'язність згенерованої задачі;
- крок нарощування зв'язності;
- мінімальна вага вершин у графі задачі;
- максимальна вага вершин у графі задачі;
- кількість графів задач для статистики.

## СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Програмне забезпечення комп'ютерних систем. Частина 2. Навчальний посібник для здобувачів ступеня магістра за спеціальністю 123 «Комп'ютерні системи та мережі» / О. Русанова., О.Корочкін – К.: КПІ ім. Ігоря Сікорського, 2022. – 112 с. Електронний ресурс. Гриф надано Метод. радою КПІ ім. Ігоря Сікорського (протокол № 5 від 26.05.2022 р.) за поданням Вченої ради ФІОТ (протокол № 8 від 18.04.2022 р.) <http://ela.kpi.ua/handle/123456789/48215>
2. Паралельні та розподілені обчислення. Вибрані розділи: Навч. посібник для здобувачів ступеня бакалавр за спеціальністю 123 «Комп'ютерні системи та мережі» / Корочкін О.В., Русанова О.В. – Київ : КПІ ім. Ігоря Сікорського, 2020. – 123с Електронний ресурс. Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 6 від 31.01.2020 р.) за поданням Вченої ради

ФІОТ (протокол № 4 від 25.11.2019 р.) <http://ela.kpi.ua/handle/123456789/48224>

3. Програмне забезпечення комп'ютерних систем. Програмування та компіляція /Русанова О.В., Корочкін О.В. – Київ : КПІ ім. Ігоря Сікорського, 2020. – 94 с. Електронний ресурс .Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 6 від 31.01.2020 р.) за поданням Вченої ради ФІОТ (протокол № 4 від 25.11.2019 р.) <https://ela.kpi.ua/handle/123456789/48296>
4. Rusanova O., Korochkin A. Scheduling Problems for Parallel and Distributed Systems. – Proceedings of SIGAda'99 Conference; American Computer Machinery, USA, 1999. ( Engl).
5. Valerii Demchyk, Vitalii Tyzun, Alexander Korochkin, Olga Rusanova. THE APPLICATION OF WCF TECHNOLOGY TO INCREASE THE EFFICIENCY OF PARALLEL COMPUTING IN CLOUD DISTRIBUTED COMPUTER SYSTEMS // Security, Fault Tolerance, Intelligence: proceedings of the International Conference ICSFTI2020, Kyiv, Ukraine, May 13, June 15, 2020. – Kyiv : Igor Sikorsky Kyiv Polytechnic Institute, Publishing House “Polytechnica”, 2020. – P.209-215
6. Olga Rusanova, Igor Boyarshin, Anna Doroshenko. ENERGY-AWARE TASK SCHEDULING ALGORITHM FOR MOBILE COMPUTING // Security, Fault Tolerance, Intelligence: proceedings of the International Conference ICSFTI2020, Kyiv, Ukraine, May 13, June 15, 2020. – Kyiv : Igor Sikorsky Kyiv Polytechnic Institute, Publishing House “Polytechnica”, 2020. – P.107-113
7. Hesham El-Rewini, Ted G.Lewis. Distributed and Parallel Computing.-Manning Publications Co., 1997.-447 p.
8. Technical Report No. 2006-504 Scheduling Algorithms for Grid Computing: State of the Art and Open Problems. Fangpeng Dong and Selim G. Akl. School of Computing, Queen's University Kingston, Ontario January 2006.
9. Fangpeng Dong. Scheduling algorithms for distributed and parallel computing:state of the art and open problems // Technical report – School of computing, Queen's university. Kingston, Ontario, 2014. – С. 12 – 19.

10. Chih-Hsueh Yang. Improving Static Task Scheduling in Heterogeneous and Homogeneous Computing Systems // Institute of Information Science, Academia Sinica Department of Computer Science, National Tsing Hua University, 2009.
11. G. Lutsky, O. Rusanova Scheduling Problems on the Parallel and Distributed Systems - an Overview, "Computer Systems and Networks: designing, application, utilization".- Poland, Rzeszow, 2000, tom 1, pp.101-105(Engl).
12. Kamaljit Kaur. Heuristics Based Genetic Algorithm for Scheduling Static Tasks in Homogeneous Parallel System // Department of Computer Science & Engineering, Guru Nanak Dev University, Amritsar- 143001, Punjab, India, 2010. – C. 212.
13. Xiaozhong Geng. A Task Scheduling Algorithm for Multi-Core Cluster Systems // Department of Computer Science & Technology Jilin University, Changchun, China. School of Electrical & Information Technology Changchun Institute of Technology, Changchun, China, 2012 – C. 127.
14. Yanyan Dai. A Synthesized Heuristic Task Scheduling Algorithm // Institute of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China, 2014. – C. 67