

НАЦІОНАЛЬНА АКАДЕМІЯ НАУК УКРАЇНИ
ІНСТИТУТ ПРОБЛЕМ МОДЕЛЮВАННЯ В ЕНЕРГЕТИЦІ
ІМ. Г.Є. ПУХОВА

Узденов Тарас Амурович



УДК 004.75 : 519.876

**МЕТОДИ ДИСПЕТЧЕРИЗАЦІЇ ЗАВДАНЬ ДЛЯ GRID-СИСТЕМ З
НЕВІДЧУЖУВАНИМИ РЕСУРСАМИ**

05.13.05 – комп'ютерні системи та компоненти

Автореферат
дисертації на здобуття наукового ступеня
кандидата технічних наук

Київ – 2021

Дисертацією є рукопис.

Робота виконана в Державному університеті «Житомирська політехніка»
МОН України.

Науковий керівник кандидат технічних наук, доцент
Данильченко Олександр Михайлович,
Державний університет
«Житомирська політехніка».

Офіційні опоненти: доктор технічних наук,
старший науковий співробітник
Гільгурт Сергій Якович,
Інститут проблем моделювання в енергетиці
ім. Г.Є. Пухова НАН України,
старший науковий співробітник відділу
математичного і економетричного моделювання;

кандидат технічних наук,
старший дослідник
Запорожець Артур Олександрович,
Інститут технічної теплофізики НАН України,
старший науковий співробітник відділу
моніторингу та оптимізації теплофізичних процесів.

Захист відбудеться «30» вересня 2021 р. о 14 годині на засіданні спеціалізованої
вченої ради Д 26.185.01 Інституту проблем моделювання в енергетиці
ім. Г.Є. Пухова НАН України за адресою: 03164, м. Київ, вул. Генерала Наумова, 15.

З дисертацією можна ознайомитись у бібліотеці Інституту проблем моделювання в
енергетиці ім. Г.Є. Пухова НАН України за адресою: 03164, м. Київ, вул. Генерала
Наумова, 15.

Автореферат розісланий «28» серпня 2021 р.

Вчений секретар
спеціалізованої вченої ради



В. В. Душеба

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність теми. GRID – географічно розподілені обчислювальні ресурси, які об'єднані в одну обчислювальну систему. Такі системи створюються для вирішення завдань, що не можуть бути вирішеними за розумний проміжок часу на одному чи навіть декількох ПК, об'єднаних в кластер. Системи GRID можуть включати в себе будь-які обчислювальні ресурси. Такі системи потребують значних фінансових та людських витрат, оскільки вартість високопродуктивного обладнання досить висока, крім того воно потребує окремих приміщень, а також залучення кваліфікованого персоналу. При цьому будь-яка сучасна організація має багато персональних комп'ютерів, на яких працює її персонал. Використання таких ПК не є максимально ефективним, так як більшість задач, що виконуються на них, не займають і 10 – 20% від максимальної продуктивності ПК. Тому є доцільність створювати на базі таких ПК обчислювальних систем, які б дозволяли виконувати інші завдання паралельно з поточними для кожного з ПК. Такі системи називаються GRID-системи з невідчужуваними ресурсами. Ще такі системи називають настільними (Desktop GRID). Перший великий волонтерський обчислювальний проект SETI@home був запущений в 1999 році, створивши основу для розвитку BOINC (Відкрита інфраструктура Берклі для мережевих обчислень).

Однією з основних задач, що виникають при створенні GRID-системи з невідчужуваними ресурсами, власне як і для GRID-систем, є задача диспетчеризації завдань. Отже, в GRID-системах повинен бути реалізований механізм планування. Він необхідний для розподілу завдань на виконання між вузлами системи, з метою мінімізації часу виконання роботи та балансування навантаження системи.

Диспетчеризація завдань в GRID-системах є достатньо складною задачею і на сьогоднішній час не існує чіткого і однозначного її рішення. Підходи, викладені в роботах Ankita Sahana S., Carastan-Santos D., De Camargo R.Y., Trystram D., Zrigui S., Dheenadayalan K., Muralidhara V.N., Srinivasaraghavan G., Haruna A.A., Jung L.T., Zakaria N., Hlaing Y.T., Yee T.T., Kaur M., Khan Z.F., Pujiyanta A., Nugroho L.E., Kumar P.P., Cordeiro D., Каляева А.І., Чернова І.А., Минухина С.В., Коровина А.В. та інших вчених є або занадто складними в реалізації (наприклад, потрібно заздалегідь знати, скільки вузлів буде в системі на протязі всього часу обчислень) або не враховують деяких важливих факторів (наприклад, того, що такі системи є досить гнучкими і постійно змінюються характеристики вузлів, може змінюватися потужність вузла, або швидкість каналу зв'язку з ним), що практично унеможливує їх застосування для GRID-систем з невідчужуваними ресурсами.

Тому розроблення та дослідження методів для вирішення задачі диспетчеризації завдань в GRID-системах з невідчужуваними ресурсами є актуальним науковим завданням, що потребує вирішення.

Зв'язок роботи з науковими програмами, планами, темами. У 2012 році колективом кафедри програмного забезпечення та обчислювальної техніки Житомирського державного технологічного університету успішно завершено виконання прикладного дослідження за державним замовленням “Розвиток методів комбінаторної оптимізації в задачах побудови замкнених маршрутів на графах та

мережах” (номер державної реєстрації: №111U001777). В якій здобувач приймав участь як один з виконавців.

Мета і задачі дослідження. Метою дисертаційної роботи є підвищення ефективності використання GRID-систем з невідчужуваними ресурсами шляхом розроблення та дослідження методів диспетчеризації завдань, що враховують різноманітність ресурсів.

Для досягнення мети були поставлені та вирішені наступні основні задачі:

- проведення порівняльного аналізу сучасних підходів вирішення задачі диспетчеризації завдань в GRID-системах з невідчужуваними ресурсами та існуючих програмних засобів для дослідження таких систем;
- розроблення нового підходу до вирішення задачі диспетчеризації завдань в GRID-системах з невідчужуваними ресурсами на основі закону балансу сил;
- розроблення методу диспетчеризації завдань, що легко можуть бути розпаралелені, в GRID-системах з невідчужуваними ресурсами;
- розроблення методу диспетчеризації завдань, що не розпаралелюються в GRID-системах з невідчужуваними ресурсами. Дослідження можливостей його модифікації за рахунок комбінації з іншими методами;
- розроблення програмного комплексу для дослідження ефективності методів диспетчеризації завдань. Проведення обчислювальних експериментів.

Об’єктом дослідження є процес диспетчеризації завдань для GRID-систем з невідчужуваними ресурсами.

Предметом дослідження є методи диспетчеризації завдань для GRID-систем з невідчужуваними ресурсами.

Методи дослідження. Методи теорії масового обслуговування, пріоритетного планування, балансування навантаження в розподілених комп’ютерних системах та мережах, об’єктно-орієнтованої методології розробки і моделювання складних систем.

Наукова новизна одержаних результатів полягає в наступному:

1) вперше запропоновано підхід до вирішення задачі диспетчеризації завдань в GRID-системах з невідчужуваними ресурсами, в якому, на відміну від існуючих, пропонується здійснювати розподіл завдань відповідно до закону балансу сил, що забезпечує розроблення простих та ефективних методів диспетчеризації;

2) розроблено метод диспетчеризації в GRID-системах з невідчужуваними ресурсами завдань, що легко розпаралелюються (FSA_P), який, на відміну від існуючих, враховує різну обчислювальну потужність вузлів, що забезпечує близьку до максимально можливої продуктивності системи;

3) розроблено метод диспетчеризації в GRID-системах з невідчужуваними ресурсами завдань, які не розпаралелюються (FSA), який, на відміну від існуючих, враховує різну обчислювальну продуктивність вузлів та різні властивості завдань, що забезпечує зменшення часу виконання черги завдань;

4) запропоновано дві модифікації методу FSA – FSA Min та FSA Max, в яких для початкового розподілу завдань використовуються ідеї методів Min-Min та Max-Min відповідно, що забезпечує зменшення часу виконання черги завдань в порівнянні з методом FSA при початковому стані системи, коли кількість завдань дорівнює або менше кількості вузлів.

Практичне значення отриманих результатів. Запропонований в роботі підхід та розроблені на його основі методи можуть бути використані для вирішення задачі диспетчеризації завдань в GRID-системах з невідчужуваними ресурсами.

Створений програмний комплекс SGridAR-1 можливо використовувати для дослідження та аналізу поведінки методів диспетчеризації при зміні різних умов: кількості та об'єму завдань, кількості та потужності вузлів. За допомогою SGridAR-1 можна проводити попередні дослідження щодо ефективності різних методів диспетчеризації для заданих архітектур комп'ютерних систем та завдань, що мають на них виконуватись.

Крім того, даний програмний комплекс має зручний та зрозумілий інтерфейс, а клієнти (вузли) реалізовані окремими програмами, всі обчислення проходять в інтерактивному режимі. Сукупність цих факторів дозволяє використовувати даний програмний продукт в освітніх цілях.

Основні результати дисертаційної роботи впроваджені в Державному університеті «Житомирська політехніка».

