

УДК 004

Є.Деркач, магістр гр. КН-22М-2

Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ ПАРАЛЕЛЬНИМИ ОБЧИСЛЕННЯМИ ДЛЯ НАУКОВИХ ДОСЛІДЖЕНЬ

У статті розроблено програмне забезпечення, яке призначено для системи управління паралельними обчисленнями для наукових досліджень. Метою розробки є дослідження та програмна реалізація системи управління паралельними обчисленнями для наукових досліджень. Об'єктом дослідження є процес управління паралельними обчисленнями для наукових досліджень. Предметом дослідження є методи управління паралельними обчисленнями для наукових досліджень. Методи дослідження базуються на методах паралелізму, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи управління паралельними обчисленнями для наукових досліджень. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Постановка проблеми. GRID-системи – це взаємопов'язані набори різнорідних і територіально розподілених ресурсів, об'єднаних для задоволення різноманітних потреб користувачів. Управління ресурсами є центральним компонентом GRID-системи. Він включає в себе управління ресурсами в системі. Його основна відповідальність полягає в тому, щоб приймати запити від користувачів, зіставляти запити користувачів з доступними ресурсами, до яких користувач має доступ, і планувати відповідні ресурси.

Управління ресурсами – це складне завдання, що включає безпеку, відмовостійкість разом із плануванням. Це спосіб, у який ресурси розподіляються, призначаються, автентифікуються, авторизуються, гарантуються, обліковуються та перевіряються. Ресурси включають традиційні ресурси, такі як обчислювальні цикли, пропускна здатність мережі, простір або система зберігання, а також такі послуги, як передача даних, моделювання тощо.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи управління паралельними обчисленнями для наукових досліджень.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи управління паралельними обчисленнями для наукових досліджень.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем управління паралельними обчисленнями для наукових досліджень.
- Дослідження системи управління паралельними обчисленнями для наукових досліджень.
- Програмна реалізація системи управління паралельними обчисленнями для наукових досліджень.

Об'єктом дослідження є процес управління паралельними обчисленнями для наукових досліджень.

Предметом дослідження є методи управління паралельними обчисленнями для наукових досліджень.

Методи дослідження базуються на методах паралелізму, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Управління ресурсами в GRID-системах ускладнюється різними факторами, такими як автономія сайту, неоднорідність ресурсів тощо. Різні фактори, що сприяють цій складності, обговорюються нижче. Система управління ресурсами GRID повинна зберігати автономію сайту. Традиційні системи управління ресурсами працюють на основі припущення, що вони повністю контролюють ресурс і, таким чином, можуть запроваджувати механізми та політику, необхідні для ефективного використання цього ресурсу. Але в системах GRID ресурси розподіляються між окремими адміністративними доменами, що призводить до неоднорідності ресурсів, відмінностей у використанні, політиках планування, механізмах безпеки. Ще одна проблема управління ресурсами, ускладнена через автономію сайту, – це спільне розміщення ресурсів. Спільне розподілення – це проблема розподілу ресурсів на різних сайтах для програми одночасно.

Різні адміністративні домени використовують різні локальні системи управління ресурсами, такі як NQE, LSF тощо. Система управління ресурсами GRID повинна мати можливість взаємодіяти та взаємодіяти з цими локальними системами управління ресурсами. Система управління ресурсами для мережі повинна підтримувати такі переговори. У сітці системні ресурси додаються та видаляються динамічно. Виконуються різні типи додатків з різними вимогами до ресурсів. Власники ресурсів встановлюють власні правила використання ресурсів і витрати. Це вимагає переговорів між користувачами ресурсів і постачальниками ресурсів. У сітці виконуються різні типи програм із широкого діапазону доменів, кожна з яких має різні вимоги до керування ресурсами. У той час як деякі типи програм вимагають якнайшвидшого планування, навіть якщо це означає зниження продуктивності, деякі класи програм потребують високої продуктивності. Структура управління ресурсами повинна дозволяти вбудовувати в неї нові політики, не вимагаючи суттєвих змін до існуючого коду. На додаток до вищезазначеного, система управління ресурсами для GRID повинна відповідати іншим питанням, таким як гнучкість і розширюваність, глобальний простір імен, безпека, відмовостійкість і якість обслуговування.

