

Т.С. Дьячук, аспір.  
Запорізький національний технічний університет

## МЕТОД ПЛАНУВАННЯ ОБЧИСЛЮВАЛЬНИХ РЕСУРСІВ У РОЗПОДІЛЕНІЙ СИСТЕМІ

(Представлено д.т.н., проф. Карпуковим Л.М.)

*Розроблено математичну модель планування розподілених обчислювальних ресурсів, враховуючи їх особливості та потреби користувачів, які отримують доступ до системи. Алгоритм здатен гарантувати точний час запуску паралельних завдань з урахуванням пріоритетів запитів і наявності вільних, або тих, що звільняються через певний час, ресурсів.*

**Постановка задачі в загальному вигляді та її актуальність.** На даний час в Україні відбувається побудова ресурсно-операційних центрів, в тому числі й на базі вищих навчальних закладів, у рамках національного Grid-проекту [1]. У роботі розглянуто GRID-систему, ресурсну базу якої складають кластеризовані ресурси (далі кластер). Під кластером розуміється об'єднання персональних комп'ютерів різної продуктивності в межах однієї організації, які одночасно використовуються як робочі станції. Таким чином, ресурси, що виділяються в Grid, не відчужуються від власника. Наприклад як кластер може розглядатися комп'ютерна аудиторія університету, приєднана до GRID-системи. У зв'язку з цим наявні університетські ресурси необхідно якомога ефективніше виділити для GRID, розробивши методи планування [2]. Під університетськими ресурсами розуміються комп'ютерні класи, що беруть участь у навчальному процесі. Ідеальним буде таке планування, при якому вимоги користувачів на виділення обчислювальних ресурсів завжди задовольняються, а ресурси повністю використовуються. Робота проводиться в рамках національного Grid-проекту [1] з побудови ресурсно-операційного центру на базі Запорізького національного технічного університету.

**Аналіз існуючих рішень в області планування ресурсів.** Існує безліч стратегій, методів та програмних засобів, де основна увага приділена плануванню ресурсів та пакетному виконанню завдань [2]. Проте різноманітність завдань та структур обчислювальних систем викликає необхідність розробки і дослідження нових підходів, які тонше враховують особливості вирішуваних обчислювальних задач та засобів для їх вирішення. Багато пакетних систем почали своє існування як системи управління завданнями на локально розподілених обчислювальних платформах. Механізм резервування ресурсів з боку користувача підтримується такими системами пакетної обробки, як PBS та LL, а також планувальником Maui [2].

В основі управління ресурсами лежить механізм черг і планування робіт, або диспетчеризація. Простим методом планування, побудованим на розділенні простору ресурсів, є метод FCFS (First Come First Served – “першим прийшов, – першим обслуговується” рис. 1) [3], згідно з яким завдання, що поступило в чергу раніше інших, має найвищий пріоритет і повинно бути запущено першим. Якщо для його запуску недостатньо ресурсів, очікується момент часу, коли накопичиться потрібний об'єм вільних ресурсів. Порядок завдань в черзі не може бути порушений: завдання запускаються одне за іншим згідно з пріоритетами. Метод гарантує запуск паралельного завдання, проте неефективно використовує ресурси: в період їх накопичення для запуску завдання з найбільшим пріоритетом частина ресурсів простоє.

Модифікація “перший відповідний” (First-Fit) метода FCFS допускає порушення порядку черги, дозволяючи низькопріоритетним завданням займати ресурси, що залишилися незавантаженими. Це підвищує ефективність їх використання, проте призводить до того, що велика частина процесорів займається дрібними завданнями – виникає фрагментація ресурсів. Так, навіть якщо завдання має найвищий пріоритет, необхідний йому обсяг ресурсів може ніколи не утворитися, а завдання може ніколи не стартувати.

Більш ефективний спосіб рішення пропонує метод зворотного заповнення (Backfill [2]), який буде розглянутий далі. Цей метод використовується у багатьох планувальниках, наприклад у планувальнику Maui [2].

**Метою досліджень** є розробка методу планування обчислювальних ресурсів у розподіленій системі, враховуючи їх особливості та потреби користувачів, які одержують доступ до системи.

**Викладення основного матеріалу досліджень.** *Модель планування ресурсів.* Існує ряд факторів та обмежень, які потребують уваги.

