

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

ВОЛК МАКСИМ ОЛЕКСАНДРОВИЧ

УДК 004.7:004.415-025.25(043.3)

**МОДЕЛІ, МЕТОДИ ТА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ
УПРАВЛІННЯ РОЗПОДІЛЕНИМ ОБЧИСЛЮВАЛЬНИМ ПРОЦЕСОМ
В ГЕТЕРОГЕННИХ КОМП'ЮТЕРНИХ СИСТЕМАХ**

05.13.06 – інформаційні технології

Автореферат дисертації на здобуття наукового ступеня
доктора технічних наук

Харків – 2019

Дисертацією є рукопис.

Роботу виконано в Харківському національному університеті радіоелектроніки Міністерства освіти і науки України.

Науковий консультант – доктор технічних наук, професор,
Рубан Ігор Вікторович,
Харківський національний університет
радіоелектроніки, проректор з науково-методичної
роботи

Офіційні опоненти: доктор технічних наук, професор
Гамаюн Ігор Петрович,
Національний технічний університет
«Харківський політехнічний інститут»,
професор кафедри програмної інженерії та
інформаційних технологій управління

доктор технічних наук, професор,
Федорович Олег Євгенович,
Харківський національний аерокосмічний
університет ім. М.Є. Жуковського «ХАІ»
завідувач кафедри інформаційних
управляючих систем

доктор технічних наук, професор,
Барабаш Олег Володимирович,
Державний університет телекомунікацій,
завідувач кафедри вищої математики

Захист відбудеться «13» лютого 2019 р. о 13³⁰ годині на засіданні спеціалізованої вченої ради Д 64.052.08 Харківського національного університету радіоелектроніки за адресою: 61166, м. Харків, пр. Науки, 14.

З дисертацією можна ознайомитись у бібліотеці Харківського національного університету радіоелектроніки за адресою: 61166, м. Харків, пр. Науки, 14.

Автореферат розісланий «09» січня 2019 р.

Учений секретар
спеціалізованої вченої ради

І.П. Плїсс

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність теми. На цей час відбувається інтенсивний розвиток інформаційних технологій, пов'язаних з розподіленою обробкою великих обсягів інформації. До найбільш масштабних з них відносять технології використання гетерогенних кластерних систем та систем хмарних обчислень. Ця тенденція викликає підвищений інтерес до розподілених обчислювальних процесів, які використовують хмарні та віддалені комп'ютерні ресурси як інструмент вирішення завдань значної складності. Окрім цього, на сьогодні в більшості сфер виробництва застосовуються інформаційні технології розподілених обчислень, за допомогою яких здійснюється діяльність окремих компаній, що дозволяє автоматизувати та вдосконалити ряд виробничих процесів.

Будь-яка розподілена система зазвичай є неоднорідною (гетерогенною), тому що будується на множині розподілених в просторі обчислювальних ресурсів з різними характеристиками. Завдання, що надходять до виконання в комп'ютерну систему, також неоднорідні, що ускладнює задачу ефективного розподілу. Тому важливе місце у ході розподілу відводять технологіям планування завдань, які виконують складання розкладу використання обчислювальних ресурсів. Багато досліджень направлено на створення методів розподілу завдань за обчислювальними ресурсами, які дозволили б мінімізувати час простою ресурсів, скоротити обсяги та час переданої інформації між ними.

Значну роль в дослідженнях цієї галузі відводять роботам вітчизняних та зарубіжних авторів: Г.М. Луцького, А.М. Бершадського, І.В. Бичкова, І.А. Голубєва, Ж.Б. Кальпєєвої, Н.В. Покусіна, Є.Ю. Селіверстова, Б.А. Телесніна, С.В. Лістрового, С.В. Мінухіна, А.І. Петренка, С.Г. Стіренка, D. Agrawal, A. Amar, R. Bolze, E. Voix, V. Amedro, V. Bodnart-chouk, L. Baduel, H. Zhao, R. Sakellariou.

Сьогодні широкого розповсюдження набули ресурсномісткі завдання, які для свого вирішення вимагають великої кількості обчислювальних ресурсів. Для таких задач використовуються територіально розподілені обчислювальні системи. Одним із прикладів таких систем є гетерогенні кластерні системи. В їх основі лежить забезпечення стабільної роботи певного набору служб, побудованих на відкритих стандартах, та проміжне програмне забезпечення. Висока вартість експлуатації розподілених обчислювальних систем і комунікаційних мереж, що їх об'єднують, вимагає ефективного управління процесами розподілених обчислень. До факторів, що впливають на вартість його проведення, відносять як параметри обчислювального середовища, так і характеристики самого завдання. Останнє обумовлює актуальність питання аналізу умов виконання та параметрів обчислювальних завдань.

Для вирішення цього питання активно залучаються засоби імітації, які вимагають наявності формального уявлення про розподілений обчислювальний процес і можливості проведення з ним певних операцій, які показують зміну основних характеристик програмного середовища в часі. Серед сучасних робіт, які спрямовані на створення такого формального апарату, можна віднести роботи В.В. Окольнішнікова, Т.В. Вознесенської. Специфіка подання розподілених програм як паралельно виконуваних процесів визначає використання в питаннях

аналізу процесної алгебри, розробленої Ч. Хоаром і Р. Мілнером. Для моделювання джерел і перетворювачів інформації, з одного боку, зазвичай використовується класична теорія інформації, розроблена К. Шеноном та істотно доповнена роботами Н. Вінера, В.А. Котельникова та А.Н. Колмогорова. Однак, з іншого боку, класична теорія інформації не враховує взаємодії та внутрішньої природи цих джерел та перетворювачів інформації.

Якість розподілення обчислювальних ресурсів визначає можливість відмови окремого ресурсу або комунікаційного каналу. Тому важливим елементом таких систем є можливість підтримки живучості або функціональної стійкості, тобто продовження обчислювального процесу при відмові одного (кількох) з розподілених програмних компонентів. Цим питанням присвячені роботи вітчизняних та зарубіжних авторів О.Г. Додонова, О.А. Машкова, Д.В. Ланде, Ю.Ю. Громова, С. Shelton, R.Ellison, J. Robert.

Існуючі натеper методи та засоби управління розподіленим обчислювальним процесом в гетерогенних кластерних та хмарних системах мають значні недоліки, що призводять до зниження ефективності роботи і підвищення витрат на її виконання. Заявною у відкритому доступі інформацією, серед причин виникнення цих недоліків слід виділити факт еволюційного розвитку таких засобів, які спиралися на використання моделей, методів та інформаційних технологій, що розроблялися в умовах локальних комп'ютерних систем, кластерів, а на більш високому рівні об'єднувалися за допомогою існуючих на той час комунікаційних засобів. Крім того, з часом виникло протиріччя між різко зростаючою складністю сучасних обчислювальних задач і потенційними технологічними можливостями існуючих інструментальних засобів управління розподіленим обчислювальним процесом, які повинні враховувати широке коло функціональних вимог до завдань, середовища виконання та ресурсів. З огляду на це виникає необхідність підвищення ефективності використання обчислювальних ресурсів та зниження часу виконання завдань в кластерних та хмарних системах. Таким чином актуальною є науково-прикладна проблема розробки моделей, методів та інформаційної технології управління розподіленим обчислювальним процесом в гетерогенних комп'ютерних системах.

Зв'язок роботи з науковими програмами, планами, темами.

Дисертаційна робота виконана відповідно до плану наукової, науково-технічної діяльності Харківського національного університету радіоелектроніки в рамках таких держбюджетних НДР: «Розробка структури Харківського ресурсно-операційного GRID-центру та його ресурсів» (№ ДР 0107U010616) та «Розробка та дослідження застосування GRID-порталу Харківського ресурсно-операційного GRID-центру» (№ ДР 0108U008261) Державної програми «Інформаційні та телекомунікаційні технології в освіті і науці» на 2006–2010 рр. (автор брав участь у виконанні робіт як відповідальний виконавець); «Створення науково-методичних основ забезпечення живучості мережних систем обміну інформацією в умовах зовнішнього потужного НВЧ випромінювання» (№ ДР 0117U003916) у розділі «Дослідження способів забезпечення живучості інформаційно-комунікаційних систем» (брав участь як виконавець).

Мета та завдання дослідження. Метою дисертаційної роботи є підвищення ефективності використання обчислювальних ресурсів та зниження часу виконання

завдань в гетерогенних комп'ютерних системах шляхом розробки та реалізації моделей, методів, інформаційної технології управління розподіленим обчислювальним процесом.

Згідно зі сформульованою метою в дисертаційній роботі необхідно вирішити такі завдання:

- провести аналіз існуючих підходів до управління обчислювальним процесом в комп'ютерних системах;
- сформулювати системні принципи методології управління розподіленим обчислювальним процесом в гетерогенних комп'ютерних системах;
- розробити теоретико-множинну модель розподіленого обчислювального середовища в гетерогенних комп'ютерних системах;
- розробити математичну модель розподіленого обчислювального процесу;
- розробити комплекс методів для оцінювання та вибору призначень програмних компонентів завдання за обчислювальними ресурсами;
- розробити методи забезпечення функціональної стійкості розподіленого обчислювального процесу;
- удосконалити метод оцінки часу виконання програмних завдань;
- удосконалити методи розподілу програмних завдань за обчислювальними ресурсами;
- розробити модель подання завдань і ресурсів у гетерогенних комп'ютерних системах;
- розробити прикладну інформаційну технологію управління розподіленим обчислювальним процесом в гетерогенних комп'ютерних системах;
- провести апробацію та впровадження результатів досліджень у ході вирішення практичних завдань.

Об'єктом дослідження є розподілений обчислювальний процес в гетерогенних комп'ютерних системах.

Предметом дослідження є методологія, моделі, методи та інформаційна технологія управління розподіленим обчислювальним процесом в гетерогенних комп'ютерних системах.

Методи дослідження: дисертаційне дослідження базується на критичному аналізі теоретичних і прикладних розробок вітчизняних і зарубіжних вчених у галузі розробки моделей та методів розподілених інформаційних систем. Для вирішення поставлених завдань використані методи системного аналізу, теорії множин (для розробки моделі розподілу завдань в гетерогенних комп'ютерних системах, моделей завдань та обчислювальних ресурсів), загальної теорії систем (для дослідження та розробки методів розподілу завдань в розподілених обчислювальних системах), теорії імітаційного моделювання (для моделювання обчислювальних процесів в гетерогенних комп'ютерних системах). Для відображення динаміки поведінки окремих моделей використаний формальний апарат процесної алгебри. В основу подання розподілених програмних моделей покладений стандарт HLA (High Level Architecture).

Наукова новизна отриманих результатів.

Вперше розроблено теоретико-множинну модель обчислювального середовища в розподілених гетерогенних комп'ютерних системах, яка уніфікує

опис завдання, комп'ютерних ресурсів та системи управління обчисленнями, що дозволяє формалізувати розподілений процес обчислень для створення комплексного підходу до підвищення ефективності обробки інформації в розподілених комп'ютерних системах.

Вперше розроблено математичну модель розподіленого обчислювального процесу, яка враховує динамічну зміну обсягів пам'яті, потоків даних між програмними компонентами завдання, час простою, методи синхронізації часу виконання, та використовує архітектуру гетерогенної комп'ютерної системи, що дозволяє забезпечити функціональну стійкість, задовольнити встановлені обмеження на обсяги оперативної пам'яті, час використання процесора, та знизити час простою обчислювальних ресурсів.

Вперше розроблено комплекс методів для оцінювання та вибору призначень програмних компонент завдання за обчислювальними ресурсами з урахуванням кількісних оцінок розподіленого обчислювального процесу, що дозволяє використати призначення з найменшим часом виконання завдання та простим обчислювальних ресурсів під час реалізації обраної стратегії управління.

Вперше розроблено методи забезпечення функціональної стійкості розподілених програмних завдань, які передбачають зберігання даних та стану програмних компонентів або історії їх змін у часі, що дозволяє автоматизувати процес відновлення працездатності розподіленого програмного забезпечення після відмови обчислювального ресурсу або комунікаційного каналу зв'язку.

Удосконалено метод оцінки часу виконання програмних завдань в гетерогенних комп'ютерних системах, що, на відміну від існуючих, враховує обсяг пам'яті програмних модулів, обсяги даних, які передаються між програмними компонентами, час простою ресурсів, час та обсяги пам'яті щодо забезпечення функціональної стійкості, що дозволяє оцінити час виконання програмних завдань при розподілі ресурсів з урахуванням обмежень на обсяг пам'яті, сумарний час простою обчислювальних ресурсів для різних методів синхронізації завдань.

Удосконалено метод розподілу завдань за обчислювальними ресурсами в пакетному режимі, який на відміну від існуючих методів, враховує параметри трафіку стосовно завантаження завдання та трафіку між програмними компонентами завдання, що дозволяє підвищити ефективність використання гетерогенних комп'ютерних систем за рахунок зменшення часу виконання пулу завдань.

Набула подальшого розвитку модель подання завдань і ресурсів у гетерогенних комп'ютерних системах, яка, на відміну від існуючих, враховує методи розподілу завдань за ресурсами та обсяги мережного трафіку в процесі завантаження і виконання завдань, що дозволяє скоротити час виконання завдань та зменшити навантаження на мережі передачі даних.

Набули подальшого розвитку методи розподілу програмних завдань за обчислювальними ресурсами, які, на відміну від існуючих, дозволяють здійснювати розподіл ресурсів з максимальним просуванням часу виконання в умовах обмежень на обсяг оперативної пам'яті, часу простою і обрати методи синхронізації та забезпечення функціональної стійкості програмних компонентів завдання, що дозволяє зменшити час виконання, обсяги пам'яті, трафік

обслуговування завдання та кількість потрібних обчислювальних ресурсів і тим самим підвищує ефективність процесу розподілених обчислень.

Набула подальшого розвитку методологія управління обчислювальним процесом в розподілених комп'ютерних системах, яка базується на системних принципах гетерогенності програмних завдань та ресурсів, вибору стратегії управління, вибору методів розподілення ресурсів, зв'язності програм та урахування трафіка у завданні, вибору методів синхронізації, функціональної стійкості та моделювання розподіленого обчислювального процесу, що забезпечує ефективне використання обчислювальних ресурсів та знижує час виконання завдань.

Практичне значення отриманих результатів. Практичне значення отриманих теоретичних результатів дисертаційної роботи підтверджено підвищенням продуктивності гетерогенних комп'ютерних систем, за рахунок скорочення простою обчислювальних ресурсів та зменшення часу виконання завданню. Зокрема, практичне вирішення теоретичних досліджень полягає у такому.

Запропоновано прикладну інформаційну технологію управління розподіленим обчисленням процесом (РОП) з дослідженням моделей програм і вибором схеми призначення, яка використовує систему імітаційного моделювання для отримання оцінок процесу виконання завдання на гетерогенних комп'ютерних ресурсах. Інформаційна технологія забезпечує: розподіл завдань за обчислювальними ресурсами в розподілених гетерогенних комп'ютерних системах, який виконується на підставі використання множини методів розподілу, що дозволяє здійснювати вибір схеми призначення з найменшим часом виконання пулу завдань та знижує простої обчислювальних ресурсів; функціональну стійкість РОП, що дозволяє продовжувати виконання програмного завдання в умовах відмови обчислювальних ресурсів або каналів передачі даних; управління РОП згідно з обраною стратегією, та має додаткові функції для ефективного розподілу (динамічного перерозподілу) завдань у випадку відмови обчислювального ресурсу, які зокрема виконують аналіз завдань та ресурсів, згортку кортежу параметрів завдання, аналіз зв'язності програмних модулів у завданні, вибір схеми призначення, імітаційне моделювання РОП, забезпечують функціональну стійкість.