Особистий внесок у роботи, виконані у співавторстві. Всі наукові результати, що подані у дисертації, одержані автором особисто. В роботах, опублікованих в співавторстві, особистий науковий внесок здобувача складає: [3] – огляд підходів для вирішення задачі масового обслуговування, при умові задіяння в системі декількох агентів; [4] – аналіз та порівняння моделей оцінки трудомісткості операцій; [5] – розробка архітектурної моделі програмного комплексу для досліджень алгоритмів диспетчеризації, побудова схеми взаємодії компонентів системи; [7] – розробка архітектури програмного комплексу для дослідження паралельних алгоритмів на кластерній системі; [8] – аналіз проблем розробки методів керування паралельними завданнями та їх алгоритмічної підтримки для актуальної форми GRID; [9] – розробка моделі процесу диспетчеризації завдань для GRID-систем.

Апробація роботи. Основні положення дисертаційної роботи доповідалися та обговорювалися на наступних наукових конференціях і семінарах: XXXIX науково-технічна конференція молодих вчених та спеціалістів Інституту проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України (м. Київ, 12 травня 2021 р.); «Workshop on the Quantum Information Technologies + Edge Computing Workshop» (м. Житомир, 1 квітня 2021 р.); міжнародна спеціалізована наукова конференція «Актуальні питання механічної та електричної інженерії, транспортних технологій, електроніки, автоматизації та ІТ» (м. Хмельницький, 5 березня 2021 р.); VIII Всеукраїнська науково-практична конференція молодих вчених «Наукова молодь-2020» (м. Київ, 2020 р.); III Всеукраїнська науково-практична інтернет-конференція здобувачів вищої освіти і молодих учених «Інформаційно-комп'ютерні технології: стан, досягнення та перспективи» (м. Житомир, 2020 р.); IV Міжнародна наукова конференція «МОДЕЛИРОВАНИЕ-2012» (м. Київ, 2012 р.); XXXVI науково-практична міжвузівська конференція, присвячена Дню науки (м. Житомир, 2011 р.); науково-практична міжвузівська конференція (м. Житомир, 2010 р.).

Публікації. Основний зміст роботи викладено в 12 наукових працях, серед яких – 2 статті в наукових періодичних виданнях, що індексуються Scopus, 4 статті у

наукових фахових виданнях, що рекомендовані МОН України, та 6 тез доповідей у збірниках матеріалів та тез конференцій.

Структура й обсяг роботи. Дисертація містить анотацію, вступ, чотири розділи, висновки, список використаних джерел із 148 найменувань та 5 додатків. Загальний обсяг дисертації становить 158 сторінок, з яких 115 сторінок основного тексту. Робота містить 34 малюнки, 6 таблиць.

ОСНОВНИЙ ЗМІСТ РОБОТИ

У вступі наведено актуальність теми, зв'язок роботи з науковими програмами і темами, сформульовано мету і задачі дослідження, описані методи дослідження, показано наукову новизну і практичне значення отриманих результатів, описано особистий внесок здобувача в праці, що виконані у співавторстві, відображено відомості про апробацію результатів і їхнє впровадження, наведено список публікацій за темою роботи.

У першому розділі проведено огляд сучасного стану розвитку GRID-систем з невідчужуваними ресурсами. Наведено поняття таких систем та їх основні характеристики. Виділені проблеми та задачі, що виникають при побудові таких систем, однією з яких є задача диспетчеризації завдань. Проаналізовано низку простих методів, що використовуються для її вирішення в таких системах. Сформульовано поняття невідчужуваності ресурсів та встановлено, що GRID-системи з невідчужуваними ресурсами можуть мати ще більш поширену назву – Desktop GRID. Проведений огляд Desktop GRID, як одного з видів GRID-систем, наведено переваги, недоліки та види таких систем. На прикладі загальновідомого проекту BOINC описана архітектура Desktop Grid. Проведено аналіз існуючих проєктів та програмних засобів для побудови GRID-систем з невідчужуваними ресурсами / Desktop GRID на предмет використання в них методів диспетчеризації, який показав, що в основному в таких системах використовується метод FCFS (*First Come First Served*).

На підставі проведених аналізів зроблено висновок, що нові методи потрібні, але одними з ключових характеристик, якими вони повинні володіти, це простота та краща продуктивність в порівнянні з FCFS. Тому розроблення методів диспетчеризації завдань для GRID-систем з невідчужуваними ресурсами є актуальним науковим завданням, що потребує вирішення.

У другому розділі сформовано постановку задачі диспетчеризації завдань в GRID-системах з невідчужуваними ресурсами. Припустимо, що є множина різних за своєю потужністю та програмно-апаратною архітектурою обчислювальних вузлів – персональних комп'ютерів (ПК). Дані ПК з'єднані між собою різними каналами зв'язку з різною швидкістю передачі даних. З іншого боку є множина завдань, які потрібно виконати. При цьому завдання можуть бути різними за своїми властивостями і потребувати різного часу виконання та різної обчислювальної потужності ПК. Між ними знаходиться регулювальник, або диспетчер, який вирішує яку задачу на який вузол потрібно поставити. Мета даного диспетчера – розподілити завдання таким чином, щоб сумарний час виконання всіх завдань був би

мінімальним. Потрібно визначити метод, за яким він буде розподіляти ці завдання. На рис. 1 схематично зображено описану вище задачу диспетчеризації.

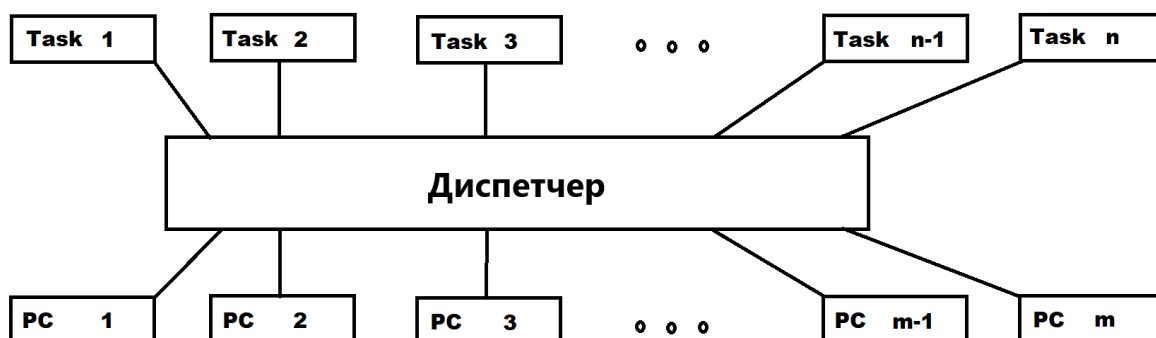


Рисунок 1 – Схематичне зображення постановки задачі диспетчеризації

В даному дослідженні автором запропоновано новий підхід до вирішення задачі диспетчеризації завдань у GRID-системах з невідчужуваними ресурсами та Desktop GRID, відповідно до якого розподіляти завдання пропонується за законом балансу сил.

На рис. 2. схематично показано основну суть запропонованого підходу.

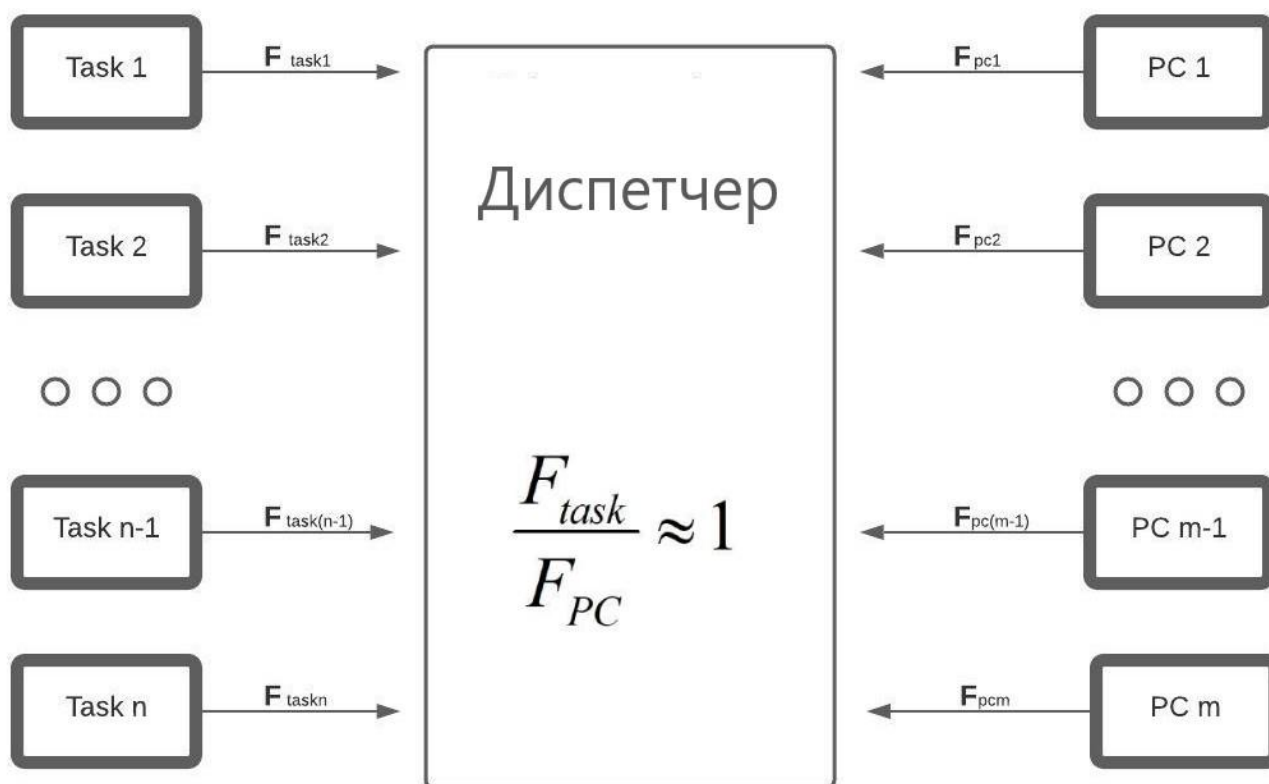


Рисунок 2 – Схематичне зображення нового підходу

Ідея полягає в тому, щоб подати завдання, які потребують вирішення, як певну фізичну силу, а обчислювальні вузли, на яких вони повинні виконуватися, як іншу силу. Дані сили протидіють одна одній. Роль диспетчера має виконувати програма,

що працює за певним алгоритмом, і розподіляє завдання таким чином, щоб підтримувати баланс сил.

