

14. Ганущак-Єфіменко Л.М. Імітаційна модель управління розвитком інноваційного потенціалу підприємств машинобудівної галузі/ Л. Ганущак-Єфіменко// Актуальні проблеми економіки. – 2009. - № 6 (96) - С.73-79.

УДК 004.02:519.7

**В.О.Кучер**

**Моделювання процесу планування виконання завдань в Grid-системах з використанням нечітко-множинного підходу**

*Використання нечітко-множинного підходу, принципу прямокутної упаковки в смугу заданої ширини, вибір агрегативної системи в якості математичної моделі досліджуваного процесу, дозволили розробити алгоритм планування виконання завдань користувачів, що забезпечує мінімізацію часу виконання завдання і збалансованість завантаження постачальників ресурсів.*

***Ключові слова:** системна мережа, черга, принцип матричної оцінки рівня факторів, кількісні та якісні шкали, лінгвістична змінна, функції приналежності, прямокутна упаковка.*

*Application of the fuzzy-set approach, the principle of square stowage into the band of a given width, choice of an aggregate system as a mathematical model of the investigated object, have allowed the development of an algorithm of scheduling the user task performance. Such an approach minimizes the time of task performance as well as balances workload of the resource suppliers.*

***Keywords:** system network, line, matrix estimation principle for a factor rate, quantitative and qualitative*

---

*scales, linguistic variable, membership functions, square stowage.*

**Актуальність.** Grid-система – це обчислювальна інфраструктура, яка забезпечує безперебійний доступ до обчислювальних потужностей і ресурсів зберігання даних, програмного забезпечення, спеціальних пристроїв, служб, які можуть динамічно входити і залишати її.

Головними суб'єктами в системі Grid є постачальники і споживачі ресурсів, у кожного з яких власні стратегії. Споживачі ресурсів застосовують стратегії розв'язання своїх прикладних завдань залежно від необхідного періоду часу і наявного бюджету. Вони мають можливість вибору постачальників, які найбільшою мірою задовольняють їх вимогам. Постачальники ресурсів зацікавлені в отриманні найбільшої вигоди від вкладених коштів і намагаються максимізувати використання своїх ресурсів.

Для управління Grid-системами зараз використовується різне ПЗ, але разом з операційними системами в них використовується проміжне програмне забезпечення (middleware) і додаткові програмні засоби, спрямовані на оптимізацію виконання завдань у системі, однією з найбільш важливих функцій якої є забезпечення механізму планування розподілу завдань. В основі кожного планувальника лежить алгоритм, від якого багато в чому залежить ефективність управління завданнями. Вибір алгоритму планування – це проблема, що розв'язується кожного разу залежно від технічних характеристик і висунутих вимог організації, а тому є актуальною проблемою.

**Аналіз останніх досліджень і публікацій.** Одним з найбільш поширених в даний час програмних засобів для

реалізації Grid є пакет Globus Toolkit [1] - [3]. Пакет Globus Toolkit розробляється, підтримується і просувається міжнародним альянсом розробників з університетів США і Великобританії, а також наукових лабораторій і обчислювальних центрів. Globus Toolkit є вільно поширюваним з відкритим вихідним кодом програмним пакетом, який пропонує базові засоби для створення Grid інфраструктури: засоби забезпечення безпеки в розподіленому середовищі, засоби надійної передачі великих обсягів даних, засоби запуску та отримання результатів виконання завдань на віддалених обчислювальних ресурсах. На базі пакету Globus Toolkit створюються промислові версії реалізацій Grid інфраструктури, наприклад, такі як Univa і Platform Globus Toolkit.

Проект NorduGrid [4] - [6] є одним з найбільших світових експериментальних випробувальних стендів, що в той же самий час знаходиться під реальним обчислювальним навантаженням експлуатаційних виробничих Grid-систем. Дослідники з 4-х країн (Данія, Фінляндія, Норвегія і Швеція) домовилися про можливе об'єднання ресурсів 12 університетів та високопродуктивних обчислювальних центрів на основі нової, розробленої ними технології ARC (Advanced Resource Connector), яка широко використовується і продовжує розвиватися.

Споживачі ресурсів Grid взаємодіють з брокерами для висунення своїх вимог. Брокер ресурсів Grid виступає в ролі сполучної ланки між користувачами і постачальниками ресурсів, використовуючи інтерфейсне ПЗ ядра для забезпечення послуг. Він відповідає за пошук ресурсів, їх вибір, прив'язку програмного забезпечення і даних до обраних ресурсів, ініціалізацію обчислень,

адаптацію до змін у системі. Брокер виступає для користувачів Grid як єдиний ресурс. Брокер включає ряд необхідних компонент.

При розв'язанні завдання планування найбільш часто застосовуються такі стратегії:

- спискові алгоритми (First Come First Served - FCFS та модифікації) [7];
- алгоритм зворотного заповнення (Backfill) [8];
- алгоритм групового складання розкладу (Gang Scheduling) [9], [10];
- алгоритм, що використовує множину черг (feedback algorithm) [11];
- генетичні алгоритми;
- змішані алгоритми.