Під запитом на ресурси розуміється паралельне складне завдання, частини якого можуть виконуватися паралельно на різній кількості обчислювальних вузлів кластера, причому на одному вузлі може виконуватися декілька гілок паралельної програми залежно від їх розподілу між вузлами. Але на

одному вузлі одночасно можуть знаходитись гілки тільки одного завдання. Завдання можуть виникати в довільні моменти часу і приходять в систему з певними потребами в ресурсах. Для виконання завдання вибираються тільки ті вузли, що мають в своєму розпорядженні необхідні ресурси. Потрібно визначити для кожного завдання час запуску та підібрати необхідні виконавчі ресурси.

Завдання мають пріоритет, завдяки якому визначається порядок виконання. Завдання з меншим пріоритетом може бути запущене тільки за умови, що воно не затримає виконання завдання з більшим пріоритетом (при очікуванні звільнення необхідних ресурсів).

Запити на ресурси бувають двох видів:

- локальні, що виникають безпосередньо в кластері [4];
- глобальні, що приходять від віддалених користувачів.

Потрібно розробити політику розподілу ресурсів між локальними та глобальними запитами з урахуванням пріоритетів запитів і наявності вільних (або тих, що звільнюються через певний час) ресурсів. Складається розклад виділення відповідних вузлів завданням, згідно з вимогами ресурсів та пріоритету.

Кожне завдання характеризується такими параметрами:

$s$  – кількість паралельних гілок завдання і відповідно максимальне число вузлів багатопроцесорної системи, яке може бути задіяне для виконання завдання. На одному вузлі кластера можуть знаходитись гілки тільки одного паралельного завдання, такий вузол вважається зайнятим для інших завдань.

$T$  – максимальний час вирішення завдання на одному еталонному ненавантаженому процесорі (припускаємо, що максимальний час виконання однієї гілки завдання  $t = T / s$ ), при запуску завдання відбувається перерахунок часу відповідно до параметрів виконавчих ресурсів.

$M$  – кількість оперативної пам'яті, яка потрібна завданню (припускаємо, що кількість оперативної пам'яті, яка потрібна для однієї гілки,  $m = M / s$ ).

Також введемо позначення:

$N$  – число комп'ютерів у кластерній системі.

$s_i$  – число гілок завдання на  $i$ -ому вузлі кластера ( $s_i \geq 0$ ,  $i = 1, \dots, N$ ).

$z_i$  – завантаженість  $i$ -го вузла, яка відповідає сумі пріоритетів процесів, що вже запущені на вузлі, і, таким чином, характеризує потенційну частину процесорного часу, яка може бути виділена новому процесу з певним пріоритетом [4].

$k_i$  – коефіцієнт уповільнення виконання завдань на  $i$ -ому вузлі, за рахунок процесів, що вже виконуються [4]:

$$k_i = z_i + s_i. \quad (1)$$

$w_i$  – нормуючий коефіцієнт для  $i$ -го вузла, оскільки вузли кластера мають різну продуктивність процесорів. Він дорівнює відношенню реальної продуктивності вузла  $wr_i$  до максимальної продуктивності  $w_{\max}$  (продуктивності деякого вузла з набору):

$$w_i = \frac{wr_i}{w_{\max}}. \quad (2)$$

Прогнозований час вирішення завдання:

$$T_{count} = \max_i(t \cdot k_i \cdot w_i) = \max_i(t \cdot (z_i + s_i) \cdot w_i). \quad (3)$$

При виконанні умов:

$$\sum_i s_i = s, \quad mem_i \geq m \cdot s_i, \quad (4)$$

де  $mem_i$  – об'єм доступної оперативної пам'яті на  $i$ -ому вузлі.

Критерієм підбору ресурсів є найшвидше отримання користувачем результатів виконання завдання. З черги вибирається завдання з найвищим пріоритетом, для якого обчислюється час, через який воно може бути виконане.

Розглядаються два випадки:

1.  $T_{count1} = T_{count}$  – час обчислення завдання за умови, що завдання буде розподілене серед вільних вузлів негайно [4];

2.  $T_{count2} = T_{wait} + T_{count}$  – час обчислення завдання за умови очікування звільнення ресурсів. Він складається з часу очікування  $T_{wait}$ , який обчислюється з розрахунку часу обчислення вже запущених завдань на необхідних вузлах, та прогнозованого часу рахунку самого завдання  $T_{count}$ .