Результати дисертаційної роботи впроваджені: в державному підприємстві Науково-дослідному технологічному інституті приладобудування (ДП НДТІП) (акт про впровадження від 17.05.2018), в державному підприємстві «Центральне конструкторське бюро «ПРОТОН» (акт про впровадження від 10.07.2018), у практичну діяльність товариства з обмеженою відповідальністю «Альтер системс» (акт про впровадження від 22.08.2018), в Харківському національному університеті радіоелектроніки на кафедрі електронних обчислювальних машин в процесі проведення лекційних занять і лабораторних робіт з курсів «Системне програмування», «Технології високопродуктивних хмарних обчислень», «Інтерфейси паралельного програмування» та «Паралельне моделювання на інноваційних НРС (High-performance computing) системах» (акт від 12.06.2018).

Особистий внесок здобувача. Всі наукові результати дисертаційної роботи, що виносяться на захист, отримані автором особисто. У роботах, опублікованих у співавторстві, особистий внесок здобувача полягає у такому: [1] запропоновано

етапи інформаційної технології розподілу завдань в гетерогенному середовищі, які ґрунтуються на використанні імітаційного моделювання у реальному часі; [2] теоретико-множинна модель програмних компонентів завдання та процесів їх виконання у розподілених обчислювальних системах; [3] математична модель розподіленого обчислювального процесу для технології аналізу розподілених програмних завдань; [4] запропоновано удосконалення існуючого методу розподілу завдань з урахуванням трафіку, який виникає всередині завдання; [5] запропоновано метод розподілення завдань за обчислювальними ресурсами з урахуванням алгоритмів синхронізації в гетерогенних комп'ютерних системах; [7] запропонована аналітична модель процесу оптимістичної синхронізації розподілених обчислень; [8] на основі аналізу сучасного стану GRID-технологій, запропоновано розширення моделей завдань для розподіленого виконання; [12] розроблено формальний апарат опису об'єктів розподілених баз даних щодо паралельного обчислювального процесу в гетерогенних комп'ютерних системах; [14] математична модель оцінки ефективності розподіленого обчислювального процесу у локальних мережах; [16] метод паралельного моделювання та розподілення обчислень на багатоядерних процесорах; [17] запропоновані методи паралельних та розподілених обчислень в задачі моделювання промислових об'єктів; [18] сформульована стратегія використання різних методів розподілу ресурсів для завдань в гетерогенних обчислювальних системах; [19] формальний математичний апарат моделювання процесів у розподілених комп'ютерних системах; [20] архітектурні елементи системи управління розподіленими обчисленнями; [21] принцип та засоби врахування трафіку завдання в комп'ютерній мережі для забезпечення ефективного розподілу завдань за обчислювальними ресурсами; [22] метод інтеграції спільних модулів у систему моделювання; [24] розширені множинні моделі системи управління розподіленими обчисленнями за рахунок потоків даних у системах імітаційного моделювання та запропоновано систему візуалізації та аналізу цих даних; [25] обґрунтовано вибір формального математичного апарату процесних алгебр щодо опису процесів управління розподіленим обчислювальним процесом та формалізовано процеси розподіленого імітаційного моделювання; [26] запропонована архітектура планувальника завдань; [27] моделі потоків завдань у гетерогенних обчислювальних системах; [28] запропоновано архітектурне рішення використання скрипкових плагинів у системі управління розподіленими обчисленнями; [29] моделі та метод аналізу розподілених імітаційних моделей з консервативними алгоритмами синхронізації з урахуванням оцінки простою обчислювальних ресурсів; [30] запропоновано принцип розподілу завдань на обчислювальні ресурси, який підтримує можливість проведення імітаційного моделювання РОП; [31] формалізація паралельних обчислень в задачах організації бізнес-процесів інформаційних систем з урахуванням їх програмної реалізації; [32] використання моделей та методів краткострокового оцінювання характеристик інформаційних систем для планувальників розподіленого обчислювального процесу; [33] метод моделювання розподіленого обчислювального процесу в умовах балансування

навантаження з урахуванням трафіку; [34] запропонована технологія синхронізації розподілених програмних компонентів завдання; [35] запропонований узагальнений критерій оцінки завдань, які поступають до системи управління розподіленими обчислюваннями; [38] запропоновано використання імітаційного моделювання в задачах управління розподіленими обчисленнями; [41] обґрунтовано можливість підключення модулів аналізу обчислювального середовища кластерних систем до системи управління розподілом завдань; [42] створення розподілених програмних моделей сучасними інтерфейсами паралельного програмування; [43] запропоновано принцип використання нових методів дослідження процесів в розподілених обчислювальних системах; [44] запропоновані нові компоненти у середовищі моделювання гетерогенних комп'ютерних систем; [45] запропоновано класифікацію методів розподілу ресурсів у гетерогенних комп'ютерних системах; [46] розроблено підсистему моніторингу трафіка гетерогенних комп'ютерних мережах; [47] розроблено метод інтеграції спільних модулів у систему моделювання; [48] розроблена архітектура підсистеми візуалізації даних моніторингу та результатів імітаційного моделювання розподіленого обчислювального процесу; [49] обґрунтовано використання імітаційної системи моделювання з метою організації управлінських рішень у процесах розподілених обчислень, запропоновано архітектуру та функціональну модель системи імітаційного моделювання гетерогенних комп'ютерних систем; [50] запропоновано засіб моніторингу трафіку в гетерогенних комп'ютерних мережах; [51] метод оптимізації передачі даних за рахунок використання протоколів низького рівня; [52] дослідження та вибір архітектури планувальника ресурсів в кластерних системах; [53] моделі потоків завдань в кластерних системах; [54] архітектура програмного комплексу з використання програмних модулів, що динамічно підключаються; [55] запропоновано метод аналізу процесу розподіленого імітаційного моделювання з урахуванням мережного трафіку; [56] запропоновано аналітичні моделі аналізу розподілених обчислень з різними методами синхронізації; [57] основні вимоги до мережних протоколів, які використовуються для імітаційного моделювання кластерних систем; [58] запропоновано концепцію менеджера пам'яті для реалізації методів автоматичного управління процесами розподілених обчислень; [59] запропонована класифікація існуючого прикладного програмного забезпечення опрацювання завдань; [60] обґрунтовано розподілення програмних компонентів з використанням мобільних платформ; [61] моделі процесів розподіленої імітації для консервативних та оптимістичних методів синхронізації; [62] запропоновані етапи технології синхронізації моделей у режимі реального часу та з використанням апріорної інформації; [63] запропоновано інформаційну технологію розподілення завдань за обчислювальними ресурсами; [64] запропоновано інформаційна технологія аналізу розподілених елементів системи управління обчислювальним процесом; [65] наведено етапи технології забезпечення розподіленого імітаційного моделювання; [66] запропоновано інформаційну технологію управління процесом розподілу завдань в розподілених комп'ютерних системах; [67] запропоновано впровадження інформаційної

технології розподілу завдань на обчислювальні ресурси в існуючі кластерні системи; [68] модель процесу розподілення ресурсів в системах імітаційного моделювання; [69] етапи інформаційної технології імітаційного моделювання у системі управління розподіленими обчисленнями; [70] застосування інформаційної технології розподілу завдань, пов'язаних з опрацюванням радіоастрономічних даних на обчислювальні ресурси у гетерогенних комп'ютерних системах; [72] запропоновано інформаційну технологію управління розподілом обчислювальних ресурсів з використанням імітаційного середовища моделювання, яке відтворює процеси, що відбуваються у гетерогенній комп'ютерній системі; [73] окремі етапи розробленого модифікованого метода Backfill; [76] інформаційна технологія розподілення пакетів завдань за обчислювальними ресурсами в гетерогенних кластерних системах; [77] метод забезпечення функціональної стійкості розподілених обчислень на основі дампа пам'яті; [78] транзакційний метод розподілення завдань за обчислювальними ресурсами; [79] моделі обчислювального процесу в розподілених мультисервісних системах; [80] етапи інформаційної технології розподілу завдань за ресурсами з альтернативними схемами призначення; [81] архітектуру фреймворка, що підтримує функціональну стійкість програмного забезпечення в розподілених комп'ютерних системах; [82] обґрунтовано вибір методів забезпечення живучості програмного забезпечення для реалізації на рівні операційної системи; [83] етапи удосконаленої інформаційної технології розподілу завдань на обчислювальні ресурси в кластерних системах; [84] запропоновано етапи методу аналізу розподілених імітаційних моделей з розширеною множиною параметрів.

Апробація результатів дисертації. Основні положення дисертаційної роботи апробовані на таких міжнародних конференціях і форумах: 1-й, 2-й Міжнародній конференції «Современные информационные системы. Проблемы и тенденции развития» (Харків-Туапсе, 2006 р., 2007 р.); 7-й, 8-й, 9-й Міжнародній науково-технічній конференції «Проблеми інформатики і моделювання», (Харків: 2007, 2008, 2009 рр.); 10-й, 11-й, 12-й, 15-й Міжнародній науково-технічній конференції «Системний аналіз та інформаційні технології», (Київ: 2008, 2009, 2010, 2013 рр.); Міжнародній науково-технічній конференції «Моделювання – 2008», (Київ: 2008р.); 1-й, 2-й Міжнародній науково-технічній конференції «Інформаційні технології в навігації й управлінні: стан та перспективи розвитку», (Київ: 2010 р.); 5-й міжнародній науково-практичній конференції «Інформаційні технології в наукових дослідженнях і навчальному процесі», (Луганск: 2010 р.); 1-й, 3-й, 5-й Міжнародній науково-технічній конференції «Сучасні напрямки розвитку інформаційно-комунікаційних технологій і способів управління», (Київ – Харків 2010 р., Полтава – Белгород – Харків – Київ – Кіровоград 2013 р., Полтава – Баку – Кіровоград – Харків 2016 р.); 3-й, 4-й, 5-й Міжнародній науково-практичній конференції «Информатика, математическое моделирование, экономика», (Смоленськ: 2013, 2014, 2015 рр.); 9-й Науковій конференції «Новітні технології – для захисту повітряного простору», (Харків: 2013 р.); 26-й Міжнародній науково-практичній конференції «Впровадження перспективних мікропроцесорних систем залізничної автоматики і засобів телекомунікацій на базі цифровізації», (Алушта: 2013 р.); 1-й, 2-й, 3-й, 4-й Міжнародній науково-технічній конференції «Проблеми

інформатизації» (Черкаси – Київ – Тольятті – Полтава 2013 р., Київ – Полтава – Катовіце – Париж – Белгород – Черкаси – Харків 2014 р., Черкаси – Баку – Бельсько-Бяла – Полтава 2015 р., Черкаси – Баку – Бельсько-Бяла – Полтава 2016 р.); 5-й Міжнародній науково-технічній конференції «Інформаційні системи та технології», (Коблево: 2016 р.); 1-й, 2-й міжнародній науково-технічній конференції «Комп'ютерні та інформаційні системи і технології», (Харків: 2017 р.); Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2017), (Novi Sad, Serbia: 2017).

Публікації. За результатами досліджень опубліковано 84 наукові праці, з них – 32 статті у фахових періодичних виданнях України з технічних наук; 17 статей входять до міжнародних наукометричних баз, з них 3 – до міжнародної наукометричної бази Scopus; 52 тези доповідей на міжнародних конференціях та симпозиумах, з них 1 – входить до міжнародної наукометричної бази Scopus; 8 публікацій англійською мовою.

Структура та обсяг роботи. Дисертація складається зі вступу, 6 розділів, висновків, списку використаних джерел, додатків. Загальний обсяг роботи складає 427 сторінок тексту, що містять 2 анотації на 35 сторінках, 35 рисунків, 7 таблиць, з яких 1 таблиця повністю займає 1 сторінку, список використаних джерел з 322 найменувань на 36 сторінках, 5 додатків на 38 сторінках.

ОСНОВНИЙ ЗМІСТ РОБОТИ

У вступі обґрунтовано актуальність дисертаційної роботи, сформульовано проблему, мету і основні завдання досліджень, наведено відомості щодо зв'язку дисертації з планами організації, де виконана робота. Дано стислу анотацію отриманих у дисертації результатів, відзначено їх наукову новизну та практичну цінність, наведено дані щодо використання результатів проведених досліджень.

У **першому розділі** на базі вивчення літературних джерел проведено аналіз предметної галузі. Розглянуто питання розвитку розподілених систем як середовища для розв'язання задач великої обчислювальної складності. Наведено огляд технологій розподілених обчислень та хмарних технологій, існуючих планувальників завдань відкритого доступу, мов опису завдань в гетерогенних розподілених комп'ютерних системах, огляд існуючого програмного забезпечення підтримки РОП, методів, засобів та існуючих стандартів розподіленого імітаційного моделювання як інструменту дослідження обчислювальних процесів у гетерогенних комп'ютерних системах, методів розподілу завдань за обчислювальними ресурсами.

Визначено, що існуючі моделі, методи та інформаційні технології управління розподіленим обчислювальним процесом в гетерогенних комп'ютерних системах мають свої переваги та недоліки. До основних недоліків відносяться: відсутність єдиного підходу до формалізації процесу управління розподіленими обчисленнями при реалізації різних стратегій управління; не враховується реальний час виконання завдань (зазвичай вказується користувачем орієнтовно, або визначається експериментально); не враховується час передачі і характер повідомлень (переданих даних) між програмами в завданні; розглядається

ідеальний випадок, коли кількість програм в завданнях дорівнює кількості процесорів, а процесори мають однакові характеристики; не враховуються алгоритми статичного і динамічного розподілу ресурсів, від яких багато в чому залежить загальний час виконання завдань; не враховуються більшість кількісних параметрів програмних компонент в завданнях і обчислювальних ресурсів: обсяги оперативної, віртуальної і зовнішньої пам'яті, фонове завантаження процесора та ін.; не розглядається варіант автоматичного вибору і зміни методів синхронізації розподілених програмних завдань; не враховується можливість виникнення тупиків, відмов як у функціонуванні програмних, так і апаратних засобів. Наприкінці розділу, виходячи з проведеного аналізу, сформульована проблема, поставлені мета та задачі наукового дослідження.

У **другому розділі** запропоновано методологічні основи управління розподіленим обчислювальним процесом (РОП) в гетерогенних комп'ютерних системах. На рис. 1 представлені елементи методології управління РОП в гетерогенних комп'ютерних системах. Сформулюємо її основні системні принципи. Принцип вибору стратегії управління РОП передбачає можливість автоматичного реконфігурування компонентів системи управління РОП і середовища виконання завдань залежно від обраної стратегії управління. В існуючих рішеннях подібні завдання вирішуються, як правило, в ручному режимі адміністраторами систем розподілених обчислень та не змінюються залежно від пулу завдань. Принцип гетерогенності завдань і ресурсів передбачає, що всі завдання, їх програмні компоненти та обчислювальні ресурси мають різні чисельні та якісні характеристики. Системи з однорідними параметрами можуть виступати окремим випадком РОП. Принцип вибору методів розподілу ресурсів передбачає, що система управління може використовувати не один, а кілька методів розподілу, оскільки для різних завдань і різного набору обчислювальних ресурсів не існує методу, який гарантовано дасть кращий результат. Принцип зв'язності програм і обліку внутрішнього трафіку в завданні передбачає використання апріорної або отриманої в ході експерименту інформації щодо обміну даними між компонентами завдання і дані стосовно каналів зв'язку між обчислювальними ресурсами. Принцип вибору методів синхронізації програм в завданні передбачає залежність ефективності розподіленого обчислювального процесу від вибору методу синхронізації програм в завданні. Принцип забезпечення функціональної стійкості РОП передбачає відновлення стану програмних компонентів у завданні при відмові обчислювального ресурсу або каналу зв'язку з ним на іншому обчислювальному ресурсі. Принцип використання імітаційного моделювання РОП передбачає автоматичне використання імітаційного моделювання для прийняття рішень в системі управління розподіленими обчисленнями в гетерогенних комп'ютерних системах.