Модифікації перелічених алгоритмів планування можуть надавати користувачеві можливість визначати для задачі необхідний мінімальний розмір оперативної пам'яті на вузлі, обсяг вільного простору на жорсткому диску і інші параметри та обмеження, що вимагає наявності додаткової інформації про наявні обчислювальні ресурси.

У реальних системах планування найбільшого поширення набули алгоритми FCFS (або їх модифікації Priority FCFS) і Backfill.

**Невирішені проблеми.** Незважаючи на те, що вже зараз пропонуються засоби створення Grid-інфраструктур, що "де-факто" стали стандартними, існує ряд важливих завдань, без розв'язання яких повномасштабне використання Grid-технологій є неможливим. Одним з актуальних завдань є ефективне управління обчислювальними ресурсами в розподіленому середовищі. Із зростанням числа ресурсних центрів, що входять в розподілену інфраструктуру, відсутність доброго планувальника, який би забезпечував управління потоком

задач, значно знижує ефективність використання всієї Grid-інфраструктури. Для розподілених систем характерним є також динамічний розвиток, що потребує відображення при розв'язанні задачі ефективного управління.

**Мета статті.** Розглядається Grid-система з багаторівневою ієрархічною структурою. На кожному з рівнів системи існують свої сервіси, що взаємодіють за допомогою певних протоколів.

Нижній рівень являє собою ресурси, спільний доступ до яких забезпечується через протоколи Grid - механізмами доступу до обчислювальних ресурсів, до ресурсів зберігання, до мережевих ресурсів, до ресурсів сховища кодів програм, до ресурсів каталогів.

Протоколи наступних двох рівнів є базовими:

- базові комунікаційні протоколи, що дозволяють здійснювати обмін даними між ресурсами нижнього рівня, та ідентифікаційні протоколи, що надають механізм для верифікації й ідентифікації користувачів та ресурсів;

- інформаційні протоколи, які використовуються для огляду інформації про структуру і стан ресурсу; протоколи менеджменту, які використовуються для надання доступу до спільних ресурсів.

На наступному за ієрархією рівні згруповані протоколи і сервіси, які не пов'язані з яким-небудь конкретним ресурсом, є більш глобальними за природою і забезпечують колективну взаємодію ресурсів. Та найверхній рівень включає в себе додаток користувача, що функціонує в середовищі Grid.

**Постановка завдання.** Мета: розробити алгоритм планування виконання завдань користувачів, що забезпечує мінімізацію часу виконання завдання і збалансованість завантаження постачальників ресурсів.

За завданням користувача повинні визначатися постачальники ресурсів, які відповідають його вимогам. З виділеної множини постачальників повинен вибиратися той, який найкращим чином відповідає поставленій меті.

Повинні бути розглянуті ситуації, коли наявні або відсутні черги до постачальників ресурсів та розроблені ефективні підходи щодо мінімізації часу обслуговування черги. Повинні розраховуватися моменти початку установки файлів для виконання завдання. А також має бути запропонований спосіб обліку відмінностей при гомогенних і гетерогенних ресурсах. Опис процесу планування повинен бути здійснений в операторній формі.

Використання нечітко-множинного підходу [12], принципу прямокутної упаковки в смугу заданої ширини, вибір агрегативної системи в якості математичної моделі досліджуваного процесу, дозволили розробити алгоритм, що відповідає всім поставленим вимогам.

**Виклад основного матеріалу.** Різноманітність властивостей і особливостей Grid і специфічні вимоги до планування виконання завдань можуть бути враховані при використанні в плануванні принципу матричної оцінки рівня факторів, що характеризують постачальників і споживачів ресурсів. Для оцінки сильних й слабких сторін постачальників ресурсу можна застосовувати як кількісні, так і якісні шкали.

Запроваджено дворівневу шкалу, що містить ряд базових факторів, які в свою чергу, характеризуються наборами своїх складових факторів:

I. Обчислювальні ресурси постачальників

$z^{li}(t)$ , ( $i = 1, 2, \dots$ )- ім'я / ip-адреса.

Складові компоненти:

$z_1^{li}(t)$ - число обчислювальних вузлів / процесорів;

$z_{2j}^{1i}(t)$ - пікова продуктивність  $j$ -го обчислювального вузла / процесора,  $\left(j = \overline{1, z_1^{1i}(t)}\right)$ , (у випадку гомогенних ресурсів:  $z_2^{1i}(t)$ );

$z_{3j}^{1i}(t)$ - продуктивність під час тесту (% від пікової),  $\left(j = \overline{1, z_1^{1i}(t)}\right)$ , (у випадку гомогенних ресурсів:  $z_3^{1i}(t)$ );

$z_{8j}^{1i}(t)$ - оперативна пам'ять;

$z_{9j}^{1i}(t)$ - розмір жорсткого диска.

