

## КЛАСИФІКАЦІЯ АЛГОРИТМІВ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ ДЛЯ ВИКОРИСТАННЯ В РОЗПОДІЛЕНИХ ГЕТЕРОГЕННИХ МЕРЕЖАХ

Алексєєв М.О., Гусак О.В.

*Інститут телекомунікаційних систем НТУУ «КПІ», Україна*

*E-mail: valavanca@gmail.com*

### **Classification load balancing algorithms for use in distributed heterogeneous networks**

This paper provides an overview of the classification load balancing algorithms for their further use and study. Also includes examples of algorithms and a brief review. Results generalized into the table.

При реалізації алгоритмів балансування потрібно знайти баланс між затримками при розподіленні завдань і обробкою самих завдань. Оптимальне рішення залежить від характеристики додатку, типу завдання, топології мережі і вузлів, вартості розгортання рішення. Для всіх цих проблем не існує ідеального алгоритму балансування який підходив би для будь-яких умов, кожний підходить для конкретних складових і націлений на розв'язання чітких цілей. Необхідно провести класифікацію існуючих алгоритмів для того щоб використовувати наявні алгоритми максимально доцільно. Розробка такої класифікації стимулюватиме розробку нових алгоритмів і прискорить вдосконалення існуючих.

Для продуктивного і більш раціонального використання мереж з розподіленими розрахунками необхідно реалізувати дві вимоги які часто конкурують. Перш за все це те що процесори повинні бути зайняті корисною роботою. Другий пункт накладає вимоги до міжпроцесорного зв'язку – він повинен бути мінімальним. Для більшості розподілених систем розрахунку вартість налагодження контакту зростає при збільшенні кількості і об'єму завдань. Важливо мінімізувати результуючу суму з них або максимум з них. Всі алгоритми балансування навантаження можна розділити на три групи: динамічні, статичні, адаптивні.

Динамічні алгоритми розподілення навантаження застосовують інформацію про стан системи для прийняття рішення про розподілення навантаження. Статичні алгоритми не слідкують за станом системи. В даній групі алгоритмів рішення про розподілення навантаження приймається заздалегідь і не змінюється в процесі роботи. Прикладом таких алгоритмів може слугувати RR (round robin), коли відбувається присвоєння задач до розрахункових вузлів по круговому циклу. Перша задача присвоюється першому розрахунковому вузлу друга другому і так до останнього вузла, потім все починається спочатку. Найбільш розповсюджена реалізація даного алгоритму на DNS-сервері шляхом повторення IP адрес вузлів в необхідній пропорції розподілення навантаження. Також до статичних відносять Simulated Annealing or Genetic Algorithms (генетичний алгоритм) який забезпечує змішану процедуру розподілу завдань за обчислювальними ресурсами з використанням методів оптимізації. Динамічне балансування передбачає перерозподіл

розрахункової потужності на вузлах в ході виконання програми і завдяки цьому можуть обігнати статичні алгоритми. Завдяки тому, що динамічні алгоритми відслідковують і аналізують такі параметри як завантаження розрахункових вузлів, пропускну здатність лінії, частоту обміну повідомленнями між логічними процесами, це призводить до збільшення накладних розрахунків на балансування в порівнянні із статичними алгоритмами.

Адаптивні ж алгоритми використовують поєднання двох вище наведених алгоритмів. В залежності від умов і поточних параметрів розрахункової системи вибирається найбільш доречний алгоритм. Наприклад Derbal[1] використовує масштабове планування основане на ентропії розрахункової системи, а вже за допомогою ланцюгів Маркова сумує динамічні служби в мережі. Для прикладу є система[2] з підтримкою поколінь планування (the introduction of the Generational Scheduling) з реплікацією завдань. Цей алгоритм адаптується до змін в продуктивності з реструктуризацією завдань.

Відповідно до цілісності передачі завдання розділяють на витісняючі (preemptive) і невитісняючі (non-preemptive) алгоритми балансування. Не витісняючі алгоритми можуть передавати задачі які ще не почали розв'язувати. Витісняючі алгоритми здатні передавати задачі які розв'язані частково. Прикладом застосування не витісняючого алгоритму – сервер NetWare, витісняючого – ОС Windows NT, OS/2, VAX/VMS, UNIX.

