

ОРГАНІЗАЦІЯ ХМАРНИХ ОБЧИСЛЕНЬ НА БАЗІ МАСИВУ ПРОГРАМОВАНИХ КОМІРОК ЛОГІКИ

© Мельник А., Козак Н., 2010

Запропоновано нову хмарну модель обчислень – ПЛІС як послуга, що покликана забезпечити масове використання ПЛІС для організації високопродуктивних обчислень.

Ключові слова: хмарні обчислення, ПЛІС, HDL-код, ПЛІС як послуга.

New cloud computing model – FPGA as a Service is offered, that called to provide more mass use of FPGA for high-performance computing.

Keywords: Cloud computing, FPGA, HDL-code, FPGA as a Service.

Вступ

Процес створення високопродуктивних рішень на базі програмованих логічних інтегральних схем (ПЛІС) гальмується великою вартістю цих кристалів. Наприклад, передові кристали лідера ринку ПЛІС компанії XILINX коштують по кілька тисяч умовних одиниць. А оскільки засоби програмованої логіки є дорогими, створити на їх основі засоби організації реконфігурованих обчислень можна лише у великих обчислювальних центрах із значним фінансуванням. Тому гнучкість таких рішень є надзвичайно низькою, оскільки користувачі, які не мають доступу до обчислювального центру, не можуть використати це обладнання. А тому і не можуть оцінити ефективність таких рішень та ризикнути закупити таке обладнання для своїх потреб. Відповідно і зацікавленість в розгортанні систем на базі програмованої логіки зменшується. Як наслідок, через малий попит зростає і ціна на ПЛІС, оскільки виробник закладає в ціну більшу частку постійних затрат на кожний кристал. Утворюється своєрідне замкнуте коло: кристали дорогі, бо їх ніхто не купує, а ніхто не купує тому, що вони надто дорогі.

Виходом з ситуації, що склалася, могло б бути більш ефективне використання засобів програмованої логіки. Але як цього досягти?

Аналізуючи останні тенденції розвитку комп'ютерної техніки, можна побачити підтвердження відомого правила, що розвиток технологій йде по спіралі. Подібно до того, як колись через велику вартість обладнання комп'ютери були багатотермінальними, так і тепер через стрімкий розвиток комп'ютерного обладнання, внаслідок чого виникає потреба в постійному його оновленні, і внаслідок розвитку всесвітньої комп'ютерної мережі і зростання швидкості, з якою користувач може доступатися до ресурсів мережі, з'явилася можливість і потреба у використанні віддаленого обчислювального центру. Більше того, з'явилася технологія grid computing, що дає змогу використовувати паралельно цілий ряд обчислювальних ресурсів. Ця технологія передбачає можливість налагодження спільної роботи ряду обчислювальних засобів за схемою, розробленою користувачем цієї технології. Та користувача загалом не має цікавити питання, як організований той чи інший обчислювальний процес – користувача має цікавити лише готовий засіб реалізації його задач, незважаючи на те, як саме організовані засоби, що виконують його задачу. Ім'я технології, яка забезпечує реалізацію цього принципу – cloud computing (хмарні обчислення).