Нижче наведено обґрунтування, згідно з яким можна перейти від рівня понять "сили завдання" та "сили вузла" на більш нижчий рівень абстракції та замінити їх на поняття відповідно "потужність завдання" та "потужність вузла".

З елементарної фізики відомо, що потужність P дорівнює відношенню роботи до часу:

$$P = \frac{A}{t}, \quad (1)$$

де A – робота;

t – час.

Робота A , в свою чергу, дорівнює силі помноженій на шлях, який потрібно пройти:

$$A = F \times s, \quad (2)$$

де F – сила, яку треба прикласти;

s – шлях, що потрібно пройти.

Враховуючи формулу (2) можемо привести формулу (1) до наступного вигляду:

$$P = \frac{F \times s}{t}.$$

Зробивши деякі перетворення, отримаємо формулу для визначення сили:

$$F = \frac{P \times t}{s}.$$

Маючи формулу для визначення сили в загальному вигляді, можемо конкретизувати її для обчислення сили обчислювального вузла (позначимо її F_{PC}):

$$F_{PC} = \frac{P_{PC} \times t_{PC}}{s_{PC}}, \quad (3)$$

де t_{PC} – час роботи обчислювального вузла;

s_{PC} – шлях, який має пройти обчислювальний вузол.

Аналогічною буде формула для визначення сили обчислювального завдання (позначимо його F_{task}):

$$F_{task} = \frac{P_{task} \times t_{task}}{s_{task}}, \quad (4)$$

де t_{task} – час виконання обчислювального завдання;

s_{task} – шлях, який має пройти обчислювальне завдання.

Відповідно до запропонованого підходу, розподіляти завдання по обчислювальним вузлам потрібно таким чином, щоб сила завдання була рівна (або ж наближеною) до сили обчислювального вузла:

$$F_{task} \approx F_{PC}. \quad (5)$$

Згідно формул (3) та (4), перепишемо формулу (5):

$$\frac{P_{task} \times t_{task}}{s_{task}} \approx \frac{P_{PC} \times t_{PC}}{s_{PC}}. \quad (6)$$

Враховуючи, що при виконанні завдання на обчислювальному вузлі, час виконання завдання t_{task} буде еквівалентний часу завантаженості вузла t_{PC} , а шлях s_{task} , який має пройти завдання (перейти від стану «невиконане» до стану «виконане»), дорівнює шляху s_{PC} , який повинен пройти обчислювальний вузол (перевести обчислювальне завдання від стану «невиконане» до стану «виконане»), то величини t_{task} , t_{PC} , s_{task} , s_{PC} можемо скоротити, і тоді формула (6) приймає вигляд:

$$P_{task} \approx P_{PC}.$$

Отже, розподіл завдань у GRID-системах з невідчужуваними ресурсами / Desktop GRID відповідно до закону балансу сил, не втрачаючи чинності, можна звести до такого, що використовує поняття потужності обчислювального завдання P_{task} та потужності обчислювального вузла P_{PC} , порівнюючи між собою кількісні величин, що відповідають даним поняттям. Дані величини є відносними і на кожній ітерації можуть бути перерахованими в залежності від зміни стану GRID-системи.

На основі описаного вище підходу автором розроблено метод диспетчеризації завдань у GRID-системах з невідчужуваними ресурсами / Desktop GRID, який отримав назву FSA (*Flows Scheduling Algorithm*).

Для реалізації цього методу нам потрібно визначити, що буде розумітися під поняттями потужності обчислювального вузла та потужності обчислювального завдання.

Потужність обчислювального завдання – відносна величина, яка є сукупністю всіх або ж тільки деяких (можливе використання лише однієї) характеристик обчислювального завдання, які вона може мати, визначених певним методом. Дана величина є змінною і залежить від набору завдань, що знаходяться в черзі.

Потужність обчислювального вузла – відносна величина, яка є сукупністю всіх або ж тільки деяких (можливе використання лише однієї) характеристик обчислювального вузла, які він може мати, визначених певним методом. Дана величина є змінною і залежить від набору задіяних обчислювальних вузлів.

Нехай є GRID-система з невідчужуваними ресурсами або Desktop GRID з n обчислювальними завданнями та m обчислювальними вузлами. Позначимо потужність обчислювального завдання як P_n , та потужність обчислювального вузла як R_m .

Введемо поняття балансу сил системи B , беручи до уваги поняття балансу сил з описаного вище підходу, тобто стан системи, при якому потужність i -го обчислювального завдання дорівнює потужності j -го обчислювального вузла:

$$B = \frac{P_i}{R_j} \approx 1.$$

Для ідеального випадку дані потужності рівні:

$$\frac{P_i}{R_j} = 1.$$

Отже, маємо потужності обчислюваних завдань $P = \{P_1, P_2, P_3, \dots, P_n\}$ та потужностей обчислювальних вузлів $R = \{R_1, R_2, R_3, \dots, R_m\}$. Потрібно розподілити завдання по вузлах таким чином, щоб для i -го завдання та j -го вузла мінімізувалось би значення виразу:

$$\left| \frac{P_i}{R_j} - 1 \right|. \quad (7)$$

Враховуючи, що баланс сил повинен бути близьким до 1, а потужності завдань та вузлів вимірюються однаковими одиницями, можна провести деякі перетворення над формулою (7). В ідеальному випадку вираз $\frac{P_i}{R_j} - 1$ повинен бути рівним нулю:

$\frac{P_i}{R_j} - 1 = 0$, тоді вираз (7), який потрібно мінімізувати, зводиться до вигляду:

$$|P_i - R_j|. \quad (8)$$

Алгоритм FSA:

- 1) встановлюємо: $B = 1$;
- 2) існує n завдань і m вузлів;
- 3) обчислюємо: $P = \{P_1, P_2, P_3, \dots, P_n\}$, $R = \{R_1, R_2, R_3, \dots, R_m\}$;
- 4) обираємо i -те завдання, де $i \in \{1, \dots, n\}$;
- 5) знаходимо пару $i-j$, де $j \in \{1, \dots, m\}$, для якої мінімізується вираз:

$$|P_i - R_j|;$$
- 6) відправляємо i -те завдання на j -ий вузол;
- 7) якщо є невідправлені завдання, переходимо до п. 2;
- 8) завершуємо роботу.

Розроблений метод FSA диспетчеризації завдань у GRID-системі з невідчужуваними ресурсами / Desktop GRID не припускає, що в черзі можуть бути завдання, які можна розділити на кілька частин та виконати ці частини окремо одна від одної на обчислювальних вузлах системи, після чого отримані результати об'єднати і отримати таким чином загальний результат для кожного завдання. Тобто, метод FSA не дозволяє розпаралелювати завдання.

Для диспетчеризації завдань, які можна легко розпаралелювати, автором розроблений метод, що отримав назву FSA_P (*Flow Scheduling Algorithm Parallel*). На базі даного методу розроблено алгоритм, що наведений нижче.

В даному випадку виконання завдань відбувається послідовно, при цьому кожне завдання розпаралелюється на низку частин, кількість яких дорівнює

кількості обчислювальних вузлів системи. Вважається, що кожне завдання має якусь відому числову характеристику (наприклад, кількість операцій, розмір файлу тощо). Дана характеристика є фізичною величиною кожного з завдань, і не може змінюватися. При цьому, використовуючи дану характеристику, можливо здійснювати поділ завдання на безліч менших за розміром завдань.

Для прикладу, розглянемо таке завдання: скажімо, що у нас є один великий файл Excel, у якому заповнена таблиця розміром 10000×1000 якимись даними, які потрібно програмно опрацювати, при цьому вони не зв'язні між собою і їх можна оброблювати паралельно на багатьох обчислювальних вузлах. І є 15 різнорідних обчислювальних вузлів, що можуть бути задіяні для обчислень. Тоді ми можемо відправити кожному з вузлів даний файл, а також задати параметри комірок для кожного з вузлів – з якої по яку виконувати обробку файлу. Ці параметри можна розраховувати вже з використанням описаного вище поняття потужності обчислювального вузла.

Враховуючи, що обчислювальні завдання можуть бути різними за своїми характеристиками, в методі використовується абстрактне поняття V – *об'єм завдання*, яке для різного завдання може означати різні властивості, головне, щоб по ньому можливо було ділити завдання.

Алгоритм FSA_P:

- 1) встановлюємо $B = 1$;
- 2) існує n завдань з об'ємом $V = \{V_1, V_2, V_3, \dots, V_n\}$ і m вузлів;
- 3) обчислюємо потужності вузлів $R = \{R_1, R_2, R_3, \dots, R_m\}$ та сумарну потужність системи: $\sum R$;
- 4) обираємо V_i , де V_i — об'єм i -го завдання з множини V , відповідно до порядку в черзі;
- 5) для всіх вузлів знаходимо: $V_{ij} = \frac{V_i \times R_j}{\sum R}$, де $j \in \{1, \dots, m\}$;
- 6) відправляємо завдання з об'ємом V_{ij} на виконання j -му вузлу;
- 7) якщо в системі є не розподілені завдання, переходимо до п. 2;
- 8) завершуємо роботу.

