

7. Лозинський А.О., Мороз В.І., Паранчук Я.С. Розв'язання задач електромеханіки в середовищах пакетів MathCAD і MATLAB: Навчальний посібник. – Львів: ДУ “Львівська політехніка”, 2000. – 166 с.

<http://doi.org/10.5281/zenodo.3859691>

Поступила 19.09.2019р.

УДК 004.272.26

В.В. Душеба, Київ

ОРГАНІЗАЦІЯ СПЕЦІАЛІЗОВАНОГО КОМУНІКАЦІЙНОГО СЕРЕДОВИЩА В СИСТЕМАХ ПАРАЛЕЛЬНОЇ ОБРОБКИ ІНФОРМАЦІЇ В РЕЖИМІ РЕАЛЬНОГО ЧАСУ

Abstract. The paper proposes the principles of organizing a specialized system of interprocessor exchanges in parallel information processing systems that allow you to develop a method for addressless data transmission in separate words, with hardware synchronization of each individual word, which is the basis for obtaining a significant increase in acceleration.

Вступ

Коло завдань, що вимагають для свого рішення застосування високопродуктивних обчислювальних ресурсів, сьогодні ще більш розширилося. Це пов'язано з тим, що відбулися фундаментальні зміни в самій організації наукових досліджень. Внаслідок прогресивного розвитку обчислювальної техніки значно посилюється напрям чисельного моделювання і чисельного експерименту. Стало можливим моделювати в реальному часі процеси інтенсивних фізико-хімічних і ядерних реакцій, глобальні атмосферні процеси, процеси економічного і промислового розвитку і т.д. Очевидно, що рішення таких масштабних завдань вимагає значних обчислювальних ресурсів [1].

Розширення кола завдань вимагає розширення типів і різновидів багатопроцесорних систем. Проте, при цьому залишаються цілі класи завдань, які взагалі не можуть бути реалізовані навіть на сучасних найпотужніших суперкомп'ютерних системах.

Організація мережі обміну в системах паралельної обробки даних

Багатопроцесорні обчислювальні системи з масовим паралелізмом (МРР) відносять до найбільш перспективних і найбільш поширених видів архітектур. Вони з'явилися, як альтернатива багатопроцесорним системам із

загальною пам'яттю, недоліком яких є наявність конфліктів при зверненні до спільної пам'яті та необхідність відстеження когерентності даних.

У MPP системах пам'ять рівномірно розподілена між усіма процесорами системи. Кожен процесор MPP наділений своєю локальною пам'яттю, яка недоступна для інших процесорів системи. Взаємодія між процесорами здійснюється за допомогою повідомлень через комутаційну мережу. До переваг MPP системи слід віднести і порівняно просте її масштабування. Масовість паралелізму передбачає можливість нарощування системи до будь-якої необхідної продуктивності. До недоліків MPP систем відносять відсутність єдиного адресного простору [2 – 6].

Останнім часом по тематиці даної роботи з'явилися публікації, які загальний стан цього наукового напрямку характеризують як незадовільний або навіть як "стан стабільної стагнації", мова йде в даному випадку про MPP системах. При цьому, чинники, що призводять до стагнації, ніде не згадуються. Деякі факти підтверджують такого роду висновки. Наприклад, починаючи з 2012 р, перше місце в списку TOP-500 займає китайський суперкомп'ютер Tianhe-2, випередивши американські суперкомп'ютери в продуктивності. Звертає на себе увагу той факт, що реальна продуктивність його становить 60% від пікового, що є посереднім показником. Це означає, що в цьому суперкомп'ютері, якому налічується більше 3-х мільйонів ядер, понад 1,2 мл, з яких не працюють за заданою програмою і, швидше за все створюють перешкоди цій роботі. У американського суперкомп'ютера Titanic, який займає 2-е місце в цьому списку реальна продуктивність складає 65% від пікового, що так само нижче середніх значень. Фірма Крея, творець комп'ютера, при цьому втратила своє лідируюче становище в світі по продуктивності систем.

