

УДК 004.94

В.В. ЛИТВИНОВ, І.В. СТЕЦЕНКО

**УПРАВЛІННЯ РОЗПОДІЛЕНИМИ РЕСУРСАМИ ГРІД-СИСТЕМИ**

**Анотація.** Розглядається управління розподілом ресурсів грід-системи серед її користувачів. Побудована Петрі-об'єктна модель розподілу ресурсів одно- та дворівневої грід-системи, що дозволяє оцінювати ефективність вибраної стратегії управління.

**Ключові слова:** грід-система, розподілена система, моделювання, мережа Петрі, об'єктно-орієнтована технологія.

**Аннотация.** Рассматривается управление распределением ресурсов грид-системы среди ее пользователей. Построена Петри-объектная модель распределения ресурсов одно- и двухуровневой грид-системы, позволяющая оценить эффективность выбранной стратегии управления.

**Ключевые слова:** грид-система, распределенная система, моделирование, сеть Петри, объектно-ориентированная технология.

**Abstract.** Management of grid system distributed resources among its users is regarded. We constructed a Petri-object model of distributed resources of one- and two-level grid system that allows you to evaluate the effectiveness of selected management strategy.

**Keywords:** grid computing, distributed system, simulation, Petri net, object-oriented technology.

**1. Вступ**

Проектування та розробка грід-систем є перспективним напрямом розвитку інформаційних систем, що надає можливість створення потужних обчислювальних систем за рахунок вільного об'єднання (можливо, тимчасового) географічно віддалених користувачів. Розрізняють обчислювальні грід-системи (Computing Grid), інформаційні грід-системи (Data Grid) та мішані грід-системи, користувачі яких мають спільний доступ як до обчислювальних ресурсів, так і до сховищ даних грід-системи.

На сьогоднішній день розроблені ефективні програмні та інструментальні засоби створення грід-систем. Найбільш відомими відкритими програмними продуктами, що забезпечують стандартний набір бібліотек для інтеграції користувачів, є Globus Toolkit та Sun Grid Engine. Комерційна версія останнього (Sun Grid Engine, Enterprise Edition) призначена для управління ресурсами підприємств і спроможна обслуговувати кілька незалежних проектів та груп користувачів.

Основним питанням, що залишається недостатньо дослідженим, є алгоритм розподілу обчислювальних та інформаційних ресурсів, які знаходяться у спільному користуванні користувачів грід-системи. Користувачі прагнуть отримувати бажані ресурси з найменшим часом очікування, а проектувальники систем управління забезпечують насамперед несуперечність прав доступу до ресурсів грід-системи, відсутність конфліктних ситуацій, відсутність відмов обслуговування.

Огляд публікацій [1–4], що розглядають розподіл ресурсів грід-систем, свідчить про різноманітність підходів до розв'язання проблеми, а також про необхідність розробки ефективних способів моделювання різних стратегій управління ресурсами розподілених систем. Існуючі системи моделювання грід-систем, такі як MicroGrid, OptorSim, GridSim, SimGrid, мають ряд недоліків, пов'язаних з вузькою спеціалізацією, обмеженістю моделюваної архітектури грід-систем, недостатньою гнучкістю у визначенні способу управління [5].

В [6] розглядається застосування базової мережі Петрі для опису процесів управління грид-системою. Але відсутність динаміки функціонування даної моделі не дозволяє оцінити часові характеристики функціонування грид-системи, такі як середній час очікування завдання, середній час обслуговування, середнє завантаження ресурсів, що являються найважливішими при виборі стратегії управління. В [7] використовуються E-мережі для розробки розподіленої системи імітаційного моделювання.

У даній публікації розглядається моделювання розподілу ресурсів з використанням Петрі-об'єктного підходу[8], що дозволило розробити динамічну модель функціонування грид-системи.

## 2. Архітектура грид-системи

Грид-системою є вільне об'єднання обчислювальних та, можливо, інформаційних ресурсів користувачів. Результатом такого об'єднання є створення спільного обчислювального ресурсу, який отримав назву віртуального обчислювального ресурсу (або грид-ресурсу). Архітектура грид – системи складається з метапланувальника системи та розподілених обчислювальних ресурсів. Якщо обчислювальні ресурси представлені комп'ютерами користувачів, а планувальник управляє розподілом ресурсів між користувачами системи, то грид-система – однорівнева. Якщо обчислювальні ресурси згруповані в обчислювальні вузли, а розподіл ресурсів здійснюється спочатку метапланувальником серед обчислювальних вузлів (кластерів) системи, а потім локальні планувальники здійснюють розподіл ресурсів серед користувачів кластера, то грид-система – дворівнева.