Контекст і функції GRMS

У GRID-системі кінцевий користувач передає системі керування завдання, яке потрібно виконати, разом із деякими обмеженнями, такими як кінцевий термін виконання завдання, максимальна вартість виконання. Функція управління ресурсами полягає в тому, щоб взяти специфікацію завдання та оцінити на основі неї вимоги до ресурсів, такі як кількість необхідних процесорів, час виконання та необхідна пам'ять. Після оцінки вимог до ресурсів RMS відповідає за виявлення доступних ресурсів і вибір відповідних ресурсів для виконання завдання. Нарешті заплануйте завдання на цих ресурсах, взаємодіючи з локальною системою управління ресурсами. RMS також відповідає за іменування ресурсів у системі, моніторинг і звітування про завдання, стан ресурсів і облік використання ресурсів. RMS взаємодіє із системою безпеки для перевірки запитів користувачів, інформаційною службою для отримання інформації про доступність ресурсів, локальною системою для планування завдань у локальній системі керування ресурсами

Таксономія на основі архітектури системи управління ресурсами обговорюється нижче. Він розділений на такі категорії: тип сітки, простір імен ресурсу, інформація про ресурс (виявлення, розповсюдження), модель планування, політика планування.

Тип GRID

Більшість сучасних GRID не забезпечують однакову підтримку для всіх типів ресурсів. GRID-системи класифікуються як обчислювальні, дані та сервісні. У обчислювальних мережах основним ресурсом, яким керує RMS, є обчислювальні цикли (тобто процесори), тоді як у мережах даних основна увага приділяється управлінню даними, розподіленими по географічних місцях. На архітектуру та послуги, які надає система керування ресурсами, впливає тип GRID-системи, у якій вона розгорнута. Ресурси можуть бути апаратними (цикл обчислення, пропускна здатність мережі та сховища даних) або програмними ресурсами (додатки та бази даних).

Організація простору імен ресурсу

GRID RMS відповідає за іменування ресурсів у GRID-системі. Іменування ресурсів впливає на інші функції керування ресурсами, такі як виявлення та розповсюдження ресурсів, а також впливає на структуру бази даних, у якій зберігається інформація про ресурси. Три підходи до організації простору імен: плоский, ієрархічний і графовий. Плоский простір імен не дуже масштабований у системі сітки. В ієрархічній організації іменування ресурсів відповідає системному системному підходу, ім'я створюється шляхом переходу вниз по ієрархії. У іменуванні на основі графів ресурси пов'язані між собою, а ім'я ресурсу створюється за посиланнями від одного об'єкта до іншого.

Протокол розповсюдження ресурсів

Розповсюдження ресурсів включає рекламу ресурсів у GRID-системі. Протоколи розповсюдження впливають на кількість даних, що передаються між системами, а також на статус бази даних ресурсної інформації. Протоколи розповсюдження можна класифікувати як періодичні або на вимогу. Періодичне розповсюдження передбачає групування інформації про зміну статусу ресурсу та періодичне оновлення інформаційної бази даних. Далі це можна класифікувати як розповсюдження, що натискає або притягує. Під час розповсюдження за запитом дані ресурсу періодично надсилаються до сховища інформації, тоді як за розповсюдження за запитом сховище інформації періодично збирає інформацію про статус ресурсів. Протоколи розповсюдження на вимогу оновлюють інформацію про ресурс, коли запускається конкретна подія або відбувається зміна статусу.

Виявлення ресурсів

Виявлення ресурсів є дуже важливою функцією управління ресурсами. Служби виявлення використовуються системою планування для отримання інформації про доступний ресурс. Підходи до виявлення ресурсів можна класифікувати як засновані на запитах або на основі агентів. Під час виявлення на основі запиту сховище інформації про ресурс запитується щодо доступності ресурсу. Більшість сучасних GRID-систем дотримуються цього підходу. При виявленні на основі агентів агенти перетинають сітку, щоб зібрати інформацію про доступність ресурсів.

Модель планування

Структура планувальника визначає структуру системи управління ресурсами та масштабованість системи. Більшість сучасних систем управління GRID-ресурсами можна класифікувати як такі, що мають централізовану, ієрархічну або децентралізовану структуру. У централізованій організації всі завдання передаються одному планувальнику, який відповідає за їх планування на доступних ресурсах. Оскільки вся інформація планування доступна в одній позиції, рішення щодо планування є оптимальним, але цей підхід не дуже масштабований у GRID-системі. У децентралізованій моделі немає центрального планувальника, планування виконується запитувачами ресурсів і власниками незалежно. Цей підхід є масштабованим і підходить для GRID-систем. Але окремі планувальники повинні співпрацювати один з одним у прийнятті рішень щодо планування, і створений графік може бути не оптимальним. Залежно від того, чи співпрацюють планувальники, їх можна далі класифікувати як кооперативні та некооперативні планувальники. В ієрархічній моделі планувальники організовані в ієрархію. Сутності ресурсів високого рівня плануються на вищих рівнях, а менші суб-сутності нижчого рівня плануються на нижчих рівнях ієрархії планувальника. Ця модель є комбінацією двох вищевказаних моделей.

