

УДК 519.68

К.т.н., доцент Волкогон В. П., магістрант Бойко О. А.

Національний технічний університету України  
«Київський політехнічний інститут»

## ДОСЛІДЖЕННЯ МЕТОДІВ І ТЕХНОЛОГІЙ ПАРАЛЕЛЬНОГО ПРОГРАМУВАННЯ

### Abstract

*Viktor P. Volkogon, assoc. prof., PhD; Olga Boiko, student  
Research of parallel programming methods and technologies*

*This paper concerns the task of effective use of multiprocessor computer systems. Three different ways of solving this problem are researched and discussed. Their advantages and disadvantages are identified. Practical importance of this research is ability to choose the most effective solving of the task.*

### Вступ

Зростання обчислювальних потужностей комп'ютерної техніки підвищує інтерес до неї все більшої кількості галузей людської діяльності. Це в свою чергу спричиняє виникнення нових задач, що потребують ще більших потужностей.

Раніше продуктивність нарощувалась підвищенням тактової частоти процесора – збільшенням кількості елементів на його кристалі. Але таке розширення втрачає свою ефективність через обмеження, накладені фізичними властивостями матеріалів. Тому зараз виробники обчислювальної техніки нарощують потужність об'єднанням кристалів.

Така нова архітектура добре працювала для багатьох малих додатків, але для вирішення однієї великої задачі себе не виправдала. Таким чином постала проблема відійти від традиційних послідовних програм до таких, що дозволять оптимально використовувати всі обчислювальні ресурси.

### Постановка задачі

Існує три основних підходи до забезпечення ефективної експлуатації комп'ютерів з паралельною архітектурою: використання паралельних мов, використання бібліотек та автоматичне розпаралелювання програм [1]. Всі ці напрями ефективно розвиваються, але кожен має ряд недоліків, що обмежують клас задач, які можна розв'язати.

Метою даної роботи є порівняння існуючих підходів для можливості вибору ефективного методу використання багатопроцесорних систем прикладним програмним забезпеченням з максимальною продуктивністю.

### **Технології паралельного програмування**

Головне призначення паралельних систем – швидке розв'язання задач. Якщо технологія програмування не дозволяє використовувати весь потенціал обчислювальної системи, використання її втрачає сенс.

Різноманітність архітектур паралельних ЕОМ зумовила різноманітність підходів до їх програмування. Якщо спробувати зробити огляд існуючих засобів, що можуть допомогти в вирішенні таких задач, то навіть поверховий аналіз приведе до списку з понад 100 найменувань. Тому вибір технології паралельного програмування стає справді складним питанням. Розглянемо декілька підходів.

### **Мови паралельного програмування**

Спеціально розроблені мови високого рівня, що включають механізми як послідовного, так і паралельного програмування. Вони забезпечують високий рівень абстракції, що дозволяє зосередитись власне на розв'язанні задачі. Крім того, послідовні і паралельні механізми можна поєднувати, щоб вони працювали разом, а аналогічні алгоритми мали схожий вигляд. Ці дві властивості полегшують створення і розуміння програм. Ще одна перевага використання таких мов полягає в контролі типів даних, що зменшує кількість помилок, дозволяє створювати короткі і стійкі програми. Головна задача розробників таких мов – розробити хороший набір абстракцій, потужний і зрозумілий набір механізмів, а також ефективну реалізацію [2].

До таких мов належать Linda, Orca, Occam, НОРМА та інші, а також паралельні розширення деяких популярних мов програмування, таких як Fortran та С.

Недоліком такого підходу є необхідність переписування програм та алгоритмів для паралельного виконання. Крім того для написання паралельних програм програмісти повинні опанувати нові мови, що часто є економічно не вигідним.

### **Бібліотеки паралельного програмування**

Представляють собою набір підпрограм, які забезпечують створення процесів, управління ними, їх взаємодію і синхронізацію. Ці підпрограми,

а особливо їх реалізація, залежать від того, який вид паралельності підтримує бібліотека – з спільними змінними чи обміном повідомленнями.

До бібліотек, оснований на передачі повідомлень, відносять MPI та PVM. Обидві мають широкодоступні реалізації, що підтримують C і Fortran. Даний вид паралельної обробки найефективніше реалізувати в розподілених системах. Наприклад в системах віддаленої обробки даних, пов'язаних через інтернет.