## II. Ресурси зберігання

$z^{2i}(t)$ ,  $(i = 1, 2, \dots)$  - ім'я / ip-адреса.

Складові компоненти:

$z_{1l}^{2i}(t)$ - тип системи зберігання даних (зберігання на одному жорсткому диску в кожному вузлі, розподіл даних між вузлами кластера і т.д.),  $(l = 1, 2, \dots)$ ;

$z_{2l}^{2i}(t)$ - об'єм системи зберігання даних,  $(l = 1, 2, \dots)$ .

## III. Системна мережа $z^3(t)$ .

Складові компоненти:

$z_1^{3mn}(t)$ - пропускна спроможність каналів зв'язку між ресурсами постачальників;

$z_2^{3mn}(t)$ - середня пропускна спроможність.

## IV. Абстрактні завдання $z^{4s}(t)$ - ідентифікація.

Складові компоненти:

$z_5^{4s}(t)$ - кількість вузлів;

$z_6^{4s}(t)$ - продуктивність;

$z_7^{4s}(t)$ - розмір оперативної пам'яті;

$z_8^{4s}(t)$ - розмір пам'яті на жорсткому диску;

$z_{9pj}^{4s}(t)$ - обсяг використовуваного проблемно-орієнтованого ПЗ;

$z_{10}^{4s}(t)$ - обсяг файлів користувача для виконання завдання.

Введена лінгвістична змінна «Рівень фактора» з підмножинами «Дуже низький рівень», «Низький рівень», «Середній рівень», «Високий рівень», «Дуже високий рівень» і введений носій  $x$  - відрізок дійсної осі  $[0,1]$  (01 - носій). Будь-який відрізок осі може бути зведений до відрізка  $[0,1]$  шляхом простого лінійного перетворення, тому виділений відрізок одиничної довжини носить універсальний характер. На цьому відрізку визначений набір функцій приналежності по всім виділеним підмножинам лінгвістичної змінної «Рівень фактора». Доцільно, щоб такі функції приналежності мали трапецієдальний вигляд (стандартна нечітка п'ятирівнева 01-класифікація).

Якщо визначити по кожному складовому фактору його рівень кількісно (від 0 до 1), то рівень базового фактора може бути визначений як кількісно, так і якісно за матричним принципом, суть якого полягає в такому. Нехай по рядках матриці відкладені складові фактори, а по стовпцях - їх рівні, виражені відповідним набором функцій приналежності. Тоді інтегральний показник сили / слабкості постачальника за базовим фактором визначається як подвійна згортка:



$$SW_i = \sum_{(j)} p_{ij} \sum_{k=1}^5 \alpha_k \mu_{ijk}(x_j), \quad (1)$$

де  $x_j$  – кількісне значення  $j$ -го складового фактора;  
 $p_{ij}$  – вага  $j$ -го складового чинника в оцінці сили / слабкості постачальника за базовим  $i$ -м фактором;  $\mu_{ijk}$  - значення  $k$ -ої функції приналежності при визначенні рівня  $j$ -го складового фактора (всього функцій п'ять, по кількості підмножин);  $\alpha_k = (0.1, 0.3, 0.5, 0.7, 0.9)$  - набір ваг станів в інтегральній згортці (так звані вузлові точки, рівномірно нанесені на 01-носії).

При чому виконується:

$$\sum_{k=1}^5 \mu_{ijk}(x) = 1 \quad (2)$$

для будь-яких значень носія  $x$ , і система ваг показників повинна в сумі давати одиницю

$$\sum_{(j)} p_{ij} = 1 \quad (3)$$

У силу (1) - (3) виконується умова  $0 < SW_i \leq 1$ , і тому отримане значення можна розпізнати за загальними правилами, визначеними для 01-носія.

Аналогічним чином можна здійснити матричну згортку при переході від окремих показників сили / слабкості постачальників по базових факторах до інтегрального показника. Потрібно тільки визначити ваги базових факторів в інтегральній оцінці.

Інтегральний показник може бути представлений також у вигляді середнього зваженого окремих показників по базовим факторам:

$$SW = \sum_{(i)} g_i SW_i, \quad (4)$$

де ваги базових факторів, як правило, розраховуються за допомогою методу простого ранжування, пропорційного методу чи методу попарного порівняння. Якщо існує можливість проранжувати всі чинники в порядку спадання їх значущості, то вагу  $i$ -го фактора можна визначити за правилом Фішберна:

$$g_i = \frac{2(N - i + 1)}{(N + 1)N}, \quad i = 1, \dots, N,$$

а в системі байдужих одне одному  $N$  альтернатив - набором рівних ваг

$$g_i = N^{-1}, \quad i = 1, \dots, N.$$

Дотримуючись викладеного, для кожного із базових чинників визначено шкали розподілу складових факторів за рівнями «Дуже низький», «Низький», «Середній», «Високий», «Дуже високий».

