

*Для забезпечення вимог користувачів з продуктивності та ефективності виконання завдань грід - система повинна реалізувати ефективний алгоритм розподілу завдань між доступними на даний час обчислювальними ресурсами. Основна мета такого балансування навантаження в грід-системі - скоротити час виконання завдання користувача і забезпечити ефективність використання обчислювальних ресурсів для виключення ситуації, коли одні ресурси простояють, а інші перевантажені виконанням завдань користувачів*

*Ключові слова: грід, NordugridARC, балансування навантаження, брокер, веб-сервіс*

*Для обеспечения требований пользователей по продуктивности и эффективности выполнения заданий грид-система должна реализовывать эффективный алгоритм распределения заданий среди доступных на данный момент вычислительных ресурсов. Основная цель такого балансирования в грид-системе – сократить время выполнения задания пользователя и обеспечить эффективность использования для исключения ситуации, в которой одни ресурсы простаивают, а другие перегружены заданиями пользователей*

*Ключевые слова: грід, Nordugrid arc, балансировка нагрузки, брокер, веб-сервис*

# ГІБРИДНИЙ АЛГОРИТМ БРОКЕРА ДЛЯ NORDUGRIDARC 2.0 ДЛЯ ВИКОРИСТАННЯ В СЕКМЕНТІ УНГ

**А.І. Петренко**

Доктор технічних наук, професор, завідувач кафедри\*

Контактний тел.: (044) 287-87-71

Email: petrenko@cad.kiev.ua

**С.Я. Свістунів**

Кандидат технічних наук, старший науковий співробітник

Інститут теоретичної фізики ім. Боголюбова НАН України

вул. Метрологічна, 14-б, м. Київ, Україна, 03680

Контактний тел.: (0044) 521-31-19

E-mail: svistunov@bitp.kiev.ua

**П.В. Свірін**

Асистент\*

\*Кафедра системного проектування

Національний технічний університет України

“Київський політехнічний інститут”

пр. Перемоги, 37, м. Київ, Україна, 03056

Контактний тел.: 063-578-00-01

Email: paul.svirin@gmail.com

## 1. Вступ

Брокер ресурсів це програмний модуль, що визначає найбільш відповідний ресурс, на якому буде можливо запустити певну задачу. В системі NordugridARC [1] даний програмний модуль знаходиться на клієнтському боці і аналізує дані, отримані з інформаційної системи.

На даний момент модуль балансування в ARC використовує виключно дані щодо кластерів, отримані з інформаційної системи ARC, дані якої дуже залежні від періодичності їх оновлення. Ще одним недоліком такого підходу є неповнота опису стану ресурсу, а саме неможливо реально оцінити поточну завантаженість ресурсу. В ARC 2.0 введено можливість розробки сервісів, що запускаються на цій платформі.

Таким чином, можливе отримання додаткових відомостей щодо обчислювальної системи кластеру (наприклад, стану окремих вузлів всередині самого кластеру) для більш ефективної політики балансування.

Таким чином, необхідно розробити методику побудови брокерів, які зможуть оцінювати стан середовища і проводити відбір ресурсів за більш комплексно.

## 2. Теоретична частина

Особливостями українського сегменту ГРІД [3, 4, 5] є невелика кількість процесорів на кластерах і невеликі об'єми пам'яті на вузлах, а також не дуже швидкі канали зв'язку між кластерами. Таким чином, для максимально раціонального використання наявних ресурсів необхідно враховувати максимальну кількість факторів.

Наразі в українській грід інфраструктурі використовується алгоритм випадкового вибору, який є статичним, тобто не враховує поточного стану обчислювального ресурсу. Отже, потрібні алгоритми з більшими можливостями. Відзначимо, що статичні алгоритми звичайно є більш простими, оскільки вони використовують середньостатистичну інформацію про ресурс, ігноруючи його поточний стан. Такі алгоритми частіше застосовуються для балансування періодичних завдань з жорсткими вимогами щодо часу завершення, але в грід-середовищі такі завдання є меншістю [8]. Чисто динамічні алгоритми, що включені в комплект ARC 2.0, більш адекватно аналізують стан грід-середовища, але, як було вказано вище, не враховують продуктивність обчислювальних ресурсів. Отже, доцільно використовувати гетерогенний підхід

до розподілу задач, який буде враховувати не тільки статичну інформацію про ресурс, а і його поточний стан.

Загальна послідовність кроків для гібридного алгоритму, що використовує сервіси Nordugrid ARC 2.0:

1) опит інформаційної системи для пошуку зареєстрованих обчислювальних ресурсів;

2) фільтрація отриманого списку ресурсів за статичними характеристиками;

3) опит сервісів, що надають розширену інформацію про ресурси;

4) ранжування ресурсів із застосуванням отриманої інформації.