Метапланувальник взаємодіє не з апаратними ресурсами, а з грид-сервісами, що представляють ці ресурси. Локальні планувальники забезпечують управління виконанням завдань локальним обчислювальним ресурсом і взаємодіють з метапланувальником. Метапланувальник та локальні планувальники забезпечують синхронізований та несуперечливий доступ та використання користувачами грид-системи спільних обчислювальних та інформаційних ресурсів. У розподілених обчислювальних вузлах та сховищах даних встановлені грид-сервіси програмної інфраструктури, які надають інформацію про поточний стан ресурсу планувальнику виконання завдань. Узагальнена архітектура грид-системи представлена на рис. 1.

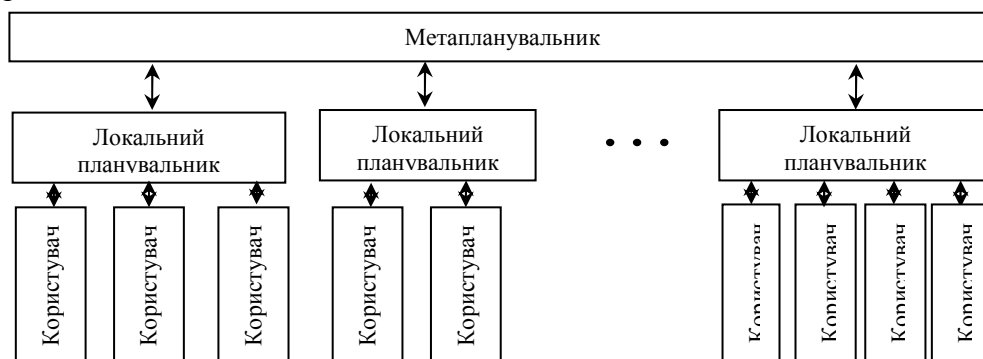


Рис. 1. Архітектура дворівневої грид-системи

Окрім несуперечливого та безвідмовного доступу до ресурсів до планувальників ставиться також задача забезпечення якісного управління ресурсами. Управління ресурсами вважається якісним, якщо досягається найбільш високий ступінь задоволення потреб користувачів та ступінь використання системних ресурсів.

## 3. Управління грид-ресурсами

Задача управління грид-ресурсами полягає в тому, щоб на основі інформації про поточний стан розподілених обчислювальних ресурсів та сховищ даних надавати дозвіл на викорис-

тання віртуального обчислювального ресурсу користувачам грид-системи в залежності від їх потреби та у відповідності до їх вкладу у віртуальний обчислювальний ресурс (або пріоритету використання ресурсу), а також надавати несуперечливий доступ до інформаційного ресурсу. Ефективність управління оцінюється такими величинами, як середній час очікування завдань, інтенсивність відмов обслуговування завдань користувача, інтенсивність споживання обчислювального ресурсу користувачами, середня кількість вільного віртуального ресурсу та ін.

Програмний пакет Sun Grid Engine, Enterprise Edition (SGEEE) надає можливість вибору однієї з чотирьох політик розподілу ресурсів між незалежно працюючими користувачами: 1) політика розділених ресурсів, 2) функціональна політика, 3) політика „роботи до строку”, 4) політика явного виділення ресурсів [9]. У першій – передбачається компенсація користувачу за переспоживання чи недоспоживання обчислювального ресурсу (у порівнянні з обсягом обчислювального ресурсу, на який претендує користувач за домовленістю) протягом певного часу. В другій – компенсація не передбачається, а кожний користувач споживає обчислювальний ресурс виключно в межах домовленості. В третій – створюються пільгові умови для користувачів, які поспішають виконати назначений обсяг робіт до указанного строку. В четвертій – використовується „ручне управління” розподілом ресурсів у відповідності з пріоритетом користувачів. В усіх запропонованих політиках користувач отримує дозвіл на використання ресурсу в залежності від наданої йому за домовленістю квоти на використання ресурсу, що визначається часткою віртуального ресурсу, на яку претендує користувач, або пріоритетом користувача на використання ресурсу.

Поняття „справедливого” управління розподілом ресурсів пов’язують з таким розподілом ресурсів грид-системи, при якому віртуальний обчислювальний ресурс розподіляється між активними користувачами пропорційно їх внеску у загальний віртуальний обчислювальний ресурс [1].

Припустимо, що деякі користувачі об’єдналися у грид-систему. Обсяг віртуального обчислювального ресурсу дорівнює сумі обсягів обчислювальних ресурсів користувачів, які вони надають у спільне користування. У договорі об’єднання прописується квота, що дорівнює відсотку загального ресурсу, на який претендує користувач. Найбільш справедливим вважається така квота, яка дорівнює внеску обчислювального ресурсу користувача у грид-систему у відсотках від загального обсягу. Але в кожний момент часу не всі користувачі активні. Отже, загальний ресурс розподіляється планувальником тільки між активними користувачами. Розподіл між активними користувачами здійснюється пропорційно їх квотам. Тому, якщо користувач має внесок 20% у загальний ресурс, але в поточний момент часу активним являється тільки він один, то він заволодіває усім віртуальним обчислювальним ресурсом.

