

УДК 681.3.06

Дерев'янченко О.В.<sup>1</sup>, к.ф.-м.н., доцент.

### Системи паралельних обчислень на комп'ютерній мережі на базі технології ПАРКС

<sup>1</sup> Київський національний університет  
імені Тараса Шевченка, 83000, м. Київ, пр-т.  
Глушкова 4д,  
e-mail: alexanderder@mail.ru

O.V. Derevianchenko<sup>1</sup>, PhD.

### Systems of parallel computing in the computer network based on the PARCS- technology

<sup>1</sup> Taras Shevchenko National University of Kyiv,  
83000, Kyiv, Glushkova st., 4d,  
e-mail: alexanderder@mail.ru

*В роботі порівнюються дві системи паралельного програмування для комп'ютерних мереж із застосуванням технології ПАРКС (Паралельні Асинхронні Рекурсивні Керуючі Системи): ПАРКС-JAVA та ПАРКС-C#. Надано опис роботи цих систем. Наведено результати тестування систем для вирішення класичної задачі множення матриць. Розглянуто можливості цих систем при вирішенні практичних задач паралельного програмування.*

*Ключові слова: паралельні обчислення, комп'ютерна мережа, ПАРКС-технологія програмування, JAVA, C#.*

*This paper compares two parallel programming systems PARCS-JAVA and PARCS-C # for computer networks using the technology PARCS (Parallel Asynchronous Recursive Control System). The basic terms of the PARCS-technology are point, programming channel and algorithmic module. Pattern of control space is a graph. Top is a point of control space and edge is programming channels. The points are connected per programming channels. Each point of control space is assigned with the algorithmic module. The algorithmic module is a procedure PARCS and extension of the basic programming languages. Algorithmic modules are executed in parallel. In practice this means that in a computer network one major computer (Server) is selected. This computer is accessible to all others (Clients). It generates a queue of all tasks. Tasks from the queue are sent for computing. Each computer obtains a task from the queue and executes it. After finishing computing the active task it sends the results to the server and selects the new task. The model is designed for local area networks. Description of these systems is provided. The results of testing systems for solving the classical problem of matrix multiplication are presented. Capabilities of these systems in solving practical problems of parallel programming were considered.*

*Keywords: parallel computing, computer network, PARCS-technology of programming, JAVA, C#.*

Статтю представив д.ф.-м.н., проф. Анісімов А.В.

#### Вступ

Метою даної роботи було дослідження, аналіз і реалізація методів прискорення обчислень паралельних систем програмування базованих на технології ПАРКС (Паралельні Асинхронні Рекурсивні Керуючі Системи). ПАРКС-технологія програмування) [2] – сукупність програмних засобів, що забезпечують процес розробки і реалізації алгоритмів паралельної обробки інформації. Ця система може використовуватися при створенні програмного забезпечення паралельних ЕОМ, а також логічної структури паралельних і спеціалізованих ЕОМ, що забезпечують найбільш ефективну реалізацію

розробленого програмного забезпечення. В роботі порівнюються дві системи для паралельних обчислень на комп'ютерній мережі із застосуванням технологій ПАРКС, а само та ПАРКС-JAVA та ПАРКС-C#.

#### Технологія ПАРКС

ПАРКС-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 алгоритмічної мови. В нашому випадку ми застосовуємо базові мови програмування JAVA та C#.

Програмні засоби розширення базових алгоритмічних мов:

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.

Таким чином, програма на деякій умовній мові алгоритмічних модулів складається з наступних частин:

1. Опис структури чи типу керуючого простору.
2. Програма налаштування керуючого простору. При виконанні цієї програми відбувається розподілення алгоритмічних модулів по точкам простору.
3. Опис локальних алгоритмічних модулів та опис функціонування всього керуючого простору.

### Технологія .NET

Microsoft .NET – сучасна технологія, що дозволяє створювати сучасні програми із застосуванням комп'ютерних мереж та паралельних обчислень. Застосування технології .NET [5] надає можливість переносу програм між різними апаратними платформами, на яких встановлене середовище виконання .NET 4.0 (фактично віртуальна машина). Однією з переваг .NET є сумісність бібліотек, написаних різними мовами. Система ПАРКС-C# написана на мові C#, 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- C#, 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-C# і надає широкий набір можливостей для розробки паралельних програм, підтримка засобів мережевого програмування, системи забезпечення пересилання даних по комп'ютерній мережі, потоковий ввід/вивід, рефлексія – процес виявлення типів під час виконання та інші.