В зосереджених системах зазвичай використовується розділення змінних. За цим принципом працюють бібліотеки Pthreads та OpenMP. Остання представляє собою набір директив компілятора, бібліотечних процедур і змінних середовища, призначених для програмування багатопоточних додатків, і на даний момент успішно розвивається і впроваджується [3].

Використання бібліотек дещо розв'язує проблеми необхідності створення нових алгоритмів і навчання програмістів, однак часто їх інтерфейси є громіздкими і ускладнюють розуміння програм.

### **Автоматичне розпаралелювання**

Основна задача розпаралелюючого компілятора – виділити максимум прихованого паралелізму з повторювальних ділянок послідовної програми, що визначають час її виконання: циклів та рекурсивних процедур. Це вимагає зміни порядку виконання операторів на цих ділянках, що можливо лише при відсутності залежності по даним між операторами.

Таким чином спершу необхідно виконати аналіз залежності: визначити, які частини послідовної програми незалежні і можуть виконуватись паралельно, а які потребують послідовного виконання або іншої форми синхронізації. Залежності по даним легко визначити в послідовному коді, що містить посилання лише на скалярні змінні. Набагато складніше це зробити в циклах і у випадку посилань на масиви, які зазвичай використовуються разом і використовують спільні змінні.

Перевірка залежності в загальному вигляді є складною (а в гіршому випадку – нерозв'язною) задачею. Ефективні перевірки існують лише для часткових випадків. Тому перед розробниками постає задача визначення алгоритму, який має бути застосований [4].

Наступним кроком є реструктуризація програми. Алгоритми розпаралелювання циклів в основному базуються на:

- декомпозиції графу залежностей на сильно-зв'язні компоненти;
- унімодулярних перетвореннях циклу(спеціальних або автоматично згенерованих);
- плануванні(одномірному або багатомірному).

Кожна з цих груп використовує свої математичні інструменти, потребують на вході особливого представлення залежностей по даним, і тому покриває конкретний клас задач. Таким чином важливою лишається задача пошуку оптимального алгоритму для кожної конкретної задачі [1].

Ідея автоматичного розпаралелювання є дуже привабливою, оскільки вирішує проблему переносу вже існуючих послідовних програм на паралельну архітектуру. Однак, через складність виявлення і реалізації прихованого паралелізму, досі не існує універсального рішення.

## Порівняння методів паралельного програмування

Порівняємо підходи до паралельного програмування визначивши їх переваги та недоліки:

### Паралельні мови програмування

#### *Переваги:*

- прозорість коду;
- порівняно короткі програми;
- концепція паралелізму прихована в самій мові.

#### *Недоліки:*

- необхідність вивчення правил написання паралельних програм;
- необхідність видозміни алгоритмів.

#### *Використання:*

Написання невеликих ефективних програм.

### Бібліотеки паралельного програмування

#### *Переваги:*

- можливість використовувати послідовні мови;
- простота написання програм.

#### *Недоліки:*

- прив'язаність бібліотеки до конкретної платформи;
- громіздкість коду;
- необхідність контролю потоків і синхронізації.

#### *Використання:*

Швидке написання паралельних програм.

### Розпаралелюючий компілятор

#### *Переваги:*

- можливість розпаралелювання вже існуючих програм;
- відсутність втручання людини процес розпаралелювання;
- непотрібність навчання.

#### *Недоліки:*

- складність реалізації;
- складність досягнення оптимальної продуктивності.

#### *Використання:*

Перенесення послідовних програм на паралельні архітектури.

## Висновки

У даній роботі розглянуто основні шляхи розв'язання проблеми ефективного використання багатопроцесорних комп'ютерних систем. Кожен з них має свої переваги і недоліки та покриває лише конкретних клас задач.

Проведений в роботі аналіз має практичний інтерес. Він дозволяє визначитись у виборі технології програмування для паралельних архітектур, яка забезпечить максимальну ефективність і віддачу.

## **Література**

1. Программные средства и математические основы информатики. – Новосибирск: Институт систем информатики им. А.П. Ершова СО РАН, 2004. – 278 с.
2. Эндрюс Г. Р. Основы многопоточного, параллельного и распределенного программирования.: Пер. с англ. – М.: «Вильямс», 2003. – 512 с.: ил.
3. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2002. – 608 с.: ил.
4. Анализ зависимостей: основные тесты на зависимость по данным – Сибирский журнал вычислительной математики, – Том 10, 2007.