Для описаних вище методів в розділі наведено псевдокод та блок-схеми.

На основі запропонованого підходу розроблено ще два методи FSA Min та FSA Max, що є комбінацією загально відомих методів Min-Min та Max-Min з методом FSA відповідно. Дані методи, як свідчать експерименти, в деяких випадках дають кращі результати ніж FSA, зокрема, при станах системи коли кількість завдань в черзі менша або дорівнює кількості вузлів. В розділі також наведено їх алгоритми, псевдокод та блок-схеми.

У *третьому розділі* проведено аналіз існуючого програмного забезпечення для моделювання GRID-систем та GRID-систем з невідчужуваними ресурсами / Desktop GRID. Зокрема, окремо проводився аналіз програмного забезпечення, яке дозволяє симулювати її роботу та досліджувати різні методи планування завдань для GRID-системи, а також розглянуто та проаналізовано програмне забезпечення, яке

дозволяє симулювати роботу Desktop GRID та досліджувати різні методи планування завдань в ній.

Серед існуючих симуляторів GRID-систем було виділено наступні: OMNET++, Bricks, MicroGrid, SimGrid, GridSim. Наведено їх короткий опис та функціональні можливості.

В свою чергу для симуляції роботи Desktop GRID виділено наступні: SimBOINC, EmBOINC, SimBA, ComBoS, Alea. Також наведено короткий опис кожного з них.

В результаті аналізу встановлено, що описані вище симулятори, без додаткового доопрацювання та розробки нових бібліотек до них, не можуть враховувати потужності завдань та потужності вузлів таким чином, як це потрібно відповідно до запропонованого підходу та розроблених на його основі методів диспетчеризації завдань, причому така доробка потребує великих витрат.

Враховуючи недоліки описаних програм, було вирішено створити програму, яка б могла симулювати роботу GRID-системи з невідчужуваними ресурсами, використовуючи розроблені алгоритми, а також продемонструвати переваги запропонованого підходу.

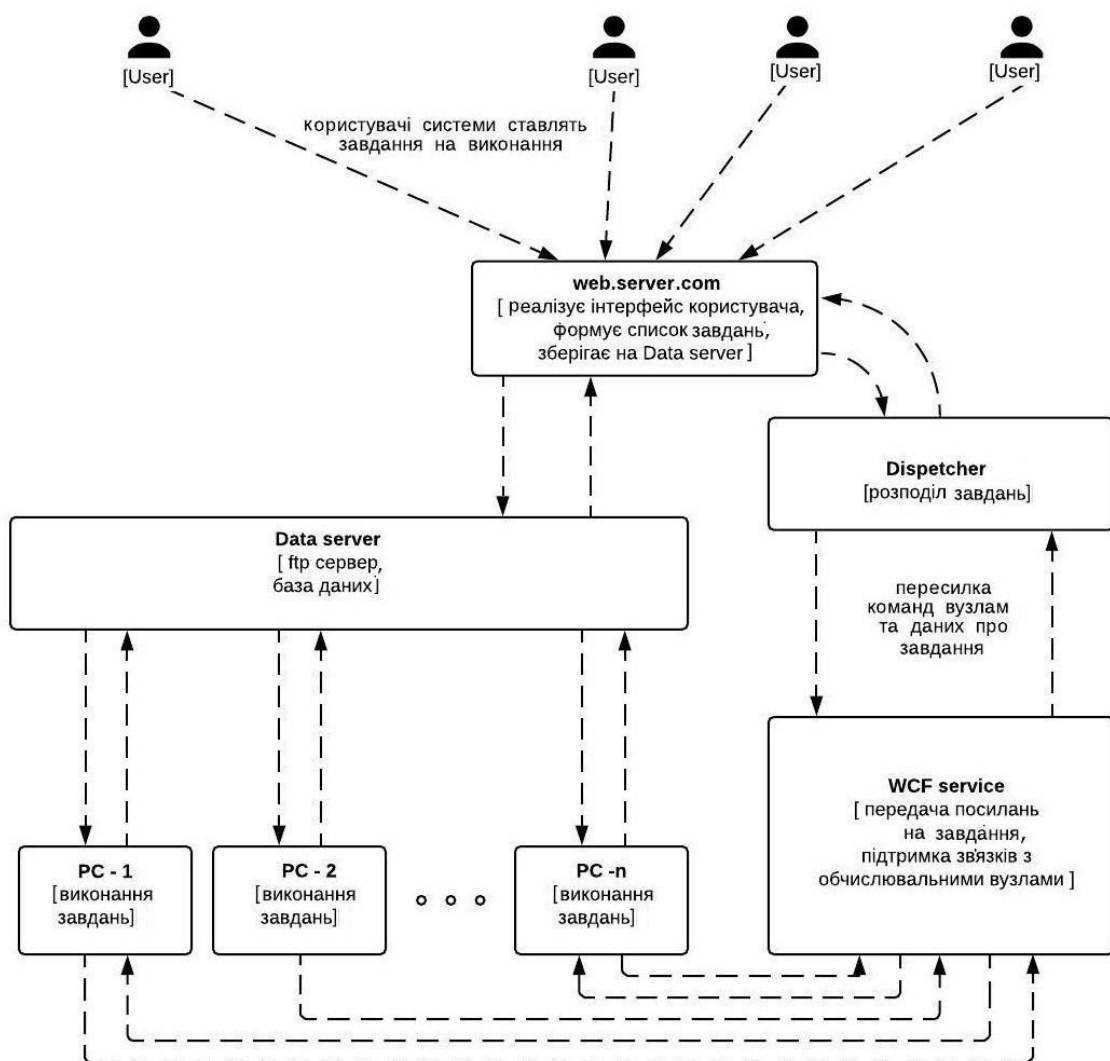


Рисунок 3 – Архітектурна модель GRID-системи з невідчужуваними ресурсами

В якості основи для створення програмного комплексу для симуляції роботи GRID-системи з невідчужуваними ресурсами та дослідження алгоритмів диспетчеризації автором запропоновано клієнт-серверну архітектурну модель програмного забезпечення для розподілених обчислень в комп'ютерних мережах та мережі Internet (рис. 3). Розробка велась мовою програмування C# з використанням платформи Windows Communication Foundation (WCF).

Запропонована автором архітектурна модель складається з трьох компонентів: Client, Server, Host.

Client – програма, що симулює роботу ПК. В системі може бути запущено безліч компонентів Client. На одному ПК також може бути запущено декілька компонентів Client. Кількість задіяних клієнтів на одному ПК обмежується лише характеристиками ПК, здебільшого об'ємом оперативної пам'яті. Наприклад, на ПК з характеристиками Intel Core i7-7500U (2.7-3.5 ГГц) / RAM 24 ГБ / SSD 512 МБ / NVidia GeForce 940MX, 2 ГБ під час проведення експериментів вдавалось запустити одночасно більш, ніж 100 клієнтів.

Server – програма, що симулює роботу сервера. Це є головна програма в системі, яка має бути запущена в одному екземплярі. Вона запускає компонент Host, до якого буде здійснюватися підключення (відключення) клієнтів, та через який буде передаватись інформація.

Host – програма, що реалізує WCF контракт на обслуговування для Client та Server, симулює роботу хоста в мережі. Вона може бути запущена як локально, так і на сервері. В останньому випадку для обчислень може бути задіяно декілька ПК.

На рис. 4 схематично зображено всі компоненти системи та основні функції, які вони виконують. Рис. 5-7 – візуальні інтерфейси даних компонентів.



Рисунок 4 – Схема компонентів симулятора GRID-системи та їх основні функції

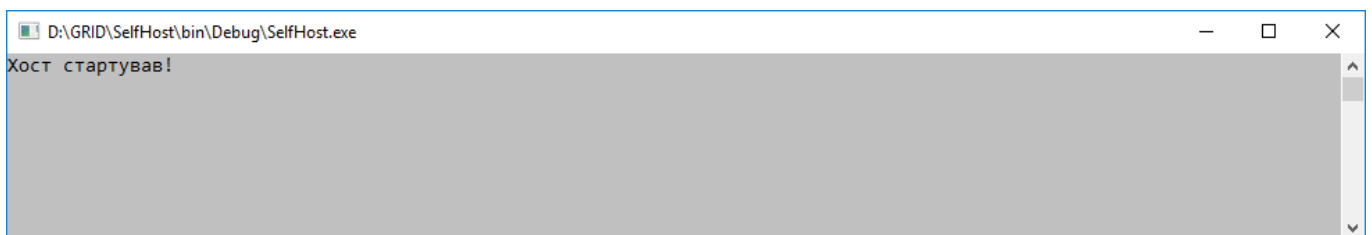


Рисунок 5 – Візуальний інтерфейс компонента Host

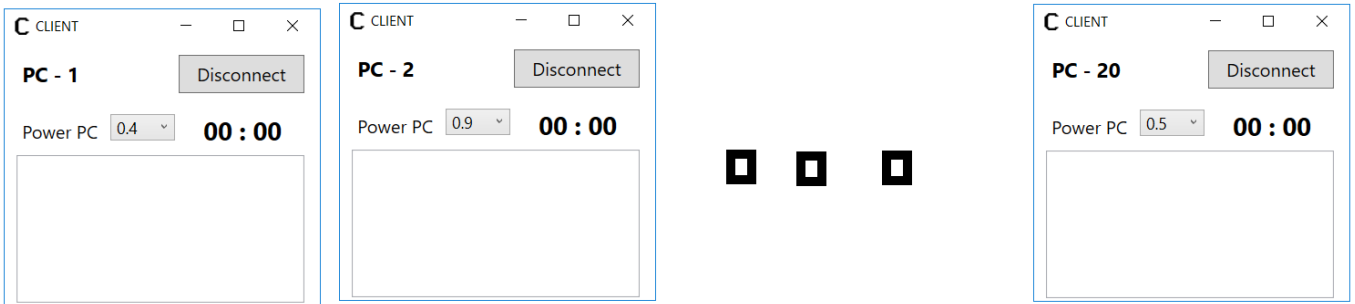


Рисунок 6 – Візуальний інтерфейс компонентів Client

Algorithm Time

Algorithm	Time
ETALON P	0
FCFS P	0
FSA P	0
ETALON	0
FCFS	0
FSA	0
FSA Min	0
FSA Max	0