Якщо час в першому випадку не більший за час другого, тобто  $T_{count1} \leq T_{count2}$ , то завдання запускається негайно і вибирається наступне завдання з черги. Інакше відбувається резервування

ресурсів через час очікування  $T_{wait}$  для початкового завдання та підбираються завдання з меншим пріоритетом, які можуть бути запущені на кластері за умови гарантованого звільнення зарезервованих ресурсів. Умовою підбору є гарантія того, що завдання з меншим пріоритетом не призведуть до затримки виконання завдань з більшим пріоритетом.

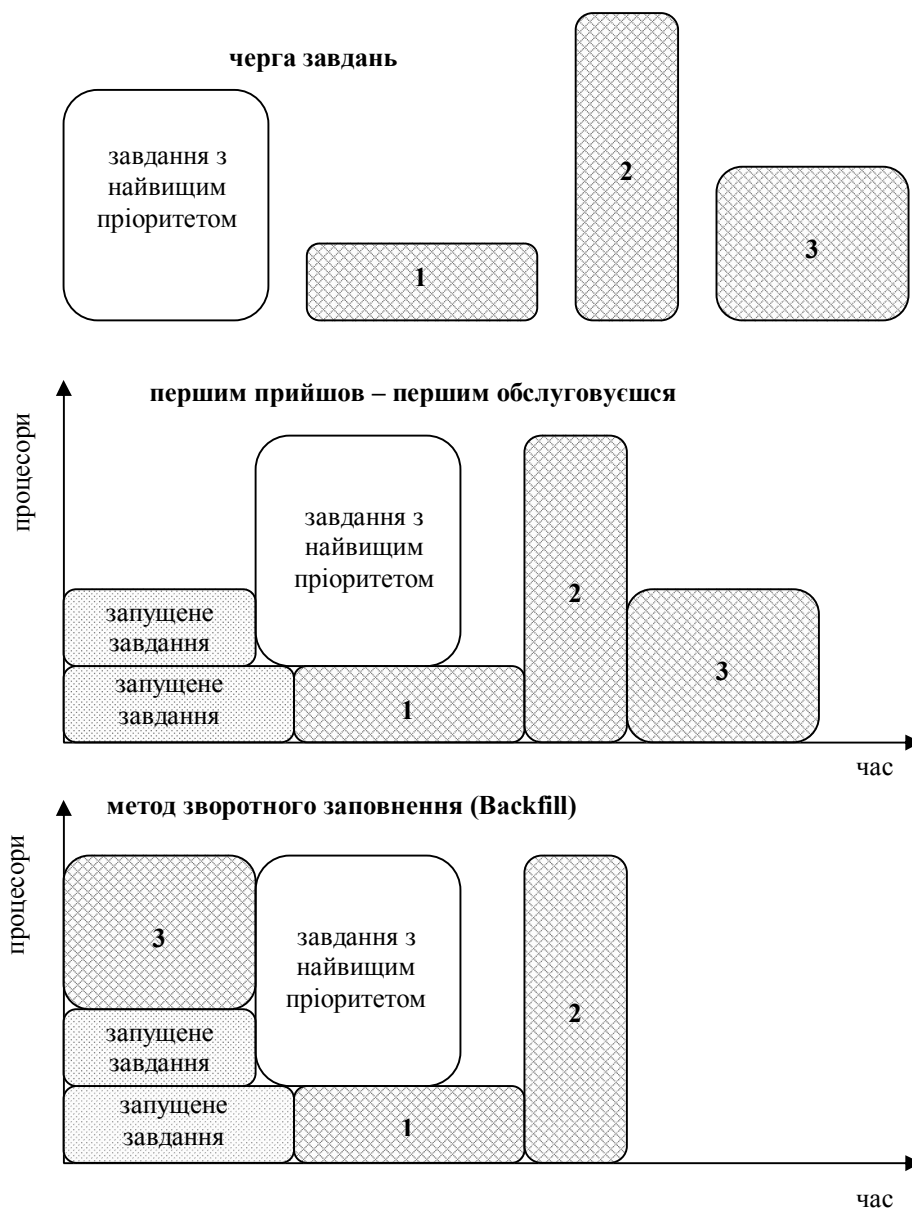


Рис. 1. Методи обробки черги завдань

Алгоритм роботи планувальника розподілених ресурсів. Опишемо механізм роботи системи. Користувач відправляє запит на ресурси, необхідні для вирішення його завдання. Запит потрапляє в чергу і згідно з його пріоритетом обробляється. Вибираються відповідні ресурси і, якщо вони в даний момент зайняті, відбувається їх резервування. Далі підбираються завдання з меншим пріоритетом, які можуть бути запущені за умови гарантованого звільнення зарезервованих ресурсів та не призведуть до затримки виконання завдань з більшим пріоритетом. Даний метод зворотного заповнення (Backfill [2]), представлений на рис. 1, де розглянутий випадок, коли завдання 3 з найменшим пріоритетом в черзі виконується раніше за інші та не призводить до їх затримки. Таким чином, ресурси використовуються більш ефективно, та користувач, який відправив завдання 3, раніше отримує результат його виконання, порівняно з класичним методом "першим прийшов, – першим обслуговується" (FCFS [3]).

Під час очікування звільнення ресурсів можливе коригування даних у разі передчасного закінчення розрахунку попередніх завдань. Далі паралельне завдання відправляється на виділені ресурси. Якщо завдання виконалося успішно, користувачеві відправляється результат. У разі збою, або зависання системи користувачеві

приходить повідомлення, а для запиту знову підбираються виконавчі ресурси. Складається розклад виділення відповідних вузлів для завдань, згідно з їх вимогами щодо ресурсів та пріоритету. Алгоритм роботи планувальника ресурсів наведено на рис. 2.