Пропонуються рішення архітектурного рівня для сервісів системи управління РОП в умовах розподілених ресурсів. Запропоновано універсальну, кластерну та гетерогенну структури системи управління розподіленими обчисленнями, наведено основні вимоги до системи та її структурних елементів. Досліджені існуючі планувальники, наведено підґрунтя формалізації опису завдань та обчислювальних ресурсів, задачі розподілу завдань за обчислювальними ресурсами.

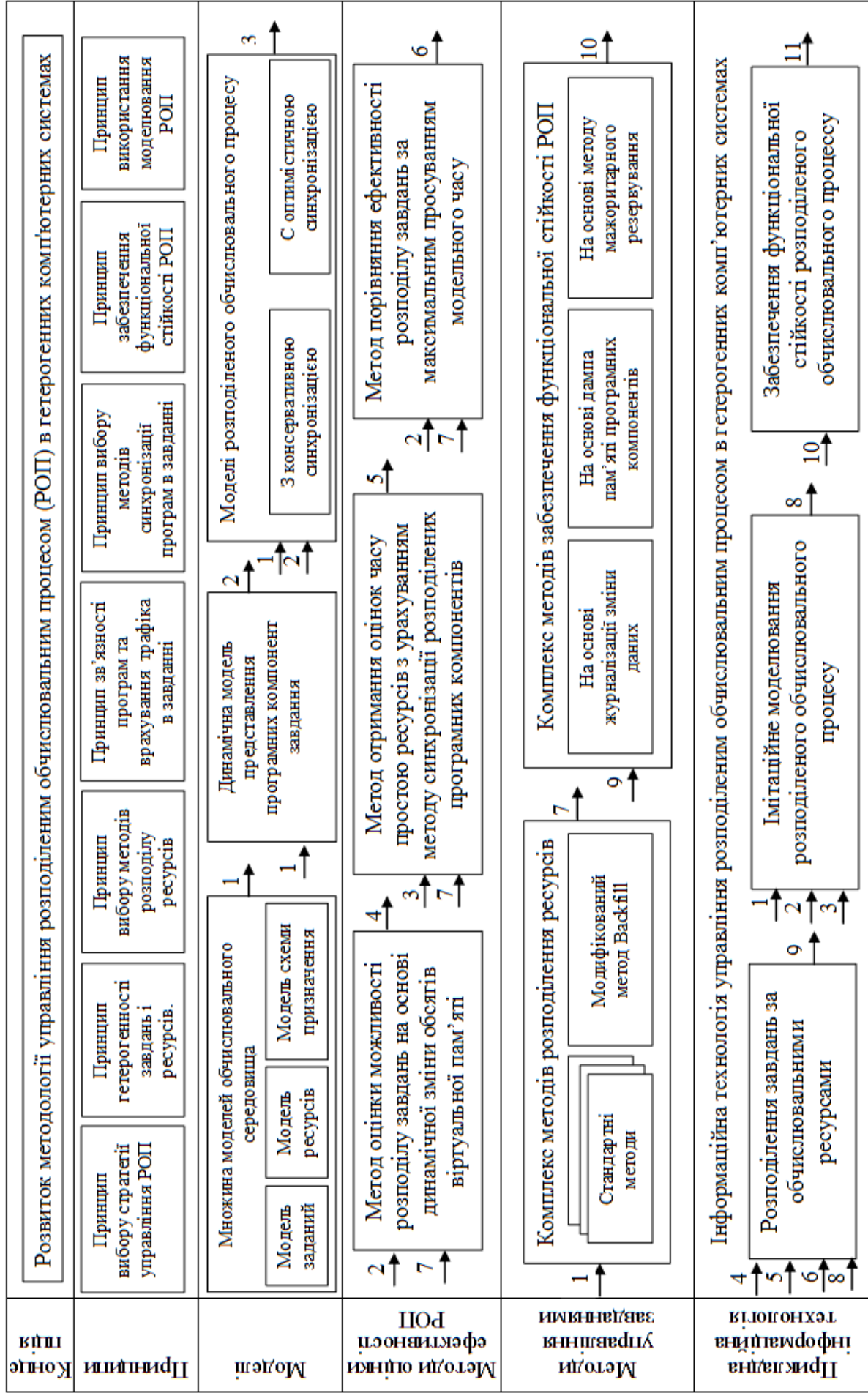


Рисунок 1 – Методологія управління розподіленим обчислювальним процесом в гетерогенних комп'ютерних системах

Запропоновано теоретико-множинну модель розподіленого обчислювального середовища під час виконання завдання в розподілених гетерогенних комп'ютерних системах:

$$\begin{aligned}
 En &= \langle \overline{Z}, \overline{R}, \overline{S} \rangle, \\
 Z_i &= \{ar_i^z, os_i^z, pc_i^z, ps_i^z, ms_i^z, dc_j^r, pr_i^z, ca_i^z, rt_i^z\} \cup \{P_i^z, M_i^z, DF_i^z, RF_i^z\}, \forall i = 1..M, (1) \\
 R_j &= \{ar_j^r, os_j^r, pc_j^r, ps_j^r, ms_j^r, dc_j^r, bw_j^r, d_j^r\}, \\
 S &= \{An, Csh, Sp, Fs, Q_k\}, Q_k = \{mn_k, lp_k\}, \forall k = 1..K.
 \end{aligned}$$

де En – обчислювальне середовище РОП; Z (job) – множина завдань, що надходять до гетерогенної комп'ютерної системи; R (resource) – множина обчислювальних ресурсів; S – множина компонентів системи управління розподіленим обчислювальним процесом, кожен з яких є окремою програмою (сервісом) операційної системи та може розташовуватися на одному або на різних комп'ютерах, які об'єднані мережним каналом зв'язку; An – аналізатор програмних завдань та обчислювальних ресурсів; Csh – підсистема вибору схеми призначення, яка використовує множину Q_k програмних реалізацій методів розподілу завдань за обчислювальними ресурсами; mn_k – програмна реалізація методу розподілу завдань за обчислювальними ресурсами (реалізується за допомогою бібліотек, що динамічно підключаються); lp – перелік вхідних параметрів, що враховуються під час розподілу; K – кількість програмних реалізацій методів розподілу, які є у наявності у системі управління РОП; Sp – підсистема моделювання процесу виконання завдань; Fs – підсистема забезпечення функціональної стійкості; описова частина завдання: ar_i (architecture) – архітектура процесора, os_i (operating system) – операційна система, pc_i (processor count) – кількість процесорів, ps_i (processor speed) – швидкість процесорів, ms_i (memory size) – об'єм оперативної пам'яті, ds_i (disk capacity) – доступний об'єм зовнішньої пам'яті, pr_i (priority) – пріоритет завдання; змістовна частина завдання: P_i^z – програмні компоненти завдання (програми, бібліотеки функцій або класів, оверлеї); M_i^z – атрибути взаємозв'язків програмних компонентів з оціночними характеристиками об'ємів даних, які передаються між програмами; DF_i^z, RF_i^z – дискриптори файлів вихідних даних та файлів результатів.

Результатом вирішення задачі розподілу ресурсів є схема призначення, де кожному програмному компоненту ставитися у відповідність обчислювальний ресурс. Схеми призначення, отримані в результаті застосування різних методів розподілу ресурсів, складають таку множину:

$$Sh_k = Q_k(Z_i, R_j) = \{Z_i \rightarrow R_j\}; i = \overline{1, M}; j = \overline{1, N}; k = \overline{1, K} \quad (2)$$

Запропоновано формалізацію процесу забезпечення функціональної стійкості розподіленого обчислювального процесу. Під функціональною стійкістю розподіленого обчислювального процесу будемо розуміти здатність системи управління продовжувати виконання завдання при відмові одного (декількох) комп'ютерного ресурсу або мережного каналу зв'язку. Знаходження

відображення зв'язків M на існуючі канали зв'язку N , як правило, забезпечується мережними сервісами і не розглядається в даній роботі. Відмови каналу зв'язку дорівнює відмові ресурсу. За цією умовою, для забезпечення функціональної стійкості, необхідно запустити програмний компонент, що працював на цьому ресурсі, на іншому комп'ютері. При цьому програмний компонент повинен бути переведений в той стан, в якому він знаходився в момент відмови.

Введена концепція використання сервісу менеджера пам'яті для реалізації функцій управління РОП. У роботі введено обґрунтування введення до менеджера пам'яті наступних функціональних елементів. Перший виконує збереження стану та даних програми (дамп пам'яті). Це забезпечує надійний механізм повернення програмних компонентів в будь-який момент в минулому, для якого існує дамп пам'яті. Цьому функціональному елементу поставимо у відповідність підпрограму $P_{dump}(t_q)$, що запам'ятовує дані програмі в момент часу t_q . Другий запам'ятовує історію змінних програми (журналізація). Якщо обсяг даних програми значно перевищує обсяг змінних значень, то можна запам'ятовувати адресу змінюваних даних і їх нове значення. Підпрограма, що відповідає цьому способу – $P_{change}(data^{adr,sz}, t_q)$, де атрибути adr і sz визначають адресу і розмір змінюваних даних відповідно в момент часу t_q . Для кожної з цих підпрограм є зворотні, тобто такі, які повертають програму в один зі станів в минулому: $P'_{dump}(t_q)$ і $P'_{change}(data^{adr,sz}, t_q)$. У роботі доведено ряд тверджень, та наведено наслідки з них, які дають підґрунтя для автоматичного управління станом програмних компонентів завдання. Доведено, що ці компоненти менеджера пам'яті повністю реалізують управління федератами стандарту HLA.

Отримані у другому розділі результати дають можливість уніфікації програмних розподілених систем, що веде до спрощення і прискорення процесу їх побудови на основі зручного інтерфейсу, а також створює передумови до створення ефективних моделей та методів аналізу обчислювальних процесів.

У **третьому розділі** запропоновано математичну модель розподіленого обчислювального процесу, яка враховує динамічну зміну обсягів пам'яті, потоків даних між програмними компонентами завдання, час простою, методи синхронізації часу виконання, методи забезпечення функціональної стійкості та архітектуру гетерогенної комп'ютерної системи. Параметрами цієї моделі є

$$Param = \{T_M, t_i^n\} \cup \{t^s, t^{s'}, t^m, V^{act}, V^{dm}, V^{Sm}\}, \quad (3)$$

де t^s – час збереження стану програми або підпрограми (дамп пам'яті); t^m – час передачі даних між розподіленими програмними компонентами, що знаходяться на різних обчислювальних ресурсах; $t^{s'}$ – час відновлення даних програми (відкат програми); V^{act} – обсяг пам'яті, що необхідний для зберігання виконуваної частини програми; V^{dm} – обсяг пам'яті даних програми; V^{Sm} – обсяг пам'яті зберігання дамсів стану програми.

На основі запропонованих характеристик пропонується математична модель, що описує динаміку розвитку обчислювального процесу в розподіленому гетерогенному комп'ютерному середовищі (відображає динаміку змін основних кількісних параметрів програмно-апаратної системи), що враховує параметри

програм в завданні і введені параметри за кількісною оцінкою обчислювальних ресурсів. Дана модель дає підставу для розробки імітаційних програмних моделей, які мають можливість відобразити динаміку поведінки системи у часі.

Математична модель розподіленого обчислювального процесу під час виконання завдання в гетерогенній комп'ютерній системі:

$$\left\{ \begin{array}{l} t_0^n = 0; V_0^n = \text{sizeof}(\Lambda^n \cup \Delta^n); V_0 = \sum_{n=1}^N V_0^n; T_0 = 0; TR_0 = 0; TR_0^n = 0; \\ t_{i+1}^n = \begin{cases} t_i^n + \xi_i^n, & \text{if } t_i^n = T_i; \\ t_i^n, & \text{if } t_i^n \neq T_i \end{cases}; \\ \Delta TR_i^n = \begin{cases} t_i^s + \sum_{k=1}^N a^{nk} \cdot t_i^m(\delta_i^{nk}, \text{sizeof}(\text{damp}_i^{nk})), & \text{if } t_i^n = T_i \\ \sum_{k=1}^N a^{nk} \cdot t_i^m(\delta_i^{nk}, \text{sizeof}(\text{damp}_i^{nk})), & \text{if } t_i^n \neq T_i \end{cases}; \\ TR_{i+1}^n = TR_i^n + \Delta TR_i^n; \\ TR_{i+1} = TR_i + \max_{n=1, N}(\text{time}(\varphi_i^n) + \Delta TR_i^n) + TR_i^{Sm} + \max_{n=1, N}(\text{time}(\text{Snd}_i^n)); \\ T_{i+1} = \min_{n=1, N}(t_{i+1}^n) + TR_{i,i+1}^{Sm}; \\ \Delta V_i^n = \begin{cases} V_i^{ns}, & \text{if } t_i^n = T_i; \\ 0, & \text{if } t_i^n \neq T_i \end{cases}; V_{i+1}^n = V_i^n + \Delta V_i^n + V_i^{Sm}; V_{i+1} = V_i + \sum_{n=1}^N \Delta V_i^n + V_{i,i+1}^{Sm}; \\ n = \overline{1, N} \end{array} \right. \quad (4)$$

де TR_i – реальний час, витрачений на виконання і кроки обчислювального процесу за все завдання; TR_i^n – реальний час, витрачений n -ю локальною програмою в завданні на i -му кроці моделювання на роботу з даними; t_i^s – час збереження стану програми на i -му кроці обчислювального процесу; t_i^m – час передачі даних; $\text{time}(\varphi_i^n)$ – час виконання поведінкової підпрограми φ_i^n , $\text{time}(\text{Snd}_i^n)$ – час передачі даних розподіленою системою управління; a^{nk} – ймовірність посилки повідомлення від однієї програми n до іншої k в завданні; N – кількість програм в завданні; TR_i^{Sm} – час роботи системи управління i -му кроці обчислювального процесу; t_i^n – локальний час n -ої програми в завданні на i -му кроці моделювання; T_i – глобальний модельний час на i -му кроці обчислювального процесу; ξ_i^n – значення часу, що характеризує внутрішню роботу процесів між двома сусідніми кроками обчислювального процесу; V_i^n – обсяг пам'яті, займаний даними n -ою програмою на i -му кроці обчислювального процесу; V_i^{ns} – обсяг дампа пам'яті для n -ої програми на i -му кроці обчислювального процесу; ΔV_i^n – приріст обсягу пам'яті n -ої програми на i -му кроці обчислювального процесу; V_i – обсяг пам'яті, займаної всією

розподіленою програмною системою в завданні на i -му кроці обчислювального процесу; TR_0 – початкове значення (зазвичай приймається рівним 0) реального часу обчислювального процесу, TR_0^n , t_0^n – початкове локальне час n -ої програми в завданні (може відрізнятись від TR_0^n), V_0^n – початкове значення виділеної оперативної пам'яті локальної програмою (в момент запуску виконуваного файлу), V_0 – початкове значення обсягу пам'яті, виділене всіма програмами в завданні. Аналогічні параметри з атрибутом Sm позначають використання часу або пам'яті децентралізованою системою управління на локальних ресурсах та головною програмою.