Task List

ID	task	status	Ptask	ETALONP	FCFSP	FSAP	ETALON	FCFS	FSA	FSAM
10	53	✓	0.828125	2944	13388	4678	53022	132525	58899	17668
11	64	✓	1	3442	16144	5633	64021	320026	64018	64554
12	46	✓	0.71875	2531	11615	4082	46029	92025	46061	76681
13	62	✓	0.96875	3274	15608	5455	62191	68915	155022	69153
14	21	✓	0.328125	1238	5379	1941	21022	35026	105021	70012
15	21	✓	0.328125	1403	5390	1948	21259	105013	70181	70235
16	59	✓	0.921875	3181	14893	5200	59023	65592	65583	98341
17	57	✓	0.890625	3042	14366	5035	57029	190029	63351	14250
18	62	✓	0.96875	3379	15659	5463	62214	103366	103341	69154
19	55	✓	0.859375	2912	13885	4849	55020	55023	110021	13750
20	55	✓	0.859375	2984	13873	4924	55232	110187	55016	11000

Connected Workers

ID	Power	status
4	0.5	✓
5	0.9	✓
6	0.6	✓
7	0.6	✓
8	0.3	✓
9	0.4	✓
10	0.4	✓
11	0.2	✓
12	0.5	✓
13	0.9	✓
14	0.6	✓
15	0.2	✓
16	0.9	✓
17	0.3	✓
18	0.6	✓
19	1	✓
20	0.5	✓

Test Results

ID	sumTask	ETALONp	FCFSP	FSAP	ETALON	FCFS	FSA	FSAMin	FSAMax
20	2195	121349	555709	196516	140448	365130	340261	360078	260040

Умовні позначки:

- 1 – сумарний об'єм всіх завдань, що необхідно вирішити;
- 2 – сумарний час виконання всіх завдань для кожного алгоритму;
- 3 – повідомлення про початок та кінець обчислень, інформація про вибраний алгоритм;
- 4 – умови для тестування: кількість завдань, мінімальне та максимальне їх значення, вибір алгоритму;
- 5 – список всіх завдань, що згенерувалися, їх об'єм та потужність, статус (виконана чи ні), час виконання за кожним алгоритмом;
- 6 – результати тестування;
- 7 – кількість ПК, що потрібно запустити;
- 8 – максимальне і мінімальне значення для вузлів системи;
- 9 – інформація про підключені до системи вузли.

Рисунок 7 – Інформативні області програми Server

Даний симулятор має візуальний інтерфейс та різні налаштування для дослідження (див. рис. 5 – 7). На рис. 6 схематично зображено три клієнти, в той час як насправді в даному випадку їх підключено 20.

На рис. 7 виділено дев'ять областей у вікні програми Server, в які під час тестування виводиться інформація про результати роботи симулятора.

Однією з позитивних властивостей даного симулятора є можливість спостерігати за роботою системи в реальному часі, за рахунок чого користувачеві добре видно, як вузли приймають завдання та виконують їх. Цю функцію реалізовано за допомогою таймера зворотного відліку часу. Ця особливість робить зручним даний комплекс програм не тільки для досліджень, але й при використанні в освітньому процесі для кращого розуміння суті розподілених обчислень та GRID-систем в цілому.

У четвертому розділі описуються обчислювальні експерименти. Виконано низку тестів для різних умов. Для порівняння ефективності розроблених методів в якості базового було обрано відомий метод диспетчеризації FCFS, який є найбільш застосовуваним для GRID-систем з невідчужуваними ресурсами. Крім того інші методи неможливо використовувати в даному випадку, так як вони не передбачають зміни потужності вузлів, а інші характеристики, такі, наприклад, як квант часу чи пріоритети було вирішено не вводити.

В програмному комплексі реалізований механізм тестування, який передбачає виконання всіх завдань, що згенеровані, використовуючи різні алгоритми диспетчеризації. Тобто, спочатку розподіляються завдання за першим алгоритмом, потім за другим і так далі, до останнього. Перелік досліджених методів наступний: ETALON_P, FCFS_P, FSA_P, ETALON, FSA, FSA Min, FSA Max (рис. 7 область 2).

Проводилось дослідження впливу *середньої потужності системи* – зваженої нерівномірності потужностей вузлів системи, на загальний час виконання всієї черги завдань для кожного з методів.

Як видно з графіків (рис. 8 – 10), в систему також заведено методи ETALON та ETALON_P і, на перший погляд, може здатися, що ефективність їх роботи значно краща, ніж методів FSA та FSA_P.

Справа в тому, що дані розрахунків за методами ETALON та ETALON_P зображені не для порівняння з іншими, а для демонстрації еталонного стану системи, коли в ній всі вузли мають потужність 100%. При збільшенні середньої потужності вузлів системи до 1 час виконання за різними методами наближається до максимально можливого еталонного значення.

Далі розглянуто три випадки:

- кількість завдань перевищує кількість ПК (тест 1);
- кількість завдань дорівнює кількості ПК (тест 2);
- кількість завдань менша за кількість ПК (тест 3).

На рис. 8 зображені діаграми, які побудовані відповідно до результатів тестування (тест 1). На їх основі можна зробити висновок, що методи FSA та FSA_P показують кращі результати ніж метод FCFS та FCFS_P. Крім того, як видно з графіка на рис. 8, а) крива методу FSA_P має плавну форму, а крива FCFS_P є ламаною, з чого можна зробити висновок, що метод FSA_P демонструє більш стабільний результат, що робить його добре прогнозованим. Знаючи потужність

системи та об'єм завдань можна складати прогноз часу завершення обчислень. В той же час, використовуючи метод FCFS_P, досить складно прогнозувати час завершення обчислень, адже багато залежить від того, на який вузол яке завдання потрапить на виконання.

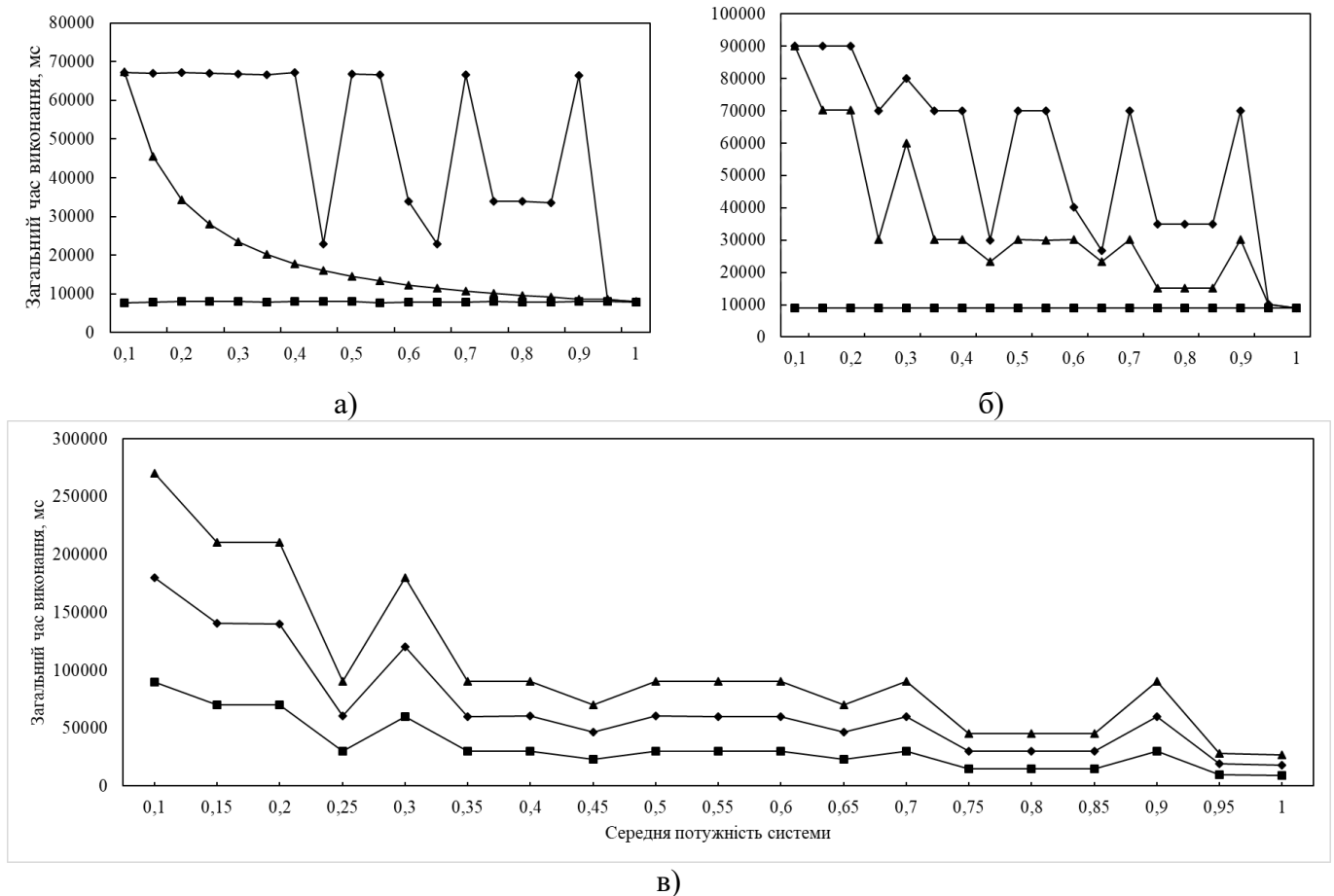


Рисунок 8 – Діаграми залежності часу виконання черги завдань від середньої потужності системи за тестом 1: а) — ■ ETALON_P, ♦ FCFS_P, ▲ FSA_P; б) — ■ ETALON, ♦ FCFS, ▲ FSA; в) — ■ FSA, ♦ FSA Min, ▲ FSA Max

