УДК 004.056.55
В.О. Ліщина
Луцький інститут розвитку людини Університету «Україна»

ПАРАЛЕЛЬНА ПАМ'ЯТЬ З ВПОРЯДКОВАНИМ ДОСТУПОМ В БАГАТОПРОЦЕСОРНИХ СИСТЕМАХ ЗІ СПІЛЬНОЮ ПАМ'ЯТТЮ

Описано системи з спільною пам'яттю для різних типів комунікаційної мережі. Показано типову структуру систем UMA, NUMA та COMA. Наведено приклади архітектури на основі цих систем

Ключові слова: *паралельна пам'ять, процесор, система UMA, система NUMA, система COMA.*

Вступ

При проектуванні багатопроцесорних систем ключовим є питання поділу даних між процесорами. Ідеальною відповіддю на це питання є застосування єдиного адресного простору. Тут процесори взаємодіють через спільні змінні, що зберігаються в єдиній пам'яті з єдиним адресним простором для усіх процесорів. В системі з спільною пам'яттю процесори взаємодіють між собою шляхом зчитування-запису інформації з комірок спільної пам'яті, яка є однаково доступною для всіх процесорів. Кожен процесор може мати регістри, буфери, кеш та локальну пам'ять, як додаткові ресурси пам'яті. Потрібно брати до уваги кілька основних проблем, які з'являються при проектуванні систем з спільною пам'яттю. Це контроль доступу до пам'яті, синхронізація, захист, безпека [1-2].

Контроль доступу визначає, для якого процесу можливий доступ до яких ресурсів. Моделі контролю доступу вимагають перевірки вмісту таблиці контролю доступу для кожного запиту доступу від процесорів до спільної пам'яті. Ця таблиця вміщує прапорці, які визначають законність кожної спроби доступу. Якщо є спроба доступу до ресурсів, то поки бажаний доступ не завершений, всі спроби доступу нехтуються, і нелегальні процеси блокуються. Запити від сумісних процесів можуть змінити вміст таблиці контролю доступу протягом їх виконання. Прапорці контролю доступу з правилами синхронізації визначають функціональність системи.

Механізми синхронізації обмежують час доступу від сумісних процесів до спільних ресурсів. Досконала синхронізація гарантує, що інформація передається належним чином та забезпечується належна системна функціональність.

Захист – це системна особливість, що перешкоджає наданню процесам довільного доступу до ресурсів, що належать іншим процесам. Сумісне використання і захист є протилежними за функціями, оскільки сумісне використання дозволяє доступ, тоді як захист обмежує його.

Основна частина

Найпростіша система з спільною пам'яттю має один модуль пам'яті, який може бути доступний від двох процесорів (рис.1). Запити надходять в модуль пам'яті через його два порти. Арбітражний блок в межах модуля пам'яті передає запити до диспетчера пам'яті. Якщо модуль пам'яті не зайнятий і надходить один запит, то арбітражний блок передає цей запит до диспетчера пам'яті і запит обслуговується. Модуль знаходиться в зайнятому стані під час обслуговування запиту. Якщо надходить новий запит коли пам'ять зайнята, обслуговуючи попередній запит, процесор P, що послав запит, може утримувати свій запит на лінії, поки пам'ять не стає вільною, або може повторювати свій запит пізніше.

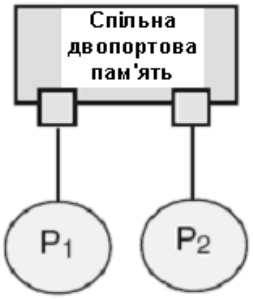

Рис.1 Система з спільною пам'яттю з двома портами.

Залежно від типу комунікаційної мережі, системи з спільною пам'яттю можна класифікувати наступним чином:

- системи UMA з однорідним доступом до пам'яті (UMA - uniform memory access),
- системи NUMA з неоднорідним доступом до пам'яті (NUMA - nonuniform memory access),
- системи COMA - архітектура лише з кеш пам'яттю (COMA - cache-only memory architecture).

У системі UMA спільна пам'ять доступна всім процесорам через комунікаційну мережу однаковим чином. Тому всі процесори мають рівний час доступу до будь-якої комірки пам'яті. Комунікаційна мережа, використовувана в системі UMA, може бути одною або множинною шиною, координатною мережею чи багатопортовою пам'яттю.

Типова структура системи з однорідним доступом до пам'яті на основі одношинної комунікаційної мережі приведена на рис.2 а.

