УДК 004

Л.А. Шувалова

Черкаський державний технологічний університет, Черкаси

ГІПЕРПОТОКОВА ОБРОБКА КОМАНД

В статті проведене дослідження технології гіперпотокової обробки команд, розробленої фірмою Intel. Технологія повинна забезпечувати максимальне завантаження функціональних блоків суперскалярних багатоконвеєрних процесорів, що сприятиме їх максимальній продуктивності та швидкодії. В статті приведені особливості реалізації гіперпотокової технології в процесорі Intel Xeon. Досліджено три підходи, які застосовуються для розподілу ресурсів фізичного процесора на кілька логічних процесорів.

***Ключові слова**: мікропроцесор, гіперпотокова обробка, суперскалярний, багатоконвеєрний, процесор, потік.*

Вступ

Постановка проблеми. Сучасні процесори в більшості своїй є суперскалярні і багатоконвеєрні, тобто виконання команд у них іде паралельно, етапами і на кількох конвеєрах відразу. Тому однією з основних проблем є розкриття всього потенціалу процесорів, що дозволить максимально завантажувати їхні функціональні блоки для досягнення максимальної продуктивності та швидкодії. Удосконалення гіперпотокової технології покликане вирішити цю проблему.

Аналіз джерел досліджень. Розв'язання вказаної проблеми започатковано в [1]. У розділі «Вычислительные системы с управлением вычислениями от потока данных» наведений опис різних архітектур потокових обчислювальних систем, у тому числі і гіперпотокової обробки. Детально пояснюються причини виникнення, а також принцип роботи технології.

Постановка задачі. Головна мета статті – дослідження технології гіперпотокової обробки команд, розробленої фірмою Intel і вперше реалізованої в мікропроцесорі Pentium 4.

Дослідження технології

Під час гіперпотокової обробки виконувана програма розбивається на два паралельних потоки (threads). Завдання компілятора (на стадії підготовки програми) та операційної системи (на етапі виконання програми) полягає у формуванні таких послідовностей незалежних команд, які процесор міг би обробляти паралельно, по можливості заповнюючи функціональні блоки, не зайняті одним з потоків, придатними командами з іншого, незалежного потоку [1].

Операційна система, що підтримує гіперпотокову технологію, сприймає фізичний суперскалярний процесор як два логічних процесора і організовує надходження на ці два процесори двох незалежних потоків команд.

Процесор із підтримкою технології hyper-threading емулює роботу двох однакових логічних процесорів, приймаючи команди, спрямовані для кожного з них. Це не означає, що в процесорі є два обчислювальних ядра – обидва логічних процесора конкурують за ресурси єдиного обчислювального ядра.

Наслідком конкуренції є більш ефективне завантаження всіх ресурсів процесора.

У процесі обчислень фізичний процесор розглядає обидва потоки команд і по черзі запускає на виконання команди то з одного, то з іншого, або відразу з двох, якщо є вільні обчислювальні ресурси. Жоден з потоків не вважається пріоритетним. При зупинці одного з потоків (в очікуванні якоїнебудь події або в результаті зациклення) процесор повністю переключається на другий потік. Можливість чергування команд з різних потоків становить принципову відмінність між гіперпотоковою і макропотоковою обробкою.

Наявність тільки одного обчислювального ядра не дозволяє досягти подвоєння продуктивності, проте за рахунок більшої віддачі від усіх внутрішніх ресурсів загальна швидкість обчислень істотно зростає. Це особливо відчувається, коли потоки містять команди різних типів, тоді уповільнення обробки в одному з них компенсується великим обсягом робіт, виконаних в іншому потоці.

Слід враховувати, що ефективність технології hyperthreading залежить від роботи операційної системи, оскільки поділ команд на потоки здійснює саме вона.

Для ілюстрації розглянемо деякі особливості реалізації гіперпотокової технології в процесорі Intel Xeon. Процесор здатний обробляти паралельно два потоки в двох логічних процесорах. Щоб виглядати для операційної системи і користувача як два логічних процесора, фізичний процесор повинен підтримувати інформацію одночасно для двох окремих та незалежних потоків, розподіляючи між ними свої ресурси. Залежно від виду ресурсу засто-

совуються три підходи: дублювання, поділ і спільне використання [1].

