УДК 004.4’232

К.т.н., доцент Марченко О. І., студент Філатов О.М.

Національний технічний університет України
«Київський політехнічний інститут»

СПОСІБ ДОДАВАННЯ ФУНКЦІОНАЛЬНИХ ВЛАСТИВОСТЕЙ ОПИСУ ПАРАЛЕЛІЗМУ ДО ПРОЦЕДУРНОЇ МОВИ

Abstract

Olexandr I. Marchenko, assoc. prof., PhD; Alexey Filatov, student
Technique for adding functional features to a procedural language

This paper is devoted to the development of the technique for adding functional features to a procedural programming language. The paper discusses ways of functional data structures description, and features of parallelizing operations performed on different types of arrays, which made it possible to create software for automatic parallelization.

Вступ

Постійне збільшення обчислювальних можливостей 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. Останнім часом, основним підходом виробників обчислювальної техніки до підвищення продуктивності стало об’єднання декількох обчислювальних ядер в один процесор. Але такий підхід не виправдав себе при виконанні однієї програми, що вирішує окрему велику задачу, бо використовувалось тільки одне ядро. Таким чином постала проблема відійти від традиційних послідовних програм до таких, що дозволяють оптимально використовувати всі обчислювальні ресурси, зокрема багатоядерні процесори.

Головне призначення паралельних систем[1] – швидке розв’язання задач. Тому, якщо існуючі для конкретної системи програмні засоби не дозволяють використовувати увесь потенціал обчислювальної системи, використання такої системи втрачає сенс. Різноманітність архітектур паралельних комп’ютерних систем зумовила різноманітність підходів до їх програмування. Серед основних можна виділити наступні: паралельні операційні системи, мови паралельного програмування, бібліотеки паралельного програмування та, так звані, розпаралелюючі компілятори, що можуть автоматично розпаралелювати послідовні програми.

Історично так склалося, що в промисловому програмуванні, в основному, використовуються процедурні мови програмування, але виконати автоматичне розпаралелення на рівні компіляції набагато легше для функціональних мов. Ця особливість витікає саме з функціонального характеру мовних конструкцій, які не вказують точної процедури виконання дій, а лише описують функціональні взаємозалежності між поняттями задачі, що вирішується. А такий спосіб програмування залишає набагато більше можливостей для розпаралелення на етапі компіляції. Тому надання функціональних властивостей поширеним процедурним мовам програмування шляхом додавання до них нових конструкцій опису функціонального характеру є важливим завданням.

Постановка задачі

Задача полягає в розробці способу додавання функціональних властивостей до конструкцій опису паралелізму існуючої процедурної мови програмування.

Термінологія

Багатозадачність – властивість системи, що дозволяє одночасно працювати з декількома наборами команд та даних, які також можуть взаємодіяти між собою.

Препроцесор – програма, яка виконує попередню обробку даних, для того, щоб вони могли використовуватись іншою програмою, такою як компілятор.

Паралельний масив – структура однотипних даних, операції над якими можна виконувати одночасно.

Опис алгоритму

Розширення існуючої процедурної мови конструкціями з функціональними властивостями має задовольняти наступним вимогам:

1) змінювати синтаксиси існуючої мови для реалізації процедурної;
2) надавати можливість роботи з різними компіляторами;
3) покращувати час виконання скомпільованих програм;
4) надавати можливість вибору способів розпаралелення;
5) надавати можливість вибору кількості паралельних потоків.

Спосіб додавання функціональних властивостей до конструкцій опису паралелізму існуючої процедурної мови програмування, що пропонується, полягає в наступному:

1) для опису можливого паралелізму до процедурної мови додається спеціалізована конструкція функціонального характеру, а саме тип

даних «паралельний масив».

2) реалізація додаткового типу даних «паралельний масив» виконується за допомогою препроцесора, що створюється для існуючої процедурної мови.
3) для виконання дій над «паралельним масивом» пропонується спеціальна методика, реалізація якої також планується в рамках препроцесора.

Тип даних «паралельний масив» характеризується тим, що його елементи є незалежними для обробки. Це виключає необхідність пошуку залежностей за даними, що, в свою чергу, суттєво спрощує процес розпаралелення його обробки. Нижче розглядається методика розпаралелення виконання операцій над даними типу «паралельний масив».

Для вирішення поставленої задачі пропонується створити препроцесор мови програмування C#, який буде здатний перетворювати тип даних «паралельний масив» та оператори його обробки у паралельний код, що може бути оброблений довільним компілятором мови програмування C#. Такий препроцесор повинен автоматично обирати способи розпаралелення операторів обробки «паралельних масивів» різної розмірності.

Методика розпаралелення обробки даних типу «паралельний масив»

Розпаралелена програма може бути ефективно виконана тільки на багатопроцесорній/багатоядерній системі. Тому розподіл елементів «паралельних масивів» між потоками так, щоб час, необхідний на обмін даними, був мінімальним, є важливим підзавданням розпаралелення.

Для розпаралелення операцій над двовимірними «паралельними масивами» («паралельними матрицями») пропонується наступна методика.

«Паралельні матриці» пропонується розподіляти кількома способами[2]:

- виконати розподіл тільки рядків;
- виконати розподіл тільки стовпчиків;
- виконати розподіл як рядків, так і стовпчиків.

Конкретний спосіб розподілу «паралельних матриць» визначається діями, які будуть виконуватися над ними. Так, при виконанні додавання та віднімання матриць, коли операції виконуються поелементно, всі три «паралельні матриці» (дві матриці операндів та матриця результату) можна розбити одним із вище наведених способів, при цьому немає необхідності пересилати дані між паралельними потоками.

Інша ситуація виникає при виконанні операції множення

«паралельних матриць»[3], коли необхідно враховувати специфіку операції множення. Для обчислення елемента результуючої матриці з індексами [i,j] потрібна i-та рядок першої матриці та j-й стовпчик іншої матриці. В цьому випадку зручніше розподілити першу матрицю за рядками, а другу – за стовпчиками.

З метою ефективного розпаралелення операції множення матриць на задану кількість процесорів/ядер поділ початкових матриць-операндів на матриці меншого розміру повинен виконуватись, враховуючи доцільну кількість потоків, в яких буде виконуватись операція множення матриць.

Розроблена методика розпаралелення дозволила збільшити швидкість виконання програми при виконанні операції множення над матрицями розмірності 100 на 100 на 5%. При виконанні операції додавання над матрицями тієї ж розмірності швидкість збільшилась на 3%. Виміри проводились на двоядерному процесорі з 64 розрядною операційною системою Microsoft Windows 7 та компілятором, що входить до програмного комплексу Microsoft Visual Studio 2010.

Висновки

Запропоновано спосіб додавання функціональних властивостей опису паралелізму до процедурної мови. Для реалізації цього способу запропоновано створення препроцесора до мови програмування C#, який буде оброблювати додатковий тип даних «паралельний масив», та автоматично розпаралелювати операції, що виконуються над даними цього типу. Крім того запропонована методика розпаралелення операцій над «паралельними матрицями» в залежності від розміру матриць та кількості наявних процесорів/ядер.

Література

1. *В.А. Евстигнеев.* Основы параллельной обработки. Учебное пособие. - Новосибирск. НГУ.- 1996.- 76 с.
2. *С.А. Лазарева.* Многоуровневое представление программ и его использование в автоматическом расспараллеливании. Т.9.- N.2.- 1997.- 235 с.
3. *М. Вольф.* Перестановка циклов. Векторизация программ: теория, методы, реализация. Сборник статей.- М.: Мир.- 1991.- 150 с.