Розподіл ресурсів здійснюється планувальником на визначений інтервал часу. Планувальник, реалізований SGEEE, розподіляє ресурси, виходячи із співвідношення білетів, „закуплених” на даний інтервал часу. Користувачі „купають квитки”, тобто посилають планувальнику замовлення ресурсу для того, щоб отримати дозвіл на використання ресурсу у наступний інтервал часу. Планувальник, виходячи з одержаних замовлень на використання ресурсів, здійснює розподіл ресурсів. Для кожного користувача підраховується усереднене фактичне використання обчислювального ресурсу, і якщо воно перебільшує (або зменшує) надану квоту, то спрацьовує механізм компенсації пере- (недо-) споживання обчислювального ресурсу. Тобто, якщо користувач певний час не використовував ресурс, то його компенсаційний коефіцієнт збільшився, і він має право отримати більшу частину ресурсу. Якщо завдання, які захопили ресурс (раніше інших завдань), використовують ресурс більше наданої квоти (більше, ніж надає квиток), то користувач лишається права отримати певну кількість квитків.

#### 4. Математична модель розподілу обчислювальних ресурсів грід-системи

Припустимо, що  $N$  користувачів  $U = \{i\}$   $|U| = N$  об'єдналися у дворівневу грід-систему. Сума обчислювальних ресурсів користувачів дорівнює віртуальному обчислювальному ресурсу (ОР) грід-системи  $S = \sum_i S_i$ . Здатність обчислювального ресурсу вимірюється, наприклад, в одиницях MIPS (Million Instruction Per Second). Отже, кожний  $i$ -ий користувач має внесок до віртуального обчислювального ресурсу, який вимірюється часткою

$$p_i = \frac{S_i}{S}, \sum_i p_i = 1. \quad (1)$$

Якщо грід-система однорівнева, то користувачі мають доступ безпосередньо до віртуального ОР і частка (1) представляє квоту користувача грід-системи. Якщо грід-система дворівнева, то користувачі мають доступ до віртуального ОР через вузли грід-системи, що представляють в загальному випадку як кластери (локальні об'єднання користувачів), так і окремих користувачів. Будемо вважати, що множина користувачів  $U = \{i\}$  складається з підмножин  $U_j$   $|\cup_j U_j = U, \cap_j U_j = \emptyset$ . Тобто кожний користувач належить одній і тільки одній підмножині  $U_j$ :  $i \in U_j$ , що відповідає вузлу грід-системи.

Здатність обчислювального ресурсу вузла грід-системи  $C_j = \sum_{k \in U_j} S_k$ ,  $j = 1, \dots, K$ .

Оскільки у дворівневій грід-системі користувач має доступ до віртуального ОР через посередництво кластера, то представимо квоту використання ресурсу користувача (1) в такому вигляді:

$$p_i = \frac{S_i}{C_j} \frac{C_j}{S}. \quad (2)$$

Частка  $\frac{C_j}{S}$  представляє квоту вузла обчислювального ресурсу:

$$r_j = \frac{C_j}{S}, \sum_j r_j = 1. \quad (3)$$

Користувач вузла грід-системи має квоту на використання обчислювальних ресурсів вузла, пропорційну його внеску в обчислювальний ресурс вузла:

$$q_i = \frac{S_i}{C_j}, \sum_i q_i = 1. \quad (4)$$

Отже, квота (2) приймає вигляд:

$$p_i = \begin{cases} q_i \cdot r_j, & i \in U_j, \\ 0, & i \notin U_j. \end{cases} \quad (5)$$

Користувач може знаходитись в активному стані і пасивному. Зміна стану відбувається з інтервалами часу, визначеними випадковою величиною з відомим законом розподілу. Позначимо множину активних користувачів грід-системи  $A \subseteq U$ . Активні користувачі отримують доступ до віртуального ОР пропорційно своїй квоті:

$$x_i = \frac{p_i}{\sum_{i \in A} p_i}, \sum_i x_i = 1. \quad (6)$$

Наприклад, користувачі  $a, b, c, d$ , які об'єднались, мають квоти  $p_a = 0,4$ ,  $p_b = 0,1$ ,  $p_c = 0,2$  і  $p_d = 0,3$ . Якщо в поточний момент часу активними є користувачі  $a, b$ , то вони отримують доступ до віртуального ОР у частках  $p_a = \frac{0,4}{0,4+0,1} = 0,8$ ,  $p_b = \frac{0,1}{0,4+0,1} = 0,2$ .