Дубльовані ресурси. Для підтримки двох цілком незалежних контекстів на кожному з логічних процесорів деякі ресурси процесора необхідно дублювати. Перш за все, це відноситься до лічильника команд (IP, Instruction Pointer), що дозволяє кожному з логічних процесорів відслідковувати адресу наступної команди потоку. Для паралельного виконання декількох процесів необхідно стільки IP, скільки потоків команд необхідно відстежувати одночасно. Іншими словами, у кожного логічного процесора повинен бути свій лічильник команд. У процесорі Xeon максимальна кількість потоків команд дорівнює двом і тому потрібно два лічильника команд. Крім того, в процесорі є дві таблиці розподілу регістрів (RAT, Register Allocation Table), кожна з яких забезпечує відображення восьми регістрів загального призначення (РЗП) і восьми регістрів з плаваючою крапкою (РПК), що відносяться до одного логічного процесора, на спільно використовуваний регістровий файл з 128 РЗП і 128 РПК. Таким чином, RAT - це дубльований ресурс, керуючий спільно використовуваним ресурсом (реєстровим файлом).

Розділені ресурси. В якості одного з видів розділених ресурсів у Xeon виступають черги (буферна пам'ять, організована за принципом FIFO), розташовані між основними ступенями конвеєра. Застосовуваний тут розподіл ресурсів можна умовно назвати статичним: кожна буферна пам'ять (черга) розбивається навпіл, і за кожним логічним процесором закріплюється своя половина черги.

Стосовно іншого виду черг – черг диспетчеризації команд (їх у процесорі три) – можна говорити про динамічний поділ. Замість того, щоб з передбачених у кожній черзі дванадцяти входів фіксовано призначити входи 0-5 логічному процесору (ЛП) 0, а входи 6-11 - логічному процесору 1, кожному ЛП дозволяється використовувати будь-які входи черги, аби тільки їх загальна кількість не перевищила шести [1].

З позицій логічного процесора і потоку між статичним і динамічним поділом немає ніякої різниці - в обох випадках кожному ЛП виділяється своя половина ресурсу.

Різниця стає істотною, якщо в якості відправної точки взяти фізичний процесор.

Відсутність прив'язки потоків до конкретних входів черги дозволяє не брати до уваги, що є два потоки, і розцінювати обидві половини як єдину чергу. Черга диспетчеризації команд просто переглядає кожну команду у загальній черзі, оцінює залежності між командами, перевіряє доступність ресурсів, необхідних для виконання команди, і планує команду до виконання.

Таким чином, видача команд на виконання не залежить від того, якому потоку вони належать. Динамічне розділення черг диспетчеризації команд запобігає монополізації черг одним з логічних процесорів.

Якщо процесор Xeon обробляє тільки один потік, то для забезпечення максимальної продуктивності цього потоку надаються всі ресурси процесора. У динамічно розділених чергах знімаються обмеження на кількість входів, доступних одному потоку, а в статичних розділених чергах скасовується їх розбиття на дві половини.

Спільно використовувані ресурси. Цей вид ресурсів у гіперпотоковій технології вважається визначальним. Чим більше ресурсів можуть спільно використовувати логічні процесори, тим більшу обчислювальну потужність можна «зняти» з одиниці площі кристала процесора. Першу групу загальних ресурсів утворюють функціональні (виконавчі) блоки: цілочисельні операційні пристрої, блоки операцій з плаваючою комою і блоки звернення (читання/запису) до пам'яті. Ці ресурси "не знають", з якого ЛП надійшла команда. Те ж саме можна сказати і про реєстрові файли – другий вид спільно використовуваних ресурсів.

Сила гіперпотокової технології – загальні ресурси – одночасно є і її слабкістю. Проблема виникає, коли один потік монополізує ключовий ресурс (такий, наприклад, як блок операцій з плаваючою крапкою), чим блокує другий потік, викликаючи його зупинку. Завдання запобігання таких ситуацій покладається на компілятор та операційну систему, які повинні утворити потоки, що складаються з команд з вимогами, що максимально розрізняються, до спільно використовуваних ресурсів.

Так, один потік може містити команди, які в основному у блоці для операцій з плаваючою крапкою, а другий – складатиметься переважно з команд цілочисельної арифметики й звернення до пам'яті.

На закінчення необхідно зупинитися на третьому виді загальних ресурсів – кеш-пам’яті. Процесор Xeon передбачає роботу з кеш-пам'яттю трьох рівнів (LI, L2 і L3) і так званою кеш-пам'яттю трасування [1]. Обидва логічних процесора спільно використовують одну й ту ж кеш-пам'ять і дані, що зберігаються в ній. Якщо потік, що обробляється логічним процесором 0, хоче прочитати деякі дані, кешовані логічним процесором 1, він може взяти їх із загального кеша. Через те, що в гіперпотоковому процесорі одну й ту ж кеш-пам'ять використовують відразу два логічних процесора, вірогідність конфліктів і, отже, імовірність зниження продуктивності зростає.