В якості варіанта реалізації алгоритму для середовища Nordugrid можна навести наступну послідовність кроків:

1) брокер опитує інформаційної системи ISIS, в якій реєструються обчислювальні ресурси;

2) проводиться фільтрація отриманого списку ресурсів за статичними характеристиками;

3) інформаційна система ISIS опитується для пошуку сервісів, що надають поточну інформацію про довжини всіх черг на системі виконання завдань (LRMS - localresourcemanagementsystem) на кожному з ресурсів;

4) опитуються знайдені сервіси для отримання інформації щодо довжин черг на ресурсах, якщо такі сервіси наявні;

5) розраховуються загальні довжина черги для кожного ресурсу:

$$N_i = \sum_{j=1}^n L_i + L_{ja},$$

де  $N_i$  - загальна довжина черги на  $i$ -му ресурсі,  $n$  - кількість черг, що існують на LRMS  $i$ -го ресурса,  $L_i$  - довжина  $j$ -ї черги на даному ресурсі,  $L_{ja}$  - довжина черги  $j$ -го ресурса на рівні Nordugrid ARC, тобто кількість задач, що лише очікують на подачу на рівень LRMS.

6) для кожного ресурсу генерується коефіцієнт, що розраховується наступним чином:

$$M = \text{rand}(0, \frac{1}{2^S}),$$

де  $S=N_i$ , якщо можливо отримати повну інформацію про черги на ресурсі з додаткового сервіса, або  $S=N_{i+1}$ , якщо такого сервіса на ресурсі немає.

7) ранжування ресурсів за спаданням отриманого коефіцієнту  $M$ .

### 3. Приклад реалізації алгоритму

Описаний вище алгоритм реалізовано в тестовому середовищі на базі Інституту теоретичної фізики ім. Боголюбова. Результат роботи алгоритму:

*VERBOSE: GeneratedEMIES target:  
https://arex3.bitp.kiev.ua:60003/arex*

*DEBUG: Setting status (SUCCESSFUL) for endpoint:  
https://arex3.bitp.kiev.ua:60003/arex (resource, emies)*

*DEBUG: Setting status (SUCCESSFUL) for endpoint:*

*https://arex3.bitp.kiev.ua:60003/arex (resource, <empty InterfaceName>)*

*VERBOSE: Generating A-REX target:  
https://arex2.bitp.kiev.ua:60001/arex*

*DEBUG: Setting status (SUCCESSFUL) for endpoint:  
https://arex2.bitp.kiev.ua:60001/arex (resource, wsrfg-lue2)*

*VERBOSE: Generated EMIES target: https://arex1.bitp.kiev.ua:60000/arex*

*DEBUG: Setting status (SUCCESSFUL) for endpoint:  
https://arex1.bitp.kiev.ua:60000/arex (resource, emies)*

*DEBUG: Setting status (SUCCESSFUL) for endpoint:  
https://arex1.bitp.kiev.ua:60000/arex (resource, <empty InterfaceName>)*

*...  
DEBUG: Querying queues service*

*INFO: Response:*

*<soap-env:Envelope*

*xmlns:echo="http://www.nordugrid.org/schemas/echo"*

*xmlns:soap-enc="http://schemas.xmlsoap.org/soap/encoding/"*

*xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"*

*xmlns:xsd="http://www.w3.org/2001/XMLSchema"*

*xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">*

*<soap-env:Body>*

*<echo:echoResponse>*

*<echo:lrmsstype>pbs</echo:lrmsstype>*

*<echo:qlen>0</echo:qlen>*

*</echo:echoResponse>*

*</soap-env:Body>*

*</soap-env:Envelope>*

*...*

*DEBUG: Mark generated for https://arex1.bitp.kiev.ua:60000/arex : 0.8292*

*DEBUG: Mark generated for https://arex2.bitp.kiev.ua:60001/arex : 0.4202*

*DEBUG: Mark generated for https://arex3.bitp.kiev.ua:60003/arex : 0.44*

*...*

*Job submitted with jobid:*

*https://arex1.bitp.kiev.ua:60000/arex/dGzMDmzPjWgnPE481mEO9xUnABFKDmABFKDm9UHKDmCBFKDmWTrwOn*

Для порівняння можна навести фрагмент виводу стандартного алгоритму, який обирає ресурси випадковим чином:

*VERBOSE: Generated EMIES target:  
https://arex3.bitp.kiev.ua:60003/arex*

*DEBUG: Setting status (SUCCESSFUL) for endpoint:  
https://arex3.bitp.kiev.ua:60003/arex (resource, emies)*

*DEBUG: Setting status (SUCCESSFUL) for endpoint:  
https://arex3.bitp.kiev.ua:60003/arex (resource, <empty InterfaceName>)*

VERBOSE: Generating A-REX target:  
<https://arex2.bitp.kiev.ua:60001/arex>

DEBUG: Setting status (SUCCESSFUL) for endpoint:  
[https://arex2.bitp.kiev.ua:60001/arex\(resource,wsrfglue2\)](https://arex2.bitp.kiev.ua:60001/arex(resource,wsrfglue2))

VERBOSE: Generated EMIES target: <https://arex1.bitp.kiev.ua:60000/arex>

DEBUG: Setting status (SUCCESSFUL) for endpoint:  
[https://arex1.bitp.kiev.ua:60000/arex\(resource,emies\)](https://arex1.bitp.kiev.ua:60000/arex(resource,emies))

DEBUG: Setting status (SUCCESSFUL) for endpoint:  
[https://arex1.bitp.kiev.ua:60000/arex\(resource,<emptyInterfaceName>\)](https://arex1.bitp.kiev.ua:60000/arex(resource,<emptyInterfaceName>))

...