Проблема обміну є слабкою ланкою для MPP систем. Як показують розрахунки, механізм обміну повідомленнями суперечить принципу масовості паралелізму. Якщо інтенсивність потоків міжпроцесорних обмінів перевищує певний поріг, прискорення вирішення таких алгоритмів може ставати менше одиниці. В даному випадку причина полягає не в алгоритмах, а в самих системах, на яких вони реалізуються.

Побудова механізмів реалізації пам'яті, що розділяється над механізмом передачі повідомлень набагато складніше. Без передбачуваної підтримки з боку апаратури всі звернення до пам'яті будуть потребувати залучення операційної системи як для забезпечення перетворення адрес і захисту пам'яті, так і для перетворення звернень до пам'яті в посилку і прийом повідомлень. Оскільки операції завантаження і запису зазвичай працюють з великим обсягом даних, то великі накладні витрати з підтримки такого обміну унеможливають чисто програмну реалізацію. При оцінці будь-якого механізму обміну критичними є три характеристики продуктивності:

– *Смуга пропускання*: в ідеалі смуга пропускання механізму обміну буде обмежена смугами пропускання процесора, пам'яті і системи міжз'єднань, а не будь-якими аспектами механізму обміну.

– *Затримка*: в ідеалі затримка повинна бути настільки мала, наскільки це можливо. Для її визначення критичні накладні витрати апаратури і програмного забезпечення, пов'язані з ініціюванням та завершенням обміну.

– *Приховування затримки*: наскільки добре механізм приховує затримку шляхом перекриття обміну з обчисленнями або з іншими обмінами.

Кожен з цих параметрів продуктивності впливає на характеристики обміну. Зокрема, затримка і смуга пропускання можуть змінюватися в залежності від розміру елемента даних. У загальному випадку, механізм, який однаково добре працює як з невеликими, так і з великими об'ємами даних буде більш гнучким і ефективним.

Таким чином, відмінності різних машин з розподіленою пам'яттю визначаються моделлю пам'яті і механізмом обміну.

Модель пам'яті передачі даних і синхронізація обміну

Проведені дослідження поточного стану розвитку суперкомп'ютерних систем, дають підстави стверджувати про необхідність використання нових підходів і принципів організації спеціалізованої системи міжвузлових обмінів. В роботах [7, 8] показана можливість мінімізації міжпроцесорного обміну (МО). Апаратною основою скасування МО служить спеціальним чином організована двохпортова пам'ять, яку ми називаємо пам'яттю передачі даних – Memoгу Exchange (ME). Пам'ять ME має послівну організацію. Окремий модуль пам'яті ME призначений для оптимізації обміну між двома взаємодіючими пристроями.

Пам'ять ME – це пам'ять з перемикаючими комірками. У кожен момент часу комірка пам'яті ME підключена тільки до одного з портів, тобто вона не має стану, коли вона підключена відразу до обох портів або не підключена до жодного з них. Організація пам'яті ME така, що пристрої, підключені до неї, можуть звертатися до довільної комірки пам'яті асинхронно, тобто в будь-якій послідовності або одночасно, на різних тактових частотах, і без попередньої перевірки стану комірки. Звертання до пам'яті ME здійснюється типовими адресними операціями. Залежно від стану комірки доступ до неї отримує тільки той із двох пристроїв, які звернулися, до порту якого підключена в даний момент комірка. При цьому, черговість звернень не має значення.

Стан комірки визначається значенням біта FE, вбудованого в кожен комірку пам'яті. При кожному зверненні до комірки у з боку будь-якого з портів, одночасно з процесом звертання, апаратними засобами самої комірки здійснюється аналіз FE, який є недоступним для програмних засобів. Він запускається адресною частиною будь-якої адресної арифметико-логічної команди. Біт FE – це лічильник по mod 2. Адресні операції з циклом читання обов'язково додають одиницю до біту FE, що рівнозначно переключенню комірки до іншого порту. Зміна FE відбувається в тому ж такті команди, в якому здійснюється і запис слова в комірку. Пристрій, записавши слово, перемикає цієї ж командою осередок на інший порт і одночасно з цим втрачає можливість зчитувати записане слово.