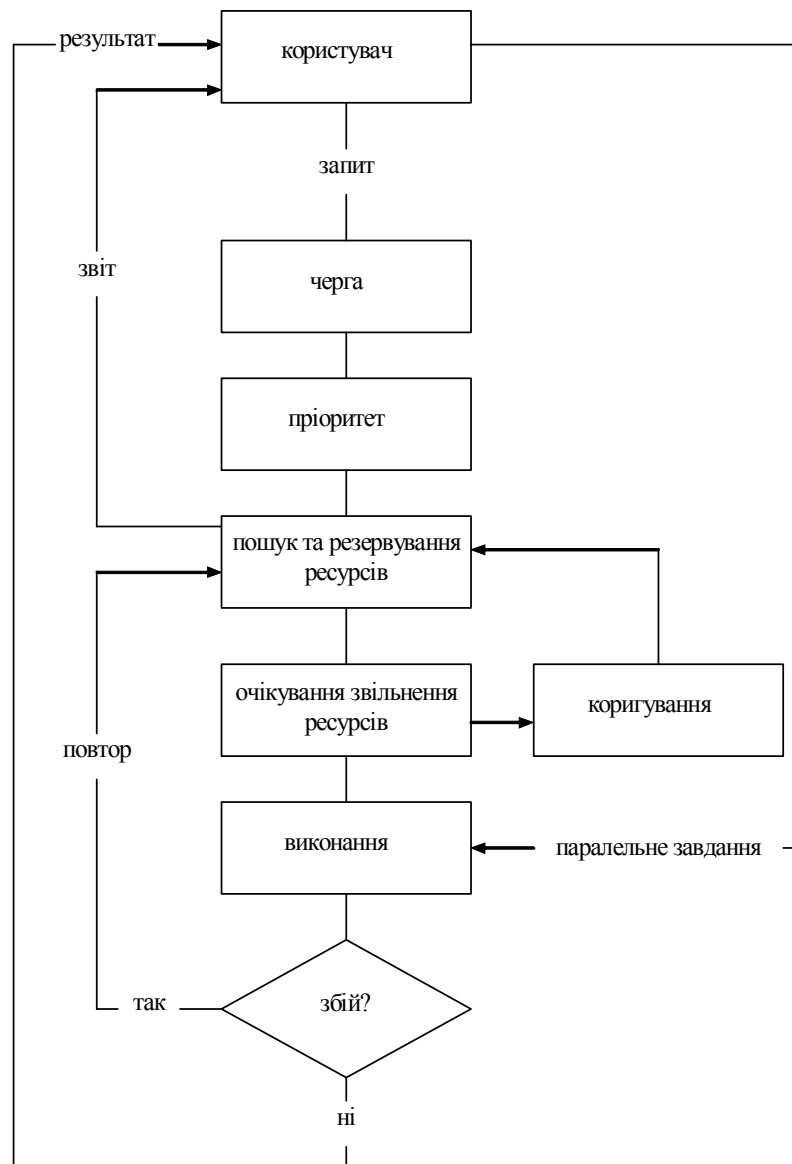


Рис. 2. Алгоритм роботи планувальника розподілених ресурсів

**Висновки.** У роботі застосовані методи: резервування, розділення часу процесора та простору ресурсів, прогнозу завантаження ресурсів, пріоритетного обслуговування завдань та випереджаючого планування. Основною перевагою розробленого алгоритму є здатність гарантувати точний час запуску паралельних завдань.

По-перше, ефективність роботи системи залежить від інформації про завдання (таку, як заданий час виконання обчислень, необхідна пам'ять тощо). Ця інформація базується на вимогах, що містяться у запитах користувача і визначається точністю їх оцінок. По-друге, ефективність залежить від інформації про ресурси (швидкість процесора, робоче навантаження та інше). Необхідно, щоб система швидко та точно реагувала на зміни стану завдань та ресурсів. Для прогнозу часу розрахунку завдань використовується заявлений завданням час вирішення  $T$ , який може не відповідати дійсності (планується надалі врахувати цей чинник). Можливий також облік вартості ресурсів та запропонованої завданнями плати.

#### ЛІТЕРАТУРА:

1. Національна GRID – інфраструктура України: <http://www.grid.ntu-kpi.kiev.ua/>
2. *Топорков В.В.* Модели распределенных вычислений – М.: ФИЗМАТЛИТ, 2004. – 320 с.
3. *Schwiegelshohn U., Yahyapour R.* Analysis of First-Come-First-Serve Parallel Job Scheduling. // In Proceedings of the 9th SIAM Symposium on Discrete Algorithms. – 1998. – Pp. 629–638. [http://www-ds.e-technik.uni-dortmund.de/~yahya/papers/cei\\_soda98.pdf](http://www-ds.e-technik.uni-dortmund.de/~yahya/papers/cei_soda98.pdf)