Розділяють динамічні алгоритми балансування які враховують топологію і які її не враховують. По централізованості динамічні алгоритми розділяють на групи: централізовані, повністю розподілені, ієрархічні, суміщені.

Потенційно централізовані алгоритми менш надійні, ніж повністю розподілені, оскільки збій в роботі центрального компоненту призведе до збою всієї системи. Також центральний компонент потрібно забезпечити великою пропускну здатністю. Ієрархічні алгоритми частково вирішують ці проблеми, уникнути їх можна лише при повністю розподілених системах. Перевагою централізованого алгоритму є спрощення контролю і налагодження. Централізований спосіб підходить тільки тоді коли завдання може виконуватися незалежно і асинхронно відповідно до центрального процесора[3]. Необхідно мінімізувати обмін інформацією між центральним і підлеглим вузлом. Прикладом може слугувати алгоритм Least Load. Це динамічний алгоритм із зворотнім зв'язком. Завдання надсилається на той розрахунковий вузол який менш всього завантажений. Завантаженість може визначатися середнім часом відповіді.

У повністю розподіленій системі з  $N$  обчислювальними ресурсами для прийняття рішення про планування ресурс має обмінятися  $2(N - 1)$  повідомленнями[4]. Частковим прикладом для балансування такого типу може слугувати алгоритм[5] Ричарда Карпа (Richard Karp) який використовується для однорангових мереж (P2P). В даному алгоритмі інформація про однорангові вузли зберігається локально в каталогах кожного «едмеїенту» мережі. Таким чином значну частину проблем балансування було знято з сервера і передано на вузли.

Прикладом ієрархічної системи є мережа з проміжним забезпеченням gLite в якому система управління потоком знань(WMS) відповідає за розподіл і управління завдань. WMS контролює процес виконання завдань і відправку

результатів замовнику. В gLite для балансування навантаження використовується алгоритм «максимальної/нечіткої оцінки» які проводять відбір ресурсів по відповідності до поставлених задач. В суміщеній розподіленій системі на список завдань кожний розрахунковий вузол формує список доступних ресурсів. Актуальну інформацію про стан системи можна отримати як від кожного вузла напряму так і від центральної інформаційної системи. Така модель реалізована в проміжному забезпеченні ARC проекту NorduGrid.

	ДИНАМІЧНИЙ				АДАПТИВНИЙ	СТАТИЧНИЙ
	Суміщений	Ієрархічний	Повністю розподілений	Централізований		
Genetic Algorithms						
RR						
Derbal						
The introduction of the Generational Scheduling						
Least Load						
алгоритм Richard Karp для P2P						
gLite (WMS)						
ARC (NorduGrid)						

Ця стаття поліпшує використання і вдосконалення алгоритмів балансування шляхом створення наглядної класифікації. Реалізовано короткий огляд найбільш розповсюджених алгоритмів з віднесенням їх до певної категорії. Кожний з підходів має свої переваги і недоліки і шляхом об'єднання і групування можна облегшити вибір необхідного алгоритму. Проявляється тенденція що чим складніша топологія і конфігурація мережі тим складніший алгоритм необхідно застосовувати.

### Література

1. Derbal, Y.: Entropic Grid scheduling. J Grid Computing 4, 373–394 (2006)
2. Lucchese, F., Huerta, E.J., Yero, Sambatti, F.S.: Anadaptive scheduler for Grids. J Grid Computing 4, 1–17 (2006).
3. Bruce Hendrickson \*, Karen Devine Dynamic load balancing in computational mechanics ст.488
4. Bokhari S. Partitioning Problems in Parallel, Pipelined, and Distributed Computing// IEEE Transactions Computers. — 1988. — 37, № 1. — P. 48–57.
5. Load Balancing in Dynamic Structured P2P Systems; Brighten Godfrey Karthik Lakshminarayanan Sonesh Surana Richard Karp Ion Stoica.