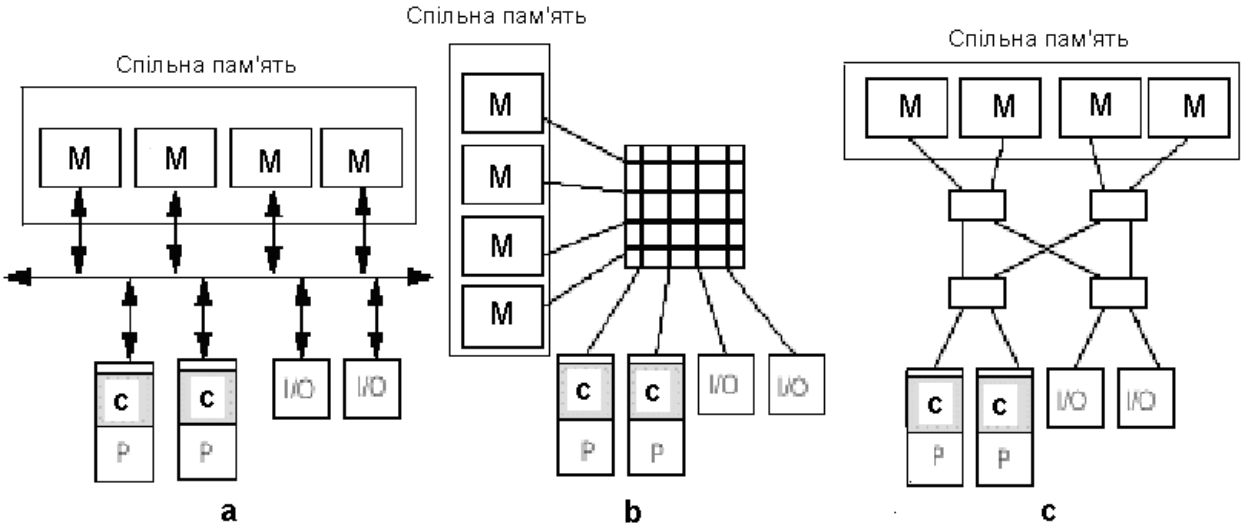

Рис.2. Багатопроцесорні системи UMA із спільною пам'яттю на основі шинної топології а), координатного комутатора b) та багатоярусної мережі с).

В граничному випадку час пересилання через шину може бути зменшений до нуля після того, як вміст кеш пам'ятей С завантажено від спільної пам'яті. Ця організація пам'яті є найпопулярнішою серед систем з спільною пам'яттю. Вона дозволяє досить просто розширити систему шляхом підключення більшої кількості процесорів. Приклади цієї архітектури - сервери Sun Starfire servers, HP V series, і Compaq AlphaServer GS. Зрозуміло, що продуктивність цієї системи обмежена часом циклу шини. Тому кожен процесор має свою кеш пам'ять, що суттєво зменшує кількість звернень до шини. Наявність багатьох кеш пам'ятей породжує проблему їх когерентності, тобто несуперечності вмісту кеш пам'яті кожного процесора із вмістом спільної основної пам'яті багатопроцесорної системи. Зазначену проблему вирішують шляхом спостереженням за шиною, що з'єднує процесори з пам'яттю, за допомогою контролера кожної кеш пам'яті разом із реалізацію в кожній кеш пам'яті наскрізного запису. Можна також в частині процесорів не використовувати кеш пам'ять.

Проблема когерентності пам'яті є однією з причин того, що багатопроцесорні системи з спільною пам'яттю на основі спільної шини мають невелику кількість процесорів. Так максимальна кількість процесорів типу Alpha 21264 в системі Compaq Alfa Server GS 140 рівна 14, процесорів PA-8500 в системі HP N9000 - 8, процесорів PoQer PC 604 в системі RS 6000 – 4 [3].

Використовувана в системі UMA комунікаційна мережа може бути одиничною чи множинною шиною, координатною мережею чи багатопортовою пам'яттю. На рис.2b та рис.2c подано типові логічні організації багатопроцесорних систем із спільною пам'яттю на основі

топології координатної комутації та багатоярусної мережі. Тут буквою С позначено локальну кеш пам'ять, I/O – пристрої введення/виведення, Р – процесори, М – модулі спільної пам'яті.

Структури на основі топології координатної комутації (рис.2a) та багатоярусної мережі (рис.2b) дозволяють вирішити проблему обмеженої пропускної здатності системи з спільною шиною. Координатний комутатор та багатоярусна мережа забезпечують множинність шляхів з'єднання процесорів та блоків пам'яті. Тому тут кількість процесорів є зазвичай більшою, ніж в системах з спільною шиною.