З графіка на рис. 8, б) видно, що послідовний метод FSA має більш гладку криву ніж FCFS, що також говорить про кращу його прогнозованість. На рис. 8, в) зображені додаткові методи FSA Min та FSA Max, в порівнянні з FSA. З графіку видно, що відмінності в результатах незначні і дають вони приблизно однаковий результат, але це лише для тесту 1.

На рис. 9 зображені діаграми, які побудовані на основі результатів виконання тесту 2. На основі них можемо зробити висновок, що запропоновані алгоритми, при умові однакової кількості ПК та завдань на виконання, дають кращі результати ніж FCFS та FCFS_P.

При цьому прогнозованість їх поведінки також зберігається. З рис. 9, в) видно, що в цьому випадку методи FSA Min та FSA Max дають кращий результат ніж FSA, зокрема, при середній потужності системи 0,15-0,35.

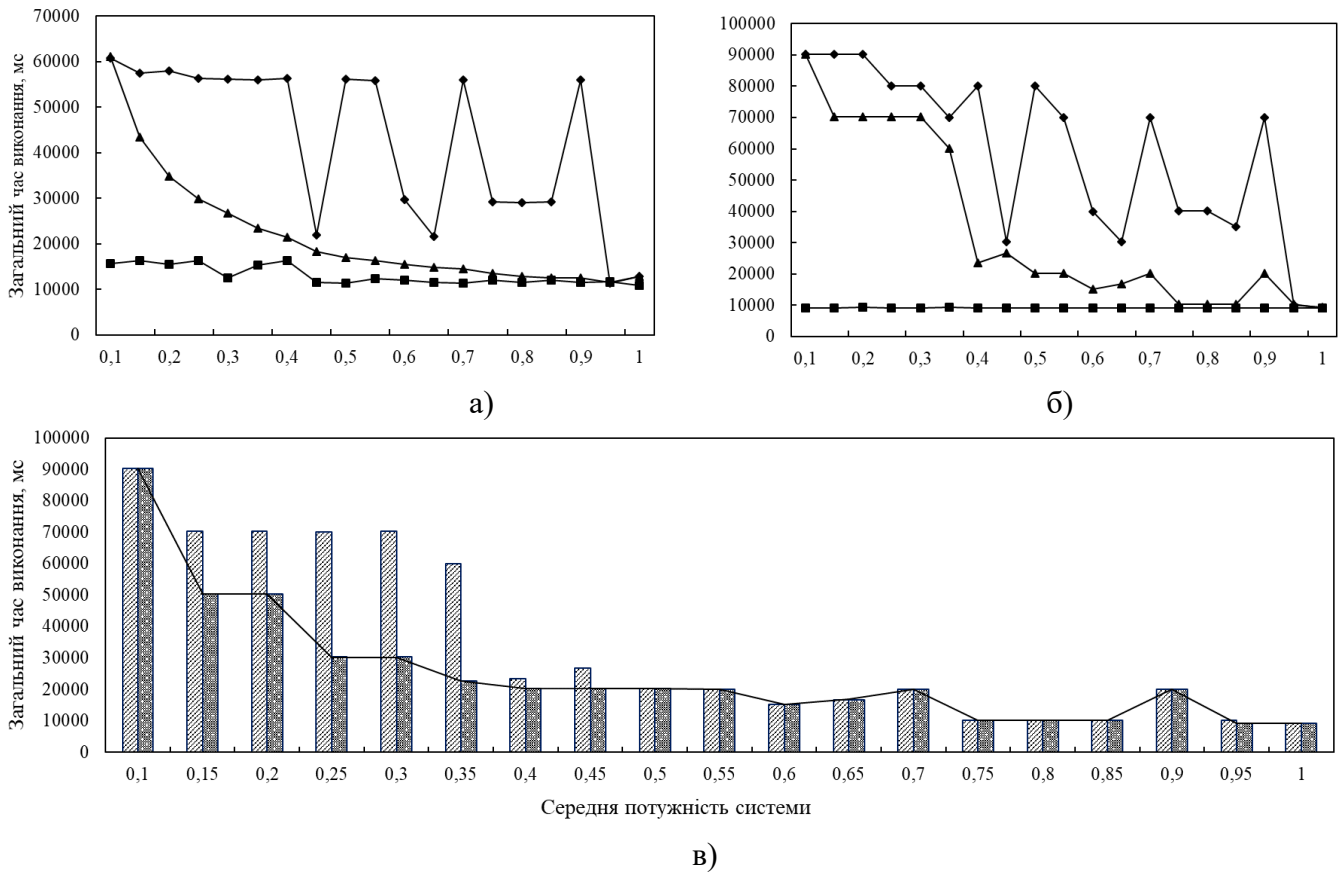


Рисунок 9 – Діаграми залежності часу виконання черги завдань від середньої потужності системи за тестом 2: а) — ■ ETALON_P, ♦ FCFS_P, ▲ FSA_P; б) — ■ ETALON, ♦ FCFS, ▲ FSA; в) — ▨ FSA, ▩ FSA Min, — FSA Max

На рис. 10, 11 зображені діаграми, які побудовані відповідно до результатів виконання тесту 3. На основі них можемо зробити висновок, що запропоновані методи дають кращі результати ніж FCFS та FCFS_P.

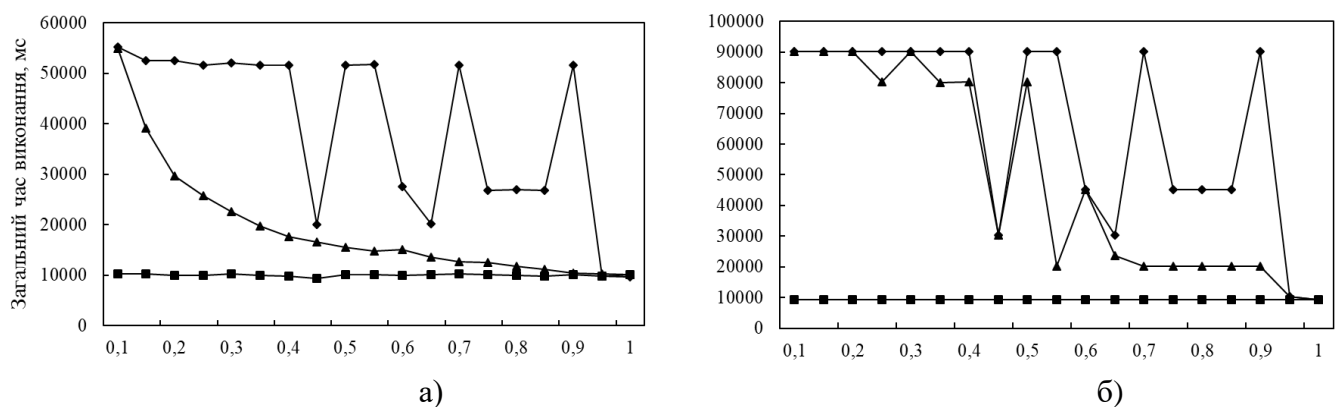


Рисунок 10 – Діаграми залежності часу виконання черги завдань від середньої потужності системи за тестом 3: а) — ■ ETALON_P, ♦ FCFS_P, ▲ FSA_P; б) — ■ ETALON, ♦ FCFS, ▲ FSA

