DOI: https://doi.org/10.33216/1998-7927-2022-275-5-21-27

УДК 004.3

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ВІРТУАЛЬНОЇ БАГАТОПОТОЧНОСТІ (2, 3, 4 ПОТОКИ) ТИПУ HYPER THREADING ПРИ ВИКОНАННІ ПОТОКІВ В ОДНАКОВИХ ТА РІЗНИХ УМОВАХ

Недзельський Д.О., Сафонова С.О., Барбарук Л.В.

RESEARCH EFFICIENCY VIRTUAL MULTITHREADING (2, 3, 4 THREADS) TYPE HYPER THREADING WHEN EXECUTION THREADS IN THE SAME AND DIFFERENT CONDITIONS

Nedzelskyi D.O., Safonova S.O., Barbaruk L.V.

В статті аналітичними методами з елементами теорії масового обслуговування досліджена ефективність ядер сучасних процесорів з використанням віртуальної багатопоточності типу технології Hyper Threading при 2-х, 3-х, 4-х потоках з урахуванням структурних особливостей ядра як при виконанні потоків в однакових умовах, так і в різних умовах. Під однаковими умовами розглядалося виконання потоків, коли використовували однакові обсяги рівнів кеш-пам'яті. Під різними умовами розглядалися ситуації, коли потоки послідовно виконувалися в умовах з використанням доступних обсягів кеш-пам'яті, а паралельно потоки виконувалися в гірших умовах використання кеш-пам'яті (з використанням нижчого рівня кеш-пам'яті або навіть оперативної пам'яті).

Для дослідження вибрано широко поширені та наочні програми: «Множення матриць», «Рішення диференціальних рівнянь у приватних похідних методом сіток». У програмах, що досліджувалися було виділено ядро, уточнені інформаційно залежні команди і команди редукції, сформовані групи команд, їх кількість та визначені часи виконання кожної групи в програмі ядра, а також визначені ймовірності появи кожної групи команд. Розроблено методику досліджень та модель ядра. Для дослідження використовувалася двофазна спрощена модель ядра процесора. Було визначено коефіцієнт навантаження універсального ФП та, в залежності від значення різних параметрів програми і ядра процесора, визначено коефіцієнт використання ПУ моделі, визначені середній час виконання ядра програми та середні часи використання окремих спеціалізованих функціональних пристроїв.

Наведено результати досліджень у вигляді формул при 2-х, 3-х та 4-х потоках в одному фізичному ядрі як при виконанні потоків в однакових умовах, так і в різних умовах. Підтверджена ефективність віртуальної багатопоточності типу Hyper Threading при двох, трьох, чотирьох потоках з відсутністю структурних конфліктів, так і при різних умовах – наявності структурних конфліктів в підсистемі кеш-пам'яті.

При виконанні потоків у різних умовах ефективність (коефіцієнт прискорення) менша, ніж при виконанні в рівних умовах. Якщо при виконанні одного потоку використовується більше половини кеш-пам'яті третього рівня або потрібна інтенсивна робота з оперативною пам'яттю, використання віртуальної багатопоточності недоцільно.

Ключові слова: *багатопотокова віртуальність, процесор, ядро, конвеєр команд, перешкоди, продуктивність, ефективність.*

Вступ. Найчастіше ефективність ядер процесорів досліджувалася експериментальним шляхом за допомогою програмних тестів. Отримані дані про час виконання тестів дозволяли визначити відношення реальної ефективності до пікової ефективності.

Проте тести не можуть відповісти на питання, чому реальна продуктивність значно менше теоретичної продуктивності і якими шляхами можна збільшити реальну продуктивність.

