

Технічні науки

УДК 004.75

Мітряєв Сергій Сергійович

студент

Харківського національного університету радіоелектроніки

Митряев Сергей Сергеевич

студент

Харьковского национального университета радиоэлектроники

Mitriaiev Sergiy

Student of the

Kharkiv National University of Radioelectronics

Андрющенко Ірина Ігорівна

студентка

Харківського національного університету радіоелектроніки

Андрющенко Ирина Игоревна

студентка

Харьковского национального университета радиоэлектроники

Andriushchenko Iryna

Student of the

Kharkiv National University of Radioelectronics

**БАЛАНСУВАННЯ НАВАНТАЖЕННЯ ПРИ ПАРАЛЕЛЬНИХ
ОБЧИСЛЕННЯХ**

**БАЛАНСИРОВАНИЕ НАГРУЗКИ ПРИ ПАРАЛЛЕЛЬНЫХ
ВЫЧИСЛЕНИЯХ**

LOAD BALANCING IN PARALLEL COMPUTATIONS

Анотація. Досліджено теоретичні питання застосування балансування навантаження при паралельних обчислень.

Ключові слова: балансування навантаження, паралельні обчислення, статичне балансування, динамічне балансування.

Аннотация. Исследованы теоретические вопросы применение балансировки нагрузки при параллельных вычислениях.

Ключевые слова: балансировка нагрузки, параллельные вычисления, статическая балансировка, динамическая балансировка.

Summary. The theoretical aspects of Load balancing in parallel computations.

Key words: load balancing, parallel computations, static balancing, dynamic balancing.

Основна причина уповільнення розгортання паралельних програм полягає в тому, що ефективні паралельні програми важко писати. Паралельне програмування додає до програмування другий вимір: не тільки коли буде виконуватися конкретна операція і який процесор буде її виконувати. Величезна кількість додатків, які розпаралелені, не мають регулярної структури для ефективного розпаралелювання. Такі додатки вимагають балансування навантаження для ефективної паралельної роботи. Навантаження в цих додатках також може змінюватися з часом, що вимагає перебалансування. Програмісту надається вибір: або безладно розподіляти обчислення, створювати погано працюють програми, або витратити більше часу на розробку, включаючи код балансування навантаження в додатку.

В останні роки з'явилися нові типи «паралельних» комп'ютерів . Мережі побутових робочих станцій роблять паралельні обчислення доступними для зростаючої групи дослідників [4, с. 206]. Мережі робочих

станцій представляють нові проблеми для прикладного програміста. Тепер, крім дисбалансу додатків, паралельна програма повинна мати справу з фоновим навантаженням від інших одночасних користувачів. Паралельні програми можуть виконуватися на кластерах робочих станцій на робочих місцях інтерактивного користувача, де основний користувач дозволяє паралельні обчислення тільки тоді, коли комп'ютер не використовується в інтерактивному режимі [5]. Нарешті, обчислювальні кластери можуть з часом розширюватиметься, але з швидким збільшенням обчислювальної потужності нові процесори, ймовірно, будуть швидше, ніж старі машини, які вони доповнюють. Щоб максимізувати пропускну здатність, балансувальник навантаження в паралельних програмах повинні враховувати всі ці фактори.

Робоча міграція - це уніфікована схема для усунення дисбалансу навантаження [2], пов'язаної як з додатками, так і з зовнішніми причинами. Труднощі з перенесенням роботи полягає в тому, що або робота перерозподіляється в залежності від програми, що лягає тягарем на програміста програми, або підтримується автоматична міграція, але з низькою точністю через відсутність знань про конкретний додаток.

Міграція об'єктів забезпечує спосіб виконання точної і детальної автоматичного балансування навантаження. У об'єктів зазвичай є невеликі чітко визначені області пам'яті, в яких вони працюють, що знижує вартість міграції. Використовуючи об'єктну модель Charm ++, система часу виконання вимірює роботу, представлену конкретними об'єктами, а не витягує час виконання з евристики конкретного додатка. Крім того, виконуюча система записує шаблони взаємодії між об'єктами.

Існує лише два типи балансування навантаження - статичне та динамічне [1].

Перше, статичне балансування виконується перед виконанням програми. Часто при цьому використовуються генетичні алгоритми, під

час розподілу логічних процесів застосовується досвід з попереднього застосування програми. Але попереднє розміщення процесів не має ніякого ефекту. Це можна пояснити тим, що обчислювальний вузол, де виконується розподілена програма може бути вже зайнятим іншим процесом та іншими обчисленнями, частка яких має можливість вирости [2]. Та у вузлі може бути змінена обчислювальна середа, будь-який вузол для обчислення може вийти з ладу.

Через будь-яку з цих причин виграш від попереднього розподілення логічних процесів по обчислювальним вузлам для паралельної обробки стає неефективним.

Динамічне балансування навантаження включає в себе перерозподіл обчислювального навантаження на вузли під час виконання додатку [3]. ПЗ, у якому реалізоване динамічне балансування можна визначити за певною пропускнуою спроможністю ліній зв'язку та частоті обміну повідомлень між різними вузлами.

Дані, зібрані о розподіленому додатку є фактором, на основі якого система приймає рішення о переносі логічних процесів з одного вузла на інший.

Література

1. Yemelianov A. Балансировка нагрузки: основные алгоритмы и методы / Andrei Yemelianov. 2015. URL: <https://habr.com/ru/company/selectel/blog/250201/>
2. Prashant J. S. Static Load Balancing / J. S. Prashant, M. V. Harrick // Introduction to Multimedia Storage Servers. 2002. URL: <https://www.sciencedirect.com/topics/computer-science/static-load-balancing>
3. Балансировка нагрузки в распределенных системах. URL: <http://masters.donntu.org/2012/fknt/volokhova/library/article3.htm>

4. Качко Е. Г. Параллельное программирование: Учебное пособие / Е. Г. Качко. Харьков: Форт, 2011. 528 с.
5. Сибгатулин М. ЕСМР и превратности балансировки на сетевом оборудовании / Марат Сибгатулин. 2019. URL: <https://nag.ru/articles/reviews/105272/ecmp-i-prevratnosti-balansirovki-na-setevom-oborudovanii.html>