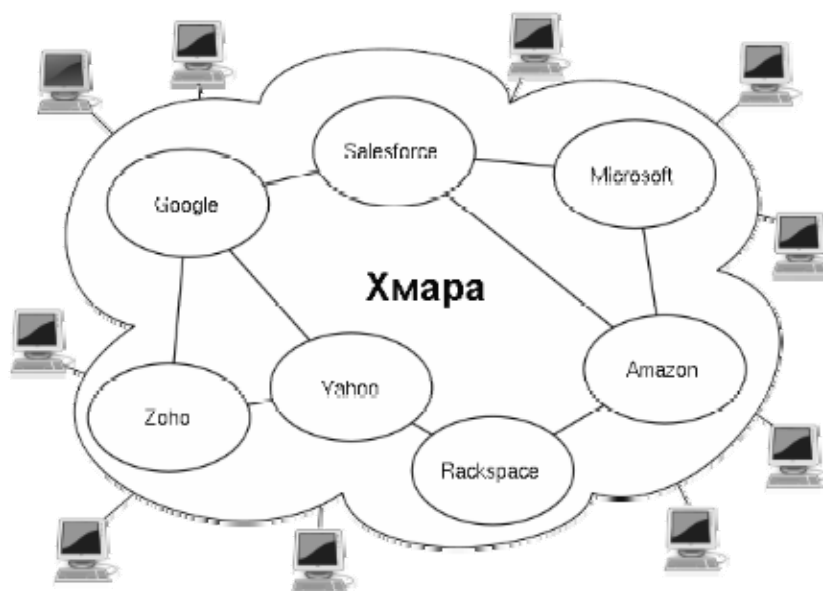


Рис. 1. Хмарні обчислення

Хмарні обчислення

Відповідно до опублікованого в 2008 році доку IEEE, «Хмарна обробка даних – це парадигма, в рамках якої інформація постійно зберігається на серверах в Інтернеті і тимчасово кешується на клієнтській стороні, наприклад, на персональних комп'ютерах, ігрових приставках, ноутбуках, смартфонах і т. д.»[1].

Хмарні обчислення, попри певні недоліки, що насамперед пов'язані з питанням приватності даних користувача, які зберігаються на віддаленому сервері, надають ряд великих переваг. По-перше, для налагодженого таким чином обчислення знижуються вимоги до обчислювальної потужності персонального комп'ютера, обов'язковою умовою є тільки наявність доступу до Інтернету. З іншого боку, це забезпечує зниження витрат на апаратне і програмне забезпечення, на обслуговування і електроенергію, а також економію дискового простору, оскільки і дані, і програми зберігаються в Інтернеті. По-друге, такий підхід дає більшу можливість для швидшої обробки даних, оскільки не вимагає при цьому модернізації клієнтської частини. По-третє, хмарні обчислення характеризуються великою надійністю, бо відмова одного із серверів, що надає послуги, не веде до краху системи.

При хмарних обчисленнях програми запускаються і видають результати роботи у вікно стандартного веб-браузера на локальному комп'ютері, при цьому, як вже було сказано, все програмне забезпечення і дані, необхідні для роботи, знаходяться на віддаленому сервері в Інтернеті.

Категорії хмарних обчислень

Існує декілька умовних категорій Cloud Computing [2]. Перша – Software as a Service (SaaS) – об'єднує закінчені, що не допускають кастомізації, продукти, наприклад, поштові служби або портали. До другої групи, іменованої Platform as a Service (PaaS), входять хмарні сервіси для розробників, які, за великим рахунком, є масштабованими і автоматично керованими хостинг-ресурсами. Яскравими прикладами таких платформ є Google App Engine і Windows Azure. Ще одна категорія – інфраструктура як сервіс (Infrastructure as a Service, IaaS), що є хостингом віртуальних машин. Останнім часом позначилася тенденція до організації ще однієї моделі Everything as a Service, але її впровадження має винятково маркетинговий характер, оскільки навіть вищезазвані три види сервісів знаходяться на етапі розвитку.

ПЛІС як послуга

Для вирішення вищеописаної проблеми недоступності програмованої логіки можна також використати принцип хмарної моделі і використовувати наявні у одного абонента мережі Інтернет-ресурси програмованої логіки як послугу для іншого абонента. Застосування цієї концепції не лише

надасть ресурс програмованої логіки тому, хто його не має, але і дасть змогу амортизувати витрати від надання послуги тому, хто нею володіє. Тому реконфігуровні обчислення подешевшають внаслідок оптимальнішого їх використання. Отож, в додаток до вже існуючих хмарних моделей пропонується нова модель – ПЛІС як послуга, FaaS (FPGA as a Service).