Параметри T_i , V_i , TR_i обчислюються незалежно один від одного. Отже, можлива реалізація паралельного обчислення цих параметрів, наприклад, на різних процесорах або ядрах одного процесора. Крім того, відсутність кількісних даних про деякі елементи виразу (4) (наприклад, часу, який характеризує внутрішню роботу процесів між двома сусідніми кроками синхронізації програм), спричинить за неможливість обчислення тільки одного з них.

Наведено опис модифікації моделі обчислювального процесу під час виконання завдання в розподіленій комп'ютерній системі під керуванням консервативних та оптимістичних алгоритмів синхронізації, удосконалено модель оцінки часу виконання розподіленого обчислювального процесу з урахуванням зміни обсягів пам'яті і часу простою ресурсів. Розроблено модель розподілу завдань в гетерогенній комп'ютерній системі. Розглянута множина показників для оцінки ефективності обчислювального процесу у гетерогенному обчислювальному середовищі.

У **четвертому розділі** запропоновані моделі були використані під час розробки ряду методів, спрямованих на підвищення ефективності обчислювального процесу в розподілених комп'ютерних системах. Критеріями ефективності виступають розміри оперативної і віртуальної пам'яті, що використовуються завданнями, час простою обчислювальних ресурсів, час виконання завдання, а також функціональна стійкість під час виконання завдання у гетерогенній комп'ютерній системі.

Розроблено комплекс методів оцінювання схем призначення програмних компонентів завдання за обчислювальними ресурсами. Наведемо етапи застосування комплексу методів (застосування методу оцінки завдань на основі динамічної зміни обсягів оперативної та віртуальної пам'яті – етапи 2–7; методу отримання оцінок простою ресурсів – етапи 8–16; методу оцінки просування часу виконання завдання – етапи 17–21):

Етап 1. Отримання множини схем призначення Sh_k , параметрів завдань та ресурсів (1). Паралельний запуск етапів 2, 8, 17.

Етап 2. На підставі множини схем призначення Sh_k , для кожної k -ої схеми призначення, виділяються підмножини програмних компонент sm_m^k розподілених на один ресурс R_m . Умовою залучення програми до цієї підмножини є її приналежність до m -го ресурсу.

Етап 3. Задати початкові параметри розподілених локальних програм відповідно до файлу-опису завдання Z і кроків розробленого алгоритму.

Етап 4. Провести обчислення кортежу $\langle \overline{T_j}, \overline{TR_j}, \overline{TP_j}, \overline{Vz_j}, \overline{Yz_j}, \overline{C_j} \rangle$ для всіх реалізацій розподілу програмних компонентів завдання.

Етап 5. Обчислити $V_{sm_m^k}$ відповідно до виразів $V_{sm_m} = \sum_{u=1}^{U_m} V_0^u$,
 $V_{sm_m} = \max_{i=1, I} \sum_{u=1}^{U_m} V_i^u$ або $V_{sm_m} = \sum_{u=1}^{U_m} \max_{i=1, I} V_i^u$ для всіх підмножин програм у завданні залежно від обраного алгоритму синхронізації.

Етап 6. Перевірити виконання умови $\forall sm_m^k \rightarrow VR_r, V_{sm_m^k} \leq VR_r^m$ для усіх можливих розподілів програм в завданні. Для цього, необхідно в циклі перебрати всі елементи вектора sm_m , і порівняти значення сукупного обсягу пам'яті всіх програмних модулів, розподілених на цей ресурс V_{sm_m} , з об'ємом пам'яті обчислювального ресурсу VR_m .

Етап 7. Усунути схеми розподілу, для яких умова з попереднього етапу не виконується хоча б для однієї з підмножин sm_m^k . В результаті отримуємо нову підмножину $Sh'_k = Sh_k \setminus Sh_k'', \forall Sh_k'' \rightarrow \exists (sm_m^k \rightarrow VR_r) V_{sm_m^k} > VR_r^m$. Перехід до етапу 23.

Етап 8. Поставити вихідні параметри моделі розподілених програм завдання згідно файлу опису завдання (Z) для параметрів $TR_0 = 0, T_0 = 0, TP_0 = 0, TR_0^n = 0, TP_0^n = 0, t_0^n = 0, V_0^n = 0, V_0 = 0, n = \overline{1, N}$.

Етап 9. Згідно розробленого алгоритму провести обчислення векторів $\overline{TR_i^n}$ та $\overline{TP_i^n}$ для усіх реалізацій розподіленого програмного забезпечення, що характеризують реальний час роботи ресурсів з обслуговування програм в завданні і час простою ресурсів. Значення TR_i^n та TP_i^n характеризують кінцевий стан параметрів на момент зупинки процесу розподілених обчислень.

Етап 10. Обчислити для кожного j -го ресурсу вартість використання в режимі виконання програм в завданні: $CR_j = TR_j^n \cdot c_j$, де n – номер програмного компонента, який розподілено за j -м ресурсом.

Етап 11. Обчислити для кожного j -го ресурсу вартість використання в режимі простою $CP_j = TP_j^n \cdot c_j$

Етап 12. Обчислити сумарну вартість використання j -го ресурсу $C_j = CR_j + CP_j$.

Етап 13. Обчислити сумарну вартість використання ресурсів, а також сумарні вартості використання ресурсів в режимі виконання програми і в режимі

простою: $C = \sum_{j=1}^R C_j, CR = \sum_{j=1}^R CR_j, CP = \sum_{j=1}^R CP_j$.

Етап 14. Обчислити коефіцієнти використання ресурсів як відношення часу обслуговування функцій програмних компонентів в завданні до загального часу

$$\text{обчислень: } \rho_j = \frac{TR_I^n}{TR_I^n + TP_I^n}.$$

Етап 15. Обчислити коефіцієнти простою ресурсів як відношення часу

$$\text{простою ресурсу до загального часу обчислень: } \zeta_j = \frac{TP_I^n}{TR_I^n + TP_I^n} = 1 - \rho_j.$$

Етап 16. Отриманий в результаті виконання пунктів 8-15 кортеж параметрів завданні $CC = \langle \overline{CR_j}, \overline{CP_j}, \overline{C_j}, \overline{CR}, \overline{CP}, \overline{C}, \overline{\rho_j}, \overline{\zeta_j} \rangle$ використати для оцінки якості застосування алгоритмів синхронізації розподіленої програмної системи. Перехід до етапу 23.

Етап 17. Отримати від системи розподілу ресурсів множину можливих схем призначення Sh_k програмних модулів на доступні обчислювальні ресурси. Тут можливо почати виконувати етапи методу, навіть коли множина Sh_k повністю не сформована. Аналіз починається з наявних методів (наприклад, з тих, що є у базі даних з попередніх виконань завдання). Паралельно з цим можливо виконання методів розподілу ресурсів з отриманням додаткових схем призначення. Поступово, нові схеми призначення долучаються до розгляду цього методу згідно наступних етапів.

Етап 18. Задати вихідні параметри розподілених імітаційних моделей згідно файлу опису завдання (Z) і системи виразів (4) для параметрів $TR_0 = 0, T_0 = 0, TP_0 = 0, TR_0^n = 0, TP_0^n = 0, t_0^n = 0, V_0^n = 0, V_0 = 0, n = \overline{1, N}$. Початкові значення параметрів можуть відрізнятися та формуватися в автоматичному та ручному режимах.

Етап 19. Згідно розробленого алгоритму провести обчислення елементів кортежу $\langle \overline{T_j}, \overline{TR_j}, \overline{TP_j}, \overline{Vz_j}, \overline{Yz_j}, \overline{C_j} \rangle$ для усіх реалізацій розподілених програм (з використанням консервативних і оптимістичних методів синхронізації).

Етап 20. Використовуючи метод оцінки розподілу програм у завданні на основі динамічної зміни обсягів віртуальної пам'яті вибрати схему призначення для розподілених програмних систем з оптимістичною синхронізацією.

Етап 21. Використовуючи метод оцінки простою ресурсів при виконанні програм в завданні вибрати схему призначення для програмних систем з консервативної синхронізацією.

Етап 22. Вибрати з двох схем призначення (отриманих в п. 19 і п. 20) реалізацій розподілених імітаційних моделей ту модель, для якої T_i (модельний час на останньому i -му кроці) є максимальним.

Етап 23. Передати отримані оціночні параметри підсистемі вибору схеми призначення.

Запропоновано метод вибору схеми призначення розподілу завдань за обчислювальними ресурсами згідно з оціночними параметрами, здобутими з використанням комплексу методів оцінювання схем призначення програмних компонентів завдання за обчислювальними ресурсами.

Етапи методу:

Етап 1. Завантаження вхідних даних завдань $\{Z_i, i = 1, 2, \dots, M\}$.

Етап 2. Отримання від підсистеми імітаційного моделювання оціночних значень часу закінчення виконання завдання $z_i^{run} \in \{Z_i, i = 1, 2, \dots, M\}$.

Етап 3. Завантаження списку ресурсів, які були підібрані для запуску кожного завдання $\{R_j^r, j = 1, 2, \dots, K\}$, згідно з критерієм оцінки завдання.

Етап 4. Сортування списку запущених завдань z_i^{run} відповідно до очікуваного часу закінчення rt , $\{Z / \text{sort}(rt_i, \forall i = 1, \dots, M)\}$.

Етап 5. Час виконання завдань зі списку t розбивається на періоди, відповідно часу закінчення завдань τ . Для кожного періоду записується кількість обчислювальних ресурсів, які використовуються в цьому періоді, $t \subset \tau, \forall \tau := pr / R_j, \forall j = 1 \dots N$. В результаті отримуємо профіль використання обчислювальних ресурсів $Prof \subset \{\tau\}$.

Етап 6. Завдання з черги упорядковано відповідно до критерію зв'язності.

Етап 7. За множиною $\{R_j^r, j = 1, 2, \dots, K\}$ формується список вікон, які можуть бути використані для запуску завдання – W_i^{run} _{$i \in 1..M$} .

Етап 8. Для кожного отриманого вікна формується перелік доступних обчислювальних ресурсів (CPU): для кожної пари (CPU-брокер) визначається затримка часу передачі пакету даних каналом (δ_n); для кожної пари (CPU – брокер) визначається пропускна здатність каналу зв'язку Y_n .

Етап 9. Для запуску задач із завдання $a, z_a \in Z_i$ набирається необхідна кількість CPU, мінімізуючи один з параметрів: якщо завдання має великий обсяг переданих вхідних даних і відповідно великий обсяг вихідних даних, то підбираються маршрути передачі даних з найбільшими значеннями пропускної здатності – Y_n ; якщо для запуску завдання не потрібна передача великого обсягу даних, то підбираються маршрути передачі даних з найменшим значенням δ_n .

Етап 10. За кожним отриманим вікном підсумовуються значення попарних відстаней і визначається найменше з них – $D_{min} = \min_{i \in 1..M} D_{W_i^{run}}$ – це і є найкращий результат розподілу.

Після того, як завдання отримало вікно на виконання, відбувається оновлення профілю $Prof \subset \{\tau\}$ з урахуванням того, що ці процесори будуть задіяні.

Удосконалено метод розподілу пулу завдань за обчислювальними ресурсами. Ефективність вибору методу розподілу з мінімальним часом виконання пулу завдань та мінімальним простим обчислювальних ресурсів гетерогенної комп'ютерної системи визначається системою критеріїв і правил їх порівняння. Вибір плану розподілу заснований на аналізі, реалізованих в середовищі моделювання методів розподілу. Аналіз побудований на порівнянні значень запропонованих критеріїв за кожним з методів. Метод, який дає кращий результат за сукупністю всіх критеріїв, буде обраний результуючим для розглянутого пулу завдань.

Удосконалення методу полягає у такому. Спочатку отримують множину планів розподілу (Sh_k) як результат виконання одного з методів розподілу Q_k . Далі отримують множину часів виконання планів (T_k) за кожним методом розподілу ($mn \in Q$). Після цього визначають відсоток простою обчислювальних ресурсів і середній час очікування завдання в черзі за кожним із запропонованих методів розподілу. Згідно з стратегією управління обирають оціночну характеристику та знаходять план розподілу з множини Sh_k за оптимальним (з наявних) значенням параметру оцінки.

Розроблена модифікація методу повного перебору під час розподілу завдань на обчислювальні ресурси. Метод перебору має найдовший час виконання, але він гарантовано дає мінімальне значення коефіцієнта використання обчислювальної системи. При цьому кількість ітерацій алгоритму фіксована й описується таким виразом: $O=M^N$, де M – кількість програм в завданні, N – кількість робочих станцій. Виходячи з цього виразу, кількість ітерацій має степеневу складність і при великій кількості завдань має малу ефективність. Етапи методу:

Етап 1. Обчислити коефіцієнт використання обчислювальної системи K для поточного розподілу робіт.

Етап 2. Порівняти цей коефіцієнт з найбільш придатним коефіцієнтом, обчисленим на попередніх ітераціях, заповнити найбільш новий придатний коефіцієнт.

Етап 3. Обчислити наступну комбінацію розподілу робіт за робочими станціями.

Етап 4. Якщо всі комбінації проаналізовані – «повернути» результат, інакше повернутися до пункту 1.

Пропонується використовувати модифікований метод перебору, який дозволяє отримати такий же результат, як і попередній, але з меншими часовими витратами. У цьому алгоритмі з розгляду відкидається велика кількість комбінацій, які, як відомо з досвіду, є неефективними. Розподіл програм в завданні вважається явно неефективним, якщо $K \geq M \cdot \sum_j PT_j$, де j аналізується

на множині нерозподілених програм в завданні. Під час обчислення K алгоритм аналізує завдання від останнього до першого і поетапно обчислює навантаження на кожну робочу станцію, підсумовуючи значення обчислювальної складності процесів, які присвоєні цій робочій станції. Якщо це навантаження перевищує значення $M \cdot \sum_j PT_j$, ітерація припиняється, всім наступним завданням

присвоюється WS_0 (статус нерозподіленого завдання), а поточне аналізоване завдання відправляється наступній робочій станції.

Таким чином, відкидається M^j свідомо неефективних комбінацій, де j – індекс поточного аналізованого завдання.

Ще одна модифікація методу дає можливість отримати результат за одну ітерацію, при цьому значення K не буде найоптимальнішим, але є прийнятним. Цей метод поетапно заповнює обчислювальні ресурси, в першу чергу, розподіляючи найскладніші завдання, а потім – більш прості.

Етапи методу такі:

Етап 1. Відсортувати робочі станції за мірою зменшення продуктивності процесорів.

Етап 2. Відсортувати завдання за мірою зменшення трудомісткості програмних компонент в завданні.

Етап 3. Вибрати робочу станцію. Поетапно, в порядку спадання трудомісткості вибирати завдання доти, доки навантаження на робочу станцію не перевищить обчислене раніше середнє значення навантаження.

Етап 4. Перейти до наступної робочої станції і повторити п. 3.

Етап 5. Якщо всі завдання розподілені за робочими станціями, здійснити вихід з методу.

Етап 6. На поточній стадії розподіл будь-якого завдання на довільну робочу станцію призведе до перевищення середнього значення навантаження на неї. Тому необхідно відсортувати робочі станції в порядку зменшення поточного значення навантаження.

Етап 7. Розподілити задачі, що залишилися у завданнях за робочими станціями: на менш завантажену станцію – більш складне завдання.

Етап 8. Результат, отриманий на виході методу може бути використано брокером ресурсів або переданим до аналізатору схем призначення.