Пристрої не забезпечені механізмом захоплення комірок. Комірка є спільним ресурсом двох пристроїв. У неї закладено принцип негайного перемикання комірки від одного пристрою до іншого, як тільки необхідність в ній відпадає. Реалізація цього принципу повністю покладена на комірку. Така організація є абсолютно безконфліктної.

Нехай пам'ять влаштована так, що при $[FE] = 0$ комірка завжди підключена до першого порту, а при $[FE] = 1$ – до другого її порту. Припустимо, що до комірки одночасно звернулися з різних портів сполучені, через даний модуль пам'яті МЕ, пристрої. З боку першого порту здійснюється запис слова, а з боку другого порту здійснюється його зчитування. Звернення відбувається при $[FE] = 0$, тобто комірка підключена до першого порту.

В цьому випадку з боку першого порту відбудеться запис слова, призначеного для сполученого пристрою. Причому, процес цього запису не має відмінностей від типової адресної операції запису. По завершенню операції запису біт $[FE] = 1$. Причому, функція зміни FE не закладена в адресних командах – це властивість пам'яті МЕ.

Сполучений пристрій, підключений з боку другого порту, звернувся до комірки для читання слова в той момент, коли воно ще не було записано, а комірка в цей момент була відключена від другого порту. У цьому випадку, щоб уникнути передчасного, помилкового читання комірки та несанкціонованої модифікації лічильника команд і внутрішніх регістрів в сполученому пристрої, у другому порту пам'яті, на підставі аналізу FE, виробляється сигнал блокування. Цей сигнал затягує в часі поточний такт синхроімпульсу, блокує операцію читання, і парний пристрій переходить в режим очікування в процесі виконання операції читання.

Як тільки значення FE стає рівним одиниці, комірка перемикається на другий порт, знімається сигнал блокування 2-го порту і відбувається завершення читання слова розпочатого до його запису. На цьому завершується цикл передачі одного слова. Таку передачу будемо називати локальною передачею даних, тому що вона можлива тільки між пристроями, що мають безпосередній зв'язок через модулі пам'яті МЕ. Організація локальної передачі даних в системі означатиме суттєву мінімізацію міжпроцесорного обміну і витрат часу на його реалізацію, і є необхідною умовою збільшення продуктивності.

Розглянемо дві модифікації пам'яті МЕ і стан біту FE після зчитування слова.

1. Комірки перемикаються операціями записи з боку будь-якого з портів (дуплексний метод передачі). У цьому випадку відразу після запису з боку одного з портів комірка із записаним операндом перемикається на інший порт і стає доступною для читання. Зчитування з неї може здійснюватися багаторазово і тільки після здійснення операції запису відбудеться повернення комірки до початкового порту.

2. Комірки перемикаються операцією запису з боку одного порту і операцією зчитування з боку іншого порту. В даному випадку має місце

односпрямована передача даних (симплексний тип передачі). У деяких класах завдань цей тип є більш ефективним, тому що операція читання може здійснюватися тільки один раз і не може гальмувати роботу пристрою підключеного до іншого порту.

Хоча наведені відмінності досить істотні, але механізми зміни FE бітів не відрізняються. Будемо розглядати тільки симплексний тип передачі.

Комірка цього типу має такі особливості. Якщо $[FE] = 0$, комірка підключена до першого порту і не містить слово (ознака "empty"); може бути виконана одна операція з циклом write. Після завершення операції write $[FE] = 1$. Якщо $[FE] = 1$, комірка підключена до другого порту і містить непрочитане слово (ознака "full"); може бути виконана одна операція з циклом read. Після завершення операції read $[FE] = 0$.

Таким чином, один пристрій тільки пише, а друге тільки читає. перший порт пам'яті має тільки входи даних і другий порт тільки виходи. Після будь-якої операції відбувається повернення комірки іншому порту. У запропонованій пам'яті ME не може статися розсинхронізація за даними навіть, якщо пристрої функціонують на різних тактових частотах.