У роботах [1,2], а також в багатьох подібних роботах, досліджувалася віртуальна двопоточна ефективність (при реалізації технології Hyper Treading) експериментальним шляхом, виконуючи різні тести. Встановлено, що в багатьох випадках віртуальна двопоточність забезпечує збільшення ефективності до 30%. У той же час, при виконанні деяких програм ефективність або не збільшується, або навіть погіршується. У всіх експериментальних дослідженнях шляхом виконання програм і фіксації часу їх виконання не вказувалися причини, які пояснили б, чому отримані такі результати. У випадках відсутності або навіть негативного ефекту це пояснювалося тим, що виконуючі дві програми конкурували за однакові ресурси без вказівки цих ресурсів і ступеня їх використання кожною програмою окремо.

В роботі [3] досліджувалась аналітичним методом ефективність віртуальної двопоточності типу

технології Hyper Threading в одному фізичному ядрі процесора при виконанні кількох розповсюджених програм (множення матриць, рішення диференціальних рівнянь з приватними похідними, швидке перетворення Фур'є) при наявності перешкод плавній роботі конвеєра виконання команд при виконанні однієї та двох однакових програм. Встановлено, що при виконанні однієї програми коефіцієнти використання окремих підсистем ядра а також оперативної пам'яті значно менші 1. Коефіцієнти використання кеш-пам'яті також далекі від максимальних. Підтверджена доцільність використання технології Hyper Threading. Продемонстровані деякі причини відсутності ефективності технології Hyper Threading.

В роботі [4] коефіцієнт прискорення (ефективність двопоточної віртуальності типу НТ) визначався як відношення часу послідовного виконання двох програм в одному фізичному ядрі до часу паралельного (одночасного) виконання цих двох програм, у цьому ж фізичному ядрі (тобто в режимі віртуальної двопоточності типу Hyper Threading). При цьому передбачалося, що для однопоточного виконання та двопоточного виконання програм фізичних ресурсів ядра достатньо. Наприклад, розміри масивів такі, що обсяги кеш-пам'яті достатні для розміщення даних виконуваних програм.

Проте, не завжди виконуються такі умови. У багатьох випадках можливі ситуації, коли обсяг кеш-пам'яті достатній для розміщення даних однопоточної програми, але не достатній для розміщення двопоточних програм. При дворазовому послідовному виконанні програми в однопотоковому режимі, робота виконується в сприятливих умовах (значно більша кількість звернень за даними здійснюється в швидкі рівні кеш-пам'яті).

При двопоточному паралельному виконанні програм, робота виконується в менш сприятливих умовах (значно більша кількість звернень за даними здійснюється на повільніші рівні кеш-пам'яті або навіть в оперативну пам'ять). Це означає, що умови виконання двох однопотокових програм можуть суттєво відрізнятись від умов паралельного виконання двопотокової програми. Відповідно результати (коефіцієнт прискорення) можуть також значно відрізнятися.

Висновки, отримані в роботі [4], сумніву не піддаються, але вони справедливі тільки для оптимальних поєднань розмірів даних у програмах та розмірів кеш-пам'яті в ядрі та в процесорі.

Також у роботі [4] не досліджувалася можливість використання трьохпоточних і чотирьохпоточних програм в одному фізичному ядрі, хоча зроблено висновки, що і під час виконання програм у двопоточному варіанті фізичні ресурси ядра використовуються не повною мірою.

Метою даної роботи є дослідження ефективності використання віртуальної багатопоточності типу Hyper Threading при 2-х, 3-х, 4-х потоках при наявності як оптимального, так і неоптимального співвідношення розмірів даних в програмах та розмірів кеш-пам'яті в ядрі (інакше кажучи структурних конфліктів в підсистемі кеш-пам'яті).

Методика досліджень. В статті використана методика досліджень з роботи [4]. У досліджуваних програмах «Множення матриць», «Розв'язання диференціальних рівнянь в приватних похідних методом сіток»:

- виділялося ядро – ділянка програми, яка забезпечує основний внесок під час виконання програми;
- розроблялися на умовному асемблері програми ядер;
- з'ясовувалися інформаційно залежні команди і команди редукції, якщо вони є;
- формувалися групи команд, які виконують інформаційно залежні ділянки команд ядра, і їх кількість;
- визначалися ймовірності появи кожної групи команд в програмі ядра;
- визначалися часи виконання кожної групи команд в програмі ядра;
- розроблялася модель ядра процесора, що виконує ядро програми;
- в моделі інформаційно залежні групи команд виконувалися підсистемою виконання послідовно згідно з алгоритмом виконання програми.