Запропоновано ряд методів, які забезпечують функціональну стійкість розподіленого обчислювального процесу.

Перший метод розроблено на основі дамів пам'яті програмних компонент. Етапи метода:

Етап 1. Реалізація схеми призначення (2). Пересилання згідно схеми призначення програмних елементів на виділені ресурси. Запуск програмних компонент, збереження початкового стану розподіленої програмної системи шляхом виконання $P_{damp\ i}(t_0)$ для усіх N програм ($q=0$).

Етап 2. Визначення підмножини вільних ресурсів після реалізації схеми призначення (пункт 1):

$$R^e = \bigcup_{m=1}^M R_m \setminus \bigcup_{i=1}^N R_r \mid r = j, \forall \{ P_i \rightarrow R_j \} \quad (5)$$

У разі, коли кількість обчислювальних ресурсів менша або дорівнює кількості програм в завданні, (5) може дати результат $R^e = \emptyset$. В цьому випадку

можна прийняти $R^e = \bigcup_{m=1}^M R_m$. Пропонується також модифікація методу, в якому

підмножина упорядковується на основі завантаженості обчислювальних ресурсів.

Етап 3. Визначити значення часу $t_q, q=q+1$, для якого буде виконаний дамп пам'яті програмних компонентів. Перевірити умову закінчення обчислювального процесу. Якщо умова виконана, перехід до пункту 8.

Етап 4. При досягненні часу t_q виконати всі підпрограми множини

$\bigcup_{i=1}^N P_{damp\ i}(t_q)$. Якщо всі підпрограми виконалися, переходимо до пункту 3.

Якщо якась із підпрограм не виконалась (відповідь від неї не одержано), переходимо до пункту 5.

Етап 5. Визначаємо підмножину програмних компонентів завдання, від яких не отримано відповіді (не виконано дампа) $P^{Err} \subset P$.

Етап 6. Для підмножини P^{Err} виконати перерозподіл ресурсів з отриманням нової схеми призначення: $Sh^{Err} = \{ P^{Err} \rightarrow R^e \}$.

Етап 7. Пересилання згідно схеми призначення програмних елементів на виділені ресурси Sh^{Err} . Виконати запуск програмних компонентів, встановити свій статус розподіленої програмної системи шляхом виконання $P'_{damp\ i}(t_q)$ для всіх програм ($q=0$) з множини P^{Err} . Перехід до пункту 5 даного методу.

Етап 8. Завершення обчислювального процесу, закриття служб підтримки функціональної стійкості.

Цей метод гарантує повне відновлення функціональності програмного забезпечення, але потребує час для здійснення перезавантаження та пошук і відновлення даних пам'яті програмних компонентів за допомогою дамів пам'яті. У цьому випадку система простоює.

Розроблено метод забезпечення функціональної стійкості розподіленого обчислювального процесу на основі протоколів історії змінних програми.

Етап 1. Реалізація схеми призначення. Пересилання згідно схеми призначення програмних елементів на виділені ресурси. Запуск програмних компонентів. Оскільки пам'ять запущеної програми завжди перебуває в одному і тому ж стані, збереження даних програми на цьому етапі не потрібно.

Етап 2. Визначення підмножини вільних ресурсів R^e після реалізації схеми призначення. Для цього етапу характерні зауваження, подані в пункті 2 методу забезпечення функціональної стійкості на основі дампа пам'яті програмних компонентів.

Етап 3. Ініціалізувати обчислювальний процес, запустивши всі програми множини P . Запустити менеджер пам'яті, перевести його в стан очікування змін даних програм. Запустити системний таймер для контролю часу t_q і синхронізувати його на всіх обчислювальних ресурсах, встановивши початкове значення t_0 .

Етап 4. Одночасно запустити сервіси системи управління обчислювальним процесом, описані в пунктах 5 та 6.

Етап 5. При виникненні змін змінних i -ої програми запустити відповідну підпрограму $P_{change,i}(data^{adr,sz}, t_q)$, яка запам'ятає зміну змінних у базі даних з атрибутом часу, у якій ця зміна відбулась. Продовжити виконання сервісу пункту 5.

Етап 6. Визначити значення часу t_q , $q=q+1$, для якого буде виконаний контроль доступності обчислювальних ресурсів. Перевірити умову закінчення обчислювального процесу. Якщо умова виконана, перехід до пункту 11.

Етап 7. При досягненні часу t_q виконати контроль працездатності всіх ресурсів (або програм). Якщо всі програми знаходяться в працездатному стані (отриманий відгук), переходимо до пункту 6. Якщо від будь-якого з ресурсів (програм) ми отримали відгук, переходимо до пункту 8.

Етап 8. Визначаємо підмножину програмних компонентів завдання, від яких не отримано відповіді (не виконано дамп) $P^{Err} \subset P$.

Етап 9. Для підмножини P^{Err} виконати перерозподіл ресурсів з отриманням нової схеми призначення: $Sh^{Err} = \{ P^{Err} \rightarrow R^e \}$.

Етап 10. Пересилання згідно схеми призначення програмних елементів на виділені ресурси Sh^{Err} . Запуск програмних компонентів, встановити свій статус розподіленої програмної системи шляхом виконання ланцюга підпрограм $P'_{change,i}(data^{adr,sz} t_q)$, що повторює зміни даних програми, для всіх програм з множини P^{Err} . Перехід до пункту 6 цього методу.

Етап 11. Завершення обчислювального процесу, закриття служб підтримки функціональної стійкості.

Зазначимо, що другий метод за значного часу безперервної роботи ланцюга підпрограм P'_{change} , які повинні викликатися для відновлення стану програм, що відмовили, може бути тривалим, тому для завдань, які виконуються довго, бажано періодично виконувати дамп пам'яті, який дозволить здійснити збереження проміжних станів завдання. В цьому випадку, пункт 10 методу забезпечення функціональної стійкості на основі історії зміни змінних програми може бути модифікований поверненням програми в один з найближчих станів в минулому шляхом виклику відповідної підпрограми виконання $P'_{damp i}(t_{q'})$, де $q' < q$, з подальшим викликом збереженого ланцюга змін пам'яті $P'_{change,i}(data^{adr,sz}, t_{q'})$ для всіх записів з тимчасовими атрибутами $q'' \in [q', q]$ в порядку зростання тимчасового атрибута.

Для забезпечення процесів, які мають значні вимоги щодо забезпечення функціональної стійкості, пропонується мажоритарний метод підтримки функціональної стійкості розподіленого обчислювального процесу, що містить наступні етапи:

Етап 1. Розширення множини P за рахунок дублювання елементів (програм). Кількість копій програм залежить від необхідного ступеня функціональної стійкості розподіленого обчислювального процесу $P^{ext} = \{ P \cup P \cup \dots \}$. Рекомендований мінімум копій програми – 3. У цьому випадку полегшено визначення однократної відмови локальної програми або обчислювального ресурсу. Для цього достатньо порівняти результати трьох паралельних програм і знайти ту з них, результат виконання якої не співпадає з іншими двома.

Етап 2. Реалізація схеми призначення для множини P^{ext} :

$$Sh_k = \{ P^{ext}_i \rightarrow R_j \}, \left| \bigcup_{i=const} \{ P^{ext}_i \rightarrow R_j \} \cap \bigcup_{j=const} \{ P^{ext}_i \rightarrow R_j \} \right| = 1, \quad (6)$$

при цьому додаткова умова говорить про те, що не можна кілька копій програми розташовувати на одному обчислювальному ресурсі, оскільки в разі його відмови, будуть втрачені відразу дві мажоритарні програми.

Етап 3. Пересилання згідно схеми призначення програмних елементів на виділені ресурси. Запуск програмних компонентів, збереження початкового

стану розподіленої програмної системи шляхом виконання $P_{damp\ i}(t_0)$ для всіх N програм ($q=0$).

Етап 4. Визначення підмножини вільних ресурсів R^e після реалізації схеми призначення (вираз 9). Для цього етапу характерні зауваження, дані в пункті 2 методу забезпечення функціональної стійкості на основі дампа пам'яті програмних компонент.

Етап 5. Визначення значення часу t_q , $q=q+1$, для якого буде виконаний дамп пам'яті програмних компонент. Перевірити умову закінчення обчислювального процесу. Якщо умова виконана, перехід до пункту 11.

Етап 6. При досягненні часу t_q виконання всіх підпрограми множини $\bigcup_{i=1}^N P_{dampi}^{ext}(t_q)$.

Етап 7. Перевіряємо умови відмови програми (програм) із завдання на основі двох умов:

1) якщо якась із підпрограм в множині не виконана (відповідь від неї не одержано);

2) дампи пам'яті однієї з однакових програм $P_i^l, P_i^l, P_i^l, \dots$ не збігається з іншими.

Етап 8. Визначення підмножину програмних компонентів завдання, від яких не отримано відповіді (не виконано дампи) або відповідь не співпадає з відповідями аналогічних програм $P^{Err} \subset P$.

Етап 9. Для підмножини P^{Err} виконати перерозподіл ресурсів з отриманням нової схеми призначення: $Sh^{Err} = \{ P^{Err} \rightarrow R^e \}$ з дотриманням обмежень.

Етап 10. Пересилання згідно схеми призначення програмних елементів на виділені ресурси Sh^{Err} . Запуск програмних компонентів, встановлення свого статусу розподіленої програмної системи шляхом виконання $P'_{damp\ i}(t_q)$ для всіх програм з множини P^{Err} . Перехід до пункту 4 цього методу.

Етап 11. Завершення обчислювального процесу, закриття служб підтримки функціональної стійкості.

Слід зазначити, що отримані результати можуть бути використані в ряді інших методів управління обчислювальним процесом в комп'ютерних системах для різних алгоритмів синхронізації. Крім цього, вони можуть використовуватися в методах статичного і динамічного розподілу ресурсів.

У **п'ятому розділі** запропонована інформаційна технологія управління обчислювальним процесом в гетерогенних комп'ютерних системах (рис. 2). Розглянемо основні етапи технології, серед яких виділімо паралельні та послідовні етапи.

Етап 1: отримання завдання на виконання від споживача у вигляді сукупності конфігураційних, виконуваних, інформаційних файлів та дескрипторів файлів-результатів:

$$n_i = \{ Z^i, \bigcup_{j=1}^N Pr_j^i, D_j^i, W^i \}. \quad (7)$$

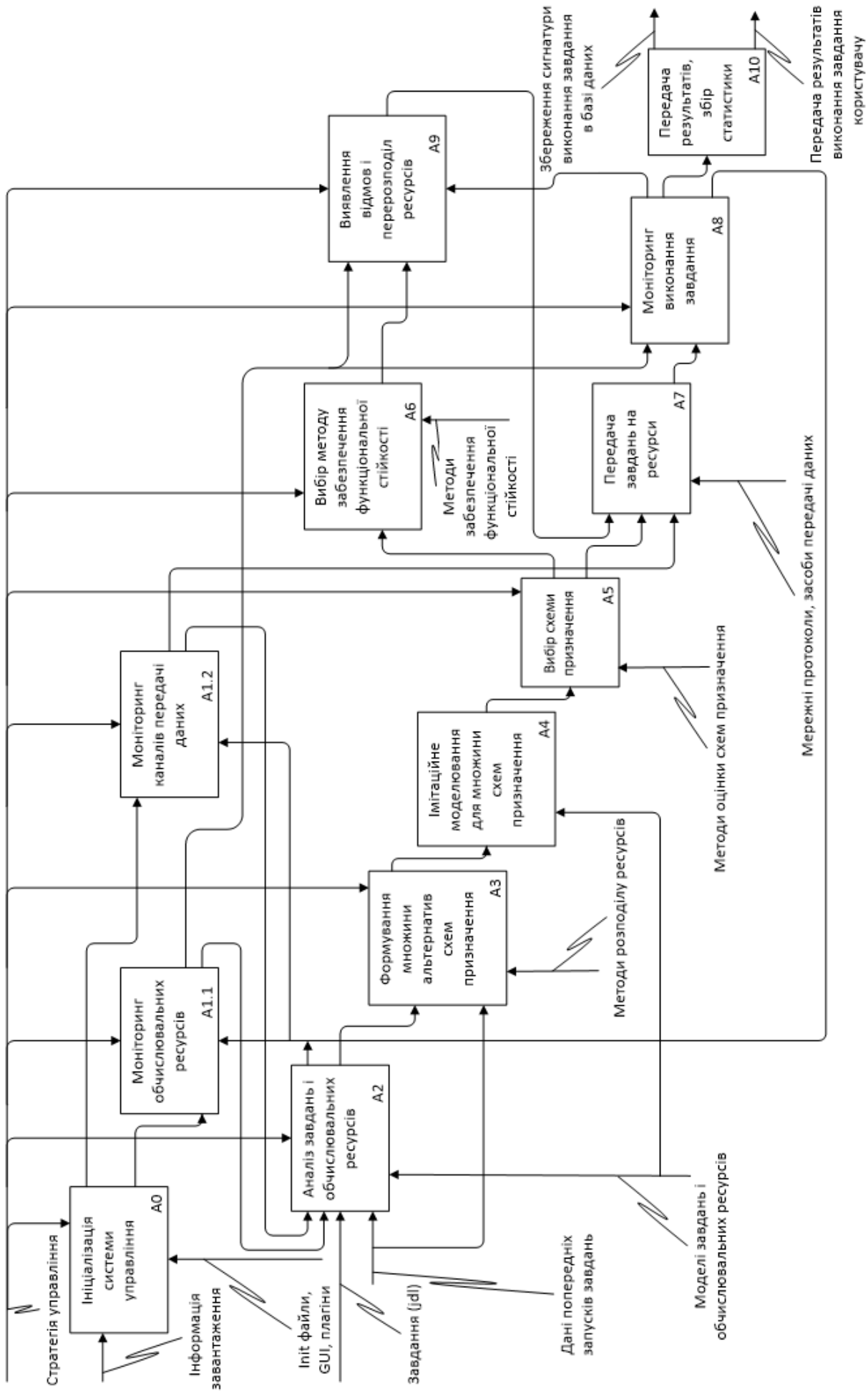


Рисунок 2 – Інформаційна технологія управління розподілим обчислювальним процесом

Запуск етапів 2 та 3 проводять паралельно.

Етап 2: на основі конфігураційного файлу завдання з множини Z проходять ініціалізацію апріорні дані програмних компонент і завдання розміщується у черзі очікування розподілу за обчислювальними ресурсами до формування пулу завдань. Далі перехід до етапу 5.

Етап 3: перевірка наявності в базі даних інформації про попередні запуски завдання (згідно атрибутам Z) на наявні або аналогічні обчислювальні ресурси, отриманої від підсистеми аналізу завдання і зібраної в результаті моніторингу попередніх процесів імітаційного моделювання. Якщо дані наявні в базі даних, то відбувається завантаження інформації з бази даних і перехід до етапу 5, інакше – перехід до етапу 4.

Етап 4: запуск програмних моделей, що моделюють обчислювальні процеси, дослідження поведінки програм в завданні з різними методами синхронізації (консервативними і оптимістичними).

Етап 5: застосування стандартних методів розподілу ресурсів з метою отримання множини допустимих схем розподілу ресурсів. Вирішується задача розподілу програмних моделей Pr_j^i на доступні обчислювальні ресурси.

Результатом виконання етапу є множина доступних схем призначення, де кожному програмному компоненту ставиться у відповідність обчислювальний ресурс $P = \bigcup_k \left\{ Pr_j^i \rightarrow R_m, \forall j = \overline{1, N} \right\}_k$ (k – кількість допустимих схем призначення).