Аналогічно визначено шкали розподілу для ресурсів зберігання і системної мережі.

Для кожного постачальника ресурсів відповідно до введених шкал розподілу складові фактори їх базових факторів згруповані за введеними рівнями. Елементи  $y_j$  отриманих матриць характеризують рівень відповідного фактора і приймають значення з множини введених рівнів: {ДВ, В, С, Н, ДН}.

Наступний крок - обчислення показника сили / слабкості постачальника за базовим  $i$ -м фактором за співвідношенням (1), а також інтегрального показника.

Якщо у постачальника гетерогенні ресурси, то діємо наступним чином:

Таблиця 1.

Обчислювальні ресурси постачальників

|                        | Дуже низький | Низький         | Середній        | Високий         | Дуже високий |
|------------------------|--------------|-----------------|-----------------|-----------------|--------------|
|                        | 0,1          | 0,3             | 0,5             | 0,7             | 0,9          |
| Число процесорів;      | $< a_1^1$    | $a_1^1 - a_2^1$ | $a_2^1 - a_3^1$ | $a_3^1 - a_4^1$ | $> a_4^1$    |
| Пікова продуктивність  | $< a_1^2$    | $a_1^2 - a_2^2$ | $a_2^2 - a_3^2$ | $a_3^2 - a_4^2$ | $> a_4^2$    |
| Продуктивність в тесті | $< a_1^3$    | $a_1^3 - a_2^3$ | $a_2^3 - a_3^3$ | $a_3^3 - a_4^3$ | $> a_4^3$    |
| Оперативна пам'ять     | $< a_1^4$    | $a_1^4 - a_2^4$ | $a_2^4 - a_3^4$ | $a_3^4 - a_4^4$ | $> a_4^4$    |
| Розмір жорсткого диска | $< a_1^5$    | $a_1^5 - a_2^5$ | $a_2^5 - a_3^5$ | $a_3^5 - a_4^5$ | $> a_4^5$    |

Таблиця 2

Матриця для оцінки  $SW_i$

| Фактори     | Вага     | Функції приналежності для рівнів складових факторів: |                    |                     |                    |                         |
|-------------|----------|--|--------------------|---------------------|--------------------|-------------------------|
|             |          | Дуже низький $\mu_1(x)$                              | Низький $\mu_2(x)$ | Середній $\mu_3(x)$ | Високий $\mu_4(x)$ | Дуже високий $\mu_5(x)$ |
| 1           | $p_{i1}$ |  |                    |                     |                    |                         |
| 2           | $p_{i2}$ |  |                    |                     |                    |                         |
| ...         | ...      |  |                    |                     |                    |                         |
| Вага рівнів |          | 0,1  | 0,3                | 0,5                 | 0,7                | 0,9                     |

- Для групи ресурсів  $z_{1k}^{1i}(t) \in z_1^{1i}(t) \left( \sum_{(k)} z_{1k}^{1i} = z_1^{1i} \right)$ , що мають однаковий рівень продуктивності  $z_{2j}^{1i}(t)$  з множини {ДВ, В, С, Н, ДН}, проводимо розрахунки з базових факторів викладеним вище чином ( $z_{2j}^{1i}(t)$  беремо рівною значенню в середині відповідної шкали табл. 1, компоненти  $z_{8j}^{1i}(t)$  і  $z_{9j}^{1i}(t)$  оцінюємо середнім значенням по виділеній групі);

- Рейтинг постачальника за базовим  $i$ -м фактором визначаємо як інтегральний показник за обчисленими показниками сформованих груп ресурсів.

Можливий інший підхід, коли постачальник гетерогенних ресурсів оцінюється не за інтегральним показником, а кожна з підгруп ресурсів, об'єднаних за рівнями факторів, розглядається незалежно як така, що має гомогенні ресурси.

Отримані в результаті проведених розрахунків рейтинги постачальників зберігаються до тих пір, поки не відбудуться зміни у їх ресурсах. При введенні \ виведенні ресурсу або входженні нових постачальників їхній рейтинг обчислюється описаним способом.

Планувальник відповідає за вибір ресурсів і розподіл завдань таким чином, щоб задовольнити всі вимоги користувачів та оптимізувати загальний час виконання завдань. Тому завдання користувача аналізується за аналогією з аналізом постачальників ресурсів. Отримані в результаті відповідних обчислень рівні базових факторів завдання дозволяють визначити постачальників ресурсів, які відповідають вимогам користувача. Це ті постачальники, рівень базових факторів яких не нижче рівня базових факторів завдання споживача. З виділеної

множини постачальників планувальник повинен вибрати того, який найкращим чином відповідає поставленій меті щодо мінімізації часу виконання завдання і збалансованості завантаження.

Оскільки більший інтерес для досліджень представляють Grid-системи виробничого типу, тобто в яких кількість користувачів значно перевищує кількість постачальників ресурсу, то при розподілі завдань необхідно враховувати наявність черг завдань на виконання.