Далі визначалися:

- середній час виконання ядра програми;
- середні часи використання окремих спеціалізованих функціональних пристроїв таких як: кеш-пам'яті першого, другого і третього рівнів; пристрої множення; пристрої складання;
- коефіцієнт навантаження універсального функціонального пристрою;
- коефіцієнт використання пристрою управління (ПУ) моделі в залежності від значення різних параметрів програми і ядра процесора;
- коефіцієнти використання спеціалізованих функціональних пристроїв.

Модель ядра процесора при виконанні потоків. Для дослідження використовувалася двофазна спрощена модель ядра процесора, що складається з пристрою управління (ПУ), підсистеми виконання груп команд і буфера між ними.

Структурну схему еквівалентної спрощеної моделі ядра надано на рисунку.

ПУ почергово читає блок командної інформації потоку i (i =1;2;3;4) з підсистеми пам'яті, дешифрує цей блок командної інформації та записує чергову групу продешифрованих команд в буфер продешифрованих груп команд потоку **i**. Імовірність генерації **i**-ї дешифрованої групи команд Wi.

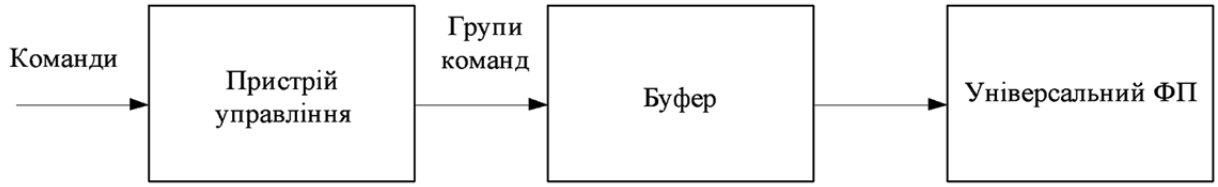

Рис. Спрощена структурна схема моделі ядра

Передбачається, що процес генерації груп команд ПУ найпростіший з показовим законом розподілу часу між згенерованими групами команд. Інтенсивність генерації груп команд визначається формулою (1)

$$\lambda_{ПУ}=1/t\text{ген}, \quad (1)$$

де tген – математичне очікування часу генерації групи команд.

Час генерації чергової групи команд ПУ залежить тільки від архітектурних і структурних особливостей ядра процесора.

Після генерації однієї групи команд та запису її в буфер потоку і пристрій управління переходить до генерації груп команд для потоку **i+1**, якщо буфер продешифрованих команд цього потоку не заповнений. Інакше пристрій управління переходить до генерації груп команд для наступного по черзі потоку **i+2**.

Якщо буфер команд потоку **i** заповнений, то пристрій управління перестає генерувати нові групи для цього потоку **i**.

Якщо буфери команд всіх потоків заповнені, то пристрій управління простоює.

Підсистема виконання груп команд складається з декількох спеціалізованих функціональних пристроїв (ФПj).

Типи ФП:

- ФП читання даних з кеш- пам’яті першого рівня L1D.
- ФП читання даних з кеш-пам’яті другого рівня L2.
- ФП читання даних з кеш-пам’яті третього рівня L3.
- ФП читання та запису даних з (в) оперативну пам’ять.
- ФП складання/віднімання.
- ФП множення.
- ФП ділення.

Інші спеціалізовані пристрої.

Команди з буферів продешифрованих команд одночасно можуть видаватися до всіх наявних ФП, при умові, що кожний ФП готовий приймати команду та готові операнди для цієї команди.

ФП читання даних з кеш-пам'яті першого рівня L1D може одночасно читати 2 операнди і записувати один результат.