Job submitted with jobid:  
<https://arex2.bitp.kiev.ua:60001/arex/v90FWLZcScLLJ-MyxPVRzo4SHXjrt37W80VVHSBn0rSNwAJPNgsEr8Bv>

Таким чином, було обрано ресурс arex2.bitp.kiev.ua, який не є оптимальним цільовим ресурсом для виконання завдання.

Слід відмітити, що для повноти інформації можливо також можливо додатково опитувати системи моніторингу типу Nagios чи Ganglia або проводити короткі тести для відкидання тих ресурсів, що надають про

себе коректну інформацію, але реальний стан ресурсу їй не відповідає. Також при обмеженій кількості типів задач можливо створити реєстр задач, в якому будуть зберігатись середній час виконання завдання на певному апаратному типі ресурсів і за допомоги таких відомостей знаходити ресурс з мінімальним часом очікування і виконання завдання.

#### 4. Висновки

Задача ефективного розподілу ресурсів пов'язана не лише з вибором механізму розподілу, але, можливо, більше визначається інформацією, доступною для брокера, і її актуальністю. В той же час, підтримка оновлення інформації про стан ресурсів може бути досить дорогою і вартість її зростає разом із розміром системи. Платформа NordugridARC 2.0 надає широкі можливості для розробки спеціальних сервісів, які можуть використовуватись в тому числі для прийняття брокером рішення щодо вибору ресурсу для виконання завдання. Такі сервіси можуть надавати розширену інформацію щодо ресурсу, що покращує управління інформаційними потоками і робить процес розподілу ресурсів більш ефективним. Це положення використані авторами для суттєвого поліпшення брокерських можливостей нового покоління проміжного забезпечення NorduGrid.

#### Література

1. NordugridARC. [Електронний ресурс]. - Режим доступу: <http://www.nordugrid.org>.
2. Abhishek Singh. Grid Resource Broker. [Електронний ресурс]. – Режим доступу: <http://www.cse.buffalo.edu/faculty/miller/Courses/Grid-Seminar/GridResourceBroker.pdf>.
3. GStat 2.0 – LDAPBrowser. [Електронний ресурс]. – Режим доступу: <http://gstat.egi.eu/gstat/ldap>.
4. GridMonitor. [Електронний ресурс]. – Режим доступу: <http://gridmon.bitp.kiev.ua>.
5. Загородний, А. Украинский академический грид [Текст] / А. Загородний, Г. Зиновьев, Е. Мартынов, С. Свистунов // Украинсько-македонський науковий збірник, Вип. 4 - Київ: Вид-во Національна бібліотека України імені В.І.Вернадського, 2009. - С.140-150.
6. Петренко, А.І. Національна Grid - інфраструктура для забезпечення наукових досліджень і освіти. [Електронний ресурс]. – Режим доступу: <http://netallted.cad.kiev.ua/downloads/Grid.pdf>.
7. Пономаренко, В. С. Методы и модели планирования ресурсов в GRID-системах [Текст] / С. В. Листровой, С. В. Минухин, С. В. Знахур. - Харків: ИНЖЭК, 2008 – 408 с.
8. Akbari Torkestani, J. A new approach to the job scheduling problem in cimoutational grids. / Akbari Toskani, J. // Cluster Computing. – 2012. – №15(3). – P. 201-210.

#### Abstract

*With the development of science one can observe an increase in the volume of experimental data to be analyzed to obtain new knowledge. The current volume of such computations often exceeds the capacity of a single cluster to compute in a satisfactory amount of time. Therefore, there is a necessity in a distributed computation of such amounts of data using the GRID-systems. For the load balancing in such systems one uses special mechanisms – brokers, which implement a certain policy, under which such system distributes tasks among computing resources according to utilization of computational resources in the system or according to the properties of the task.*

*To implement the algorithm of balancing a broker should take into account enough number of factors to find the most appropriate task that can perform it with the lowest possible load or in minimum amount of time. One can use regular information sources of software that maintain the GRID-system, as well as additional ones that are not included in the standard package of this software.*

*The article is dedicated to the problems of a broker construction using heterogeneous information sources.*

*The proposed method is implemented on the basis of the Ukrainian segment GRID using the software package NordugridARC 2.0. Also, this technique can be implemented on other similar software packages for the GRID*

**Keywords:** grid, NordugridARC, load balancing, broker, web service