УДК 621.316

**О.М. Данильченко, к.т.н., доц.**
**Т.А. Узденов, аспір.**
*Житомирський державний технологічний університет*

# ПРОЦЕС ДИСПЕТЧЕРИЗАЦІЇ ДЛЯ GRID-СИСТЕМ З НЕВІДЧУЖУВАНИМИ РЕСУРСАМИ

*У даній статті описано структуру розробленої моделі процесу диспетчеризації та загальні принципи її роботи для організації розподілених обчислень на GRID-системах з невідчужуваними ресурсами. Дана модель має ряд переваг серед інших. Вона спроектована на основі принципів об'єктно-орієнтованого програмування.*

**Вступ.** Під процесом диспетчеризації розподілених обчислень для GRID-систем з невідчужуваними ресурсами розуміють розподіл обчислювальних задач, які необхідно розв'язати доступними обчислювальними елементами (комп'ютерами). При цьому обчислювальні елементи географічно розподілені, тобто знаходяться в різних місцях і зв'язані між собою лініями зв'язку (комп'ютерними мережами). Також вважається, що дані обчислювальні елементи можуть одночасно використовуватися їх власниками, тобто на одному комп'ютері може виконуватись декілька процесів.

**Аналіз літературних джерел та останніх досліджень.** Як відомо [1], дана тема роботи набуває все більшої актуальності. Це зв'язано з тим, що з кожним днем все більше і більше людей приєднуються до всесвітньої мережі інтернет. При цьому потужності персональних ЕОМ на стільки виросли, що рідко який користувач використовує їх на всі сто відсотків. З [2] слідує, що величезні ресурси мережі інтернет можна використати для вирішення важливих для людства завдань.

**Мета роботи** – побудова моделі процесу диспетчеризації для GRID-систем з невідчужуваними ресурсами.

**Викладення основного матеріалу.** Програмний комплекс організації розподілених обчислень, побудований за загальноприйнятою для даного класу програм схемою. Система заснована на клієнт-серверній архітектурі. У даній архітектурі в мережі присутній комп'ютер-сервер, який перебуває в стані очікування запитів від клієнтів. Одержавши такий запит, сервер обробляє його й посилає результат клієнтові. Модель взаємодії процесів у системі – керуючі-робітники. У цій моделі в системі є спеціальна ЕОМ-керуючий, яка володіє набором підзадач – портфелем. Ця ЕОМ розподіляє підзадачі по доступних обчислювальних елементах – робітниках – і потім збирає результати [1]. Це варіант архітектури клієнт-сервер, у якому обчислювальні елементи відіграють роль серверів, а керуюча ЕОМ є клієнтом.

Загальну схему комплексу представлено на рисунку 1.

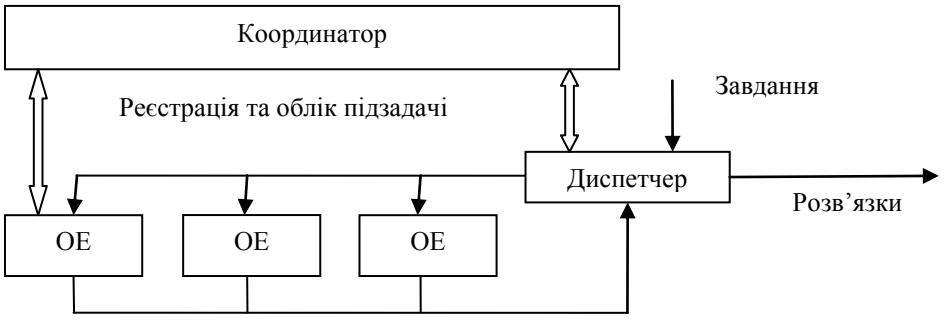


*Рис. 1. Загальна схема комплексу*

Програмно-алгоритмічний комплекс складається із трьох компонентів, відповідних до названих ролей ЕОМ. Кожний компонент реалізований у вигляді окремого програмного продукту [2].

Координатор – це програма, відповідальна за реєстрацію обчислювальних елементів, що належать до мережі і виділення їх для розв'язання задач при одержанні запиту від програми-диспетчера. У мережі може бути присутнім тільки один активний координатор.