Ефективність розподілу визначається мінімізацією часу імітації або вартістю використання ресурсів.

Етап 6: застосування методів оцінки завдання. У роботі розглянуто комплекс методів, який містить три методи оцінки: метод порівняння ефективності розподілених за максимальним просуванням модельного часу, метод оцінки можливості розподілу програмних компонентів на основі динамічної зміни обсягів віртуальної пам'яті, метод отримання простою ресурсів під час використання різних методів синхронізації розподілених програмних компонентів. Крім того, існує можливість використання нових методів, що підключаються паралельно та виконують оцінку додаткових параметрів завдань та ресурсів.

Метою виконання етапу є знаходження схеми призначення, що має мінімальний час виконання з обмеженнями на обсяги пам'яті, займані програмними компонентами, час простою обчислювальних ресурсів, з урахуванням різних методів синхронізації. Така схема призначення визначається за допомогою методу вибору. В кінці етапу проводиться збереження інформації, отриманої в результаті застосування методів оцінки схем розподілу, в базі даних.

Етап 7: виклик сервісів передачі даних брокера ресурсів, що реалізують фізичний розподіл локальних програмних компонентів на обчислювальні ресурси відповідно до обраної схеми розподілу.

Етап 8: використання засобів моніторингу системи управління РОП і операційної системи для отримання динамічних параметрів процесу імітації та збереження їх в базі даних для повторного використання. Збереження і встановлення зв'язків в базі даних між апріорними параметрами експерименту,

з одного боку, і отриманими в результаті застосування технології аналізу і збору статистичних даних від систем моніторингу середовища виконання програмних модулів, з іншого боку.

Етап 9: передача результатів виконання завдання користувачеві у вигляді множини файлів-результатів *W*. Аналіз інформації, отриманої в результаті обчислювального процесу (паралельного виконання програмних компонентів) проводиться користувачем і не входить до переліку питань, розглянутих в цій роботі.

В основі багатьох процесів в управлінні обчислювальним процесом значну роль відіграє система імітаційного моделювання, опис та особливості інтегрування якої наведено у розділі.

Практична реалізація інформаційної технології та експериментальні дослідження наведені у **шостому розділі**. Розглянуто питання обґрунтування вимог та особливості реалізації програмних засобів підтримки обчислювального процесу в гетерогенних комп'ютерних системах, особливості програмної реалізації конфігураційних файлів в середовищі моделювання GRASS, програмної реалізації інформаційної технології розподілу завдань за обчислювальними ресурсами, результати експериментальних досліджень розробленої системи.

На рис. 3 представлено результати моніторингу оціночних параметрів РОП для різних методів розподілення ресурсів.

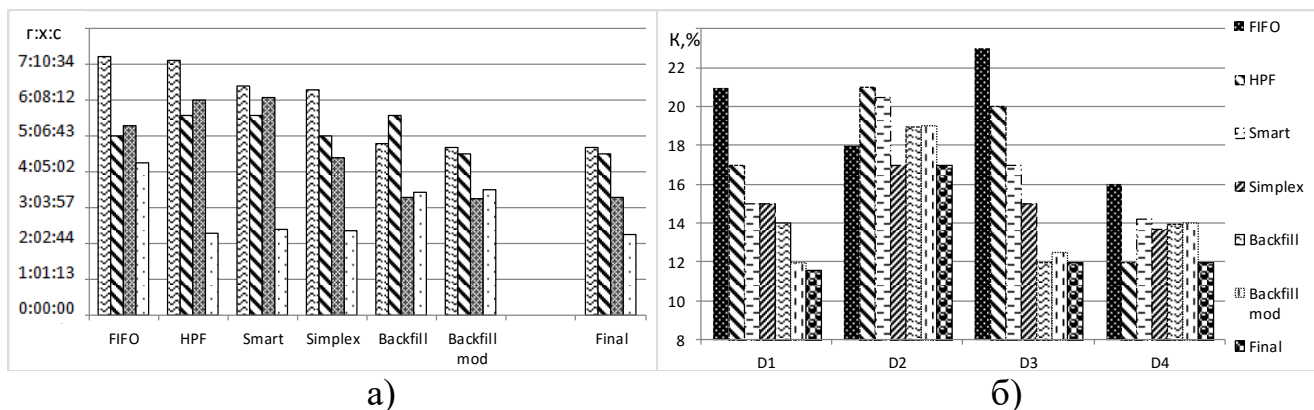


Рисунок 3 – Результати моніторингу оціночних параметрів РОП:
а) час виконання завдань; б) сукупний критерій ефективності

Наведено результати вибору системи управління (Final) для чотирьох типів завдань, які відрізняються взаємовідношенням обчислювальної складності та зв'язністю програмних компонентів в завданні (D1 – мала зв'язність, велика обчислювальна складність; D2 – велика зв'язність, велика обчислювальна складність; D3 – мала зв'язність, велика обчислювальна складність; D4 – мала зв'язність, мала обчислювальна складність). Плани розподілу завдань за ресурсами паралельно отримані з використанням наступних методів розподілу: FIFO (first in, first out), HPF (highest priority first), Smart (перший на вільний ресурс, що відповідає обмеженням), Simplex (симплекс-метод), стандартний та розроблений модифікований метод Backfill. Результати свідчать, що система керування у всіх випадках обирала найліпший план розподілу ресурсів з мінімальним часом виконання (рисунок 3.а) та сукупним критерієм ефективності

K (рисунок 3.б), який враховує різницю часу виконання завдання, часу простою ресурсів та затрати віртуальної пам'яті ($K = a_1 \Delta TR + a_2 \Delta TP + a_2 \Delta V$, де $\Delta TR, \Delta TP, \Delta V$ – процент перевищення мінімальних значень часу виконання, часу простою та обсягу пам'яті відповідно для обраного методу розподілу, a_i – ваговий коефіцієнт, який залежить від обраної стратегії управління; у даному випадку вагові коефіцієнти рівні). Проведені експерименти з використання запропонованої інформаційної технології на наявних комп'ютерних ресурсах показали підвищення ефективності РОП згідно з обраними критеріями та обмеженнями (для відповідних класів завдань: D1 – до 60%, D2 – до 12%, D3 – до 74%, D4 – до 40%) у порівнянні з виконанням тих же завдань в існуючих кластерних системах та системах хмарних обчислень відкритого доступу.

У додатках наведено акти про впровадження результатів дисертаційної роботи, графічний матеріал, конфігураційні файли системи управління розподіленими обчисленнями, список публікацій.

ВИСНОВКИ

У дисертаційній роботі вирішено актуальну науково-прикладну проблему розробки моделей, методів та інформаційної технології управління розподіленим обчислювальним процесом в гетерогенних комп'ютерних системах.

В результаті проведених досліджень отримано такі наукові і практичні результати:

1. Виконано аналіз існуючих підходів до управління розподіленим обчислювальним процесом в гетерогенних комп'ютерних системах, технологій розподілених обчислень, хмарних технологій, існуючих планувальників завдань, мов опису завдань, існуючого програмного забезпечення підтримки розподілених обчислень, методів та засобів імітаційного моделювання як інструменту дослідження обчислювальних процесів у комп'ютерних системах, методів розподілу завдань. Наведені формалізації опису завдань та обчислювальних ресурсів, задачі розподілу завдань за обчислювальними ресурсами, процесу забезпечення функціональної стійкості РОП. Введена концепція використання сервісу менеджера пам'яті для реалізації функцій управління РОП Розроблені концептуальні основи побудови системи управління РОП, що базуються на комплексі методів аналізу завдань та обчислювальних ресурсів, імітаційному моделюванні обчислювальних процесів, комплекси методів підтримки функціональної стійкості РОП.

2. Набула подальшого розвитку методологія управління обчислювальним процесом в розподілених комп'ютерних системах, яка базується на системних принципах гетерогенності програмних завдань та ресурсів, вибору стратегії управління, вибору методів розподілення ресурсів, зв'язності програм та урахування трафіка у завданні, вибору методів синхронізації, функціональної стійкості та моделювання розподіленого обчислювального процесу, що забезпечує ефективне використання обчислювальних ресурсів та знижує час виконання завдань.

3. Розроблено теоретико-множинну модель розподіленого обчислювального середовища під час виконання завдання в розподілених гетерогенних комп'ютерних

системах, яка уніфікує опис обчислювального завдання, комп'ютерних ресурсів та системи управління розподіленими обчисленнями, що дозволяє формалізувати процес розподілених обчислень та реалізувати комплексний підхід до підвищення ефективності розподіленого обчислювального процесу.

4. Розроблено математичну модель розподіленого обчислювального процесу, яка враховує динамічну зміну обсягів пам'яті, потоків даних між програмними компонентами завдання, час простою, методи синхронізації часу виконання, та використовує архітектуру гетерогенної комп'ютерної системи, що дозволяє забезпечити функціональну стійкість, задовольнити встановлені обмеження на обсяги оперативної пам'яті, час використання процесора, та знизити час простою обчислювальних ресурсів.

5. Розроблено комплекс методів для оцінювання та вибору призначень програмних компонент завдання за обчислювальними ресурсами з урахуванням кількісних оцінок розподіленого обчислювального процесу, що дозволяє використати призначення з найменшим часом виконання завдання та простим обчислювальних ресурсів під час реалізації обраної стратегії управління.

6. Розроблено методи забезпечення функціональної стійкості розподілених програмних завдань, які регулярно виконують моніторинг і зберігання стану програмних компонентів або історії їх змін у часі, що дозволяє автоматизувати процес відновлення працездатності розподіленого програмного забезпечення після відмови обчислювального ресурсу або комунікаційного каналу зв'язку.

7. Удосконалено метод оцінки часу виконання програмних завдань в гетерогенних комп'ютерних системах, що, на відміну від існуючих, враховує обсяг пам'яті програмних модулів, обсяги даних, які передаються між програмними компонентами, час простою ресурсів, час та обсяги пам'яті щодо забезпечення функціональної стійкості, що дозволяє оцінити час виконання програмних завдань при розподілі ресурсів з урахуванням обмежень на обсяг пам'яті, сумарний час простою обчислювальних ресурсів для різних методів синхронізації завдань.

8. Удосконалено метод розподілу завдань за обчислювальними ресурсами в пакетному режимі, який на відміну від існуючих методів, враховує параметри трафіку стосовно завантаження завдання та трафіку між програмними компонентами завдання, що дозволяє підвищити ефективність використання гетерогенних комп'ютерних систем за рахунок зменшення часу виконання пулу завдань.

9. Набули подальшого розвитку методи розподілу програмних завдань за обчислювальними ресурсами, які, на відміну від існуючих, передбачають розподіл ресурсів з максимальним просуванням часу виконання в умовах обмежень на обсяг оперативної пам'яті та часу простою, вибір найбільш оптимальних методів синхронізації та забезпечення функціональної стійкості програмних компонентів завдання, що дозволяє зменшити час виконання, обсяги пам'яті, трафік обслуговування завдання та кількість обчислювальних ресурсів і тим самим підвищує ефективність процесу розподілених обчислень.

10. Набула подальшого розвитку модель подання завдань і ресурсів у гетерогенних комп'ютерних системах, яка, на відміну від існуючих, враховує методи розподілу завдань за ресурсами та обсяги мережного трафіку в процесі

завантаження і виконання завдань, що дозволяє скоротити час виконання завдань та зменшує навантаження на мережі передачі даних. Використання у моделях сумарної затримки часу передачі пакетів даних і пропускної здатності каналу зв'язку дозволяють скоротити час виконання пулу завдань, що підвищує ефективність використання обчислювальних ресурсів в гетерогенних комп'ютерних системах.

11. Розроблено інформаційну технологію управління розподіленим обчислювальним процесом в гетерогенних комп'ютерних системах, програмні засоби системи управління розподіленим обчислювальним процесом. Експериментальні дослідження показують ефективність запропонованих рішень: зменшення часу виконання завдань від 7% до 52%, і підвищення ефективності використання обчислювальних ресурсів у середньому від 12% до 74% в залежності від типу завдань, наявних обчислювальних ресурсів та стратегії управління.

12. Практичні результати, що отримано, підтверджені актами впровадження та доводять коректність теоретичних положень дисертаційної роботи, високу ефективність розроблених моделей, методів та інформаційної технології, Результати роботи впроваджені: в Державному підприємстві Науково-дослідному технологічному інституті приладобудування, в Державному підприємстві «Центральне конструкторське бюро «ПРОТОН», в товаристві з обмеженою відповідальністю «Альтер системс», в освітній процес Харківського національного університету радіоелектроніки.

Подальші дослідження методології, моделей, методів і технологій управління розподіленим обчислювальним процесом обумовлені постійним розвитком та появою нових технологій розподілених обчислень, наприклад, таких як «туманні обчислення» (Fog Computing), Dew Computing, питаннями, пов'язаними з використанням в обчислювальних процесах розподілених баз даних (для розміщення даних завдань та даних системи управління), комп'ютерних мереж, спецпроцесорів та ін.

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. Filimonchuk T., Volk M., Ruban I., Tkachov V. Development of information technology of tasks distribution for grid-systems using the GRASS simulation environment. Eastern-European Journal of Enterprise Technologies. Information and controlling system, 2016. Vol.3/9 (81). P. 45–53. (Входить до міжнародної наукометричної бази Scopus).

2. Волк М.А., Гридель Р.Н., Филимончук Т.В., Ал Шиблак Муаз. Формализация процессов распределенной имитации информационных систем. Системы обработки информации. Харків: ХУПС, 2013. №4(111). С. 89–93. (Входить до міжнародних наукометричних баз BASE, WorldCat).

3. Волк М.О., Гридель Р.М., Саранча С.Н., Гавриш Д.А. Алгоритмічна модель процесу розподіленої імітації для технології аналізу розподілених імітаційних моделей. Східно-Європейський журнал передових технологій. Харків, 2013. №. 2(66). С. 32–36. (Входить до міжнародних наукометричних баз BASE, Index Copernicus).

4. Волк М.А., Филимончук Т.В. Разработка модифицированного метода обратного заполнения Backfill для консервативного резервирования. Системы обработки інформації. Харків: ХУПС, 2017. №1(147). С. 33–37. (Входит до міжнародної наукометричної бази BASE).

5. Гавриш Д.А., Саранча С.Н., Волк М.А. Метод распределения задач с учетом затрат синхронизации при параллельном моделировании сложных цифровых систем в гетерогенной вычислительной среде. Системы обработки інформації, 2015. Вип. 5 (130). С. 122–128. (Входит до міжнародної наукометричної бази BASE).

6. Волк М.А. Анализ распределенных имитационных моделей в гетерогенных вычислительных системах. Науковий вісник Чернівецького національного університету імені Юрія Федьковича. Серія: Комп'ютерні системи та компоненти. Том 1, випуск 2. Чернівці: ЧНУ, 2010. С.35–39.

7. Волк М.О., Грідель Р.М., Муаз Ал Шиблак. Анализ распределенных имитационных моделей с оптимистическими алгоритмами. Системы обработки інформації. Харків, 2013. Вип. 1(108). С.35–40. (Входит до міжнародних наукометричних баз BASE, WorldCat).

8. Волк М.А., Филимончук Т.В., Ал Шиблак Муаз. Анализ современного состояния и развития GRID-технологий и языков описания заданий. Збірник наукових праць ХУПС. Харків: ХУПС, 2013. №2 (35). С. 75–81. (Входит до міжнародних наукометричних баз BASE, WorldCat).