Будь-який вид кеш-пам'яті однаково трактує всі звернення для читання або запису, незалежно від

того, який з логічних процесорів дане звернення виконує.

Це дозволяє будь-якому потоку монополізувати будь-який кеш, причому ніяким захистом від монополізації, як це має місце у випадку черг диспетчеризації команд, процесор не володіє. Іншими словами, фізичний процесор не в змозі змусити логічні процесори співпрацювати при їх зверненні до кеш-пам'яті.

В цілому, серед спільно використовуваних ресурсів в технології hyperthreading кеш-пам'ять виявляється найбільш критичним місцем, і конфлікти за володіння цим ресурсом позначаються на загальній продуктивності процесора найбільш відчутно.

За результатами досліджень [1], при використанні гіперпотокової технології приріст швидкості обчислень в деяких випадках може досягати 25-35%. У додатках, орієнтованих на багатозадачність, програми прискорюються на 15-20%. Можливі, однак, ситуації, коли приріст у швидкодії може бути непомітний і навіть бути негативним. Це відбувається внаслідок конфліктів, які виникають через спільне використання кеш-пам'яті кількома логічними процесорами. Запобігти цьому може розробка архітектури процесорів, в якій адресний простір кеш-пам'яті розділений на кілька незалежних відділів, кожному з яких відповідатиме свій логічний процесор.

Висновки

Ефективність технології знаходиться в прямій залежності від характеру реалізованого програмного забезпечення. Максимальна віддача досягається при роботі серверних додатків за рахунок розмаїтості процесорних операцій.

На даний момент апаратна підтримка технології закладена в мікропроцесори Pentium 4, Core 2 Duo, Core 2 Quad, Core i7 [1]. Також підтримка технології буде присутня в процесорах, які на даний момент знаходяться на стадії розробки.

Програмна підтримка технології передбачена в операційних системах Windows 2000, Windows XP, Windows Vista, Windows 7 і Windows Server (в попередніх ОС Windows така можливість відсутня).

Підвищення ефективності технології пропонується за рахунок розподілу адресного простору кеш-пам'яті на кілька незалежних відділів, кожному з яких відповідатиме свій логічний процесор, утворення компілятором та операційною системою окремих потоків команд, кожен з яких містить свій власний тип команд, а також за рахунок розвитку багато ядерності процесорів.

Список літератури

1. Цилькер Б.Я. Организация ЭВМ и систем / Б.Я. Цилькер, С.А. Орлов. – М.:, 2004. – 667 с.

2. Комп'ютерна техніка, Комп'ютерні технології: Підручник для студентів вищих навч. закладів. – Ред. Пушкарь О.І. – К: Академія, 2002. – 704 с. – (Альма-матер). – 270 с.

3. Наукові праці Донецького національного технічного університету. Сер.: Обчислювальна техніка та автоматизація. – Донецьк: ДонНТУ, 2003. – 280 с.

Надійшла до редколегії 31.05.2010

Рецензент: д-р техн. наук, проф. В.М. Рудницький, Черкаський державний технологічний університет, Черкаси.

ГИПЕРПОТОКОВАЯ ОБРАБОТКА КОМАНД

Л.А. Шувалова

В статье проведено исследование технологии гиперпотоковой обработки команд, разработанной фирмой Intel. Технология должна обеспечивать максимальную загрузку функциональных блоков суперскалярных многоконвеерных процессоров, что будет способствовать их максимальной производительности и быстродействию. В статье приведены особенности реализации гиперпотоковой технологии в процессоре Intel Xeon. Исследованы три подхода, которые применяются для распределения ресурсов физического процессора на несколько логических процессоров.

***Ключевые слова**: микропроцессор, гиперпотоковая обработка, суперскалярный, многоконвеерный, процессор, поток.*

HYPERTHREADING TECHNOLOGY

L.A. Shuvalova

Hyperthreading technology developed by Intel was explored in this article. The technology should provide a maximal loading of functional units of superscalar multivector processors, providing maximal productivity and performance. The article mentioned features of hyperthreading technology in the processor Intel Xeon. Three methods used for resource allocation of physical on logical processors were explored.

***Keywords**: microprocessor, hyperthreading technology, process, superscalar, processor, flow.*