Диспетчер – це програма, що запускається на кожній з ЕОМ, що належать мережі. Диспетчер відповідає за організацію процесу обчислень:

- формує запит до координатора на виділення ОЕ для проведення обчислень;
- формує окремі підзадачі, відправляє їх на обчислювальні станції, виділені координатором, зберігає й аналізує результати обчислень.

Обчислювальний агент – програма, що виконується на обчислювальних елементах. Вона вирішує три завдання:

- здійснює реєстрацію на координаторові при підключенні обчислювального елемента до мережі;
- збирає статистичні дані про функціонування обчислювального елемента;
- одержує дані про підзадачу від диспетчера, здійснює необхідні обчислення й відсилає результат назад диспетчерові.

Для розв'язання конкретної обчислювальної задачі із застосуванням комплексу необхідно створити набір модулів, що реалізують алгоритм розв'язку, відповідно до специфікації, що накладається комплексом.

Обчислювальна задача реалізується у вигляді бібліотеки, що динамічно завантажується (dll) та здійснює формування підзадач, збереження й аналіз отриманих результатів їх розв'язку. Даний модуль завантажується й виконується на ЕОМ, що відіграє роль диспетчера.

Підзадача є комбінацією вхідних даних і програмного модуля, що реалізує необхідний обчислювальний алгоритм. Модуль є бібліотекою, що динамічно завантажується (dll). Модуль завантажується й виконується на ЕОМ, що відіграє роль ОЕ.

Опишемо протокол взаємодії всіх трьох основних компонентів:

- у момент запуску програми-агента ОЕ здійснює підключення до зазначеного в настроюваннях агента координаторові, реєструється на ньому й переходить у режим очікування завдань;
- диспетчер повідомляє координатора про початок обчислювальної сесії;
- диспетчер здійснює запит до координатора на виділення ОЕ;
- координатор передає адреси вільних ОЕ диспетчерові й виключає їх зі списку вільних;
- диспетчер зв'язується з ОЕ;
- диспетчер запитує у ОЕ набір даних і оцінок, необхідних для здійснення процесу планування обчислень на даній ітерації;
- ОЕ обчислюють необхідні оцінки й відправляють їхньому диспетчерові;
- диспетчер робить побудову плану обчислень;
- диспетчер робить відправлення підзадач на ОЕ, відповідно до побудованого плану;
- ОЕ здійснюють необхідні обчислення й повертають диспетчерові результат;
- по закінченню обчислень диспетчер повідомляє координатора про завершення обчислювальної сесії;
- диспетчер звільняє виділені йому ОЕ;
- вільні ОЕ реєструються на координаторові як вільні й знову переходять у режим очікування завдань.

На координаторові в один момент часу може бути зареєстрована тільки одна обчислювальна сесія. У цей час інші диспетчери не можуть почати процес обчислень і очікують завершення поточної сесії. Протягом сесії всі ОЕ, що реєструються на координаторові, перенаправляються на диспетчер, що створив поточну сесію.