Якщо ж активними є користувачі  $a, c$  і  $d$ , то  $p_a = \frac{0,4}{0,4+0,2+0,3} = 0,44$ ,  $p_c = \frac{0,2}{0,4+0,2+0,3} = 0,22$ ,  $p_d = \frac{0,3}{0,4+0,2+0,3} = 0,33$ . Це означає, що частка доступного віртуального ОР користувача у випадку однорівневої грид-системи розраховується за формулою

$$\forall i \quad x_i = \begin{cases} \frac{p_i}{\sum_{i \in A} p_i}, i \in A, \\ 0, i \notin A, \end{cases} \quad (7)$$

де  $A \subseteq U$  – підмножина активних користувачів грид-системи.

Якщо грид-система дворівнева, то частка доступного віртуального ОР розраховується з урахуванням активності вузла грид-системи (вузол активний, якщо хоч один його користувач активний):

$$\forall i \in U, j \in K \quad x_i = \begin{cases} \frac{q_i}{\sum_{i \in A \cap U_j} q_i} \cdot \frac{r_j}{\sum_{j \in K_A} r_j}, (i \in A) \wedge (i \in U_j), \\ 0, (i \notin A) \vee (i \notin U_j), \end{cases} \quad (8)$$

де  $K_A \subseteq K$  – підмножина усіх активних вузлів,  $A \cap U_j$  – підмножина активних користувачів грид-вузла, до якого належить користувач  $i$ .

Обсяг доступного віртуального ресурсу користувача дорівнює  $Z_i = x_i \cdot S$ . Оскільки  $\sum_i x_i = 1$ , то  $\sum_i Z_i = S$ .

Користувач, якщо він активний, генерує завдання, які вимагають для свого виконання певний обсяг обчислювального ресурсу, і розміщує їх у буфері завдань. Завдання, яке знаходиться у буфері завдань, очікує дозволу від планувальника грид-ресурсів на захоплення ресурсу. Оскільки буфер завдань користувача обмежений, то завдання може отримати відмову. На початку кожного такту планування завдання, обсяг вимоги ОР яких не перевищує обсяг доступного віртуального ресурсу користувача  $Z_i$ , отримують дозвіл на захоплення віртуального ОР і захоплюють ресурс на час, рівний тривалості виконання завдання.

Локальний планувальник отримує від метапланувальника інформацію про частку доступного віртуального ОР  $r_j$  на поточний такт управління, а від користувачів – інформацію про поточний стан користувача. Локальний планувальник розподіляє доступний ві-

ртуальний ресурс вузла грід-системи  $\frac{r_j}{\sum_{j \in K_A} r_j} \cdot S$  між усіма своїми активними користувачами

$i \in U_j \cap A$ :

$$\forall j \in K_A \quad \frac{r_j}{\sum_{j \in K_A} r_j} \cdot S = \sum_{i \in U_j \cap A} Z_i. \quad (9)$$

Якщо у буфері завдань користувача є завдання, обсяг потреби ОР якого не перевищує обсяг доступного ОР користувача, то користувач отримує дозвіл на захоплення ресурсу. Ясно, що швидше доступ отримують користувачі, які переглядаються планувальником „у першу чергу”. Звідси локальний планувальник переглядає користувачів у порядку спадання значення пріоритету користувача. Очевидно, що від пріоритетів сильно залежить використання ресурсу. Наприклад, якщо користувачі з маленькими потребами ресурсу стоять у черзі першими, то користувачі з великими потребами ОР так і не отримують дозвіл на використання ресурсу через його недостатність. Звідси слідує, що інформація про потік завдань від користувача необхідна для ефективного управління розподілом ресурсів. Замість пріоритетів може реалізовуватись також перегляд користувачів у випадковому порядку. У випадках, коли потік завдань змінюється в часі, такий спосіб може бути більш ефективним, ніж пріоритети користувачів.

## 5. Петрі-об’єктна модель розподілу ресурсів грід-системи

Для побудови моделі використаємо технологію Петрі-об’єктного моделювання, запропоновану в [8]. Об’єктно-орієнтований аналіз грід-системи дозволяє виділити такі складові об’єкти системи: Планувальник, Користувач, Завдання. Усі об’єкти є динамічними, тому розробимо відповідні Петрі-об’єкти. Динаміка функціонування Петрі-об’єкта задається мережею Петрі і передається у відповідне поле Петрі-об’єкта під час конструювання екземплярів Петрі-об’єктів. Для опису динаміки Петрі-об’єктів використовується стохастична мережа Петрі з часовими затримками, багатоканальними та конфліктними переходами, інформаційними зв’язками [10].