### Опис системи ПАРКС-C#

Архітектура системи складається з декількох частин:

1. *Parcs* – основна бібліотека класів системи. Містить всі основні класи, що будуть використовуватись усіма іншими частинами, в тому числі і алгоритмічними модулями.
2. *Daemon* – демон (клієнт). Являє собою програму, на якій будуть проводитись обчислення.
3. *HostServer* – сервер. Має список доступних клієнтів, а також інформацію про поточні задачі та точки. При запуску модуля починає роздавати завдання клієнтам в залежності від кількості процесорів клієнтів та їх завантаженості.
4. *Алгоритмічний модуль* – окремих проект, що має в собі посилання на *Parcs* та реалізує інтерфейс *IModule*.

### Опис системи ПАРКС-JAVA

Застосування мови програмування JAVA надає можливість переносу програм між різними апаратними платформами, на яких встановлена JavaVM – віртуальна машина.

До позитивних якостей мови програмування JAVA, належать:

- проста об'єктна модель, що дозволяє легко проводити зміни в коді програми, доповнювати її та документувати;
- детермінованість коду – властивість мови, що не дає програмісту робити помилки, наприклад, пов'язані з несанкціонованими діями, такими, як різні виклики;
- система програмування JAVA відповідає сучасним вимогам: підтримка засобів мережевого програмування (сокетів), потоковий ввід/вивід, графічний інтерфейс користувача, системи забезпечення пересилки даних по комп'ютерній мережі, технології, що дозволяють уніфікувати клієнт-серверний (RMI) та інші.

**Архітектура системи** складається з наступних частин [3]:

- 1) *Parcs* – основна бібліотека класів системи, що використовуються всіма компонентами системи, в тому числі і алгоритмічними модулями;
- 2) *Daemon* – програма запускається в фоновому режимі на машинах, які підключаються до обчислюваної мережі (хости). Вона виконує запуск алгоритмічних модулів обчислювальних задач, а також тестування продуктивності машин.
- 3) *HostsServer* – сервер, що є централізованою службою для управління та обліку задач та точок. Сервер зберігає інформацію про підключені машини, продуктивність та кількість точок на кожному хості, і згідно цієї інформації виділяє машини для створення нових точок, а також відмічає видалення точок, початок та завершення задач.
- 4) Обчислювальна задача, що поставляється у вигляді одного чи декількох jar-файлів, або окремих класів алгоритмічних модулів. На обчислення запускається початковий алгоритмічний модуль задачі. В процесі виконання початкового АМ він завантажує з фізичного диску програмний код інших алгоритмічних модулів.
- 5) RMI-сервер *ParcsLauncher* використовується для віддаленого запуску програми на виконання. RMI-клієнт для запуску програми повинен передати код обчислювальної задачі, запакований в jar-архів, та може вказати чи потрібно чекати кінець обчислення перш, ніж повернути управління клієнту.

### Робота систем ПАРКС

Робота обох систем ПАРКС-С# та ПАРКС-JAVA схожа в цілому, але відрізняється деякими властивостями реалізації відповідних мов програмування С# та JAVA.

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* (ПАРКС-С#) використовується початковим АМ для знаходження комп'ютеру, на якому запущений *HostServer* (IP-адреса).

Сімейство протоколів TCP/IP було обрано за основу для передачі інформації та служить для об'єднання комп'ютерів у локальні мережі та мережі Internet.

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

1. При запуску модуля сервер перевіряє всі зазначені у файлі *hosts.txt* (ПАРКС-С#) 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 exe-файл (ПАРКС-С#) або jar-файл (ПАРКС-JAVA), в якому міститься інформація про всі АМ, що будуть запущені під час виконання поточної задачі. За допомогою рефлексії *Daemon* (ПАРКС-С#) створює екземпляр класу АМ і викликає метод *Run()*. Таким чином запускаються необхідні обчислення.
4. У файлі даних (одному або декількох) знаходяться вхідні дані для виконання обчислень, результати записуються в файл(и) результату. При створенні нової точки алгоритмічні модулі звертаються до сервера, щоб отримати адресу машини (найменш завантаженої) для запуску нового АМ. Список доступних машин зберігається в файлі *hosts.list* (ПАРКС-JAVA) та *hosts.txt* (ПАРКС-С#).
5. 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.