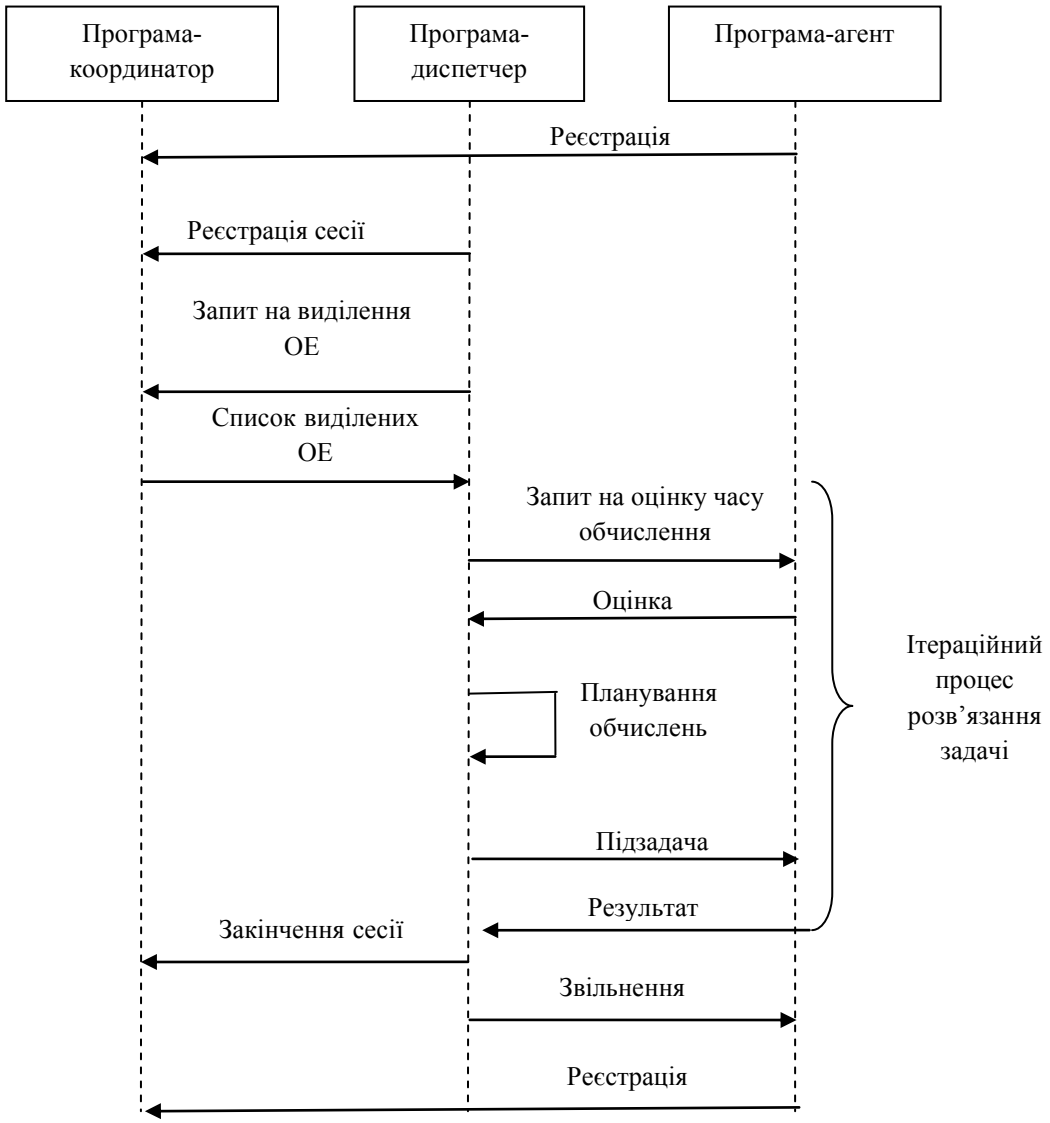


*Рис. 2. Взаємодія компонентів комплексу*

У загальному вигляді протокол взаємодії компонентів комплексу відображений у вигляді UML-діаграми на рисунку 2.

Опис програми-координатора

Як можна помітити, координатор безпосередньо не бере участі у процесі розв'язання задачі. Його призначенням є облік доступних ОЕ, тобто він вирішує завдання підтримки цілісності системи й керування ресурсами мережі.

Після запуску програма-координатор відображається у вигляді іконки в області нотифікацій на панелі задач. При клику по іконці відображається контекстне меню, що забезпечує доступ до спеціальних функцій додатка:

- виклик вікна настроювань програми;
- висновок списку вільних у цей момент ОЕ.

Опис програми-диспетчера

Програма-диспетчер відповідає за організацію процесу розподілу обчислень. З погляду внутрішньої структури диспетчер містить: пул підзадач, де перебувають ще не розв'язані підзадачі, планувальник плану, що здійснює складання призначень підзадач на доступні обчислювальні елементи, мережеві інтерфейси й завантажування модуля завдань.