Оцінка стану

Користувачі надсилають заявку та вхідні дані разом із деякими вимогами до якості обслуговування, як-от кінцевий термін, до системи керування. Система управління повинна оцінити конкретні вимоги до ресурсів для запуску програми. Є три підходи до цієї проблеми: теоретичне передбачення, прогнозування на основі історії та прогнозування на основі тестових випадків. У теоретичному прогнозуванні вимоги до програми оцінюються на основі аналізу моделі програмування програми та предметної області. Ефективність цього підходу залежить від того, наскільки добре зрозуміла обчислювальна модель для програми. У підході

на основі історії передбачення робиться на основі попередніх запусків програм. Цей підхід підходить для програм, які, швидше за все, не зміняться з часом або зазвичай виконуються на певному наборі ресурсів, де минула продуктивність є сильним показником майбутньої продуктивності програми на тих самих ресурсах. У прогнозуванні на основі тестових випадків програма перевіряється на обмежений набір випадків на типових машинах. Це підходить для програм із невідомими характеристиками або програм, які змінюються з часом.

Розробка структурної схеми

Паралельне завдання визначається як програма, реалізована у вигляді декількох компонентів, кожна з яких запускається на окремому процесорному вузлі й у ході свого виконання може взаємодіяти з іншими компонентами, а також уводиться поняття планування як основного етапу керування паралельними завданнями.

Існують дві групи методів планування, використовуваних у сучасних паралельних архітектурах і кластерних системах:

– Методи першої групи використовують поділ часу процесорів між декількома завданнями. Відзначається основний недолік цих методів, що складає в тому, що компоненти одного завдання можуть бути перервані й перезапущені в різні моменти часу, що приводить до сильного зниження загальної ефективності використання ресурсів.

– Друга група методів планування (FCFS, First-Fit, Backfill) заснована на ідеї поділу простору ресурсів між завданнями, відповідно до якої кожне завдання одержує необхідний обсяг ресурсів на необхідний час в ексклюзивному режимі. Докладно розглядається широко застосовується на практиці метод зворотного заповнення Backfill, що гарантує запуск паралельного завдання й разом з тим ефективно використовує ресурси завдяки виділенню ресурсів завданням не безпосередньо в момент звільнення, а завчасно.

Розглянемо технології розподіленого комп'ютерингу, зокрема, концепція й базові принципи GRID як способу організації інфраструктури високопродуктивного комп'ютерингу. Сучасний і найпоширеніший підхід до такої організації складається в побудові GRID, що застосовується для проведення наукових досліджень, із просторово розподілених багатопроекторних установок, таких як масивно-паралельні комп'ютери (MPP) і кластери. Відзначається, що в існуючих проектах (EGEE, GRID2003, NorduGRID і ін.) переважає спосіб створення GRID, коли самі власники не використовують свої ресурси, а віддають їх цілком в GRID. Такий спосіб організації ресурсів, називаний GRID з відчужуваними ресурсами, не може претендувати на універсальність, однак може бути корисний у спеціальних умовах застосування, наприклад, коли власники самі не використовують свої ресурси, а виступають у ролі провайдерів. На практиці представляється більше цікавим спосіб, при якому ресурси використовуються спільно їхніми власниками й користувачами GRID. Цей спосіб організації називають GRID з невідчужуваними ресурсами. Його перевага полягає в тому, що такий GRID можна створити лише на якийсь час, необхідне для рішення конкретного завдання, і без формування спеціальної ресурсної бази: досить лише утягнути вже наявні ресурси, не завантажуючи повністю свої власні.

Розглянемо специфіку керування паралельними завданнями в GRID у порівнянні з паралельними комп'ютерними архітектурами й кластерними системами. Найбільш значимі відмітні властивості GRID, що ускладнюють завдання коаллокації:

- Автономність.
- Гетерогенність.
- Просторова розподіленість.
- Дворівнева організація ресурсів.