ФП читання даних із кеш-пам'яті другого рівня L2, ФП читання даних із кеш-пам'яті третього рівня L3, та ФП читання та запису даних з (в) оперативну пам'ять комбінаційного типу та у будь-який момент часу можуть виконувати лише одну команду. До виконання наступної команди вони можуть розпочинати тільки після завершення попередньої команди.

Якщо ФП вільний і в буферах продешифрованих команд немає готових до виконання команд, ФП простоює.

Наприклад, при виконанні ядра програми «Множення матриць» група команд, що складається:

- з команд читання 2-х операндів з кеш-пам'яті даних першого рівня (L1D), команди множення прочитаних операндів і команди нагромаджуючого додавання результатів множення виконується з інтенсивністю $\mu_1=1/t\text{скл}$, де tскл - повний час виконання операції складання.
- з команд читання операнда з кеш-пам'яті даних першого рівня (L1D), команди читання операнда з оперативної пам'яті, команди множення прочитаних операндів і команди нагромаджуючого додавання результатів групи, виконується з інтенсивністю $\mu_{ОП}=1/t_{ОП}$, де $t_{ОП}$- час читання операнда з оперативної пам'яті;
- з команд читання операнда з кеш-пам'яті даних другого рівня L2, команди читання операнда з кеш-пам'яті другого рівня L2, команди множення прочитаних операндів і команди нагромаджуючого додавання результатів множення, виконується з інтенсивністю, що визначається часом читання операнда з кеш-пам'яті другого рівня L2 $\mu_{L2}=1/t_{L2}$, де t_{L2} час читання операнда з кеш-пам’яті другого рівня L2.
- з команд читання операнда з кеш-пам'яті даних першого рівня L1D, команди читання операнда з кеш-пам'яті третього рівня L3, команди множення прочитаних операндів і команди нагромаджуючого додавання результатів множення (тип групи L1; L3; Множ.; Дод.) виконується з інтенсивністю, що визначається часом читання операнда з кеш-пам'яті третього рівня L3 $\mu_{L3}=1/t_{L3}$, де t_{L3} час читання операнда з кеш-пам’яті третього рівня L3.

Результати досліджень.

Таблиця 1

Дослідження програми «Множення матриць»

Потоки виконуються в рівних умовах (структурні конфлікти відсутні)		
Варіант	**Коефіцієнти прискорення**	**Приклади коефіцієнтів прискорення**
всі дані в L1	$Sk \cong \frac{2/m + 15t_{\text{ОБЧ}}/t_{\text{ОП}}}{2/m + 15t_{\text{ОБЧ}}/\text{к}t_{\text{ОП}}}$	при $t_{\text{ОП}} = 256\tau$; $t_{\text{ОБЧ}} = 4\tau$; m=64 $S2 = 1.79$; $S3 = 2.43$; $S4 = 2.95$.
всі дані в L2	$Sk \cong \frac{2/m + t_{L2}/t_{\text{ОП}} + 15 * t_{\text{ОБЧ}}/t_{\text{ОП}}}{2/m + t_{L2}/t_{\text{ОП}} + 15t_{\text{ОБЧ}}/kt_{\text{ОП}}}$	при $t_{L2} = 12\tau$; $t_{\text{ОП}} = 256\tau$; $t_{\text{ОБЧ}} = 4\tau$; m=256 $S2 = 1.68$; $S3 = 2.18$; $S4 = 2.55$.
всі дані в L3	$Sk \cong \frac{2/m + t_{L3}/t_{\text{ОП}} + 15 * t_{\text{ОБЧ}}/t_{\text{ОП}}}{2/m + t_{L3}/t_{\text{ОП}} + 15t_{\text{ОБЧ}}/kt_{\text{ОП}}}$	при $t_{L3} = 36\tau$; $t_{\text{ОП}} = 256\tau$; $t_{\text{ОБЧ}} = 4\tau$; m=512 $S2 = 1.45$; $S3 = 1.70$; $S4 = 1.87$.
всі дані в ОП	Прискорення відсутнє	m>обсягу L3
Потоки виконуються в нерівних умовах (структурні конфлікти є)		
Структурний конфлікт L1/L2	$Sk \cong \frac{15t_{\text{ОБЧ}}}{t_{L2}+15t_{\text{ОБЧ}}/k}$	при $t_{\text{ОП}} = 256\tau$; $t_{\text{ОБЧ}} = 4\tau$; m=100; S2=1.43; S3=1.88; S4=2.22
Структурний конфлікт L2/L3	$Sk \cong \frac{t_{L2}+15t_{\text{ОБЧ}}}{t_{L3}+15t_{\text{ОБЧ}}/k}$	при $t_{L2} = 12\tau$; $t_{\text{ОП}} = 256\tau$; $t_{\text{ОБЧ}} = 4\tau$; m=256 S2=1.03; S3=1.2; S4=1.31.