Таблиця 1

Час множення матриць для різної кількості потоків для систем ПАРКС-JAVA та ПАРКС- С#.

Матриці Комп./поток	1000x1000		2000x2000		4000x4000	
	ПАРКС-С#	ПАРКС-JAVA	ПАРКС-С#	ПАРКС-JAVA	ПАРКС-С#	ПАРКС-JAVA
1/1	17.9	18.2	140.3	129.3	1095.1	1210.4
1/2	10.1	11.4	74.6	62.6	569.8	624.2
2/4	6.1	7.4	41.1	39.7	308.1	317.7
4/8	3.8	4.5	26.3	24.5	173.2	185.1

### Тестування систем ПАРКС

Для тестування роботи системи був реалізований паралельний алгоритм множення матриць, описаний за допомогою керуючого простору [4]. Обчислення проводились на комп'ютерній мережі із застосуванням 4-х двоядерних процесорів (Intel Core2 Duo CPU 2.53GHz). Результати тестування систем ПАРКС-JAVA та ПАРКС-C# наведені в Таблиці 1. Час тестування заданий в с.

### Висновки

В даній роботі було порівняно створені на факультеті кібернетики Київського національного університету системи для паралельних обчислень на комп'ютерній мережі ПАРКС-JAVA та ПАРКС-C#. Було проведено тестування систем на класичній задачі паралельного множення матриць. При виконанні обчислень на комп'ютерній мережі із застосуванням 4-х двоядерних процесорів було досягнуто покращення в часі більше ніж в 5 разів, для обох систем, що є досить непоганим результатом.

З результатів тестування видно, що системи ПАРКС-JAVA та ПАРКС-C# мають для тестової задачі множення матриць приблизно однакову ефективність розпаралелювання, але час виконання обчислень при збільшенні розмірності матриць для системи ПАРКС-JAVA зростає, що пов'язано з використанням пам'яті для JavaVM. Таким чином використання системи ПАРКС-C# може бути більш доцільним при застосуванні паралельних обчислень для задач з великими об'ємами даних.

Результати роботи можуть бути використаними при створенні різноманітних мережевих обчислювальних комплексів. Реалізація даної системи може застосовуватися для розв'язування широкого класу задач, що потребують паралельної обробки.

На цьому роботі над вдосконаленням систем не завершується. Планується робота над розробкою 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., Кулябко П.П. Особенности ПАРУС-технологии. – 1993. – №3. – С.128-137.
3. Анисимов А.В., Деревянченко А.В. Система ПАРУС-JAVA для параллельных вычислений на компьютерных сетях // Кибернетика и системный анализ. – 2005. – №1. – С.25-36
4. Деревянченко О.В. Моделирование параллельных программ за допомогою системи ПАРКС-JAVA // Наукові записки НаУКМА, Комп'ютерні науки. – 2005. – С.47-58.
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.

### References

1. GLUSHKOV V.M., ANISIMOV A.V. (1980), Controlling spaces in asynchronous parallel computations, *Kybernetika* #5, pp. 1-9.
2. ANISIMOV A.V., KULYABKO P.P. (1993), Features PARCS-technology, *Cybernetics and Systems Analysis* # 3, pp. 128-137.
3. ANISIMOV A.V., DEREVIANCHENKO A.V. (2005), The System PARCS-JAVA for Parallel Computations on Computer Networks, *Cybernetics and Systems Analysis* # 1, pp. 25-36.
4. DEREVIANCHENKO O.V. (2005) Simulation of parallel programs using the system PARCS-JAVA, *Scientific notes Kyiv-Mohyla Academy, Computer Science*, pp.47 -58.
5. RICHTER J. (2010), CLR via C#, *Third Edition* , Microsoft Press, 896 p.

Надійшла до редколегії 17.04.14