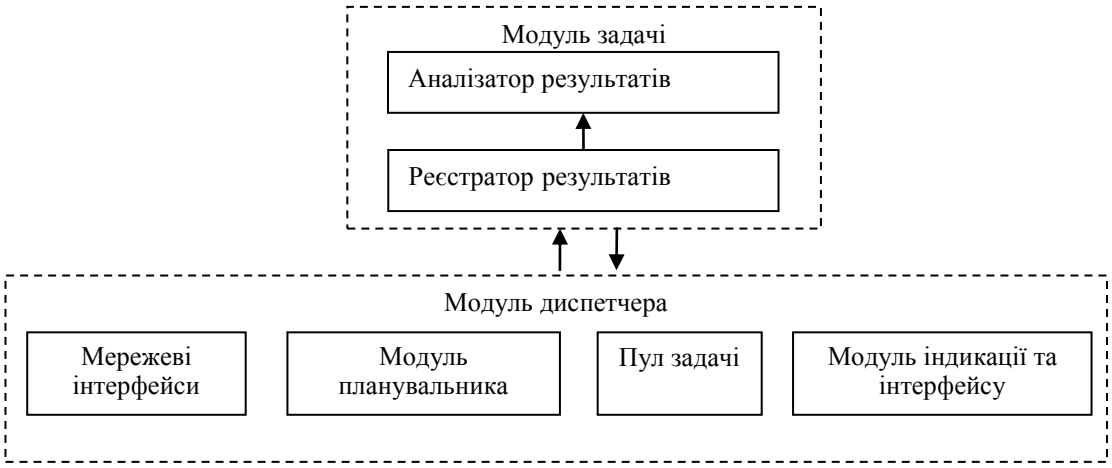


*Рис. 3. Структура програми-диспетчера*

Модуль завдання реалізує алгоритм розв'язку задачі, який містить формування підзадач, здійснює реєстрацію й аналіз результатів, отриманих у результаті розв'язання підзадач. На основі аналізу модуль завдання формує нові підзадачі або ухвалює рішення щодо припинення обчислень.

Схематично структуру програми-диспетчера зображено на рисунку 3.

Програма-диспетчер реалізована у вигляді віконного додатка. З головного вікна програми доступні такі функції:

- вказівка модуля завдання;
- запуск та зупинка обчислень;
- візуальна індикація прогресу вирішення завдання (на основі даних, що надає модуль завдання);
- вивід списку використовуваних у цей момент ОЕ;
- вивід журналу подій;
- виклик настроювань програми.

Опис програми-агента

Ключові моменти запропонованої методики, в основному, є в алгоритмах, реалізованих на боці програми-агента, що виконується на кожному ОЕ. Через це розглянемо структуру програми-агента більш докладно. Програма-агент виконує три функції:

- збір, обробка й аналіз статистичних даних;
- прогнозування;
- обслуговування запитів на розрахунки.

Загальну схему програми наведено на рисунку 4:

- модуль прогнозування за запитом від модуля обслуговування обчислень здійснює аналіз зібраних статистичних даних і формує оцінки запитаних величин;
- модуль обслуговування обчислень реалізує процес розрахунків і обмін даних із зовнішніми системами (розв'язок підзадач і обмін даними з координатором і диспетчером).

Після запуску програма-агент відображається у вигляді іконки в області нотифікацій на панелі задач. При клику по іконці відображається контекстне меню, що забезпечує доступ до спеціальних функцій додатка:

- виклик вікна настроювань програми;
- вивід графіка завантаження ЕОМ завданнями користувача;
- вивід виявлених паттернів завантаження;
- вивід журналу подій (відкриття текстового файла стандартною програмою виводу текстових файлів);
- зупинка й поновлення процесу обслуговування розподілених обчислень.