\

Примітки:
t_{L1} – латентність кеш-пам'яті першого рівня;
t_{L2} – латентність кеш-пам'яті другого рівня;
t_{L3} – латентність кеш-пам'яті третього рівня;
$t_{\text{ОП}}$ – латентність оперативної пам'яті;
$t_{\text{ОБЧ}}$ – час обчислення однієї операції;
m – розмір квадратної матриці.

Передбачається також, що:

- якщо ФП вільний, то чергова згенерована група команд з відповідного буферу продешифрованих команд відразу надходить на виконання в ФП;
- вибирає групи команд з буфера груп команд згідно дисципліни FIFO.
- Якщо ФП вільний і в буфері груп команд немає заявок, то ФП простоює.

Потік виконаних в ФП груп команд найпростіший, закон розподілу показовий з інтенсивністю $\mu_{\text{ФП}}=1/t_{\text{ФП}}$, де $t_{\text{ФП}}$ – середній час виконання групи команд.

Висновки. 1. При виконанні потоків в рівних умовах коефіцієнти прискорення змінюються так:

1.79 - для двопоточності; 2.43 - для трьохпоточності; 2.95 = для чотирьохпоточності при виконанні всіх потоків з використанням кеш-пам'яті першого рівня L1D.

1.68 - для двопоточності; 2.18 - для трьохпоточності; 2.55 = для чотирьохпоточності при виконанні всіх потоків з використанням кеш-пам'яті другого рівня L2.

1.45 - для двопоточності; 1.70 - для трьохпоточності; 1.87 = для чотирьохпоточності при виконанні всіх потоків з використанням кеш-пам'яті третього рівня L3.

Прискорення відсутнє, якщо дані навіть одного потоку більші обсягу кеш-пам'яті третього рівня L3.

2. Якщо потоки виконуються в нерівних умовах (структурні конфлікти є) коефіцієнти прискорення змінюються так:

Структурний конфлікт L1/L2 (один потік може виконуватися через L1, а вже 2, 3, 4 потоки будуть виконуватися через L2) - 1.43 - для двопоточності; 1.88 - для трьохпоточності; 2.22 для чотирьохпоточності.

Структурний конфлікт L2/L3 (один потік може виконуватися через L2, а вже 2, 3, 4 потоки будуть виконуватися через L3) - 1.03 - для двопоточності; 1.20 - для трьохпоточності; 1.31 для чотирьохпоточності.

Структурний конфлікт L3/ОП (один потік може виконуватися через L3, а вже 2, 3, 4 потоки будуть виконуватися через оперативну пам'ять прискорення відсутнє.

3. Наявність структурних конфліктів при використанні підсистеми кеш-пам'яті значно зменшує коефіцієнти прискорення. Наприклад: при двопоточності - від 1.79 до 1.45 – при виконанні потоків в рівних умовах; до 1.03 та відсутності прискорення; при трьохпоточності від 2.43 до 1.70 – при виконанні потоків в рівних умовах; до 1.20 та відсутності прискорення при виконанні потоків в нерівних умовах; для чотирьохпоточності від 2.95 до 1.31 при виконанні потоків в рівних умовах та відсутності прискорення при виконанні потоків в нерівних умовах.