Спочатку роботу з виділення апаратного ресурсу можна буде здійснити на якомусь одному сервері. Пізніше один HDL-запит можна буде виконувати на ряді розподілених серверних локалізацій у всьому світі.

Базовою одиницею обладнання запропонованої системи буде звичайна мережа плат з інтерфейсом Ethernet, на яких розміщуватимуться кристали ПЛІС. Також буде потрібна як мінімум одна серверна станція на кожну локалізацію базового обладнання для виконання системних функцій. Відповідно, кожна така структурна одиниця матиме порти для зв'язку з локальною мережею обчислювального центру та порти для під'єднання до мережі Інтернет.

Клієнтській стороні буде потрібне відповідне програмне забезпечення для користування цим хмарним сервісом, виконане у вигляді web-додатка і завантажуване з відповідного сервера. Іншим варіантом є інтеграція таких функцій в хмарну операційну систему, однією з яких стане Google Chrom OS, перший реліз якої заплановано на кінець цього року. Недавно було продемонстровано і перший хмарний нетбук Litl Webbook, який працюватиме на подібних операційних системах.

З погляду обміну даними між частинами організованого хмарним методом обчислювача існуватимуть три рівні організації його роботи.

В останні роки швидкість передачі даних через всесвітню павутину виросла в рази [3,4], і навіть мережі мобільного доступу до Інтернет вже досягають швидкодії до 14,4 МБ/с, а в недалекому майбутньому обмін даними стане ще швидшим [5]. Інша справа – латентність, оскільки навіть якщо в найближчому майбутньому і будуть розроблені маршрутизатори і супутне обладнання з дуже малою латентністю, то вже саме поширення сигналу, швидкість якого хоча і значна, але скінченна, у масштабах планети вносить значну затримку. Тому швидше за все в найближчому майбутньому цю проблему вирішити не вдасться, і *на верхньому рівні* обмін даними між великими сегментами пристрою відбуватиметься з великою затримкою, оскільки реально це будуть географічно віддалені сервери. Тому розробник HDL-опису повинен добре сегментувати своє рішення – створити його з якомога більшої кількості паралельних частин, які не потребуватимуть швидкого обміну інформацією між собою. Потрібно зазначити, що навіть якщо і не вдасться організувати сегментування, то така система все одно матиме принаймні два сегменти, оскільки клієнт, який користується послугою через Інтернет, фактично буде сегментом вводу/виводу. І саме клієнтський інтерфейс виконуватиме первинну обробку даних.

Верхній рівень і надаватиме проектованій системі властивостей хмарної.

Середній рівень буде мережею плат, на яких розміщуватимуться кристали програмованої логіки, яка географічно сконцентрується в межах одного обчислювального центру і функціонуватиме під його керуванням. На цьому рівні обмін даними відбуватиметься або за допомогою комуруючої мережі, яка організована на одному з кристалів, у межах друкованої плати, або за допомогою локальної мережі для організації відповідного кластеру. На цьому рівні організація обчислювального процесу відбувається аналогічно організації роботи багатопроекторної системи.

Нижній рівень. На цьому рівні можна буде реалізовувати весь спектр архітектур, оскільки система виконуватиметься в межах окремо взятого кристала.

Нижній рівень надасть проектованій системі властивості систем із конфігурованою архітектурою.

Процес обробки клієнтського HDL-коду

Відповідно до описаних вище рівнів організації системи процес імплементації клієнтського запиту відбуватиметься по чергово для кожного рівня (рис. 2). Під час кожного етапу спочатку відбуватиметься аналіз доступності ресурсів відповідного рівня, після чого проходитиме декомпозиція HDL-коду, а на останньому етапі – його відображення у конфігураційний файл відповідного кристала.