Звернення до пам'яті ME будь якою адресною операцією з циклом запису стає одночасно і передачею слова іншому пристрою, а кожна адресна операція з циклом читання є операцією обробки прийнятого слова. Водночас, будь-яка операція з пам'яттю є ефективною процедурою синхронізації, тому що, така модель не дозволяє записати слово поверх іншого слова і прочитати незаписане слово. За когерентністю даних стежать не протоколи обміну, а множина FE бітів пам'яті ME, на виході якої поява некогерентних даних повністю виключена.

Висновки

Запропоновані принципи організації спеціалізованої системи міжпроцесорних обмінів в системах паралельної обробки інформації дозволяють розробити метод безадресної передачі даних окремими словами, а не масивами, з апаратною синхронізацією кожного окремого слова, що є основою отримання суттєвого росту прискорення. Мінімізація обміну можлива тільки при передачі даних між паралельними фрагментами розпаралеленої програми. Для повного перекриття обміну в системах паралельної обробки інформації необхідним є організація мережі зв'язків кожного обчислювального вузла з кожним, тобто така система відображається повним графом, кожної спрямованої дузі якого в реальній системі ставиться у відповідність один модуль міжпроцесорного обміну. Один із способів організації зв'язку "кожного з кожним" реалізується в ортогональних багатопроцесорних системах. Такий підхід дозволяє збільшити прискорення всіх паралельних алгоритмів, бо виключає величезні втрати, закладені в механізмах передачі повідомлень.

1. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. - М.: Мир, 1982. - 416 с.

2. Гергель В.П., Линёв А.В. Проблемы и перспективы достижения экзафлопного уровня производительности суперкомпьютерных систем / Информационные технологии Вестник Нижегородского университета им. Н.И. Лобачевского, 2012, № 3 (1), С. 189-198.
3. Tumeo A., Secchi S., Villa O. Designing Next-Generation Massively Multithreaded Architectures for Irregular Applications. IEEE Computer, August 2012.
4. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. - СПб.: Питер, 2013. – 816 с.
5. Sterling T. Architecture Paths to Exaflops Computing. Is Multicore the next Moor's Law? What about Memory? Invited Presentation to the DOE E3SGS Town Hall Meeting, April 18, 2007.
6. Ferreira da Silva R., Callaghan S., Deelman E. On the use of burst buffers for accelerating data-intensive scientific workflows // Proceedings of the 12th Workshop on Workflows in Support of Large-Scale Science, WORKS '17. ACM, 2017. P. 2:1–2:9.
7. Сигарев А.А. Методология упразднения межпроцессорного обмена в МВС со множественным потоком команд. // Информационные технологии. № 7 / 2009.
8. Сигарев А.А., Душеба В.В. Способ упразднения межпроцессорного обмена в макроконвейерах // Электрон. моделирование. – 2006. – Т.28, № 6. – С. 71 – 89.

<http://doi.org/10.5281/zenodo.3859693>

Поступила 3.10.2019р.

УДК681. 625.23:004.942

М.І. Верхола, д.т.н., проф., Українська академія друкарства, м. Львів
М.І. Калитка, ст.викладач, Українська академія друкарства, м. Львів

МОДЕЛЮВАННЯ ТА АНАЛІЗ ВПЛИВУ ТЕХНОЛОГІЧНИХ ПАРАМЕТРІВ НА РОЗПОДІЛ ОБ'ЄМІВ ПОТОКІВ ФАРБИ У ФАРБОДРУКАРСЬКІЙ СИСТЕМІ ПОСЛІДОВНО-ПАРАЛЕЛЬНОЇ СТРУКТУРИ

Abstract. To perform this study, a mathematical model was developed and a signal graph and a color printing system simulator were built. With the help of simulation modeling the nature of the influence of changing the parameters of the printing form and ink on the process of ink transfer and filling during printing of the ink printing system with ink is established. The need to determine the amount of ink that accumulates in the system is justified, as it affects both the parameters of the input task and the amount of ink consumption for the manufacture of printed products.

Постановка проблеми. Фарбодрукарські системи офсетних машин повинні забезпечувати нанесення рівномірного шару фарби заданої товщини на друкувальні елементи фарби і передачу його через офсетний циліндр на матеріал, що задруковується. Для вирівнювання товщини фарби в

© М.І. Верхола, М.І. Калитка 225