4. Підтверджуються експериментальні дані тестів ефективності віртуальної двопоточності про досить значне можливе прискорення, так і про його відсутність.

5. Віртуальні трьох- та чотирьохпоточності ефективні.

Таблиця 2

Дослідження програми «Розв'язання диференціальних рівнянь в приватних похідних»

Потоки виконуються в рівних умовах (структурні конфлікти відсутні)		
Варіант	**Коефіцієнти прискорення**	**Приклади**
всі дані в L1	$Sk \cong \frac{3t_{ОП}/n+16t^2_{ОБЧ}}{3t_{ОП}/n+16t^2_{ОБЧ}/k}$	n=128; $t_{ОП}$ =256τ ; $t^1_{ОБЧ} = 12\tau$; $t^2_{ОБЧ} = 24\tau$ S2=1.97; S3=2.91, S4=3.82.
всі дані в L2	$Sk \cong \frac{3t_{ОП}/n + 2t_{L2} + t^1_{ОБЧ} + 15t^2_{ОБЧ}}{3t_{ОП} + 2t_{L2} + t^1_{ОБЧ} + 15t^2_{ОБЧ}/k}$	n=128; $t_{ОП}$ =256τ ; $t^1_{ОБЧ} = 12\tau$; $t^2_{ОБЧ} = 24\tau$; $t_{L2} = 12\tau$ $S2 \cong 1.78$; $S3 \cong 2.39$; S4 ≅ 2.90.
всі дані в L3	$Sk \cong \frac{3t_{ОП}/n + 2t_{L3} + t^1_{ОБЧ} + 15t^2_{ОБЧ}}{3t_{ОП}/n + 2t_{L3} + t^1_{ОБЧ} + 15t^2_{ОБЧ}/k}$	n=128; $t_{ОП}$ =256τ ; $t^1_{ОБЧ} = 12\tau$; $t^2_{ОБЧ} = 24\tau$; $t_{L3} = 36\tau$ $S2 \cong 1.67$; $S3 \cong 2.15$; S4 ≅ 2.5.
всі дані в ОП	$Sk \cong \frac{3t_{ОП} + t^1_{ОБЧ} + 15t^2_{ОБЧ}}{3t_{ОП} + t^1_{ОБЧ} + 15t^2_{ОБЧ}/k}$	m>обсягу L3 n=128; $t_{ОП}$ =256τ ; $t^1_{ОБЧ} = 12\tau$; $t^2_{ОБЧ} = 24\tau$ $S2 \cong 1.19$; $S3 \cong 1.27$; S4 ≅ 1.31.
Потоки виконуються в нерівних умовах (структурні конфлікти є)		
Структурний конфлікт L1/L2	$Sk \cong \frac{3t_{ОП}/n + t^1_{ОБЧ} + 15t^2_{ОБЧ}}{3t_{ОП}/n + 2t_{L2} + t^1_{ОБЧ} + 15t^2_{ОБЧ}/k}$	m=64; n=128, tоп=256 τ, $t^1_{ОБЧ}$=12 τ, $t^2_{ОБЧ}$=24 τ, t_{L2}=12 τ S2=1.65; S3=2.26; S4=2.78.
Структурний конфлікт L2/L3	$Sk \cong \frac{3t_{ОП/n} + 2t_{L2} + t^1_{ОБЧ} + 15t^2_{ОБЧ}}{3t_{ОП}/n + 2t_{L3} + t^1_{ОБЧ} + 15t^2_{ОБЧ/k}}$	n=128, t_{L2}=12 τ, tоп=256 τ, $t^1_{ОБЧ}$=12 τ, $t^2_{ОБЧ}$=24 τ, t_{L3}=40 τ S2=1.45; S3=1.85; S4=2.14.
Структурний конфлікт L3/ОП	$S_k \cong \frac{3t_{ОП}/n + 2t_{L3} + t^1_{ОБЧ} + 15t^2_{ОБЧ}}{3t_{ОП} + t^1_{ОБЧ} + 15t^2_{ОБЧ}/k}$	m>1024 Прискорення відсутнє

Примітки:
k – кількість потоків; n – кількість ітерацій;
m – розмір квадратної сітки;
t_{L2} – латентність кеш-пам'яті другого рівня;
t_{L3} – латентність кеш-пам'яті третього рівня;
$t_{ОП}$ – латентність оперативної пам'яті;
$t^1_{ОБЧ}$– час обчислення першої порції операцій;
$t^2_{ОБЧ}$– час обчислення другої порції операцій.