4. *Дьячук Т.С., Кудерметов Р.К.* Распределение заданий в многопроцессорной системе // Радиоэлектроника. Информатика. Управление. – Запоріжжя: ЗНТУ. – 2007. – № 2. – С. 51–54.

ДЬЯЧУК Тетяна Сергіївна – магістр з комп'ютерних систем та мереж, аспірант, асистент кафедри комп'ютерних систем та мереж Запорізького національного технічного університету.

Наукові інтереси:

- паралельні та розподілені обчислення;
- оптимізація обчислень;
- методи планування розподілених ресурсів;
- балансування навантаження в багатопроцесорних системах.

Тел. (моб.): +38(066)7126323.

E-mail: [diacht@zntu.edu.ua](mailto:diacht@zntu.edu.ua)

Подано 14.11.2008

Дьячук Т.С. Метод планування обчислювальних ресурсів у розподіленій системі  
Дьячук Т.С. Метод планирования вычислительных ресурсов в распределенной системе  
Diachuk T.S. Method of computing resources scheduling on the distributed system

УДК 004.75

**Метод планирования вычислительных ресурсов в распределенной системе / Т.С. Дьячук**

Разработана математическая модель планирования распределенных вычислительных ресурсов, учитывающая их особенности и потребности пользователей, получающих доступ к системе. Алгоритм способен гарантировать точное время запуска параллельных заданий с учётом приоритетов заданий и наличия свободных, или освобождающихся через определённое время, ресурсов.

УДК 004.75

**Method of computing resources scheduling on the distributed system / T.S. Diachuk**

The mathematical model of the distributed computing resources scheduling, taking into account their features and users requirements receiving access to system is developed. The algorithm is capable to guarantee an exact times of parallel tasks start in view of tasks priorities and presence free, or exempted through certain time, resources.