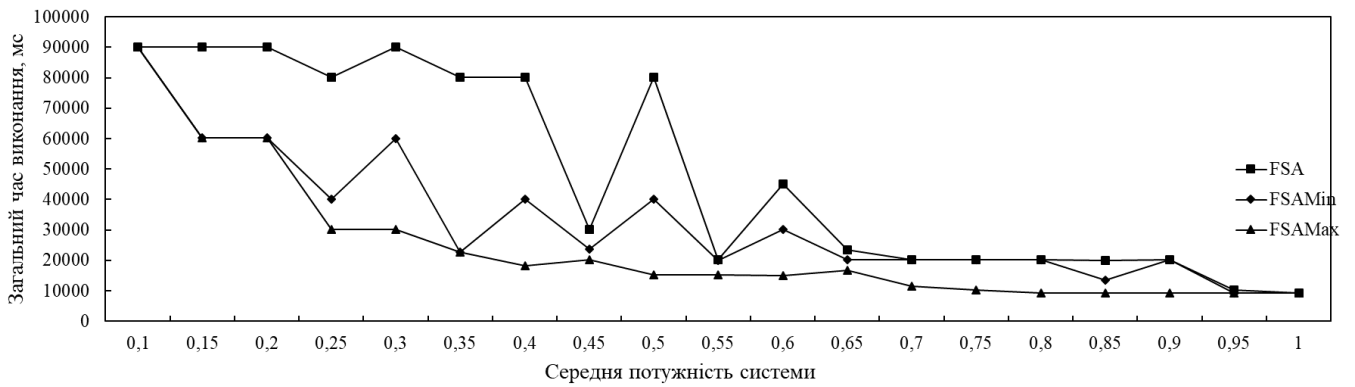
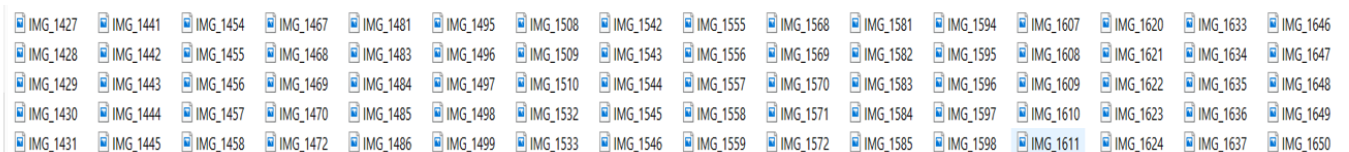


Рисунок 11 – Діаграми залежності часу виконання черги завдань від середньої потужності системи за тестом 3 для методів: ■ FSA, ◆ FSA Min, ▲ FSA Max

При цьому, прогнозованість їх поведінки також зберігається. З рис. 11 видно, що в цьому випадку методи FSA Min та FSA Max дають кращий результат ніж FSA. Найкращий з результатів належить методу FSA Max.

Приклад практичної задачі оптимізації розміру зображень.

В якості прикладу використання запропонованих методів у GRID-системах з невідчувуваними ресурсами / Desktop GRID в роботі розглянуто задачу, схематично наведену на рис. 12. Скажімо, у нас є 1000 зображень, які потрібно оптимізувати відповідно до рекомендацій Google PageSpeed. Загальний розмір файлів – 5 Гб. Якщо виконувати цю задачу на одному ПК, то це займе багато часу. Можливо, навіть цілий день. Все залежить від швидкості роботи ПК, на якому її запустити. Тому, щоб пришвидшити виконання цієї роботи, було б доцільно поділити її між різними ПК, які ми маємо.



Існує 1000 зображень розміром від 100 KB до 10 MB

Задача

Зменшити розмір зображень за допомогою оптимізатора, який вбудований в кожен клієнт, що розташований на вузлах

Існує 15 вузлів з потужністю від 0,1 до 1



Рисунок 12 – Схема простої практичної задачі оптимізації розміру зображень

Припустимо, що у нас є низка ПК, які ми можемо використовувати для вирішення таких задач. І всі ці обчислювальні вузли пов'язані між собою за допомогою програмного забезпечення Desktop GRID.

Основний недолік запропонованого методу полягає в складності розрахунку потужності завдань і вузлів. Але для цього практичного завдання ця проблема вирішується досить просто. Враховуючи, що всі зображення відрізняються одне від одного лише за розміром, найбільшому зображенню (S_{\max}) присвоюється потужність $P_{\max} = 1$, а решта ($n-1$) потужностей завдань обчислюється за формулою:

$$P_i = \frac{S_i \times P_{\max}}{S_{\max}}. \quad (9)$$

Отже, нехай ми маємо n зображень та m ПК.

Враховуючи, що $P_{\max} = 1$ формулу (9) можемо привести до наступного вигляду:

$$P_i = \frac{S_i}{S_{\max}} \quad (10)$$

З іншого боку, у нас є m ПК (обчислювальних вузлів), які відрізняються один від одного, наприклад, лише тактовою частотою процесора. Тоді, подібно до розрахунку потужності завдань, ми можемо пропорційно знайти потужність вузлів.

Вузлу з найбільшою тактовою частотою (F_{\max}) присвоюється потужність $R_{\max} = 1$. Решта ($m-1$) потужностей вузлів обчислюється за формулою:

$$R_j = \frac{F_j \times R_{\max}}{F_{\max}} \quad (11)$$

Враховуючи, що $R_{\max} = 1$ ми можемо скоротити вираз:

$$R_j = \frac{F_j}{F_{\max}}. \quad (12)$$

Формули (10) та (12) використовуються на третьому кроці алгоритму планування завдань FSA згідно його визначення.

Таким чином показано, що розрахувати потужність завдань і потужність вузлів не є складною задачею, коли потрібно розподілити зображення між вузлами, щоб оптимізувати їх розмір. Якщо ж використовуються вузли з різною швидкістю з'єднання з мережею, то їх потужності необхідно обчислювати якимось іншим способом, оскільки не завжди вузол з процесором з найбільшою тактовою частотою матиме найбільшу потужність. І якщо цей момент не враховано, то розподіл буде не таким ефективним, як у першому випадку.

ВИСНОВКИ

У дисертаційній роботі наведено нове вирішення актуальної науково-практичної задачі, що пов'язана з диспетчеризацією завдань в GRID-системах з невідчужуваними ресурсами. Дана задача полягає в подоланні низької ефективності простих і високій вартості та ненадійності складних підходів до створення методів диспетчеризації, зокрема у системах з різнорідними ресурсами. Вирішується дана

задача шляхом розроблення методів диспетчеризації завдань, що враховують різномірність ресурсів.

У роботі отримано наступні наукові та практичні результати:

1) проведено порівняльний аналіз сучасних підходів до вирішення задачі диспетчеризації завдань в GRID-системах з невідчужуваними ресурсами, проаналізовано існуючі програмні засоби для дослідження таких систем;

2) вперше запропоновано підхід до вирішення задачі диспетчеризації завдань в GRID-системах з невідчужуваними ресурсами, який, на відміну від існуючих, пропонує здійснювати розподіл завдань відповідно до закону балансу сил, що забезпечує розроблення простих та ефективних методів диспетчеризації;

3) розроблено метод диспетчеризації завдань (що легко розпаралелюються) в GRID-системах з невідчужуваними ресурсами (FSA_P), який, на відміну від існуючих, враховує різну обчислювальну потужність вузлів, що забезпечує близьку до максимально можливої продуктивності системи;

4) розроблено метод диспетчеризації завдань (що не розпаралелюються) в GRID-системах з невідчужуваними ресурсами (FSA), який, на відміну від існуючих, враховує різну обчислювальну потужність вузлів та різні властивості завдань, що забезпечує зменшення часу виконання черги завдань.

5) запропоновано дві модифікації методу FSA – FSA Min та FSA Max, в яких для початкового розподілу завдань використовуються ідеї методів Min-Min та Max-Min відповідно, що забезпечує зменшення часу виконання черги завдань в порівнянні з методом FSA при середній потужності системи від 0,15 до 0,35 та початковому стані системи, коли кількість завдань дорівнює кількості вузлів. Таким чином, встановлено, що запропонований метод FSA можливо ефективно комбінувати із загальновідомими методами;

б) розроблено програмний комплекс «Симулятор GRID-системи з невідчужуваними ресурсами» (SGridAR-1), який, на відміну від існуючих, побудований з використанням WCF служби, що дозволяє проводити дослідження методів диспетчеризації, як на одному ПК, так і на багатьох (за умови їх підключення до локальної або глобальної комп'ютерної мережі). Як свідчать експерименти, ПК з характеристиками: Intel Core i7-7500U (2.7-3.5 ГГц) / RAM 24 Гб / SSD 512 МБ / Nvidia GeForce 940MX, 2 Гб дозволяє симулювати GRID-систему розміром до 100 вузлів.

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. Uzdenov T. (2021) A New Task Scheduling Algorithm for GRID Systems with Non-alienable Resources. *Studies in Systems, Decision and Control*, vol 346, pp. 207-220. https://doi.org/10.1007/978-3-030-69189-9_12. (*Scopus, ISSN 2198-4182*)

2. Uzdenov T.A. (2021) Task scheduling in Desktop GRID by FSA method: a practical example. *CEUR Workshop Proceedings*, vol. 2850, pp. 97-109. <http://ceur-ws.org/Vol-2850>. (*Scopus, ISSN 1613-0073*)

3. Данильченко О.М. Моделі системи масового обслуговування для розрахунку навантаженості багатоагентних систем / О.М. Данильченко, Д.Г. Літвинчук, Т.А. Узденов // Вісник ЖДТУ №4(55) – ЖДТУ, 2010. – Т.ІІ. – С. 86-

93.

4. Данильченко О.М. Моделі розрахунку трудомісткості операцій комунікації / О.М. Данильченко, Д.Г. Літвинчук, Т.А. Узденов // Моделювання та інформаційні технології, 2011. – Вип. 59. – С. 72-76.

5. Данильченко О.М. Процес диспетчеризації для GRID-системи з невідчужуваними ресурсами / О.М. Данильченко, Т.А. Узденов // Вісник ЖДТУ №2(61) – ЖДТУ, 2012. – Т.ІІ. – С. 147-154.

6. Узденов Т.А. Симулятор процесу диспетчеризації задач в GRID-системах з невідчужуваними ресурсами / Т.А. Узденов // Електронне моделювання. 2021. Т. 43. № 1, С. 87-97.