Завдання коаллокації – центральне завдання планування паралельних завдань, що складає в розміщенні набору компонент паралельного завдання на безлічі ресурсів. Для умов GRID вона формулюється в такий спосіб. Як ресурси розглядаються обчислювальні ресурси GRID – кластери $R = \{R_1, R_2, \dots, R_N\}$, кожний з яких являє собою деяку кількість процесорних вузлів – процесорів і пов'язану з кожним з них оперативну й дискову пам'ять. На ці ресурси

необхідно розмістити P компонент паралельного завдання $J = \{J_1, J_2, \dots, J_P\}$, синхронно виділяючи їм ресурси з безлічі ресурсів R на час виконання T .

У результаті рішення завдання визначається час старту завдання й набір аллокацій ресурсів: $A = \{A_x(r_{ji}, t_0, t_0 + T), \dots, A_K(r_{jK}, t_0, t_0 + T)\}$, $K = 1, P$, де r_{ji} – ресурси, використовувані для аллокації такі, що:

$$r_{ji} \subseteq R_j, i = 1, K, r_{j1} + r_{j2} + \dots + r_{joo} = P,$$

де r_{ji} – кількість аллоційованих процесорних вузлів на ресурсі. Всі аллокації починаються в один час і мають однакову тривалість. Крім того, забезпечується виконання двох умов:

- характеристики виконавчих ресурсів відповідають вимогам компонентів завдання;
- виконавчі ресурси в період $[t_0, t_0 + T]$ вільні, і політика поділу ресурсів не перешкоджає їхньому виділенню для завдання.

Перераховані вище властивості GRID роблять можливість побудови точних аллокацій A_1, \dots, A_K , $K = 1, P$, у яких визначається безліч виконавчих процесорів r_{h1}, \dots, r_{jK} і часовий інтервал $[t_0, t_0 + T]$, на який вони приділяються цьому завданню, проблематично. Замість цього в багатьох практичних реалізаціях результатом планування є аллокації, у яких час початку й ресурси точно не визначені. Для паралельних завдань це спричиняє серйозні дефекти при їхній обробці, такі як «зависання» завдань у черзі кластера.

Робиться вивід про те, що для запобігання небажаних ефектів планування в GRID, що застосовується для проведення наукових досліджень, повинне бути детермінованим, тобто повинне будувати точні аллокації, і, крім того, запуск завдань на виконавчих ресурсах повинен здійснюватися відповідно до побудованого аллокаціями.

Обґрунтуємо вибір на користь пріоритетного планування для застосування в GRID, що застосовується для проведення наукових досліджень, і розглянемо класифікацію методів такого планування. Як показує практика, методи поділу часу важко застосовні в умовах GRID. Крім того, ці методи спрямовані на поліпшення середнього часу обробки завдань, тоді як в GRID планування в першу чергу повинне мати на меті забезпечення якості обслуговування окремих користувачів. Для цього більше підходять методи поділу простору ресурсів, які в цей час переважно використовуються в GRID, що застосовується для проведення наукових досліджень. Серед них виділяються методи пріоритетного планування й методи, засновані на інтегральних критеріях. Останні здатні оптимізувати завантаження ресурсів або загальний час рахунку пакета завдань. Представляється, що планування в GRID у першу чергу повинне бути спрямоване на справедливий поділ ресурсів між завданнями, забезпечуючи при цьому можливість керування порядком виділення ресурсів для окремих завдань. Для таких цілей найбільш підходящим є пріоритетне планування. Спосіб рішення завдання коаллокації залежить значною мірою від того, яка інформація про ресурси для планування є в наявності. За цим критерієм сучасні методи пріоритетного планування розділяються на два класи. До першого класу належать методи, засновані на використанні черг завдань, основний принцип яких полягає у виділенні ресурсів для вартих у черзі завдань, виходячи з поточного стану ресурсів.

Застосування цих методів для паралельних завдань в GRID неефективно по двох причинах:

- По-перше, нагромадження ресурсів, необхідних для запуску завдання, може бути здійснено лише за допомогою блокування вільних ресурсів, що приводить до їхнього простою протягом невизначеного часу.
- По-друге, механізм попереднього резервування, призначений для забезпечення детермінованого планування, важко застосовуємо через відсутність інформації про час звільнення необхідного завданню обсягу ресурсів.