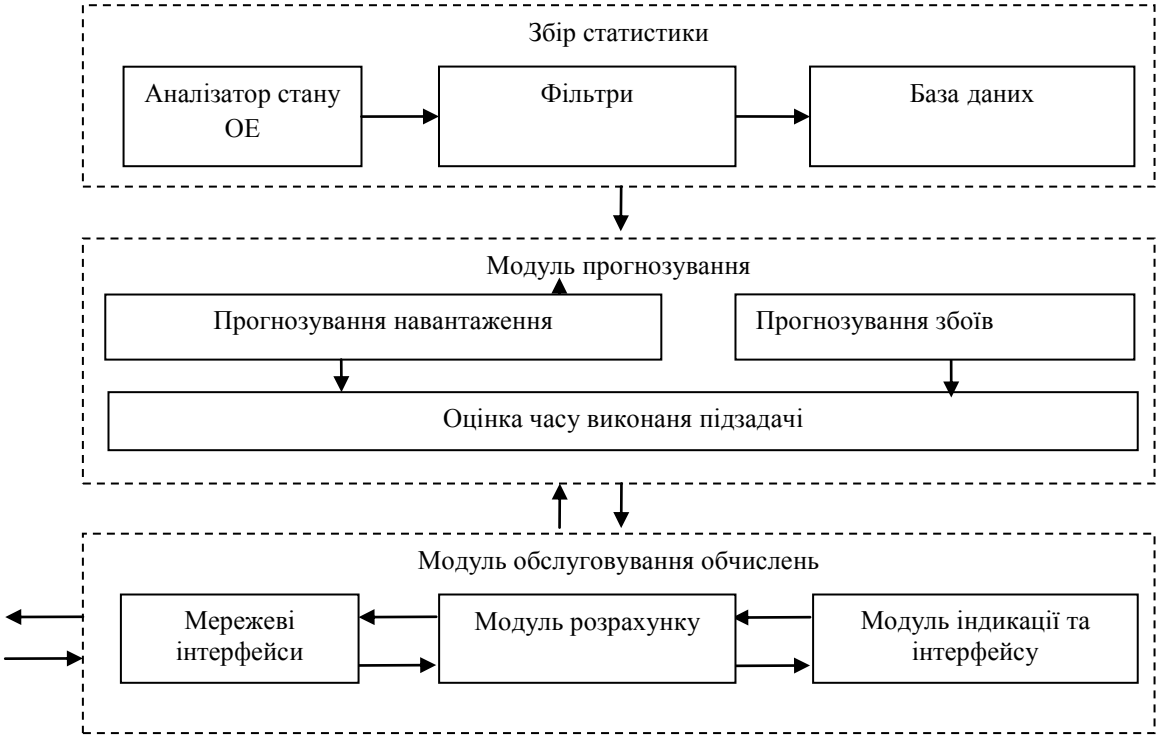


*Рис. 4. Структура програми-агента*

*Опис процесу розв'язання задачі*

Дамо формальний опис процесу розподіленого розв'язання задачі.

У кожний момент часу є черга обчислювальних підзадач і набір доступних обчислювальних елементів.

Підзадачі поділяться на три види, відповідно до типу події, у результаті настання якого дана підзадача була породжена:

- стартові (ініціювання процесу обчислень);
- породжені (завершення однієї з обчислювальних підзадач);
- повторні (збій обчислювального елемента).

Стартові підзадачі – це підзадачі, які ставляться у чергу в момент початкового старту обчислень, тобто вони відповідають початковій розбивці завдання.

Породжені підзадачі створюються на основі аналізу вже завершених підзадач, наприклад, у випадку недостатньої точності отриманого результату або невиконанні умови закінчення обчислень.

Повторні підзадачі – це підзадачі, які вже були передані на обчислювальні елементи, однак через якісь причини не були завершені. Умови відмови можуть бути як явні (повідомлення про припинення обчислень у результаті програмного збою), так і визначаються побічно (втрата зв'язку з елементом на тривалий проміжок часу).

Відсутність підзадач у черзі означає закінчення процесу обчислень за умови, що всі призначені елементи підзадачі вже завершені [3].

Підзадача може бути виключена із черги в результаті одного із двох подій:

- призначення підзадачі обчислювальному елементу;
- примусове припинення обчислень.

Після розв'язку підзадачі на ОЕ результат вертається диспетчерові.

Протокол взаємодії диспетчера й ОЕ при розв'язку окремої підзадачі представлений у вигляді UML-діаграми на рисунку 5. Стадії запиту оцінок про стан ОЕ й побудови плану обчислень на діаграмі опущені, тому що вони вже були розглянуті раніше.

