

УДК 681.301

**К.т.н, доцент Тесленко О. К., магістрант Белінський Д. І.,
аспірант Яновська О.Ю.**

**Національний технічний університет України
«Київський політехнічний інститут»**

МЕТОД GRID-ОРІЄНТОВАНИХ РЕКОНФІГУРОВНИХ ОБЧИСЛЕНЬ

Abstract

*Oleksandr C. Teslenko, assoc. prof., PhD; Dmytro Belinskiy, student;
Olena Yanovska postgraduate*

Grid-oriented method of reconfigurable computing

This paper concerns the idea for creating Grid oriented method of reconfigurable computing. An idea of distributed computing systems using IRL technology and computing capabilities of FPGA was introduced. It was also suggested that one or more substitutions can be the most effective for certain types of data.

Вступ

Сьогодні існує велика кількість різноманітних задач, що потребують значних комп'ютерних ресурсів для свого вирішення. Серед них є задачі, які досить довго та не ефективно вирішуються навіть на суперкомп'ютерах та у середовищі розподілених обчислень Grid. Підвищити швидкість вирішення таких задач може спеціально створений процесор, але на його створення буде витрачено багато часу та коштів. Завдяки своїй гнучкій архітектурі плати ПЛІС дають можливість створити на своїй базі обчислювальний модуль з будь якою архітектурою, який зможе ефективно вирішувати поставлені задачі. Таким чином, плати ПЛІС дозволяють звичайним користувачам ефективно вирішувати свої задачі.

В патенті [1] компанії Майкрософт запропонована система та спосіб для виявлення та опису обчислювальних ресурсів, які знаходяться у різних обчислювальних пристроях, пов'язаних ідентичністю, а також для розкриття цих ресурсів як служб, адресованих програмами, так, що програми в пристроях можуть використовувати ресурси іншого пристрою для підвищення ефективності обробки інформації.

Запатентована технологія компанії Xilinx IRL дозволяє дистанційно налаштовувати плати ПЛІС для подальшої роботи.

Вказані технології дозволяють створювати розподілені системи, в яких існує можливість дистанційного налаштування специфічного обладнання з метою проведення на ньому спеціалізованих обчислень.

Можливість створення подібної системи залежить від стану розвитку Grid середовища та існуючої програмної та процесуальної бази.

Постановка задачі

Метою даної роботи є розробка методу та дослідження можливості створення Grid орієнтованої системи реконфігурованих обчислень на прикладі задачі підвищення ефективності ущільнення великих об'ємів даних різноманітних форматів.

Система Grid орієнтованих реконфігурованих обчислень

Пропонується розширення для системи та способу [1] виявлення та опису обчислювальних ресурсів, які знаходяться у різних обчислювальних пристроях, а саме можливістю користувача самостійно конфігурувати віддалені ресурси, якщо ці ресурси підтримують таку модель. Це дозволить конфігурувати віддалені ресурси під свою конкретну задачу, або проект, що в свою чергу відобразиться на швидкості вирішення поставлених задач.

Обчислювальні ресурси, що дозволяють змінювати свої обчислювальні можливості, можуть представляти собою, наприклад, програмовані логічні інтегральні схеми ПЛІС, логіка роботи яких, на відміну звичайних цифрових мікросхем, не визначається при виготовленні, а задається користувачем.

Враховуючи стан розвитку Grid-інфраструктури в Україні, створення та впровадження подібної системи на базі інфраструктури Grid на сьогоднішній день неможливо, оскільки в мережі Grid використовуються лише кластери. Але вже почалися розробки в напрямку підключення до Grid окремих комп'ютерів користувачів. Таким чином буде створена мережа розподілених обчислень, в якій користувач може запропонувати власні ресурси, пам'ять, обчислювальну потужність, периферійні пристрої, плати ПЛІС, тощо. Подібна система дозволить не лише надавати доступ, але і створювати віддалені ресурси для вирішення своїх задач.

Отже, створення системи розподілених спеціалізованих обчислень з використанням IRL-технології та обчислювальних можливостей плат ПЛІС справа недалекого майбутнього. На базі подібної системи з'явиться можливість вирішувати широкий спектр складних задач. Одна з можливих задач - це задача ущільнення великих обсягів різноманітних даних. Суттю

цієї задачі є досягнення найбільшого ефекту ущільнення в порівнянні з існуючими засобами шляхом використання можливостей гнучкої архітектури плат ПЛІС. Як приклад, може використовуватись попередня обробка даних за допомогою апаратної реалізації підстановок довільної розрядності. Припускається, що для однакових типів даних найефективнішою може бути одна підстанова, або декілька з певного набору підстановок.