Висновки. 1. При виконанні потоків в рівних умовах коефіцієнти прискорення змінюються так:

1.97 - для двопоточності; 2.91 - для трьохпоточності; 3.82 = для чотирьохпоточності при виконанні всіх потоків з використанням кеш-пам'яті першого рівня L1D.

1.78 - для двопоточності; 2.39 - для трьохпоточності; 2.90 = для чотирьохпоточності при виконанні всіх потоків з використанням кеш-пам'яті другого рівня L2.

1.67 - для двопоточності; 2.15 - для трьохпоточності; 2.5 = для чотирьохпоточності при виконанні всіх потоків з використанням кеш-пам'яті третього рівня L3.

1.19 - для двопоточності; 1.27 - для трьохпоточності; 1.31 = для чотирьохпоточності при виконанні всіх потоків з використанням виключно оперативної пам'яті

2. Якщо потоки виконуються в нерівних умовах (структурні конфлікти є) коефіцієнти прискорення змінюються так:

Структурний конфлікт L1/L2 (один потік може виконуватися через L1, а вже 2, 3, 4 потоки будуть виконуватися через L2) - 1.65 - для двопоточності; 2.26 - для трьохпоточності; 2.78 для чотирьохпоточності.

Структурний конфлікт L2/L3 (один потік може виконуватися через L2, а вже 2, 3, 4 потоки будуть виконуватися через L3) – 1.45 - для двопоточності; 1.85 - для трьохпоточності; 2.14 - для чотирьохпоточності.

Структурний конфлікт L3/ОП (один потік може виконуватися через L3, а вже 2, 3, 4 потоки будуть виконуватися через оперативну пам'ять - прискорення відсутнє.

3. Наявність структурних конфліктів при використанні підсистеми кеш-пам'яті зменшує коефіцієнти прискорення.

4. Підтверджуються експериментальні дані тестів ефективності віртуальної двопоточності про досить значне можливе прискорення, так і про його відсутність.

5. Віртуальні трьох- та чотирьохпоточності ефективні.

Загальні висновки. Отримані аналітичні вирази для коефіцієнтів прискорення.

Вони підтверджують ефективність віртуальної багатопоточності при 2-х, 3-х, 4-х потоках в одному фізичному ядрі як при виконанні потоків в рівних, так і в нерівних умовах.

Добре корелюються з експериментальними даними тестів про ефективність віртуальної двопоточності та можливості відсутності прискорення.

Якщо при виконанні одного потоку використовується більше половини обсягу кеш-пам'яті третього рівня або необхідна інтенсивна робота з оперативною пам'яттю, використання віртуальної багатопоточності залежить від типу програм, які будуть виконуватися, і може бути недоцільним.

Література

1. Є.О. Шквар. Ефективність використання технології віртуальної багатопоточності при паралельному розв'язанні рівняння Пуассона. Вісник НАУ, Київ. 2012 N3 (101), том 2, С.138- 141.
2. Буділовський С. Simultaneous Multithreading (SMT) в топовом AMD Ryzen 7 2700X: тестування в синтетиці і іграх. Постійний URL: https://ru.gecid.com/cpu/simultaneous_multithreading_smt_v_amd_ryzen_7_2700x
3. Недзельський Д.О. Дослідження ефективності підсистеми генерації команд в ядрах сучасних процесорів. Луганськ: Вісник Східноукраїнського національного університету ім. В. Даля, 2017. - №8 (238), - С.64-66.
4. Недзельський Д.О., Сафонова С.О., Барбарук Л.В. Аналітичне дослідження ефективності ядер процесорів при наявності «перешкод» з використанням технології Hyper Threading Наукові вісті Далівського університету. Електронне наукове фахове видання. - 2021. - №21. doi: https://doi.org/10.33216/2222-3428-2021-21-3