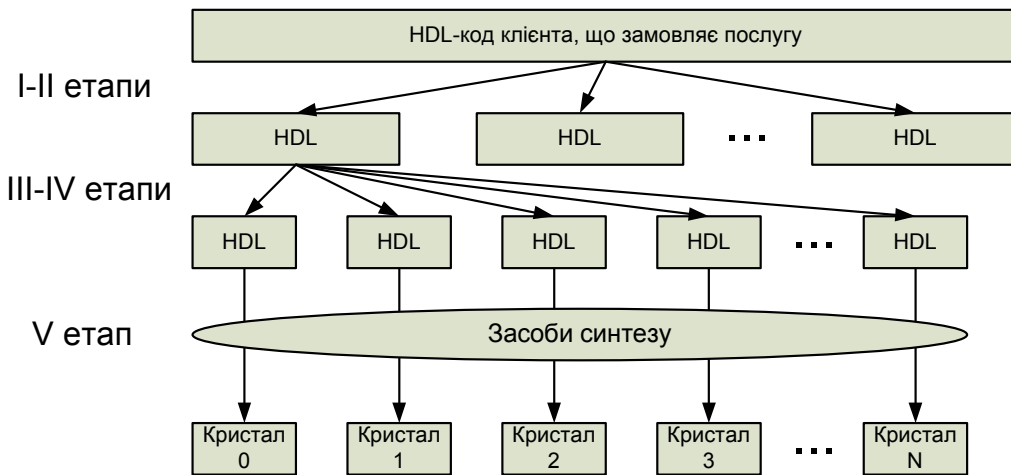


Рис. 2. Імплементція HDL-коду клієнта

Отож, на першому етапі має бути виконаний аналіз доступності вільного обладнання з Інтернету із створенням відповідного графу, в якому кожен вузол – це обчислювальний центр, що має в наявності вільні ресурси програмованої логіки. Вага кожного вузла вказуватиме на кількість вільних еквівалентних логічних вентилів для конфігурації. Вітки графу, що зв’язують вершини, вказуватимуть на латентність і швидкість, з якою інформація може передаватися між відповідними обчислювальними центрами.

На другому етапі виконуватиметься декомпозиція HDL-коду відповідно до отриманого графу.

Третій етап передбачає виконання аналізу вільного сегмента програмованої логіки обраних обчислювальних центрів. На базі цього аналізу можна буде сформувати еквівалентний граф вільних ресурсів кристалів для кожного обчислювального центру.

Четвертий етап – декомпозиція HDL-коду відповідно до отриманого графу вільних кристалів.

Після цього на п'ятому етапі можна буде отримати кінцевий набір HDL-файлів для генерації прошивок, які разом з додатковими файлами налаштування системи стануть кінцевим продуктом її імплементції.

Що ж до процесу генерації самих прошивок, то для цільових кристалів він відбуватиметься відповідно до технології обраної фірми виробника кристалів програмованої логіки. Наприклад, для кристалів XILINX він може відбуватися за допомогою відповідних консольних утиліт [6,7].

Можливість використання ресурсів кристалів програмованої логіки з різною архітектурою не є суперечливою, адже файл для окремо взятого кристала імплементують спеціально розробленим для цього синтезатором. З одного боку, це якраз і підкреслюється універсальністю в силу абстрактності мов опису апаратних засобів (HDL – Hardware Description Language). З іншого боку, це буде також і ознакою хмарної технології, одним із принципів якої є прозорість стосовно того, “як це реалізовано”.

Висновок

Запропонована нова модель хмарного обчислення – ПЛІС як послуга, яка вирішує проблему недоступності великих кристалів ПЛІС в силу їх великої вартості. Запропоновані алгоритми планування роботи цієї системи можна розвинути для кращої організації обчислювального процесу, а запропонований сервіс можна розширити іншими технологіями.

Потрібно також зазначити, що оскільки індустрія хмарних обчислень зараз стрімко розвивається і, за прогнозами аналітиків, до 2012 року на її частку припадатиме 9% всіх витрат на ІТ, то завдяки цій технології реконфігуровані обчислення можуть вийти на новий рівень свого розвитку і в майбутньому стати дуже поширеними.