З метою мінімізації часу обслуговування черги вибирається підхід прямокутної упаковки в смугу заданої ширини. Суть цього підходу в такому.

Нехай постачальник ресурсу має  $G$  обчислювальних вузлів. У черзі до нього знаходиться  $M$  завдань. Завдання  $m \in M$  вимагає для свого виконання  $r_m$  вузлів. Передбачувану тривалість виконання завдання  $m$  позначимо  $T_m$ .

Тоді завдання  $m$  можна інтерпретувати як прямокутник, ширина якого  $r_m$ , довжина -  $T_m$ .

Потрібно знайти таку упаковку напівнескінченої смуги шириною  $G$ , для якої довжина зайнятої частини смуги (тривалість проміжку часу, необхідного для обслуговування  $M$  завдань) досягала б мінімуму  $T$  і виконувалися б такі умови:

1. Сторони прямокутників паралельні сторонам смуги в упаковці;
2. Прямокутники взаємно не перекривали б один одного;
3. Прямокутники не перекривали б сторони смуги.

Для розв'язання завдання прямокутної упаковки розроблені евристичні методи. Їх ефективність можна

оцінити за допомогою числових експериментів на розроблюваній моделі.

Використовуючи зазначений підхід, планувальник накопичує певну кількість завдань і здійснює їх прямокутну упаковку в смугу заданої ширини. Таким чином частина черги завдань упакована, а інша частина - накопичується для упаковки. Цей факт враховується при розподілі завдань: вибирається кластер з мінімальним сумарним часом, що є необхідним: 1) для виконання ще не обслугованих на момент часу  $t$  завдань з упакованої черги і 2) для виконання завдань неупакованої черги, час на обслуговування яких оцінюється за допомогою мінімальної загальної площі цих завдань:  $\sum_{i=1}^{N+1} (r_i T_i) / G$ , де  $N$  - число завдань, які стоять в неупакованій черзі,  $N+1$  - номер завдання, яке відправляється;  $G$  - число процесорів кластеру.

Для опису черги завдань введемо базову компоненту.

V. Черга завдань  $z^{5i}(t)$  до постачальника ресурсів  $z^i(t)$ .

Складові компоненти:

$z_2^{5i}(t)$  - кількість завдань в черзі в момент часу  $t$ ;

$z_{3s}^{5i}(t)$  - координата довжини завдання;

$z_{4s}^{5i}(t)$  - координата ширини завдання  $z^{4s}$ .

Незважаючи на те, що даний підхід до планування орієнтований на Grid-системи виробничого типу, тобто в яких кількість користувачів значно перевищує кількість постачальників ресурсу, необхідно домовитися про поведінку планувальника у разі відсутності черг до постачальників ресурсів і наявності вільних ресурсів.

У такій ситуації можна запропонувати різні шляхи. Після того, як визначено множину постачальників, ресурси яких відповідають запиту користувача:

1. Виділяється підмножина тих, у яких кількість вільних вузлів не менше числа вузлів, запитуваних завданням. Завдання направляється до того постачальника з цієї підмножини, час передачі файлів до якого є мінімальним.

2. Якщо підмножина пункту 1 є порожньою, завдання направляється тому постачальнику, в якого швидше за все звільниться необхідна кількість вузлів.

Результати різних підходів до розподілу в значній мірі залежать від вхідного потоку завдань, від характеристик ресурсів, від обсягів даних, що передаються, і ін. У такому випадку кращий варіант доцільно визначати шляхом тестування.

При плануванні необхідно визначати моменти початку установки файлів для завдань. Передчасна установка файлів призводить до неефективного використання ресурсів зберігання, з іншого ж боку файли повинні бути доступні перед початком виконання завдання. Час, який потрібний для копіювання файлів, залежить від розміру файлів і швидкості мережі.

Використовуючи складові компоненти введених базових компонент, далі будуть визначені співвідношення для знаходження моментів початку установки файлів для завдань.

Якщо при виконанні завдання  $s$ , його статус стає «помилка» або «призупинено», то зберігається досягнутий стан виконання, завдання переноситься на перше місце неупакованої черги і при надходженні на обслуговування продовжується його виконання.

Розглянутий процес планування можна подати як нескінчену послідовність змінюючих одна одну тривалостей безперервних і миттєвих дискретних поведінок. Математичною моделлю такого безперервно-дискретного процесу може бути агрегативна система.

Опис агрегату як перетворювача інформації можна дати таким чином:

$$A = \{T, X, \Gamma, Y, Z, H, G\},$$

де елементи зазначених множин називаються так:  $t \in T$  - момент часу;  $x \in X$  - вхідний сигнал;  $g \in \Gamma$  - керуючий сигнал;  $y \in Y$  - вихідний сигнал;  $z \in Z$  - стан;  $z(t) \in H$  - оператор переходів;  $y(t) \in G$  - оператор виходів.

Модель агрегату може бути використана як модель всього безперервно-дискретного процесу або його елемента. В останньому випадку процес представляється мережею агрегатів з фіксованими каналами зв'язку.