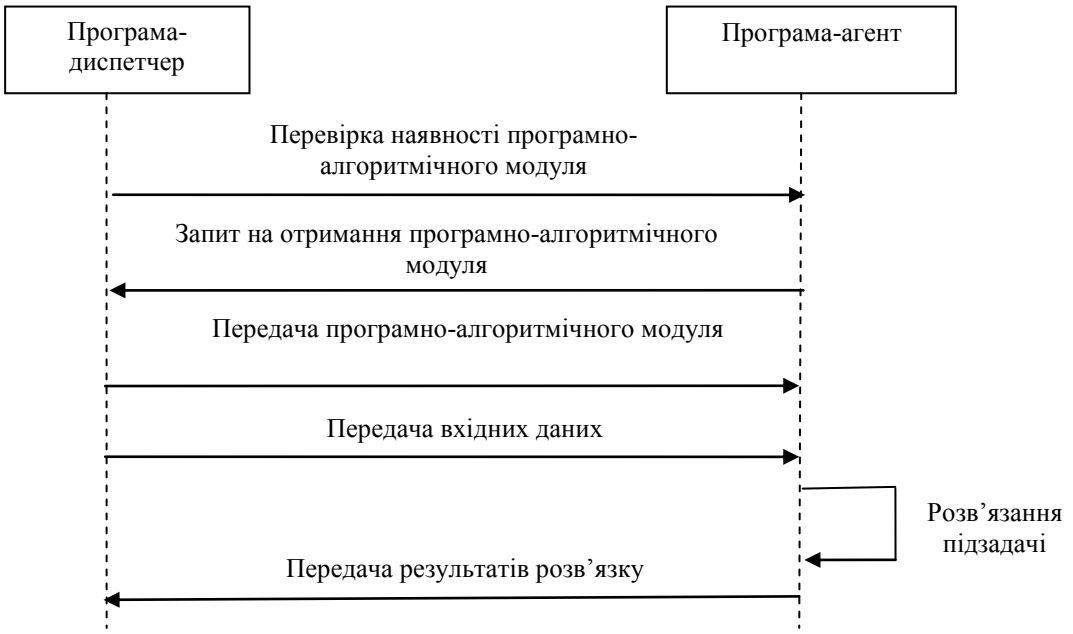


*Рис. 5. Протокол взаємодії диспетчера й ОЕ при призначенні підзадачі*

Розглянемо протокол взаємодії диспетчера й ОЕ детально.

1. На першому етапі диспетчер здійснює перевірку наявності на ОЕ програмного модуля, що реалізує розв'язання призначеної на ОЕ підзадачі. Модуль є бібліотекою, що динамічно завантажується (dll). Бібліотека однозначно ідентифікується іменем файла і його контрольною сумою (CRC). Додаткова перевірка контрольної суми дозволяє відрізняти однойменні програмні модулі різних версій. Зазначені дані передаються від диспетчера на ОЕ.

На диску ОЕ організована спеціальна ієрархія зберігання програмних модулів: кожний програмний модуль належить директорії, що збігається з іменем файла модуля, і потім у вкладену директорію з іменем, відповідним до строкової контрольної суми файла. Наприклад, за наявності двох версій модуля kinetics.dll з контрольними сумами 5A22702C і 85EC87F1 структура файлів на диску буде представлена таким чином:

- modules\Kinetics.dll\5A22702C\Kinetics.dll;
- modules\Kinetics. dll\85EC8 7F1 \Kinetics. dll.

2. У тому випадку, коли на диску ОЕ немає версії програмного модуля, що відповідає даним запиту, ОЕ відправляє запит диспетчера на одержання модуля.
3. У відповідь на цей запит диспетчер робить передачу модуля по мережі.
4. ОЕ, одержавши модуль, зберігає його на диску, відповідно до описаного принципу.
5. Проводиться передача вхідних даних підзадачі з диспетчера на ОЕ.
6. Одержавши дані, ОЕ завантажує програмний модуль і передає до нього дані на обробку.
7. По закінченню обчислень ОЕ передає результати диспетчерові. При цьому якщо в процесі обчислень виникає збій, то ОЕ припиняє обчислення й передає диспетчерові повідомлення про відмову. Повідомлення бувають двох видів: з вини ОЕ й програмного модуля розв'язання підзадачі. У випадку виникнення відмови з вини ОЕ підзадача вертається в чергу, де очікують, тобто стає повторною. Відмова через збій у програмному модулі розцінюється як непоправна ситуація й диспетчер ініціює припинення процесу обчислень поточної підзадачі.
8. На підставі отриманих результатів, диспетчер ухвалює рішення щодо необхідності продовження обчислень або їх закінчення.