1. Hewitt C. ORGs for Scalable, Robust, Privacy-Friendly Client Cloud Computing // Massachusetts Institute of Technology (vol. 12 no. 5) pp. 96-99, September/October 2008 2. Francesco Maria Aymerich, Gianni Fenu, Simone Surcis, A real time financial system based on grid and cloud computing //

Proceedings of the 2009 ACM symposium on Applied Computing, March 08-12, 2009. 3. Thrasyvoulos Spyropoulos, Serge Fdida, Scott Kirkpatrick, *Future internet: fundamentals and measurement // SIGCOMM Computer Communication Review, March 2007.* 4. S. Zhou. *Understanding the Evolution Dynamics of Internet Topology // Physical Review E, vol. 74, 2006.* 5. Kyung-Ho Kim, *Key technologies for the next generation wireless communications // CODES+ISSS '06: Proceedings of the 4th international conference on Hardware/software codesign and system synthesis, October 2006.* 6. Xilinx, Inc., *The Development System Reference Guide // www.xilinx.com/support/documentation, 2009.* 7. Xilinx, Inc., *The Programmable Logic Data Book // www.xilinx.com/support/documentation, 2009.*

УДК 531.36+534

М. Назаркевич, А. Шпекторова
Національний університет “Львівська політехніка”,
кафедра автоматизованих систем управління

РОЗРОБЛЕННЯ КАЛІГРАФІЧНОГО ШРИФТУ ДЛЯ УКРАЇНСЬКОЇ АБЕТКИ

© Назаркевич М., Шпекторова А., 2010

Розроблено рукописний шрифт, що належить до групи декоративних шрифтів. Проведено класифікацію комп'ютерних шрифтів. Написано абетку кирилиці: рядкові і прописні літери, знаки пунктуації, цифри, які потім відскановано та обмальовано у векторному редакторі CorelDraw. Їх вміщено у редактор шрифтів Fontlab. Побудовано альтернативні символи та показано можливість їх підключення.

Ключові слова: шрифт, абетка, кирилиця, символ

A cursive is developed, that belongs to the group of decorative fonts. Classification of computer fonts is conducted. The alphabet of cyrillic is written: capital letters small and, and signs of punctuation, numbers which scan then, and draw along contour in the vectorial editor CorelDraw. They are placed in the editor of the Fontlab fonts. Alternative characters are built and possibility of their connecting is shown.

Keywords: font, alphabet, Cyrillic, symbol

1. Вступ

З виникненням і розвитком цифрових технологій у додрукарській підготовці у розробленні шрифтів відбулися значні зміни. По-перше, відпала необхідність трудомісткої операції втілення шрифту в металі. Сучасні шрифти існують у цифровому вигляді і не потребують матеріальних носіїв. А по-друге, кількість шрифтів різко збільшилася. З одного боку, це розширило можливості для розроблення, а з іншого – виникло багато неякісних шрифтів, а часто їх великий вибір часто перешкоджає роботі і нерідко приводить до браку. Отже, шрифтом називають набір друкованих чи текстових символів певного стилю (наприклад, жирний або курсив) і певного розміру, який має певне зображення [1].

Ототожнюючи різні види класифікацій шрифтів, можна сказати, що сучасні шрифти можна поділити на чотири основні групи [2] – рублені шрифти (ще їх називають Sans Serif, що в перекладі означає “без засічок”), шрифти з засічками, декоративні шрифти і спеціальні символні шрифти, які замість букв містять символи і картинки.

Зокрема до групи шрифтів з засічками належать:

➤ Ренесансна (стара) антиква. Це група малоконтрастних шрифтів з похиленим положенням пера (25–45о), які мають співвідношення основних і з'єднуючих штрихів 1:2 ... 1:4. До цієї групи