Для розробленої моделі потрібен ряд додаткових компонент, що характеризують завдання і політику вибору вузлів, а також розширюють список вимог до даних, які необхідні для прийняття рішення:

$z_5^{1i}(t)$  - кількість вільних процесорів у постачальника  $z^i(t)$ ;

$z_{6j}^{1i}(t)$  - залишкова тривалість передбачуваної зайнятості  $j$ -го процесора  $\left(j = \overline{1, z_1^{1i}(t)}\right)$  у постачальника

$z^i(t)$ ;

$z_{7s}^{1i}(t)$  - тривалість доставки файлів до постачальника  $z^i(t)$  для виконання завдання;



$z_{3j}^{2i}(t)$ - тип проблемно-орієнтованого ПЗ, що зберігається у постачальника  $z^{2i}(t)$ ;

$z_1^{3k}(t)$ - пропускна спроможність сервісної мережі постачальника  $z^k(t)$ ;

$z_2^{3k}(t)$ - середня пропускна спроможність сервісної мережі постачальника  $z^k(t)$ ;

$z_{11}^{4s}(t)$ - передбачувана тривалість виконання завдання  $z^{4s}(t)$ ;

$z_{1i}^{4s}(t)$ - обраний постачальник ресурсів для виконання завдання  $z^{4s}(t)$ ;

$z_4^{4s}(t)$ - статус завдання  $z^{4s}(t)$  (1 - у черзі, 2 - активно, 3 - завершено, 4 - зупинка);

$z_3^{4s1i}(t)$ - залишковий проміжок часу до передбачуваного завершення завдання  $z^{4s}(t)$  у постачальника  $z_{1i}^{4s}(t)$ ;

$z_{2i}^{4s}(t)$ - моменти початку установки файлів для завдання  $z^{4s}(t)$ , виконуваного постачальником  $z_{1i}^{4s}(t)$ ;

$z_1^{5i}(t)$ - кількість упаковок в черзі до постачальника  $z^i(t)$ ;

$z_{1l}^{5i}(t)$ ,  $l = \overline{1, z_1^{5i}}$  - моменти прямокутної упаковки завдань в черзі до постачальника  $z^i(t)$ ;

$z_2^{5i}(t)$ - кількість завдань в неупакованій частині черги до постачальника  $z^i(t)$  в момент часу  $t$ ;

$z_{5l}^{5i}(t)$ - довжини зайнятих частин смуги упакованих черг у постачальника  $z^i(t)$ ;

$z_6^{5i}(t)$ - передбачуваний час обслуговування черги постачальника  $z^i(t)$ ;

$z_{7l}^{5i}(t)$ ,  $l = \overline{1, z_1^{5i}}$  - ідентифікація упаковки черги постачальника  $z^i(t)$  ;

$z_{8z_{7l}^{5i}z^{4s}}^{5i}(t)$ ,  $l = \overline{1, z_1^{5i}}$  - координати лівих нижніх кутів завдання  $z^{4s}$  упаковки  $z_{7l}^{5i}$  (черговість завдань в упаковках).

Будемо називати вузловими ті моменти часу, коли відбуваються події, що вимагають реакції планувальника. Перелічимо ці множини вузлових моментів:

$\{t_j^1\}$  - надходження завдань;

$\{t_j^2\}$  - зміна статусу завдань (в черзі, активно, завершено, призупинено);

$\{t_j^3\}$  - введення \ виведення ресурсів;

$\{t_j^4\}$  - моменти початку установки файлів для завдань.

$\{t_j^5\}$  - моменти прямокутної упаковки завдань у смугу заданої ширини;

$\{t_j^{6l}\}$  - моменти початку виконання завдань  $l$ -ої з упакованих частин.

Продемонструємо опис процесу планування в операторній формі в моменти надходження завдань  $\{t_j^1\}$  :

Маємо впорядковані у порядку зростання рейтинги постачальників ресурсів за базовими факторами

$$SW_i = \sum_{(j)} p_{ij} \sum_{k=1}^5 \alpha_k \mu_{ijk}(x_j):$$

$$\{ SW_i^{z^{1k}} \}, i = 1, 2, \dots; z^{1jk} \in \{ z^{1i} \}.$$

- Обчислюємо рейтинг надійшовшого завдання за базовими факторами:  $RS_i, i = 1, 2, \dots$

- Виділяємо множину постачальників ресурсів для надійшовшого завдання за базовими факторами:

$$\left\{ z^{jk} / SW_i^{z^{1jk}} \geq RS_i \right\}.$$

- Обчислюємо рейтинг завдання за інтегральним показником  $RS = \sum_{(i)} p_i \sum_{k=1}^5 \alpha_k \mu_{ik}(RS_i)$ .