9. Волк М.А. Процессное представление состояний распределенных имитационных моделей с учетом специфики их программной реализации. Вісник Національного технічного університету «Харківський політехнічний інститут». Збірник наукових праць. Тематичний випуск: Інформатика і моделювання. Харків: НТУ «ХПІ», 2009. №13. С. 23–33.

10. Волк М.А. Структура программного комплекса имитационного моделирования элементов GRID-систем для научных исследований. Системы обработки інформації. 2009. Вип. 3(77). С.125–128. (Входит до міжнародних наукометричних баз BASE, WorldCat).

11. Волк М.А. Журнализация состояний программных распределенных мооделей и ее использование в оптимистических алгоритмах синхронизации. Збірник наукових праць Харківського університету Повітряних Сил. 2010, випуск 1 (23). С.104–107. (Входит до міжнародних наукометричних баз BASE, WorldCat).

12. Кобылин О.А., Волк М.А., Саранча С.Н. Формальный аппарат построения параллельных алгоритмов распознавания объектов распределенных баз данных. Проблемы бионики, 2000. №53. С. 36–38.

13. Волк М.А. Использование технологии .NET для построения распределенных имитационных моделей. Автоматизированные системы управления и приборы автоматики, 2002. Вып.121. С. 57–60.

14. Волк М.А., Кожушко Д.Г. Оценка эффективности распараллеливания вычислительных задач в локальных компьютерных сетях. Автоматизированные системы управления и приборы автоматики, 2004. №127. С. 149–154.

15. Волк М.А. Структурная организация поведенческого имитационного моделирования в grid. Системи обробки інформації. 2007. Вип. 9(67). С. 41–45. (Входить до міжнародних наукометричних баз BASE, WorldCat).
16. Волк М.А., Остроушко А.П., Колыбельникова Н.И. Имитационное моделирование методом просмотра активностей для архитектуры Multi-core. Автоматизированные системы управления и приборы автоматики. 2008. Вып. 143. С. 73–80.
17. Волк М.А., Бугрий А.Н., Скоромный А.Л. Алгоритм и программа расчета и визуализации двухмерного температурного поля вертикального плитового холодильника. Электронное моделирование. 2009. Т. 31. № 3. С. 65–72.
18. Волк М.А., Филимончук Т.В., Гридель Р.Н. Методы распределения ресурсов для GRID-систем. Збірник наукових праць ХУПС. Харків: ХУПС, 2009. №1(19). С. 100–104. (Входить до міжнародних наукометричних баз BASE, WorldCat).
19. Олищук С.О., Волк М.А. The Process Algebra Usage for Simulation Purpose. Radioelectronics & Informatics. 2009. №2(45). Р. 68–70.
20. Волк М.А., Дема К.В., Зозуля В.В. Архитектура модулей генерации потоков задач и очереди в распределенной системе имитационного моделирования GRID. Системи обробки інформації. 2010. Вип. 2(83). С.27–32 (Входить до міжнародних наукометричних баз BASE, WorldCat).
21. Руденко О.Г., Волк М.А., Филимончук М.А., Филимончук Т.В. Архітектура системи моніторингу трафіку в GRID. Право і безпека. Харків: ХНУВС, 2010. №1(33). С. 226–229.
22. Волк М.О., Горенков А.С., Гридель Р.М. Архитектура имитационной модели GRID-системы, основанная на подключаемых модулях. Системи обробки інформації. Харків, 2010. Вип. 1(82). С.17–20.
23. Волк М.А. Парадокс момента модельного времени в распределенных поведенческих имитационных моделях. Вісник Національного технічного університету «Харківський політехнічний інститут». Збірник наукових праць. Харків: НТУ «ХПІ», 2010. № 21. С. 30–37.
24. Волк М.О., Гридель Р.М., Олендаренко А.С. Архитектура подсистемы визуализации данных распределенной имитационной системы моделирования GRID. Автоматизированные системы управления и приборы автоматики. Харків, 2010. Вып. 150. С. 89–92.
25. Волк М.А., Олищук С.А. Анализ процессных алгебр и их использование в задачах распределенного имитационного моделирования. Системи управління, навігації та зв'язку. Київ. 2010. Вип. 4(16). С.150–153.
26. Дема К.В., Волк М.А., Филимончук М.А. Анализ архитектур и характеристик современных планировщиков заданий в GRID-системах. Збірник наук. праць ХУПС. 2011. Вип. 2(28), С. 123–128. (Входить до міжнародних наукометричних баз BASE, WorldCat)
27. Волк М.А., Зозуля В.В., Аль Шиблак М. Моделирование временных характеристик потоков заданий в GRID-системах. Системи управління навігації та зв'язку. К. 2011, вип. 2(18). С.248–250.

28. Волк М.А., Горенков А.С., Тучин О.В. Использование скриптов для повышения гибкости имитационной модели GRID-системы. Системы обробки інформації. 2011. Вип. 5(95). С.180–184. (Входит до міжнародної наукометричної бази BASE).

29. Волк М.А., Филимончук М.А., Муаз Ал Шиблак, Гридель Р.Н. Анализ распределенных имитационных моделей с консервативными алгоритмами синхронизации. Збірник наук. праць ХУПС.2012, Вип. 1(30) С. 95–98. (Входит до міжнародних наукометричних баз BASE, WorldCat).

30. Волк М.А., Филимончук М.А., Филимончук Т.В. Модуль распределения заданий в GRID-системах. Системы обробки інформації. Харків: ХУПС, 2012. №2(100). С. 177–182. (Входит до міжнародних наукометричних баз BASE, WorldCat).

31. Shostak I., Danova M., Romanenkov Y., Bugaienko O., Volk M., Karminska-Bielobrova M. Organization of information support for business processes at aviation enterprises by means of ontological engineering. Eastern-European Journal of Enterprise Technologies, [S.l.], v. 2, n. 2 (92), P. 45–55. (Входит до міжнародної наукометричної бази Scopus)

32. Romanenkov Y., Danova M., Kashcheyeva V., Bugaienko O., Volk M., Karminska-Bielobrova M., Lobach O. Complexification methods of interval forecast estimates in the problems on short-term prediction. Eastern-European Journal Of Enterprise Technologies. v.3 n.3 (93). P. 50–58. (Входит до міжнародної наукометричної бази Scopus).

33. Ivanisenko I., Volk M. Simulation methods for load balancing in distributed computing. Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2017), Novi Sad, Serbia, September 27 – October 2, 2017. P. 690–695. (Входит до міжнародної наукометричної бази Scopus).

34. Волк М.О., Гридель Р.М., Муаз Ал Шиблак. Технологии синхронизации распределенных компьютерных моделей. Информатика, математическое моделирование, экономика: Сб. научн. статей по итогам Международной научно-практической конференции. Смоленск, 22-24 апреля 2013. Том 1. С. 54–59.

35. Волк М.А., Филимончук Т.В. Обобщенный критерий оценки задания для технологии планирования заданий в GRID. Информатика, математическое моделирование, экономика: Сборник научных статей по итогам третьей международной научно-практической конференции (24-26 апреля 2013 г., Смоленск). В 3-х томах. Том 2. Смоленск: Смоленский филиал Российского университета кооперации, 2013. С. 172–176.

36. Волк М.А. Оценка эффективности распараллеливания вычислительных задач в локальных компьютерных сетях. Материалы первой Международной конференции «Глобальные информационные системы. Проблемы и тенденции развития», Харьков–Туапсе, 2006. С. 219–220.

37. Волк М.А. Условия синхронизации распределенных имитационных моделей в гетерогенной GRID. Материалы 2-й международной конференции «Современные информационные системы. Проблемы и тенденции развития». – Харьков–Туапсе, 2007. С. 475–476.

38. Волк М.А. Остроушко А.П., Бугрий А.Н. Анализ использования GRID для целей имитационного моделирования. Материалы 2-й международной конференции «Современные информационные системы. Проблемы и тенденции развития». – Харьков–Туапсе, 2007. С. 477–478.

39. Волк М.А. Перспективы создания харьковского ресурсно-операционного GRID-центра. Проблемы информатики и моделирования. Материалы 7-й международной научно-технической конференции. Харьков: НТУ «ХПИ», 2007. С. 29.

40. Волк М.А. Управление событиями в распределенной имитационной системе моделирования для GRID-инфраструктуры. Системний аналіз та інформаційні технології: Матеріали X Міжнародної науково-технічної конференції (20–24 травня 2008 р., Київ). К.: НТУУ «КПІ», 2008. С. 289.

41. Волк М.А., Филимончук М.А., Корниенко Т.В. (Филимончук Т.В.) Анализ использования искусственных иммунных систем в GRID-инфраструктуре. Системний аналіз та інформаційні технології: Матеріали X Міжнародної науково-технічної конференції. К.: НТУУ «КПІ», 2008. С. 290.

42. Волк М.А. Дьяченко К.Ю. Применение технологий распределенного программирования в имитационном моделировании. Проблеми інформатики і моделювання. Матеріали восьмої міжнародної науково-технічної конференції. Х.: НТУ «ХПИ», 2008. С. 20.

43. Волк М.А., Филимончук М.А., Филимончук Т.В. Анализ алгоритмов распределения ресурсов в имитационных системах моделирования, ориентированных на GRID системы. Проблеми інформатики і моделювання. Матеріали восьмої міжнародної науково-технічної конференції. Х.: НТУ «ХПИ», 2008. С. 58.

44. Волк М.О., Грідель Р.М., Дьяченко К.Ю. Элементы распределенной имитационной модели GRID-систем. Системний аналіз інформаційні технології: Материалы XI Международной научно-технической конференции. Киев: УНК «ИПСА» НТУУ «КПИ», 26–30 мая 2009. С. 422.

45. Волк М.А., Филимончук М.А., Филимончук Т.В. Исследование методов распределения заданий для GRID-систем. Системний аналіз і інформаційні технології: Материалы XI Международной научно-технической конференции (26–30 мая 2009 г., Киев). К.: УНК «ИПСА» НТУУ «КПИ», 2009. С. 423.

46. Волк М.А., Филимончук М.А., Филимончук Т.В. Система мониторинга трафика для GRID. Проблеми інформатики і моделювання. Матеріали дев'ятої міжнародної науково-технічної конференції. Х.: НТУ «ХПИ», 2009. С. 42.

47. Волк М.А., Грідель Р.М. Архитектура имитационной модели GRID-системы, основанная на подключаемых модулях. Проблеми інформатики і моделювання. Матеріали 9-тої міжнародної науково-технічної конференції. Х.: НТУ «ХПИ», 2009. С. 42.

48. Волк М.А., Олендаренко С.С., Горенков А.С. Архитектура подсистемы визуализации данных в распределенной имитационной модели GRID – системы. Проблеми інформатики і моделювання. Матеріали дев'ятої міжнародної науково-технічної конференції. Х.: НТУ «ХПИ», 2009. С. 43.

49. Волк М.А., Филимончук Т.В., Грідель Р.Н. Имитационная система моделирования GRID-инфраструктуры GRASS. Системний аналіз і інформаційні технології: Материалы XII Международной научно-технической конференции. Киев: УНК «ИПСА» НТУУ «КПИ», 2009. С. 424.

технологии: Материалы XII Международной научно-технической конференции (25–29 мая 2010 г., Киев). К.: УНК «ИПСА» НТУУ «КПИ», 2010. С. 359.

50. Волк М.А., Олищук С.А. Анализ мультимедийного трафика в гетерогенных сетях. Сучасні напрями розвитку інформаційно-крмунікаційних технологій та засобів управління. Х.:ДП «ХНДІ ТМ»; К.: ДП «ЦНДІ НіУ», 2010. С. 67

51. Волк М.А., Буртыка А.М., Иващенко Д.А. Анализ сетевых протоколов для оптимизации передачи данных интерфейса параллельного программирования MPI. Сучасні напрями розвитку інформаційно-крмунікаційних технологій та засобів управління. Х.: ДП «ХНДІ ТМ»; К.: ДП «ЦНДІ НіУ», 2010. С.74.

52. Волк М.А. Дема К.В. Сучасні напрями розвитку інформаційно-крмунікаційних технологій та засобів управління. Х.:ДП «ХНДІ ТМ»; К.: ДП «ЦНДІ НіУ», 2010. С. 745

53. Волк М.О., Грідель Р.М., Зозуля В.В. Анализ моделей потоков заданий для моделирования рабочей нагрузки в GRID-системах. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління. Х.: ДП «ХНДІ ТМ»; К.: ДП «ЦНДІ НіУ», 2010. С.75.

54. Олищук С.А., Волк М.А. Программный комплекс моделирования GRID-систем на основе подключаемых модулей. Системний аналіз та інформаційні технології: матеріали Міжнародної науково-технічної конференції SAIT 2011, Київ, 23–28 травня 2011 р. НК «ИПСА» НТУУ «КПИ». К., 2011. С. 383.

55. Волк М.О., Грідель Р.М., Муаз Ал Шиблак. Методы анализа распределенного имитационного моделирования с учетом сетевого трафика. Інформаційні технології в навігації і управлінні: стан та перспективи розвитку. Матеріали другої міжнародної науково-технічної конференції. К.: ДП «ЦНДІ НіУ», 2011. С.45.

56. Волк М.О. Грідель Р.М. Расширенный анализ распределенных имитационных моделей с оптимистическими и консервативными алгоритмами синхронизации. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління. Матеріали другої міжнародної науково-технічної конференції. К.: ДП «ЦНДІ НіУ», Х.: ДП «ХНДІ ТМ»; К.: КДАВТ, 2011. С. 36.

57. Волк М.О., Муаз Ал Шиблак. Анализ сетевых протоколов для распределенного имитационного моделирования. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління. Матеріали другої міжнародної науково-технічної конференції К.: ДП «ЦНДІ НіУ», Х.: ДП «ХНДІ ТМ»; К.:КДАВТ, 2011. С.36.

58. Волк М.О. Грідель Р.М., Муаз Ал Шиблак. Менеджер памяти распределенных имитационных моделей. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління. Матеріали другої міжнародної науково-технічної конференції. Полтава: ПНТУ; Білгород: НДУ «БілДУ»; Харків: ДП «ХНДІ», Київ: НТУ; Кіровоград: КЛА НАУ, 2013. С.34.

59. Волк М.А., Филимончук Т.В. Анализ существующего прикладного программного обеспечения GRID-систем и языков описания заданий. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління: Матеріали третьої міжнародної науково-технічної конференції. Полтава: ПНТУ;

Белгород: НДУ «БілДУ»; Харків: ДП «ХНДІ ТМ», Київ: НТУ; Кіровоград: КЛА НАУ, 2013. С. 49.

60. Кобылин И.О., Волк М.А. Использование мобильных платформ для оценки финансово-экономического состояния предприятия. Дев'ята наукова конференція Харківського університету Повітряних Сил імені Івана Кожедуба «Новітні технології – для захисту повітряного простору»: тези доповідей, 17–18 квітня 2013 року. Х.: ХУПС ім. Кожедуба, 2013. С. 217.

61. Волк М.О., Муаз Ал Шиблак. Модели процессов распределенной имитации для консервативных и оптимистических методов синхронизации. Системний аналіз та інформаційні технології: матеріали 15-ї Міжнародної науково-технічної конференції SAIT 2013, Київ, (27-31 мая 2013 р.). ННК «ІПСА» НТУУ «КПІ». К.: ННК «ІПСА» НТУУ «КПІ», 2013. С. 347.

62. Волк М.О., Грідель Р.М., Муаз Ал Шиблак. Технология синхронизации в распределенном имитационном моделировании. 26-я научно-практическая конференция «Внедрение перспективных микропроцессорных систем железнодорожной автоматики и средств телекоммуникаций на базе цифровизации». – Алушта, 2013. С. 63.