Другий клас включає методи, що використовують для розподілу ресурсів повноцінний план на майбутнє або розклад. До них ставиться метод зворотного заповнення (Backfill). З їхньою допомогою автоматично забезпечується нагромадження ресурсів і

природно реалізується механізм попереднього резервування ресурсів, що дозволяє використовувати методи цього класу для обслуговування паралельних завдань в GRID.

Опишемо запропонований метод керування паралельними завданнями за допомогою випереджального планування для практично важливої форми організації GRID, що застосовується для проведення наукових досліджень, з невідчужуваними ресурсами, коли вони не виділяються в GRID повністю, а використовуються в режимі поділу з їхніми власниками. У цих умовах завдання надходять не тільки з GRID (глобальний рівень), але й від користувачів кластера (локальний рівень) (рис. 1).

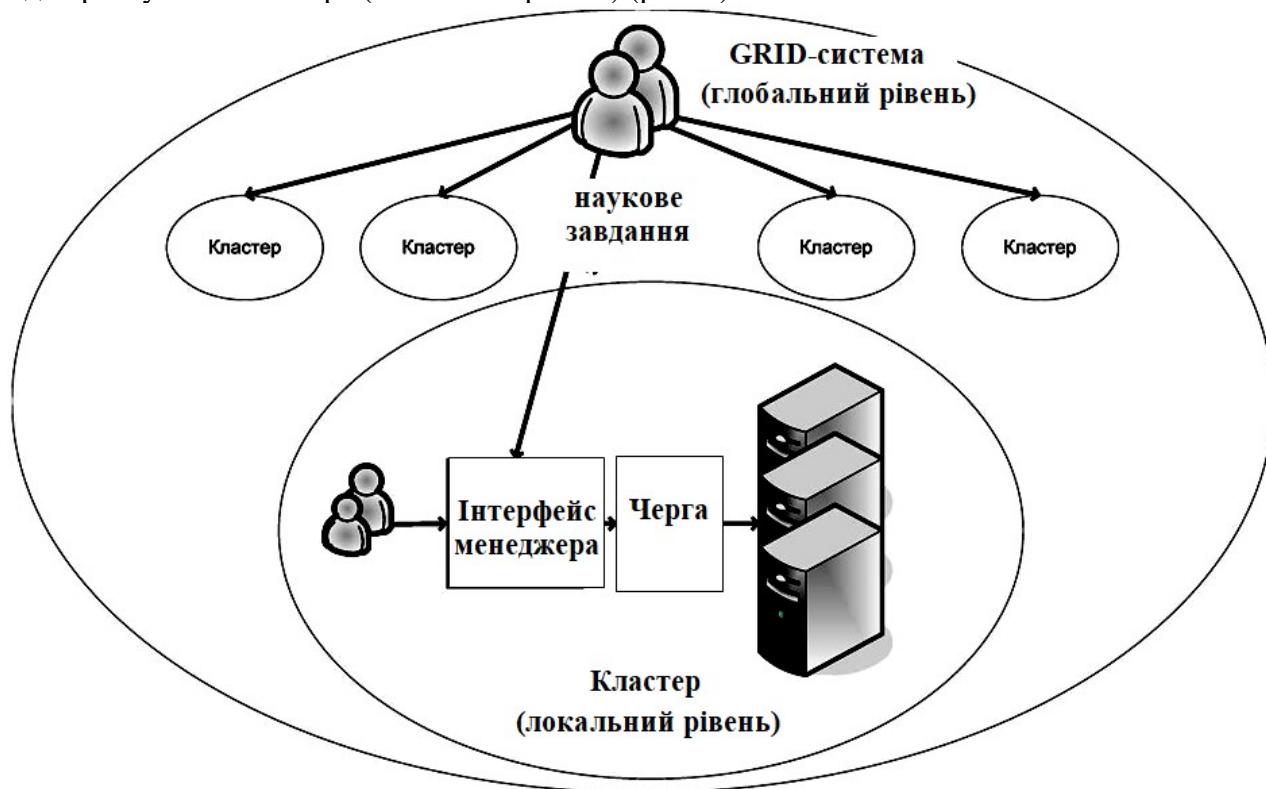


Рисунок 1 – Структурна схема системи

Особливістю запропонованого рішення є те, що ресурси розподіляються не в момент їхнього звільнення, а будується розклад їхнього використання на деякий період часу у вигляді безлічі тимчасових слотів, де кожний слот відповідає початку й кінцю деякого завдання або вільній ділянці розкладу. Виконання розкладу забезпечується за допомогою попереднього резервування ресурсів. Завдяки цьому планування в умовах невідчужуваності ресурсів є детермінованим.