- Обчислюємо рейтинги постачальників з виділеної множини за інтегральним показником при тих же вагах базових факторів, які використовувалися при обчисленні рейтингу завдання:  $\{ SW^{z^{1jl}} \},$

$$z^{jl} \in \left\{ z^{jk} / SW_i^{z^{1jk}} \geq RS_i \right\}.$$

- Виділяємо за сукупністю факторів множину постачальників ресурсів для надійшовшого завдання:

$$\left\{ z^{jl} \right\} = \left\{ z^{jk} / SW^{z^{1jk}} \geq RS \right\}.$$

- Розраховуємо час обслуговування черг для виділених постачальників:

$$z_6^{5i}(t) = \max \left\{ 0, \left[ z_{51}^{5i} - (t - t_j^{51}) \right] + \sum_{l=2}^{5i} z_{51}^{5i} + \sum_{s=1}^{5i+1} z_{4s}^{5i} z_{3s}^{5i} / z_1^{1i} \right\}$$

- Виділяємо постачальників з необхідною кількістю вільних процесорів з множини, визначеної за сукупністю

$$\text{факторів: } \left\{ z^{i_l} \right\} = \left\{ \left( 1 - \underset{(i_l)}{\text{sign max}} \left\{ 0, z_5^{4s} - z_5^{1i_l} \right\} \right) z^{i_l} \right\}.$$

- Кількість постачальників з достатньою кількістю затребуваних ресурсів:

$$z_6^{1i} = \sum_{i_l} \left( 1 - \underset{i_l}{\text{sign max}} \left\{ 0, z_5^{4s} - z_5^{1i_l} \right\} \right).$$

- Вибір постачальника для даного завдання ( $\xi^{i_l}$  - рівномірно розподілена випадкова величина з рядом розподілу  $\left\{ p^{i_l} = 1/z_6^{1i} \right\}$ , що забезпечує рівноправний вибір

постачальника з наявного множини;  $\left\{ z_{6j_k}^{1i} \right\}$  - послідовність проміжків часу до звільнення потрібної кількості процесорів (розташованих за зростанням);

$k = (z_5^{4s} - z_5^{1i}) / z_7s^{1i}$  - послідовність проміжків часу доставки файлів для завдання (розташованих за зростанням)):

$$z_{1i}^{4s} (t_j^1 + 0) = R_1 z^{i_l} \rightarrow z_6^{5i_l} = \underset{(i)}{\text{min}} z_6^{5i} + (1 - R_1) \cdot \left\{ (1 - R_2) \xi^{i_l} + \right. \\ \left. + R_2 \left\{ (1 - R_3) z^{i_l} \rightarrow z_{6j_k}^{1i} = \underset{(i)}{\text{min}} z_{6j_k}^{1i} + R_3 z^{i_l} \rightarrow z_{7s}^{1i} = \underset{(i)}{\text{min}} z_{7s}^{1i} \right\} \right\},$$

$$\text{де } R_1 = \underset{(i)}{\text{sign}} \left\{ \underset{(i)}{\text{min}} z^{5i} \right\}, R_2 = \underset{(i)}{\text{sign max}} \left\{ 0, z_5^{4s} - z_5^{1i} \right\},$$

$$R_3 = \underset{(i)}{\text{sign max}} \left\{ 0, z_{7s}^{1i} - z_{6j_k}^{1i} \right\}.$$

$$\text{-Статус завдання: } z_4^{4s} (t_j^1 + 0) = \underset{(i)}{\text{sign max}} \left\{ 0, z_5^{4s} - z_5^{1i} \right\} \cdot 1 +$$

$$+ \left( 1 - \text{sign} \left\{ \min_{(i)} z^{5i} \right\} \right) \left\{ \left( 1 - \text{sign} \max_{(i)} \left\{ 0, z_5^{4s} - z_5^{1i} \right\} \right) \cdot 2 \right\}.$$

- Черга у вибраного постачальника:

$$z^{5z_{1i}^{4s}}(t_j^1 + 0) = \left[ 1 - \text{sign} \left( z_4^{4s}(t_j^1 + 0) - 1 \right) \right] * \left[ z^{5z_{1i}^{4s}}(t_j^1) + 1 \right] + \\ + \left[ \text{sign} \left( z_4^{4s}(t_j^1 + 0) - 1 \right) \right] * z^{5z_{1i}^{4s}}(t_j^1).$$

- Кількість упаковок в черзі:  $z_1^{5z_{1i}^{4s}}(t_j^1 + 0) =$

$$\text{signmax} \left\{ 0, M - \left( z_2^{5z_{1i}^{4s}}(t_j^1) + 1 \right) \right\} * \\ * z_1^{5z_{1i}^{4s}}(t_j^1) + \left[ 1 - \text{signmax} \left\{ 0, M - \left( z_2^{5z_{1i}^{4s}}(t_j^1) + 1 \right) \right\} \right] * \\ * \left[ z_1^{5z_{1i}^{4s}}(t_j^1) + 1 \right].$$