Петрі-об’єкт Планувальник містить поля і методи, що встановлюють вибір стратегії планування (формули розрахунку частки доступного ресурсу кожного користувача) та основні параметри планування (тривалість такту управління, обсяг вільного віртуального обчислювального ресурсу у початковий момент часу). Мережа Петрі об’єкта Планувальник представлена на рис. 2 (переходи, які мають більший пріоритет, позначені на рисунках жирною лінією) на прикладі розподілу між чотирма користувачами  $A, B, C, D$ . Розрахунок частки доступного ресурсу кожного користувача здійснюється у відповідності до вибраної стратегії за інформацією про кількість активних користувачів, наприклад, за формулою (7). За результатом розрахованої частки доступного ресурсу задаються ймовірності запуску переходів „A”, „B”, „C”, „D”:  $R_i = x_i, i \in \{A, B, C, D\}$ , де  $R_A, R_B, R_C, R_D$  – ймовірності запуску переходів „A”, „B”, „C”, „D” відповідно,  $x_A, x_B, x_C, x_D$  – частки доступного ресурсу користувачів  $A, B, C, D$ .

Якщо стратегія управління передбачає пріоритет користувачів, то ймовірність запуску відповідного переходу встановлюється, і весь вільний віртуальний ОР надходить у використання користувача з найбільшим пріоритетом.

Зауважимо, що використовуються багатоканальні переходи мережі Петрі, тому перехід „Передати інформацію про обсяг вільного віртуального ОР” при виконанні умови запуску переходу спрацьовує стільки разів, скільки в ньому маркерів на поточний момент часу (рис. 2).