Розглянемо економічний підхід до планування в GRID, що застосовується для проведення наукових досліджень, для керування паралельними завданнями. Відповідно до цього підходу процес розподілу ресурсів організується на базі моделі ринку: власники ресурсів виступають як продавці, а користувачі – як покупці цих ресурсів. Реалізуючи такий підхід, пропонуються дві угоди по поділу ресурсів і способи їхньої підтримки при плануванні.

Перша угода стосується поділу ресурсів між користувачами. Основна вимога полягає в тому, що користувач повинен мати можливість управляти швидкістю одержання ресурсів при плануванні. Для цього їм установлюється плата за виконання кожного завдання в рамках виділеного йому бюджету. При збільшенні плати шанси швидше виконати завдання підвищуються, однак при цьому обмеження бюджету не дозволяє користувачеві монополізувати весь ресурсний пул.

Для поділу ресурсів між двома потоками завдань, що надходять із локального й глобального рівнів, пропонується друга угода про поділ ресурсів: глобальне завдання може

зайняти ресурси за умови, що плата, призначена користувачем за виконання завдання, не менше їхньої вартості. Така угода дозволяє здійснювати динамічне балансування потоків завдань, забезпечуючи конкуренцію глобальних і локальних завдань, і є ключовим для GRID з невідчужуваними ресурсами. Відповідно до цього підходу глобальні завдання можуть розміщатися не тільки на вільних ресурсах, але й на ресурсах, на якому претендують локальні завдання, за умови, що плата за ці завдання не нижче вартості ресурсів. Таким чином, планування паралельних завдань полягає в підборі безлічі «підходящих» слотів, які:

- можна синхронно виділити в кількості, необхідній завданню;
- задовольняють ресурсному запиту завдання;
- підходять по вартості.

Приведемо алгоритм планування, що вирішує завдання коаллокації ресурсів з урахуванням угод по їхньому поділу, а також проведемо його обґрунтування й експериментальну оцінку, проведену на розробленому прототипі диспетчера завдань. Один із кращих на сьогоднішній день методів планування паралельних завдань – метод зворотного заповнення (Backfill) – в умовах невідчуждаємості ресурсів має квадратичну складність від загальної кількості слотів. Для цих умов у роботі пропонується алгоритм планування лінійної складності і його варіант, що реалізує критерій якнайшвидшого завершення завдань.

Алгоритм розрахований на наступні умови:

– Усі компоненти паралельного завдання мають ті самі вимоги до ресурсів комп'ютера, виражені в ресурсному запиту.

– Плата за кожний компонент завдання становить RC / P , де P – кількість компонентів, а RC – вартість завдання в цілому. Таким чином, відповідно до принципу поділу ресурсів, завданню можуть бути виділені тільки ті з них, які мають вартість, меншу RC / P .

Алгоритм підбирає слоти для одного завдання у два етапи й заснований на спільному використанні двох подань розкладу у вигляді списків L_x і L_2 , у першому з яких слоти відсортовані по зростанню часу початку, а в другому – по зростанню часу кінця. Це сортування виконується один раз для всієї черги завдань перед початком роботи алгоритму.

На першому етапі алгоритму підраховується кількість підходящих слотів у кожний момент часу. Зміна цього числа відбувається тільки в точках часової осі, що відповідають початку або кінцю якого-небудь слоту. Для підрахунку виконується паралельний прохід по списках L_1 і L_2 . При переході до нового слоту зі списку L_1 , час початку аллокації t встановлюється рівним початку цього слоту. Здійснюється перевірка, чи є слот підходящим для запуску завдання. Якщо це так, то кількість підходящих слотів збільшується. Після цього проходяться слоти другого списку, кінець яких менше $t + T$, і відповідно зменшується кількість підходящих слотів. Як тільки набирається необхідна кількість підходящих слотів P , цей етап роботи алгоритму завершується, і виходить час $I_0 = t$, починаючи з якого для завдання можна синхронно виділити необхідну кількість слотів.

Якщо на першому етапі необхідна кількість слотів знайдена, то всі слоти, що лежать на інтервалі $[t_0, t_0 + T]$ і які є підходящими, можуть бути використані для його запуску. Для їхнього відбору відбувається ще один прохід по першому списку (другий етап). Отриманий набір слотів (або їхніх частин) є шуканим набором аллокацій.

