

СИСТЕМА ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ НА КОМП'ЮТЕРНІЙ МЕРЕЖІ ПАРКС-.NET

Деревянченко О.В.¹⁾, Хавро А.Ю.²⁾

Київський національний університет імені Тараса Шевченка

^{1)к.ф.-м.н., доцент; 2)студент}

I. Вступ

В роботі розглядається проектування системи для паралельних обчислень на комп'ютерній мережі із застосуванням технологій ПАРКС та .NET. На факультеті кібернетики Київського національного університету імені Тараса Шевченка технологія ПАРКС [1-3] застосовується для навчання студентів паралельному програмуванню [4]. Ця робота присвячена розширенню (паралельні властивості) множини алгоритмічних мов програмування (Pascal, C, JAVA[3]), які вже реалізовані за допомоги технології ПАРКС, новими за допомогою технології .NET та мови програмування C#[5].

II. Технологія Microsoft .NET Framework 4

Microsoft .NET Framework 4 – сучасна програмна платформа, що дозволяє створювати програми із застосуванням комп'ютерних мереж та паралельних обчислень, а також надає можливість переносу програм між різними апаратними платформами, на яких встановлене середовище Microsoft .NET Framework 4 (фактично віртуальна машина). Однією з переваг .NET є сумісність бібліотек, написаних різними мовами. Система ПАРКС-.NET написана на мові C# [5], але її можна використовувати і для інших мов .NET, наприклад Visual Basic або C++.

До позитивних якостей технології .NET, що вплинули на її вибір для проектування системи ПАРКС-.NET, також належать:

- наявність мови C#, яка набуває все більшої популярності завдяки простій об'єктній моделі та постійному розвитку;
- відповідність сучасним вимогам: бібліотека TPL (Task Parallel Library), яка використовується в системі ПАРКС-.NET і надає широкий набір можливостей для розробки паралельних програм, підтримка засобів мережевого програмування, забезпечення пересилання даних по комп'ютерній мережі, рефлексія – процес виявлення типів під час виконання та інше.

III. Технологія ПАРКС

ПАРКС (паралельні асинхронні рекурсивні керовані системи) - технологія програмування, що являє собою деяку сукупність програмних засобів, які забезпечують процес розробки і реалізації алгоритмів паралельної обробки інформації. Вона базується на концепції керуючого простору (КП)[1] – динамічний граф, який використовується для опису логічної та комунікаційної структури досліджуваної задачі.

Основні терміни ПАРКС-технології:

- точка;
- програмний канал (ПК);
- алгоритмічний модуль (АМ).

Структура КП – граф, вершини якого – точки КП, ребра – ПК. До кожної точки КП приписаний АМ, який є процедурою ПАРКС-розширення базової мови. В нашому випадку ми застосовуємо базову мову програмування C#.

Програмні засоби розширення базових алгоритмічних мов програмування [1] :

1. Програмні засоби, що забезпечують побудову і модифікацію КП.
2. Програмні засоби, що забезпечують зберігання та передачу інформації за допомогою КП.
3. Програмні засоби, що забезпечують роботу з АМ.
4. Програмні засоби, що забезпечують процедурно – об'єктно-орієнтоване програмування.

IV. Система ПАРКС-.NET

Архітектура системи складається з наступних частин:

1. *Parcs* – основна бібліотека класів системи. Містить всі основні класи, що будуть використовуватись для моделювання паралельних обчислень.
2. *Daemon* – клієнт. Являє собою програму, за допомогою якої будуть проводитись обчислення.

3. *HostServer* – сервер. Має список доступних клієнтів, а також інформацію про поточні задачі та точки. При запуску модуль починає роздавати завдання клієнтам в залежності від кількості ядер процесорів (потужності) та їх завантаженості.
4. АМ – окремий проект, що використовує бібліотеку *Parcs* та реалізує інтерфейс *Module*.

На всіх машинах, що призначені для обчислень, запускається програма *Daemon*, а початковий АМ та *HostServer* можуть запускатися або на тих же, або на інших машинах. В окремому випадку навіть всі три елементи системи можуть бути запущені в одному місці. Файл *server.txt* використовується початковим АМ для знаходження комп'ютера, на якому запущений *HostServer* (IP-адреса).

У створеній реалізації клієнта та сервера зв'язок проходить у такій послідовності:

- 1) При запуску модуля сервер перевіряє всі зазначені у файлі *hosts.txt* і видаляє ті, з якими не вдалося з'єднатися. Після цього сервер запитує у кожного з клієнтів кількість ядер процесора і зберігає собі ці результати.
- 2) Сервер розподіляє точки, що створюються в алгоритмічному модулі, по комп'ютерах, на яких виконується програма *Daemon*.
- 3) Після цього на комп'ютері, де буде створена нова точка, початковий АМ передає ехе-файл, в якому міститься інформація про всі АМ, що будуть запущені під час виконання поточної задачі. За допомогою рефлексії *Daemon* створює екземпляр класу АМ і викликає метод *Run()*. Таким чином запускаються необхідні обчислення.
- 4) Після завершення обчислень *Daemon* відсилає результат назад на початковий АМ. Після того, як завершилися обчислення на всіх машинах, початковий АМ оброблює всі результати, виводить загальний результат на екран та зберігає його в файл.

V. Тестування системи ПАРКС-.NET

Для тестування роботи системи був реалізований класичний алгоритм множення матриць. Обчислення проводились на комп'ютерній мережі із застосуванням 4-х двоядерних процесорів (Intel Core2 Duo CPU 2.53GHz). Результати тестування системи ПАРКС-.NET наведені в таблиці 1.

Таблиця 1

Час (с.) множення матриць для різної кількості потоків

Матриці Комп./потік	1000x1000	2000x2000	4000x4000
1/1	17.9	140.3	1095.1
1/2	10.1	74.6	569.8
2/4	6.1	41.1	308.1
4/8	3.6	26.3	173.2

Висновки

В даній роботі була представлена система для паралельних обчислень на комп'ютерній мережі ПАРКС-.NET. Було проведено тестування системи на класичній задачі множення матриць. При виконанні обчислень на комп'ютерній мережі із застосуванням 4-х двоядерних процесорів було досягнуто покращення в часі в 5-6 разів, при збільшенні розмірності обчислень маємо збільшення ефективності застосування системи ПАРКС-.NET, що на практиці дає досить непоганий результат.

Результати роботи можуть бути використаними при створенні різноманітних мережевих обчислювальних комплексів. На цьому робота над даною системою не завершується. Як шлях вдосконалення системи планується розробка підсистеми диспетчеризації та захисту від збоїв.

Список використаних джерел

1. Глушков В.М., Анисимов А.В. Управляющие пространства в асинхронных параллельных вычислениях. // Кибернетика. – 1980. – №5. – С. 1-9.
2. Анисимов А.В., Деревянченко А.В. Система ПАРУС-JAVA для параллельных вычислений на компьютерных сетях // Кибернетика и системный анализ. – 2005. – №1. – С.25-36.
3. Деревянченко О.В. Моделирование параллельных программ за допомогою системи ПАРКС-JAVA // Наукові записки НАУКМА, Комп'ютерні науки. – 2005. – С.47-58.
4. Анисимов А.В., Деревянченко О.В. Паралельне програмування із застосуванням технології ПАРКС-JAVA // К.: ТОВ «Компанія ВАІТЕ». – 2013. – 78с.
5. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C#. 3-е изд. // СПб.: Питер. – 2012. – 928 с.