References

1. E.O. Shkvar Effectiveness of using virtual multithreading technology in parallel solution of Poisson's equation. Bulletin of NAU, Kyiv. 2012 N3 (101), T 2, pp. 138-141.
2. Budilovskyi S. Simultaneous Multithreading (SMT) in the top AMD Ryzen 7 2700X: testing in synthetics and games. Permanent URL: ttps://ru.gecid.com/cpu/simultaneous_multithreading_smt_v_amd_ryzen_7_2700x.
3. Nedzelskyi D.O. Study of the effectiveness of the command generation subsystem in the cores of modern processors. Luhansk: Bulletin of the East Ukrainian National University named after V. Dalya, 2017. - N. 8 (238), - P.64-66.
4. Nedzelskyi D.O., Safonova S.O., Barbaruk L.V. Analytical study effectiveness processor cores in the presence of "interference" using Hyper Threading technology. Scientific news of Daliv University. Electronic scientific publication. - 2021. - No. 21. doi: https://doi.org/10.33216/2222-3428-2021-21-3

Nedzelskyi D.O., Safonova S.O., Barabruk L.V. Research efficiency virtual multithreading (2, 3, 4 threads) type Hyper Threading when execution threads in the same and different conditions.

The article uses analytical methods with elements of the mass service theory to investigate the efficiency of the cores of modern processors using virtual multithreading of the Hyper Threading type with 2, 3, and 4 threads, taking into account the structural features of the core both when executing threads under the same conditions, and in different conditions. Under the same conditions, the execution of threads was considered when they used the same amount of cache memory levels. Under different conditions, situations were considered when threads were executed sequentially in conditions using available cache memory volumes, and parallel threads were executed in worse cache memory usage conditions (using a lower level of cache memory or even RAM).

Widely used and visual programs were selected for the study: "Multiplication of matrices", "Solution of differential equations in partial derivatives by the grid method". In the studied programs, the kernel was identified, information-dependent

commands and reduction commands were specified, groups of commands were formed, their number and execution times of each group in the kernel program were determined, and the probabilities of occurrence of each group of commands were determined. A research methodology and a model of the core have been developed. The study used a two-phase simplified model of the processor core. The load factor of the universal functional device was determined and, depending on the value of various parameters of the program and the processor core, the utilization rate of the model control device was determined, the average execution time of the program core and the average utilization times of individual specialized functional devices were determined.

The results of studies with 2, 3 and 4 threads in the form of formulas in one physical core are given, both when threads are executed under the same conditions, and under different conditions. The effectiveness of virtual multithreading such as Hyper Threading with two, three, four threads with no structural conflicts and under different conditions - the presence of structural conflicts in the of cache memory - has been confirmed.

When performing threads under different conditions, the efficiency (acceleration factor) is less than when performing under equal conditions. If a single thread uses more than half of the third-level cache memory or requires intensive work with RAM, then the use of virtual multithreading is inappropriate.

Keywords: *multi-threaded virtuality, processor, core, instruction pipeline, interference, performance, efficiency.*

Недзельський Дмитро Олександрович – к.т.н., доцент, доцент кафедри комп'ютерних наук та інженерії Східноукраїнського національного університету імені Володимира Даля, e-mail: nedzelsky@snu.edu.ua

Сафонова Світлана Олександрівна – к.т.н., доцент, доцент кафедри комп'ютерних наук та інженерії Східноукраїнського національного університету імені Володимира Даля, e-mail: safonova@snu.edu.ua

Барбарук Ліна Вікторівна – к.т.н., доцент кафедри комп'ютерних наук та інженерії Східноукраїнського національного університету імені Володимира Даля, e-mail: barbaruk_a@snu.edu.ua

Стаття подана 19.10.2022