Представлений алгоритм визначає самий ранній час старту завдання. Якщо в системі є процесорні вузли з різною продуктивністю, то більше корисним є критерій якнайшвидшого завершення завдання. Для реалізації цього критерію пропонується модифікація алгоритму, що намагається в першу чергу підібрати слоти на найбільш продуктивних процесорах, щоб виконати завдання якнайшвидше.

Для аналізу ефективності розробленого алгоритму планування реалізований планувальник, розглянутий далі і підготовлене експериментальне середовище, заснована на подійному моделюванні. З його допомогою отримані показники, використовувані для оцінки ефективності алгоритмів планування: середній час очікування, час рахунку пакета завдань і ступінь завантаження ресурсів, а також побудована залежність середнього часу планування

від кількості завдань у системі. На основі отриманих результатів проведено порівняння розробленого алгоритму планування з алгоритмом FCFS в умовах GRID з невідчуваними ресурсами, що показав істотно більш високі показники ефективності планування першого алгоритму. Також досліджений вплив зміни вартості ресурсів на час відгуку й масштабованість планувальника.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів управління паралельними обчисленнями для наукових досліджень. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем управління паралельними обчисленнями для наукових досліджень; Досліджена система управління паралельними обчисленнями для наукових досліджень; На основі отриманих результатів досліджень створена програмна реалізація системи управління паралельними обчисленнями для наукових досліджень. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання управління паралельними обчисленнями для наукових досліджень. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.
2. Smirnov O., Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.
3. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.
4. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.
5. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.
6. Kuznetsova, T., «Code-Based Schemes for Post-Quantum Digital Signatures», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P. 707-712.
7. Smirnov, O., Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Krasnobaev, V., Kuznetsova K. «Information Hiding Using 3D-Printing Technology», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P.701-706.
8. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.
9. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів IEC60880 та IEC62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». Системи управління, навігації та зв'язку, 2023, вип. 3(73), С. 155-166.
10. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». Кібербезпека: освіта, наука, техніка. 2024. №3(23), С. 111-131.
11. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». Системи управління, навігації та зв'язку, 2023, вип. 2(72), С. 170-178.
12. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». Сучасні інформаційні системи, 2023, том 7, № 2, С. 49-56.
13. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А. «Дослідження нормативної документації та стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для

- безпеки». VI міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 20-21 квітня 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 35-36.
14. Смірнов, О.А., Усік П.С., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю. «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.
 15. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.
 16. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.
 17. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». Центральнотраїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.
 18. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
 19. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральнотраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.
 20. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018
 21. O. Smirnov, O. Kovalenko, A. Kovalenko, S. Smirnov, V. Vialkova. The mathematical model of the testing technology for Dom Xss vulnerabilities. Scientific & practical cyber security journal (SPCSJ) Vol 2 Issue 1, 22-28 pp. [Електронний Журнал]. Georgia. Tbilisi: SCSA – 2018.
 22. Oleksii Smirnov, Oleksandr Kovalenko, Jamil Al-Azzeh, Anna Kovalenko, Serhii Smirnov. Qualitative risk analysis of software development. Asian Journal of Information Technology. – Volume 17(3). – Medwell Journals. – 2018. – P. 218-230.
 23. Смірнов О.А., Коваленко О.В., Коваленко А.С., Смірнов С.А. Розробка методу передтестової компіляції й розподілу доступу. Збірник наукових праць III міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 19-20 квітня 2018р. – Кропивницький: ЦНТУ. – 2018. – С. 214-215
 24. Smirnov Oleksii, Kovalenko Oleksandr, Kovalenko Anna, Smirnov Serhii. Method of testing the dom xss vulnerability. International Conference «information technologies, systems and networks ITSN-2017». Chisinau, Republic of Moldova. 17 – 18 October 2017. – Chisinau: Academy of Sciences of Moldova, Military Academy of Armed Forces “Alexandru cel Bun”. 2017. P7.
 25. Смірнов О.А., Смірнов С.А., Коваленко О.В., Коваленко А.С. Технологія тестування DOM XSS уразливості. Науково-практичний журнал кібер безпеки (SPCSJ) № 1. [Електронний журнал]. Грузія. Тбілісі: SCSA - 2017.