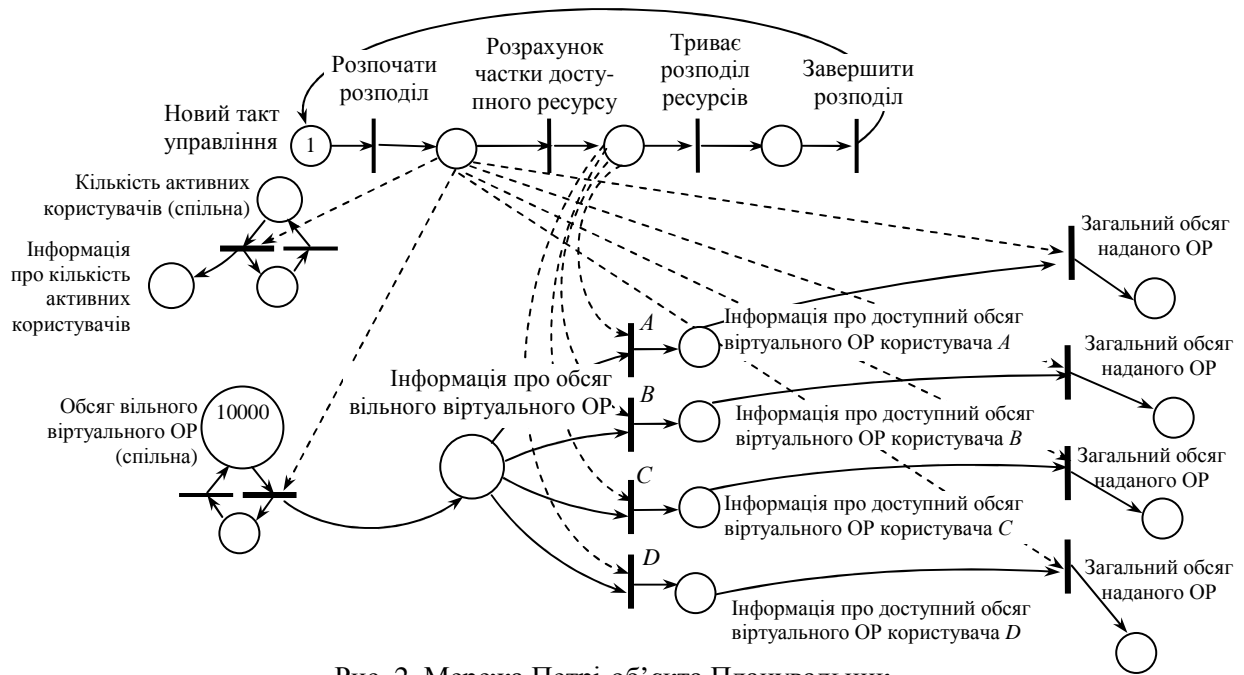


Рис. 2. Мережа Петрі-об'єкта Планувальник

Петрі-об'єкт Завдання моделює виконання одного завдання користувача з використанням обчислювального ресурсу грид-системи. Інформація про доступний обсяг віртуального ОР надходить до цього об'єкта з Петрі-об'єкта Планувальник. Оскільки позиція є спільною для всіх Петрі-об'єктів Завдання користувача та Петрі-об'єкта Планувальник, то доступний ресурс користувача використовується усіма завданнями. Кількість завдань дорівнює розміру буфера завдань грид-системи. Після багатократного запуску переходу „Найменше зі значень” в одній із вхідних позицій цього переходу досягається нульове маркірування. Якщо ненульовим виявилось маркірування в позиції „Потреба завдання в ОР”, то потреба перевищує доступний обсяг ресурсу, і спрацьовує перехід „Недостатньо”. Якщо ж виявилось ненульовим маркірування в позиції „Інформація про доступний обсяг віртуального ОР користувача”, то потреба завдання в ОР менша за доступний обсяг, і спрацьовує перехід „Достатньо”. При надходженні завдання на виконання здійснюється захоплення ОР з віртуального ОР в обсязі, що дорівнює потребі завдання, на час, що дорівнює тривалості виконання завдання. Якщо ресурсу недостатньо, то відбувається вивільнення маркерів з позицій „Найменше зі значень” і „Потреба завдання в ОР”. Відповідна мережа Петрі об'єкта Завдання представлена на рис. 3.

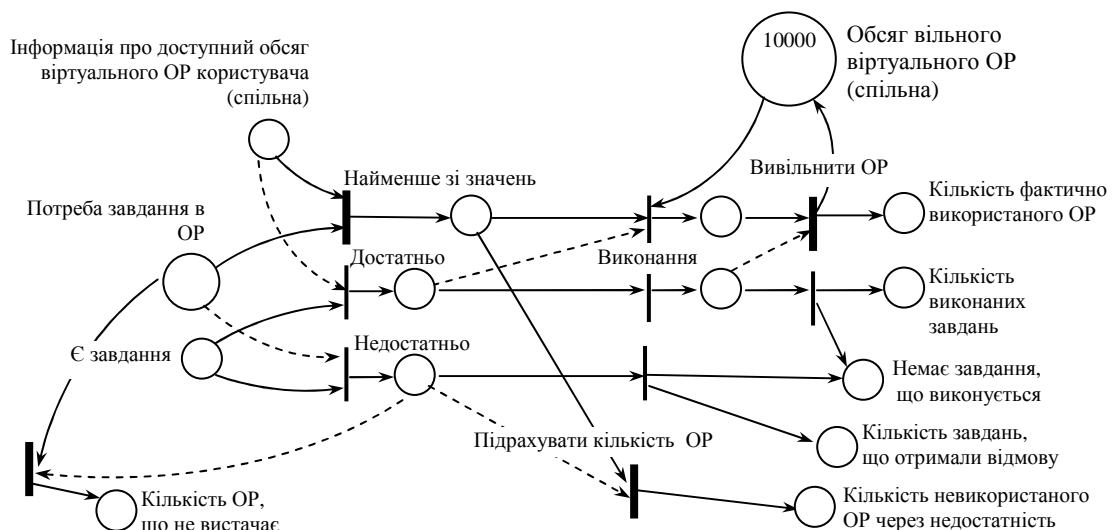


Рис. 3. Мережа Петрі-об'єкта Завдання

Петрі-об'єкт Користувач містить інформацію про параметри, що характеризують індивідуальність користувача (інтервали часу активності та пасивності, інтенсивність генерування потреби в ОР), та інформацію, що характеризує якість обслуговування користувача (кількість відмов, обсяг невдоволеної потреби користувача в ОР). Користувач, який знаходиться в активному стані, генерує потребу в ОР із заданою інтенсивністю. Сформоване завдання направляється на виконання в Петрі-об'єкт Завдання за умови, що в ньому немає завдання, яке виконується. У протилежному випадку завдання направляється в буфер або на відмову, якщо всі місця в буфері зайняті. Переходи „Поставити першим у буфер завдань”,... „Поставити  $n$ -им у буфер завдань” мають значення пріоритету, що відповідають правилу обслуговування черги завдань. Мережа Петрі об'єкта Користувач представлена на рис. 4. З'єднання спільних позицій Петрі-об'єктів Користувач і Завдання здійснюється в конструкторі Петрі-об'єкта Користувач, якщо в аргументі конструктора передається список Петрі-об'єктів Завдання.