63. Волк М.А., Филимончук Т.В. Информационная технология распределения заданий по ресурсам. Проблеми інформатизації: Матеріали першої міжнародної науково-технічної конференції. Черкаси: ЧДТУ; Київ: ДУТ; Тольятті: ТДУ; Полтава: ПНТУ, 2013. С. 20–21.

64. Волк М.О., Грідель Р.М., Муаз Ал Шиблак. Информационная технология анализа распределенных имитационных моделей. Проблеми інформатизації: Матеріали першої міжнародної науково-технічної конференції. Черкаси: ЧДТУ; Київ: ДУТ; Тольятті: ТДУ; Полтава: ПНТУ, 2013. С. 29.

65. Волк М.О., Грідель Р.М. Информационные технологии обеспечения распределенного имитационного моделирования. Проблеми інформатизації: Матеріали другої міжнародної науково-технічної конференції. Київ: ДУТ; Полтава: ПНТУ; Катовице: Катовицький економічний університет; Париж: Університет Париж VII Винсент-Сен-Дзі; Черкаси: ЧДТУ; Харків: ХНДІТМ, 2014. С. 56.

66. Волк М.А., Филимончук Т.В. Информационная технология управления заданиями в распределенных системах. Проблеми інформатизації: Матеріали другої міжнародної науково-технічної конференції. Київ: ДУТ; Полтава: ПНТУ; Катовице: Катовицький економічний університет; Париж: Університет Париж VII Венсент-Сен-Дені; Білгород: НДУ «БДУ»; Черкаси: ЧДТУ; Харків: ХНДІТМ, 2014. С. 59–60.

67. Волк М.А., Филимончук Т.В. Информационная технология управления заданиями в GRID-системах. Информатика, математическое моделирование, экономика: Сборник научных статей по итогам четвертой международной научно-практической конференции (23-25 апреля 2014 г., Смоленск). В 2-х томах. Том 1. Смоленск: Смоленский филиал Российского университета кооперации, 2014. С. 379–383.

68. Волк М.О., Грідель Р.М. Функциональная модель системы распределения ресурсов в задачах распределенного имитационного моделирования. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління.

Матеріали четвертої міжнародної науково-технічної конференції. Полтава: ПНТУ; Баку: ВА ЗС АР; Білгород: НДУ «БілДУ»; Кіровоград: КЛА НАУ; Харків: ДП «ХНДІ», 2014. С. 17.

69. Волк М.О., Грідель Р.М. Информационная технология распределенного имитационного моделирования. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління. Матеріали п'ятої міжнародної науково-технічної конференції. Полтава: ПНТУ; Баку: ВА ЗС АР; Кіровоград: КЛА НАУ; Харків: ДП «ХНДІ ТМ», 2015. С.17.

70. Волк М.А., Филимончук Т.В. Информационная технология распределения заданий на вычислительные ресурсы для обработки радиоастрономических данных в GRID-системах. Проблеми інформатизації: Матеріали третьої міжнародної науково-технічної конференції. Черкаси: ЧДУТ; Баку: ВА ЗС АР; Бельсько-Бяла: УТіГН; Полтава: ПНТУ, 2015. С. 23.

71. Volk M.O. Simulation framework architecture for GRID and Cloud systems. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління. Матеріали п'ятої міжнародної науково-технічної конференції. Полтава: ПНТУ; Баку: ВА ЗС АР; Білгород: НДУ «БілДУ»; Кіровоград: КЛА НАУ; Харків: ДП «ХНДІ», 2016. С.44.

72. Volk M.O., Filimonchuk T.V. Information technology for job distribution in GRID-systems Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління: матеріали шостої міжнародної науково-технічної конференції. Полтава: ПНТУ; Баку: ВА ЗС АР; Кіровоград: КЛА НАУ; Харків: ДП «ХНДІ ТМ», 2016. С. 45–46.

73. Волк М.А., Филимончук Т.В. Модифицированный метод Backfill с консервативным резервированием. Проблеми інформатизації: Матеріали четвертої міжнародної науково-технічної конференції. Черкаси: ЧДУТ; Баку: ВА ЗС АР; Бельсько-Бяла: УТіГН; Полтава: ПНТУ, 2016. С. 20.

74. Volk M.O. Information technology for functional stability of distributed simulation. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління. Матеріали сьомої міжнародної науково-технічної конференції. Полтава: ПНТУ; Баку: Кіровоград: КЛА НАУ; Харків: ДП «ХНДІ», 2017. С. 17.

75. Волк М.О. Обеспечение функциональной устойчивости систем распределенного имитационного моделирования. Матеріали першої міжнародної науково-технічної конференції «Комп'ютерні та інформаційні системи і технології». Збірка наукових праць. Харків: ХНУРЕ. 2017. Р. С. 15.

76. Волк М.А., Филимончук Т.В. Информационная технология распределения заданий на вычислительные ресурсы в гетерогенных GRID-системах. Проблеми інформатизації: Матеріали п'ятої міжнародної науково-технічної конференції. Черкаси, Баку, Бельсько-Бяла, Полтава; 13–15 листопада 2017. С. 34.

77. Volk M.O., Klenov A.E., Shkruty D.A. Programs survivability method for distributed computing systems. Проблеми інформатизації: Матеріали п'ятої міжнародної науково-технічної конференції. Черкаси, Баку, Бельсько-Бяла, Полтава; 13–15 листопада 2017. С. 27.

78. Волк М.А., Филимончук Т.В., Луничкин А.Г., Рисухин М.В. Информационная технология распределения заданий на вычислительные ресурсы

на основі транзакційного методу. Матеріали другої міжнародної науково-технічної конференції «Комп'ютерні та інформаційні системи і технології». Збірка наукових праць. Харків: ХНУРЕ. 2018. С. 55.

79. Волк М.О., Рісухін М.В., Копійка М.В. Моделі обчислювального процесу в розподіленій мультисервісній системі з різними методами синхронізації. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління. Матеріали восьмої міжнародної науково-технічної конференції – Полтава – Баку – Харків – Жиліна. 2018. С. 21.

80. Волк М.О., Рісухін М.В., Новицький Р.В. Інформаційна технологія розподілу завдань у розподіленому обчислювальному процесі з альтернативними схемами призначення. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління. Матеріали восьмої міжнародної науково-технічної конференції – Полтава – Баку – Харків – Жиліна. 2018. С. 29.

81. Волк М.О., Шкруть Д.А. Исследование архитектуры фреймворка для системного программного обеспечения. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління. Матеріали восьмої міжнародної науково-технічної конференції – Полтава – Баку – Харків – Жиліна. 2018. С. 39.

82. Волк М.А., Клёнов А.Е. Исследование методов обеспечения живучести системного программного обеспечения. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління. Матеріали восьмої міжнародної науково-технічної конференції – Полтава – Баку – Харків – Жиліна. 2018. С. 40.

83. Волк М. А., Филимончук Т. В. Использование системы моделирования GRASS в задачах распределения заданий в GRID-системах. 20-й Ювілейний міжнародний форум «Радіоелектроніка та молодь у XXI столітті». Харків: ХНУРЕ, 2016. Т.4. С. 171–172.

84. Волк М.О., Грідель Р.М., Муаз Ал Шиблак. Метод анализа распределенных имитационных моделей. Інформаційні технології в навігації і управлінні: стан та перспективи розвитку. Матеріали першої міжнародної науково-технічної конференції. К.: ДП "ЦНДІ НіУ", 2010. С.46.

АНОТАЦІЯ

Волк М.О. Моделі, методи та інформаційна технологія управління розподіленим обчислювальним процесом в гетерогенних комп'ютерних системах. – На правах рукопису.

Дисертація на здобуття наукового ступеня доктора технічних наук за спеціальністю 05.13.06 – інформаційні технології. – Харківський національний університет радіоелектроніки, Міністерство освіти і науки України, Харків, 2018.

У дисертаційній роботі вирішено актуальну науково-прикладну проблему розробки моделей, методів та інформаційної технології управління розподіленим обчислювальним процесом в гетерогенних комп'ютерних системах.

Розроблено теоретико-множинну модель розподіленого обчислювального середовища під час виконання завдання в розподілених гетерогенних комп'ютерних системах, математичну модель розподіленого обчислювального процесу. Розроблено комплекс методів оцінювання та вибору схем призначення

програмних компонентів завдання за обчислювальними ресурсами, методи забезпечення функціональної стійкості розподілених програмних завдань. Удосконалено метод оцінки часу виконання програмних завдань та метод розподілу пулу завдань за обчислювальними ресурсами. Набули подальшого розвитку методологія управління розподіленим обчислювальним процесом в гетерогенних комп'ютерних системах, модель подання завдань і ресурсів та методи розподілу програмних завдань за обчислювальними ресурсами. Розроблено інформаційну технологію управління розподіленим обчислювальним процесом в гетерогенних комп'ютерних системах, програмні засоби системи управління. Отримані практичні результати підтверджені актами впровадження і доводять коректність теоретичних положень дисертаційної роботи, високу ефективність розроблених моделей, методів та інформаційної технології.

Ключові слова: інформаційні технології, розподілені обчислення, гетерогенні комп'ютерні системи, програмні завдання, розподілення ресурсів, функціональна стійкість, система управління.

АННОТАЦІЯ

Волк М.А. Модели, методы и информационная технология управления распределенным вычислительным процессом в гетерогенных компьютерных системах. – На правах рукописи.

Диссертация на соискание ученой степени доктора технических наук по специальности 05.13.06 – информационные технологии. – Харьковский национальный университет радиоэлектроники, Министерство образования и науки Украины, Харьков, 2018.

В диссертационной работе решена актуальная научно-прикладная проблема разработки моделей, методов и информационной технологии управления распределенным вычислительным процессом в гетерогенных компьютерных системах.

Разработана теоретико-множественная модель распределенной вычислительной среды во время выполнения задания в распределенных гетерогенных компьютерных системах, которая унифицирует описание задания, компьютерных ресурсов и системы управления распределенными вычислениями, позволяет формализовать процесс распределенных вычислений и реализовать комплексный подход к повышению эффективности распределенного вычислительного процесса.

Разработана математическая модель распределенного вычислительного процесса, учитывающая динамическое изменение объемов памяти, потоков данных между программными компонентами задачи, время простоя, методы синхронизации, методы обеспечения функциональной устойчивости и архитектуру гетерогенной компьютерной системы, что позволяет выполнить установленные ограничения на объемы оперативной памяти, время использования процессора, и минимизировать время простоя вычислительных ресурсов для различных условий синхронизации программных компонентов задания.

Предложен комплекс методов оценки и выбора схем назначения программных компонентов задания по вычислительным ресурсам, который,

за счет использования предложенной математической модели распределенного вычислительного процесса, учитывает параметры задачи, ресурсов и выбранной стратегии управления, что позволяет получить количественные параметры оценки эффективности распределенного вычислительного процесса, применить схему назначения с минимальным временем выполнения и простым вычислительных ресурсов при реализации выбранной стратегии управления.

Разработаны методы обеспечения функциональной устойчивости распределенных программных задач, которые позволяют обеспечить сохранение данных и состояния программных компонентов или истории их изменений во времени, способствует автоматизации процесса восстановления работоспособности распределенного программного обеспечения после отказа вычислительного ресурса или коммуникационного канала связи. Выбор метода обеспечения функциональной стабильности зависит от стратегии управления.

Усовершенствован метод оценки времени выполнения программных заданий, который, в отличие от существующих, учитывает объем памяти программных модулей, объемы данных, передаваемых между программными компонентами, время простоя ресурсов, время и объемы памяти для обеспечения функциональной устойчивости, позволяет оценить время выполнения программных задач для различных схем распределения ресурсов с учетом ограничений на объем памяти, суммарное время простоя вычислительных ресурсов и различных методов синхронизации программных компонентов задачи.

Усовершенствован метод распределения пула задач на вычислительные ресурсы, который в отличие от существующих методов, учитывает параметры трафика процесса загрузки задачи и трафика между программными компонентами задачи, позволяет повысить эффективность использования гетерогенных компьютерных систем за счет уменьшения времени выполнения пула задач.

Получила дальнейшее развитие модель представления задач и ресурсов в гетерогенных компьютерных системах, которая, в отличие от существующих, учитывает методы распределения задач по ресурсам и объемы сетевого трафика в процессе загрузки и выполнения задач, что позволяет сократить время выполнения задач и уменьшает нагрузку на сети передачи данных.

Получили дальнейшее развитие методы распределения программных задач на вычислительные ресурсы, которые, в отличие от существующих, позволяют осуществлять распределение ресурсов с максимальным продвижением времени выполнения в условиях ограничений на объем оперативной памяти, времени простоя и выбрать наиболее оптимальные методы синхронизации и обеспечения функциональной устойчивости программных компонентов задачи.

Получила дальнейшее развитие методология управления вычислительным процессом в распределенных компьютерных системах, основанная на системных принципах гетерогенности программных задач и ресурсов, выбора стратегии управления, выбора методов распределения ресурсов, связности программ и учета трафика в задании, выбора методов синхронизации, функциональной устойчивости и имитационного моделирования распределенного вычислительного процесса, которая обеспечивает эффективное использование вычислительных ресурсов и снижает время выполнения заданий.

Разработана информационная технология управления распределенным вычислительным процессом в гетерогенных компьютерных системах, программные средства системы управления. Экспериментальные исследования показывают эффективность предложенных решений. Полученные практические результаты подтверждены актами внедрения и доказывают корректность теоретических положений диссертационной работы, высокую эффективность разработанных моделей, методов и информационной технологии.

Ключевые слова: информационные технологии, распределенные вычисления, гетерогенные компьютерные системы, программные задачи, распределения ресурсов, функциональная устойчивость, система управления.

ABSTRACT

Volk M.O. Models, methods and information technology of the distributed computing process management in heterogeneous computer systems. – Manuscript.

Dissertation for the Doctor of Technical Sciences degree in the specialty 05.13.06 – Information Technologies. – Kharkiv National University of Radio Electronics, Ministry of Education and Science of Ukraine, Kharkiv, 2018.

In thesis the solution of the scientific and applied problem of the development of models, methods and information technology of the management of the computing process in distributed heterogeneous computer systems was proposed.

The set-theoretic models of the computing environment during the task execution in distributed heterogeneous computer systems, the distributed computing process mathematical model were developed. The set of methods for estimating the schemes of job program assignment on computing resources was proposed, the method of distribution scheme selection for job distribution on computing resources, methods for ensuring the functional stability of distributed software jobs were developed. The method of estimating the time of execution of program tasks and the method for jobs pool allocation on computing resources were improved. The methodology of distributed computing management in heterogeneous computer systems, model for representing tasks and resources and methods for distributing program jobs on computing resources were further developed. The information technology of distributed computing process management in heterogeneous computer systems, software tools of control system were developed. The obtained practical results are confirmed by the acts of implementation and prove the correctness of the theoretical positions of the dissertation work, high efficiency of the developed models, methods and information technology.

Keywords: information technologies, distributed computing, heterogeneous computer systems, program tasks, resource allocation, functional stability, management system.

Підп. до друку 02.01.19.
Умов.друк.арк. 2,3.
Ціна договірна

Формат 60x84 $\frac{1}{16}$.
Облік. вид.арк. 2,1.
Зам № 2-2.

Спосіб друку – ризографія.
Тираж 100 прим.

ХНУРЕ. Україна. 61166, Харків, просп. Науки, 14

Віддруковано в редакційно-видавничому відділі ХНУРЕ
61166, Харків, просп. Науки, 14