Оскільки на сьогодні не має можливості впровадити та дослідити подібну систему на практиці, пропонується дослідити на кластері припущення, що одна або декілька, підстановок можуть бути найефективнішими для певних типів даних.

Моделювання на кластері проблеми удосконалення методів ущільнення даних

Підстановки необмеженої розрядності реалізуються на ПЛІС шляхом використання одновимірних каскадів конструктивних модулів (ОККМ) [2,3].

Загальна ідея ущільнення файлу, полягає у поетапному перетворенні файлу за допомогою багаторозрядної підстановки, з подальшим ущільненням будь якими відомими способом початкового файлу та файлу після підстановки. Таким чином, можна виконати декілька ітерації підстановок та порівняти отримані результати ущільнення.

Детальніше процес ущільнення виглядає наступним чином. Беруть деякий файл A , довжиною $\#A$ байт, та за допомогою багаторозрядної підстановки перетворюють в файл $B=P(A)$. Файли A та B ущільнюються за допомогою будь-якого архіватора, наприклад RAR, таким чином, що нові файли: $A_{rar}=RAR(A)$, $B_{rar}=RAR(B)$, та визначають їх довжини $\#A_{rar}$ та $\#B_{rar}$. Далі файли A_{rar} та B_{rar} за допомогою багаторозрядної підстановки P перетворюють у файли $C=P(A_{rar})$ та $D=P(B_{rar})$. Файли C та D ущільнюються за допомогою будь якого архіватора, наприклад RAR і тоді отримують нові файли: $C_{rar}=RAR(C)$ та $D_{rar}=RAR(D)$.

Шляхом перебору всіх можливих підстановок, які реалізуються на регулярних простих ОККМ, визначають ті, для яких $\#B_{rar} \ll \#A_{rar}$ або $\#C_{rar} \ll \#A_{rar}$ або $\#D_{rar} \ll \#A_{rar}$, та кращі значення зберігають для подальшого аналізу.

Задача виявлення ефективних багаторозрядних підстановки легко піддається розпаралелюванню при моделюванні ОККМ на комп'ютері. Враховуючи те, що при моделюванні виконується перебір по всіх можливих структурах конструктивних модулів [4], доцільно буде використати розпаралелювання по елементах таких структур. Під час

паралельних обчислень важливо чітко синхронізувати роботу всіх циклів, та коректне зберігання отриманих результатів.

Отже, виконуючи подібну задачу на кластері, який знаходиться в мережі Grid, практично моделюємо роботу в Grid орієнтованої системи реконфігурованих обчислень. Дані, які будуть отримані в результаті дослідження, можуть довести припущення щодо ефективності певних підстановок на різних типах даних.

Висновок

Враховуючи потужний потенціал методу Grid-орієнтованих реконфігурованих обчислень необхідно, не чекаючи завершення реалізації низки технічних та організаційних проблем, проводити дослідження та підготовлювати необхідні дані для його використання з метою підвищення ефективності спеціалізованих обчислень. Доцільно, при цьому, використовувати технологію кластерів для моделювання майбутніх застосувань.

Література

1. Заявка: №2003119524/09, від 27.06.2003. Аллес Девіс (US), Мур Джордж (US) – Розподілені обчислення на основі ідентичності для ресурсів пристроїв.
2. *Тарасенко В.П. , Тесленко О.К. Яновська О. Ю.* Реалізація повних підстановок за допомогою багатомодульного каскаду найпростіших конструктивних модулів // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні, випуск 2 (17), 2008.
3. *В.П.Тарасенко, О.К.Тесленко, О.Ю.Яновська.* Властивості повних підстановок, які реалізуються найпростішим однонаправленим регулярним ОККМ // Радіоелектронні і комп'ютерні системи. №6, 2010 с.123-128
4. *В.П. Тарасенко, О.К. Тесленко, О.Ю. Яновська.* Реалізація повних підстановок на простому двомодульному каскаді конструктивних модулів. Інформаційні технології та комп'ютерна інженерія.№1(11)2008, с.88-97