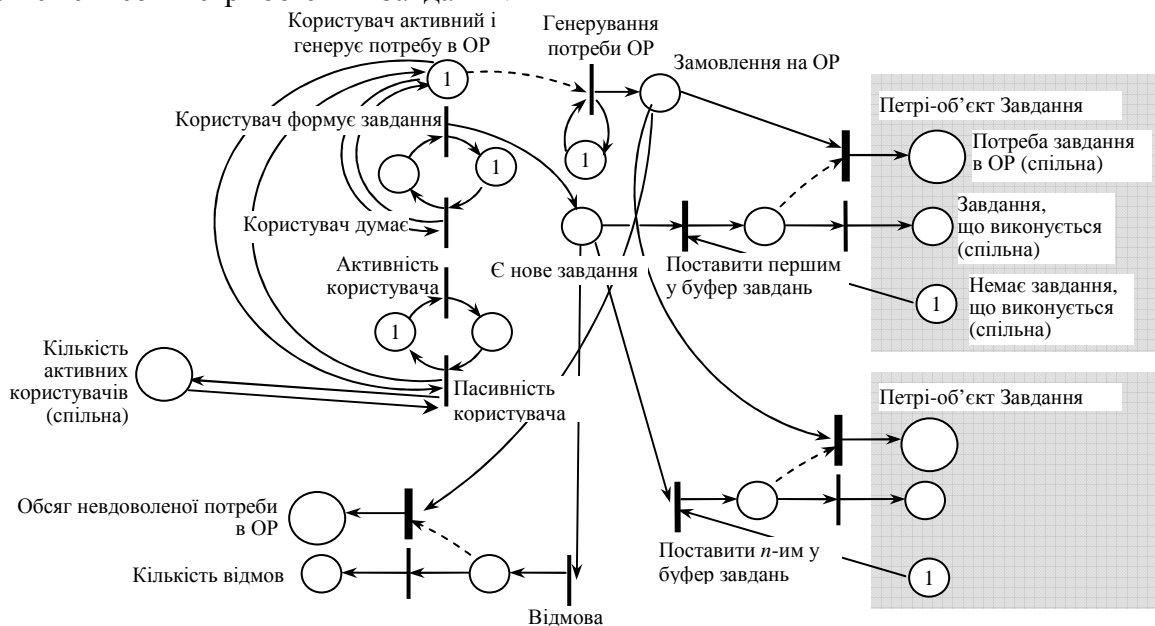


Рис. 4. Мережа Петрі-об'єкта Користувач

Якщо користувачі грид-системи використовують, окрім обчислювальних ресурсів, інформаційні ресурси (ІР), то доступ до кожного ІР надається в кожний такт управління тільки одному користувачу. До Петрі-об'єкта Користувач додається генерування потреби в ІР, а до Петрі-об'єкта Планувальник додається позиція Вільний ІР, маркер якої захоплюється на початку такту управління одним із користувачів на час, що дорівнює тривалості такту управління. Якщо грид-система дворівнева, то додається Петрі-об'єкт Метапланувальник з динамікою функціонування, що описується мережею Петрі, яка аналогічна мережі Петрі об'єкта Планувальник. Структура взаємозв'язків Петрі-об'єктів представлена на рис. 5.

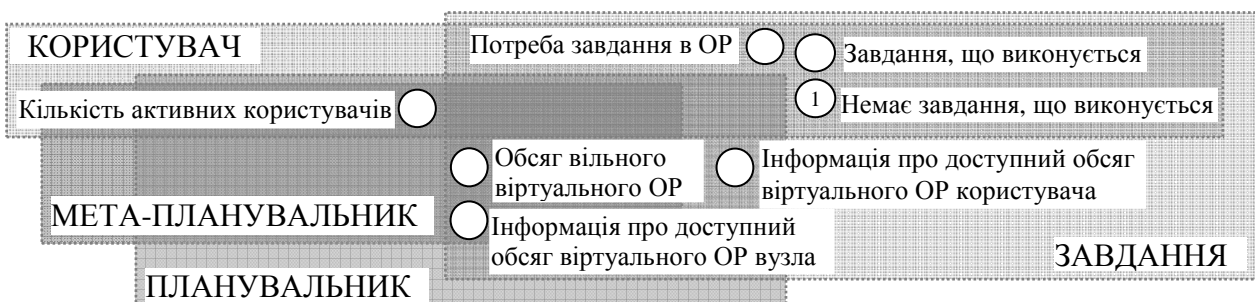


Рис. 5. Зв'язки між Петрі-об'єктами Планувальник, Користувач, Завдання



Реалізація моделі виконана з використанням бібліотеки Java-класів PetriObj [11]. За результатами моделювання визначаються середній час очікування завдання користувача; середня кількість завдань, що очікують доступу до грід-ресурсів; кількість виконаних завдань; кількість завдань, що отримали відмову через недостатність місця в буфері завдань; кількість завдань, що отримали відмову через недостатність ресурсу; кількість фактично використаного ресурсу; кількість доступного ресурсу; кількість ресурсу, що не вистачає для безвідмовної роботи системи. Модель може бути використана для пошуку найбільш ефективних стратегій управління розподілом ресурсів, що задовольняють вимогам користувачів грід-системи. Наприклад, на рис. 6 наведені результати дослідження впливу статичного та динамічного типів управління. Динамічне управління розподілом ресурсів передбачає визначення частки ресурсу користувачу грід-системи з урахуванням кількості активних користувачів за формулою (7), а статичне управління – без урахування кількості активних користувачів ( $x_i = p_i = const$ ). Критерієм, що визначає відповідність вимогам користувачів, обрана відносна пропускна спроможність системи.