Якщо під час проведення обчислювальним елементом розрахунків пропадає підключення до диспетчера, то після закінчення кожні 5 хвилин проводяться спроби відновити підключення (часовий інтервал обраний, виходячи із середнього часу розв'язання підзадачі). Якщо до моменту закінчення розрахунків підключення не було відновлено, результати обчислень відкидаються і ОЕ переходить у вільний стан, знову реєструючись на координаторові.

**Висновки.** Розроблено модель процесу диспетчеризації для GRID-систем з невідчужуваними ресурсами. Розроблено протоколи функціонування й взаємодії компонентів системи. Простота розробки модулів розв'язання обчислювальних завдань забезпечена за допомогою надання розробнику шаблонованого модуля задачі й підзадачі. На основі даної моделі реалізований програмний комплекс.

Він спроектований і реалізований на основі принципів об'єктно-орієнтованого програмування, що підвищує зручність його застосування й полегшує подальший розвиток. Простота розробки модулів розв'язку обчислювальних завдань забезпечена за допомогою надання розробнику шаблонованого модуля задачі й підзадачі.

**Список використаної літератури:**


1. *Данильченко О.М.* Проблеми розробки методів керування паралельними завданнями та їх алгоритмічної підтримки для актуальної форми GRID / *О.М. Данильченко, Т.А. Узденов* // тези XXXVI наук.-практ. міжвуз. конф., присвяченої Дню науки 12–13 травня 2011 р. – Житомир : ЖДТУ, 2011. ─ Т. І. ─С. 51–52.
2. Грид-диспетчер: реализация службы диспетчеризации заданий в грид / *В.Н. Коваленко, Е.И. Коваленко, Д.А. Корягин и др.* // труды Междунар. конф. "Распределенные вычисления и Грид-технологии в науке и образовании" (29 июня–2 июля 2004 г, г. Дубна). – Дубна : ОИЯИ, 2004. – С. 133–139.
3. *Данильченко О.М.* Моделювання програмного комплексу для дослідження паралельних алгоритмів на кластерній системі / *О.М. Данильченко, Т.А. Узденов* // тези наук.-практ. міжвуз. конф., 18–19 січня 2010 р. – Житомир : ЖДТУ, 2010. – С. 36–37.



ДАНИЛЬЧЕНКО Олександр Михайлович – кандидат технічних наук, доцент, завідувач кафедри програмного забезпечення обчислювальної техніки, член Вченої ради Житомирського державного технологічного університету.

Наукові інтереси:
- теорія розкладів;
- теорія складності екстремальних задач, бази даних.

УЗДЕНОВ Тарас Амурович – аспірант кафедри ПЗОТ Житомирського державного технологічного університету.

Наукові інтереси:
- теорія розкладів;
- розподілені системи;
- паралельне програмування.

Тел.: 0632823417.

E-mail: uzdenov_taras@mail.ru


Стаття надійшла до редакції 25.05.2012

**Данильченко О.М., Узденов Т.А.** Процес диспетчеризації для GRID-системи з невідчужуваними ресурсами
**Данильченко О.М., Узденов Т.А.** Процес диспетчеризации для grid-систем с неотчуждаемыми ресурсами
**Данильченко О.М., Uzdenov Ò.À.** Proces of the control centralized traffic for the grid-systems with inalienable resources

УДК 621.316

**Процес диспетчеризации для grid-систем с неотчуждаемыми ресурсами / О.М. Данильченко, Т.А. Узденов**

В данной статье расписывается структура разработанной модели процесса диспетчеризации и основные принципы ее работы для организации распределенных вычислений на GRID-системах с неотчуждаемыми ресурсами. Эта модель имеет ряд преимуществ перед другими. Она спроэктована на основе принципов объектно-ориентированного программирования.

УДК 621.316

**Proces of the control centralized traffic for the grid-systems with inalienable resources / O.M. Данильченко, Ò.À. Uzdenov**

In this article the structure of developed model of process of the control centralized traffic and basic principles of its work is painted for organization of the up-diffused calculations on the grid-systems with inalienable resources. This model has a row of advantages before other. It developed on the basis of principles of the object-oriented programming.