Потрібно відзначити, що в наведених вище багатопроцесорних системах UMA із спільною пам'яттю на основі шинної топології a), координатного комутатора b) та багатоярусної мережі c) в ролі спільної памяті з успіхом можна використати ПВД, замінивши нею крім самої пам'яті і шини, і координатний комутатор, і багатоярусну мережу. При цьому може бути отримано значний виграш як в затратах обладнання, так і в продуктивності.

У системі NUMA кожен процесор має частину спільної пам'яті (рис.3). Ця пам'ять має єдиний адресний простір. Тому, будь-який процесор може звернутися до будь-якої комірки спільної пам'яті безпосередньо, використовуючи її адресу. Проте, час доступу до модулів спільної пам'яті залежить від їх відстані від процесора. Це приводить до різного (неоднорідного) часу доступу до пам'яті від різних процесорів. Використовується кілька архітектур для підключення процесорів до модулів пам'яті в системі NUMA.

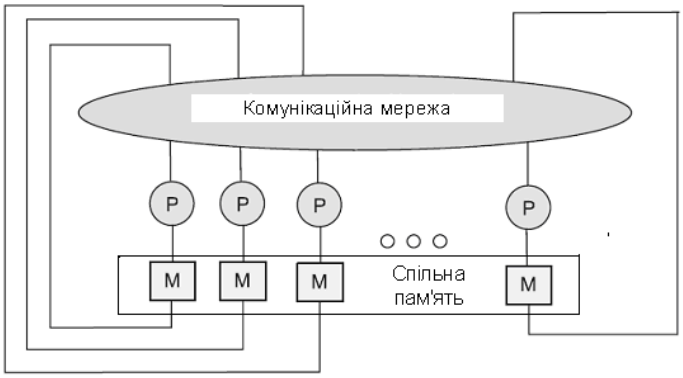

Рис.3. Система NUMA з спільною пам'яттю.

,

Як правило, система NUMA розрахована на застосування великої кількості процесорів. UMA є більш вибагливою щодо швидкодії підсистеми пам'яті. Тобто, реалізація UMA є порівняно простішою та розповсюдженішою (багатопроцесорні сервери фірми Compaqяк приклад). Архітектуру NUMA реалізовано, наприклад, в суперкомп'ютері ORIGIN 2000 фірми SGI (більш точно – тут є архітектура CC-NUMA, або NUMA із множиною когерентних кеш пам'ятей).

У симетричних багатопроцесорних комп'ютерних системах з спільною пам'яттю має місце практична межа числа їх процесорів. Ефективна схема з кеш пам'яттю зменшує навантаження на шину між процесором і основною пам'яттю, але з збільшенням числа процесорів навантаження на шину також зростає. Оскільки шина використовується також для передачі сигналів, що забезпечують когерентність, ситуація з навантаженням на шину ще більш ускладнюється. З якогось моменту в плані продуктивності шина перетворюється на вузьке місце. Для систем з спільною пам'яттю такою межею стає число процесорів в межах від 16 до 64. Наприклад, кількість процесорів системи Silicon Graphics PoQer Challenge обмежена 64 процесорами типу R10000, оскільки при подальшому збільшенні числа процесорів продуктивність падає.

Обмеження на число процесорів в архітектурі з спільною пам'яттю служить спонукальним мотивом для розвитку кластерних систем. У останніх кожен вузол має локальну основну пам'ять, тобто додатки «не бачать» спільної основної пам'яті. По суті, когерентність підтримується не стільки апаратурою, скільки програмним забезпеченням, що не кращим чином позначається на продуктивності. Одним з шляхів створення великомасштабних комп'ютерних систем є технологія CC-NUMA. Наприклад, система NUMA Silicon Graphics Origin підтримує до 1024 процесорів R10000. Технологія CC-NUMA передбачає включення множини незалежних вузлів, кожний з яких може бути, наприклад, системою із спільною пам'яттю. Таким чином, вузол містить множину процесорів, у кожного з яких присутні локальні кеш пам'яті першого і другого рівнів. У вузлі є і основна пам'ять, спільна для всіх процесорів цього вузла, але така, що розглядається як частина

спільної основної пам'яті системи. У архітектурі CC-NUMA вузол виступає основним будівельним блоком. Наприклад, кожен вузол в системі Silicon Graphics Origin містить два мікропроцесори MIPS R10000. Вузли об'єднуються за допомогою якої-небудь комунікаційної мережі, яка представлена координатним комутатором, кільцем, або має іншу топологію.