Рис. 6. Дослідження впливу типу управління на ефективність функціонування системи

З результатів дослідження слідує, що при малих обсягах ресурсів використання динамічного управління надає можливість збільшити відносну пропускну спроможність системи у кілька разів, а при великих обсягах ресурсів ефект від використання динамічного управління хоч і присутній, але незначний.

## 6. Висновки

Таким чином, у результаті наукового дослідження розроблена імітаційна модель управління розподілом ресурсів грід-системи, яка дозволяє враховувати індивідуальні властивості користувачів та динамічне захоплення віртуального обчислювального ресурсу користувачами грід-системи в залежності від обраної стратегії управління.

Формалізація моделі засобами Петрі-об'єктної технології надає можливість конструювання складної грід-системи з довільною кількістю користувачів та кількістю завдань, що зберігаються в буфері завдань. Однією з переваг застосування Петрі-об'єктного підходу є можливість конструювання великої системи зі складових елементів. Відтворення процесів функціонування грід-системи з використанням стохастичних мереж Петрі з часовими затримками, багатоканальними та конфліктними переходами дозволяє максимально деталізувати формальний опис системи, а розбиття системи на Петрі-об'єкти забезпечує гнучкість при відтворенні структури грід-системи. За результатами моделювання визначаються тип управління та параметри управління розподіленими ресурсами, які найбільш задовольняють вимогам користувачів грід-системи.

## СПИСОК ЛІТЕРАТУРИ

1. Коваленко В.Н. Организация ресурсов грид [Электронный ресурс] / В.Н. Коваленко, Д.А. Корягин // Preprint, Inst. Appl. Math., the Russian Academy of Science. – Москва, 2004. – Режим доступа: [http://www.keldysh.ru/papers/2004/prep63/prep2004\\_63.html#\\_Toc85204258](http://www.keldysh.ru/papers/2004/prep63/prep2004_63.html#_Toc85204258).

2. Шелестов А.Ю. Имитационная модель взаимодействия GRID-узлов с очередью доступа к общей памяти / А.Ю. Шелестов // Автоматика. Автоматизация. Электротехнические комплексы и системы. – 2008. – № 1 (21). – С. 32 – 40.
3. Эвристики распределения задач для брокера ресурсов Grid / А.И. Аветисян, С.С. Гайсарян, Д.А. Грушин [и др.] // Труды Института системного программирования РАН. – Москва, 2004. – Т. 5. – С. 269 – 280.
4. Петренко А.І. Комп'ютерне моделювання грід-систем / А.І. Петренко // Електроніка і зв'язь. Тематичний вип. 5: Електроніка і нанотехнології. – 2010. – № 5. – С. 40 – 48.
5. A toolkit for modeling and simulating Data Grids: an extension to GridSim / A. Sulistio, U. Cibej, S. Venugopal [et al.] / Concurrency and Computation: Practice and Experience – Concurrency. – 2008. – Vol. 20, N 13. – P. 1591 – 1609.
6. Шелестов А.Ю. Моделирование Grid-узла на основе сетей Петри / А.Ю. Шелестов // Проблемы управления и информатики. – 2008. – № 1. – С. 104 – 113.
7. Литвинов В.В. Распределенная система имитационного моделирования на основе архитектуры CORBA / В.В. Литвинов, В.В. Казимир, И.Б. Гавсиевич // Математичні машини і системи. – 2000. – № 2, 3. – С. 111 – 114.
8. Стеценко И.В. Формальное описание систем средствами Петри-объектных моделей / И.В. Стеценко // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: зб. наук. пр. – 2011. – № 53. – С. 74 – 81.
9. Анни П. Этот Grid – неспроста... [Электронный ресурс] / П. Анни // Открытые системы. – 2003. – № 1. – Режим доступа: <http://citforum.ru/nets/articles/egridn.shtml>.
10. Стеценко І.В. Моделювання управляючих систем засобами мереж Петрі з інформаційними зв'язками / І.В. Стеценко // Вісник Черкаського державного технологічного університету. – 2011. – № 3. – С. 3 – 9.
11. Стеценко И.В. Алгоритм имитации Петри-объектной модели / И.В. Стеценко // Математичні машини і системи. – 2012. – № 1. – С. 154 – 165.

*Стаття надійшла до редакції 22.11.2011*