- Моменти прямокутної упаковки завдань:

$$z^{5z_{1i}^{4s}}(t_j^1 + 0) = \\ z_1^{5z_{1i}^{4s}} \\ = \left[ 1 - \text{signmax} \left\{ 0, M - \left( z_2^{5z_{1i}^{4s}}(t_j^1) + 1 \right) \right\} \right] t_j^1.$$

- Довжина упакованої частини черги:  $z_2^{5z_{1i}^{4s}}(t_j^1 + 0) =$

$$= \text{signmax} \left\{ 0, M - \left( z_2^{5z_{1i}^{4s}}(t_j^1) + 1 \right) \right\} *$$

$$* \left[ z_2^{5z_{1i}^{4s}}(t_j^1) + 1 \right] + \left[ 1 - \text{signmax} \left\{ 0, M - \left( z_2^{5z_{1i}^{4s}}(t_j^1) + 1 \right) \right\} \right] * \\ * \left[ z_2^{5z_{1i}^{4s}}(t_j^1) + 1 - M \right].$$

- Вільні процесори:  $z_5^{1z_{1i}^{4s}}(t_j^1 + 0) = z_5^{1z_{1i}^{4s}}(t_j^1) - z_5^{4s}(t_j^1)$ .

- Моменти початку установки файлів для завдання:

$$z_{2z_1}^{4s} z_5^{4s} z_{1i}^{4s}(t_j^1 + 0) = \\ = \max \left\{ t_j^1, \left( z_6^{5z_{1i}^{4s}}(t) - \max \left\{ z_{10}^{4s} / z_2^{3z_{1i}^{4s}}, \left\{ z_{9z_{1p}}^{4s} / z_1^{3z_{1i}^{4s}} z_{1i}^{4s} \right\} \right\} \right) \right\}.$$

У роботі проведено опис процесу планування в операторній формі для всіх типів вузлових моментів часу.

**Висновки.** Розроблена модель дозволяє оцінити ефективність запропонованого алгоритму. Чисельні експерименти на моделі дозволяють порівняти оцінки варіантів при проектуванні, при виборі режимів функціонування з точки зору обраного критерію ефективності тощо. Модель також підтримує можливість проведення експериментів, що складається з послідовних запусків моделі зі зміною деяких параметрів при кожному наступному запуску. Наприклад, може змінюватися потік завдань, конфігурація кластерів, мережових з'єднань і т.ін. Це дозволяє подивитися на динаміку зміни ефективності системи та визначити вузькі місця.

#### Література

1. Foster, C. Kesselmann, S. Tuecke, J.M.Nick «The Physiology of the Grid», 2002, - p. 1 - 31.
2. M. Ellert, A.Konstantinov, B. Kónya, O.Smirnova, A.Wäänänen. Architecture Proposal, NORDUGRID-TECH-1, 2002, - p 1-7.
3. <http://www.globus.org>

4. <http://www.nordugrid.org>
5. <http://www.gridcomputing.com>
6. <http://www.gridclub.ru>
7. Saeed Iqbal, Rinku Gupta, Yung-Chin Fang. Planning Considerations for Job Scheduling in HPC Clusters // Dell Power Solutions , 2005, - p. 133-136.
8. David Jackson, Quinn Snell, Mark Clement. Core Algorithms of the Maui Scheduler // Lecture Notes in Computer Science Job Scheduling Strategies for Parallel Processing: 7th International Workshop, JSSPP 2001. Cambridge, MA, USA, 16 June, 2001. Revised Paper, Volume 2221, 2001, - p. 87-102.
9. Uwe Schwiegelshohn, Ramin Yahyapour Improving First Come First Served job scheduling by gang scheduling // Computer Engineering Institute, University Dortmund, Germany: JSSPP'98, LNCS 1459, 1998, – p. 180-109.
10. Julita Corbalan Conzalez. Coordinated Scheduling and Dynamic Performance Analysis in Multiprocessor Systems — [[http://www.tdr.cesca.es/TESIS\\_UPC/AVAILABLE/TDX-0723102-094622//07Jcg07dc08.pdf](http://www.tdr.cesca.es/TESIS_UPC/AVAILABLE/TDX-0723102-094622//07Jcg07dc08.pdf)].
11. O.Sename, D.Simon, D.Robert. Feedback scheduling for real-time control of systems with communication delays // ETFA'03 9th IEEE International Conference on Emerging Technologies and Factory Automation, Lisbonne. Volume 2, 2003, - p. 454-461.
12. Alexey Nedosekin. FUZZY FINANCIAL MANAGEMENT. Russia, Moscow, AFA Library, 2003, - p. 186.

УДК 330.46

**Г. А. Цихан**

### **Економічний аналіз рядів динаміки за допомогою теорії фракталів**

*У статті обґрунтована можливість використання математичної теорії фракталів для економічного аналізу рядів динаміки. Визначені основні показники фрактального аналізу часових рядів для вирішення задачі економічного прогнозування, проаналізовані головні ознаки класифікації часових рядів за показником Херста.*