7. Данильченко О.М. Моделювання програмного комплексу для дослідження паралельних алгоритмів на кластерній системі / О.М. Данильченко, Т.А. Узденов // Тези науково-практичної міжвузівської конференції, 18-19 січня 2010 року. Житомир: ЖДТУ, 2010. – С. 36-37.

8. Данильченко О.М. Проблеми розробки методів керування паралельними завданнями та їх алгоритмічної підтримки для актуальної форми GRID / О.М. Данильченко, Т.А. Узденов // Тези XXXVI науково-практичної міжвузівської конференції, присвяченої Дню науки, 12-13 травня 2011 року. Житомир: ЖДТУ, 2011. – Т.І. – С. 51-52.

9. Узденов Т.А. Модель процесу диспетчеризації для організації розподілених обчислень на GRID-системах з невідчужуваними ресурсами / Т.А. Узденов, В.О. Артемчук // IV Международная научная конференция «МОДЕЛИРОВАНИЕ-2012». Сборник трудов конференции., К., 2012. – С. 71-74.

10. Узденов Т.А. Алгоритми диспетчеризації для GRID систем з невідчужуваними ресурсами / Т.А. Узденов // Збірник матеріалів VIII Всеукраїнської науково-практичної конференції молодих вчених «Наукова молодь-2020» (Київ, 21 жовтня 2020 р.). – К.: ФОП Ямчинський О.В. – С. 186-189.

11. Узденов Т.А. Огляд програм-симуляторів для дослідження алгоритмів диспетчеризації для GRID-систем / Т.А. Узденов // Тези доповідей III Всеукраїнської науково-технічної конференції «Комп'ютерні технології: інновації, проблеми, рішення», м. Житомир, 26-27 листопада 2020 р. – Житомир: Житомирська політехніка, 2020. – С. 21-22.

12. Узденов Т.А. Планування обчислень в Desktop GRID / Т.А. Узденов // Матеріали I міжнародної спеціалізованої наукової конференції «Актуальні питання механічної та електричної інженерії, транспортних технологій, електроніки, автоматизації та ІТ». 5 березня 2021 р., Хмельницький – МЦНД – С. 35-36.

АНОТАЦІЯ

Узденов Т.А. Методи диспетчеризації завдань для GRID-систем з невідчужуваними ресурсами. – На правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.05 – комп'ютерні системи та компоненти. – Інститут проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України, Київ, 2021.

В роботі автором запропоновано новий підхід до вирішення задачі диспетчеризації завдань в GRID-системах з невідчужуваними ресурсами, який пропонує здійснювати розподіл завдань між обчислювальними вузлами відповідно до закону балансу сил, що забезпечує розроблення простих та ефективних методів диспетчеризації.

Автором пропонується чотири методи, що розроблені на його основі, для вирішення даної задачі. Один для завдань, що легко розпаралелюються, інші три – для послідовних завдань.

Пропонується розроблена автором клієнт-серверна архітектурна модель для створення програмного забезпечення для розподілених обчислень в комп'ютерних мережах, а також для створення GRID-систем з невідчужуваними ресурсами. Дана модель побудована на базі WCF сервісу.

Автором розроблено програмний комплекс, що дозволяє симулювати роботу GRID-системи з невідчужуваними ресурсами та досліджувати роботу різних алгоритмів в різних умовах, а також спостерігати за зміною поведінки системи при зміні кількості та об'єму виконуваних завдань, потужності та кількості вузлів, та інші.

Автором проведено дослідження ефективності запропонованих методів, яке показало, що всі вони дають результат кращий ніж FCFS. Таким чином, використання запропонованого підходу забезпечує створення ефективних програмних засобів.

Ключові слова: диспетчеризація завдань, методи планування, FCFS, GRID-система, програма-симулятор, розподілені обчислення.

АННОТАЦИЯ

Узденов Т.А. Методы диспетчеризации заданий для GRID-систем с неотчуждаемыми ресурсами. – На правах рукописи.

Диссертация на соискание ученой степени кандидата технических наук по специальности 05.13.05 – компьютерные системы и компоненты. – Институт проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины, Киев, 2021.

Диссертационная работа посвящена повышению эффективности использования GRID-систем с неотчуждаемыми ресурсами путем исследования и разработки методов для решения задачи диспетчеризации заданий, как одной из основных задач, которая возникает при построении таких систем.

Для достижения цели автором был проведен сравнительный анализ современных подходов к решению задачи диспетчеризации заданий в GRID-системах с неотчуждаемыми ресурсами, который показал, что исследователями со всего мира было предложено множество новых методов для решения данной задачи,

но в реально действующих системах чаще всего используется метод FCFS (*First Come First Served*). И связано это с тем, что данный метод является очень простым в разработке и надежным в работе. Использование же более сложных методов сильно усложняет систему, делает ее менее надежной. Учитывая, что такие системы и так являются не стабильными, из-за многих факторов, то понятно, почему разработчики отказываются от сложных методов и предпочитают FCFS. Из этого следует вывод, что новые методы нужны, но ключевыми характеристиками, которыми они должны обладать, это простота реализации, и лучшая производительность, по сравнению с FCFS. Поэтому разработка и исследование методов диспетчеризации заданий для GRID-систем с неотчуждаемыми ресурсами является актуальной научной задачей, требующей решения.

В работе автором предложен новый подход к решению задачи диспетчеризации заданий для GRID-систем с неотчуждаемыми ресурсами, который предлагает осуществлять распределение заданий между вычислительными узлами в соответствии с законом баланса сил, что обеспечивает разработку простых и эффективных методов диспетчеризации.

На основе предложенного подхода автором разработан метод диспетчеризации заданий (FSA), которые нельзя распараллелить. Рассматривается очередь заданий, требующих решения, каждая из которых может быть выполнена только на одном из узлов. Распределение происходит в соответствии с соотношением мощностей каждого из узлов и мощностей заданий.

Также автором найдены две возможные модификации метода FSA (FSA Min, FSA Max), за счет комбинации с методами Min-Min и Max-Min.

На основе предложенного подхода автором разработан метод диспетчеризации заданий (FSA_P), которые легко могут быть распараллелены. Рассматривается очередь заданий, требующих решения, каждая из которых может быть распараллелена между узлами системы. Распараллеливание происходит пропорционально величине мощностей каждого из узлов.

Предлагается разработанная автором клиент-серверная архитектурная модель для создания программного обеспечения для распределенных вычислений в компьютерных сетях, а также для создания GRID-систем с неотчуждаемыми ресурсами. Данная модель построена на основе WCF сервиса.

На основе предложенной архитектурной модели автором разработан программный комплекс «Симулятор GRID-системы с неотчуждаемыми ресурсами» (SGridAR-1), который позволяет симулировать работу GRID-системы с неотчуждаемыми ресурсами и исследовать работу различных алгоритмов в различных условиях, а также наблюдать за изменением поведения системы при изменении количества и объема выполняемых заданий, мощности и количества узлов, и другие. Особенностью данного программного комплекса является то, что проводить исследования можно не только на одном ПК, а и на многих, соединенных между собой локальной или глобальной компьютерной сетью.

Автором проведено исследование эффективности предложенных методов, которое показало, что все они дают лучший результат чем FCFS. Таким образом, использование предложенного подхода обеспечивает создание эффективных программных средств.

Практическое значение полученных результатов заключается в том, что предложенный в работе подход и разработанные на его основе методы могут быть использованы для решения задачи диспетчеризации заданий в GRID-системах с неотчуждаемыми ресурсами.

Созданный программный комплекс SGridAR-1 можно использовать для: исследования и анализа поведения методов диспетчеризации при изменении различных условий (количества и объема заданий, количества и мощности узлов); проведения предварительных исследования по эффективности различных методов диспетчеризации для заданных архитектур компьютерных систем и заданий, что будут на них выполняться; образовательных целей; проведения тех или иных вычислений, как полноценную GRID-систему с неотчуждаемыми ресурсами (после некоторой доработки).

Предложенную в работе клиент-серверную архитектурную модель можно использовать для построения на ее основе новых программных решений, требующих распределенных вычислений для своего полноценного функционирования.

Ключевые слова: диспетчеризация заданий, методы планирования, FCFS, GRID-система, программа-симулятор, распределенные вычисления.

ABSTRACT

Uzdenov T.A. Task scheduling methods for GRID-systems with inalienable resources. – As a Manuscript.

Dissertation for the degree of candidate of technical sciences in specialty 05.13.05 – computer systems and components. – Pukhov Institute for Modelling in Energy Engineering, National Academy of Sciences of Ukraine, Kyiv, 2021.

The author proposes a new approach to solving the problem of task scheduling in GRID-systems with inalienable resources, which proposes to distribute tasks between computing nodes in accordance with the law of balance of forces, which provides the development of simple and effective methods of scheduling.

The author proposes four methods developed on its basis to solve this problem. One for easily parallel tasks, the other three for sequential tasks.

The author's client-server architectural model for creating software for distributed computing in computer networks and the Internet. And also for creation of GRID-systems with inalienable resources. This model is based on the WCF service.

The author has developed a software package has been developed to simulate the operation of a GRID-system with inalienable resources and to study the operation of different algorithms in different conditions, as well as to monitor changes in system behavior when changing the number and volume of tasks, power and number of nodes, and others.

A study of the effectiveness of the proposed methods, which showed that they all give better results than FCFS. Thus, the use of the proposed approach provides the creation of effective software.

Keywords: task scheduling, scheduling methods, FCFS, GRID-system, simulator program, distributed computing.