Відповідно до технології CC-NUMA, кожен вузол в системі володіє власною основною пам'яттю, але з погляду процесорів має місце спільна пам'ять, де кожен елемент будь-якої локальної основної пам'яті має унікальну системну адресу. Коли процесор ініціює доступ до пам'яті і потрібна комірка відсутня в його локальній кеш пам'яті, кеш пам'ять другого рівня процесора організує операцію вибірки. Якщо потрібна комірка знаходиться в локальній основній пам'яті, вибірка проводиться з використанням локальної шини. Якщо ж необхідна комірка зберігається у віддаленій секції спільної пам'яті, то автоматично формується запит, що посилається по комунікаційній мережі на потрібну локальну шину, і вже по ній до підключеної до даної локальної шини кеш пам'яті. Всі ці дії виконуються автоматично, прозорі для процесора і його кеш пам'яті.

У даній конфігурації головна турбота - когерентність кеш пам'ятей. Хоча окремі реалізації і відрізняються в деталях, загальним є те, що кожен вузол має довідник, в якому зберігається інформація про місцезнаходження в системі кожної складової спільної пам'яті, а також про стан кеш пам'яті.

Як видно, і в системі CC-NUMA в ролі спільної памяті можна використати ПВД, замінивши нею крім самої пам'яті і комутуючу мережу.

Подібно до NUMA, в системі COMA кожен процесор має частину спільної пам'яті (рис.4). Проте, в даному випадку спільна пам'ять є кеш пам'яттю. Система COMA вимагає, щоб дані мігрували до процесора, що запросив їх.

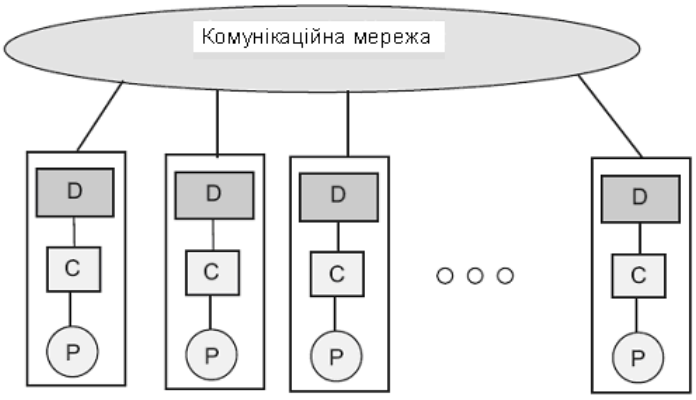

Рис.4. Система COMA з спільною пам'яттю.

Тут відсутня ієрархія пам'яті, а адресний простір визначається ємністю кеш пам'яті. Крім того, тут наявна директорія D кеш пам'яті, яка допомагає у віддаленому доступі до кеш пам'яті. Прикладом цієї архітектури є система KSR-1 фірми Kendall Square Recearch. ПВД тут доцільна на місці комунікаційної мережі.

Висновки

Таким чином, як видно з результатів аналізу місця та функцій пам'яті в комп'ютері, вона в значній мірі визначає його архітектурні особливості. Потрібно відзначити, що з розвитком комп'ютерів суттєві зміни відбулися і в будові пам'яті, значно покращилися її технічні характеристики. Разом з тим, при виконанні ресурсномістких задач потрібно забезпечувати зберігання масивів даних, що надходять з багатьох каналів, одночасно з зчитуванням раніше прийнятих масивів даних для опрацювання в багатоблокових операційних пристроях процесорів, виконувати операції реорганізації та впорядкування даних в масивах, що для існуючих типів пам'яті є доволі складною, а часто і не вирішуваною з прийнятними характеристиками задачею [4-5]. Це зумовлено їх потенційними обмеженнями, які суттєво ускладнюють організацію роботи процесора та приводять до сповільнення його роботи. Тому виникає потреба в створенні пам'яті з новими властивостями, якими не завжди володіють існуючі типи пам'яті [6]. Вимоги до пам'яті сучасного комп'ютера та основні критерії її ефективності:

- одночасний доступ до даних з багатьох портів,
- забезпечення одночасного запису даних, які поступають на її входи, та зчитування на виходи раніше записаних даних,
- здатність забезпечувати можливість виконання операцій реорганізації масивів та впорядкування даних в масивах,
- забезпечення доступу до пам'яті лише з однократним зверненням без потреби пам'ятати адресу запису даного,
- пам'ять не повинна вимагати звернення за даними як при їх записі, так і при зчитуванні,
- організація пам'яті не повинна передбачати необхідність звернення до комірки пам'яті при записі і при зчитуванні числа,
- організація пам'яті не повинна передбачати потребу зберігання інформації про місце знаходження даного в пам'яті.

1. Ashok K. Sharma, Semiconductor Memories: Technology, Testing and Reliability.- IEEE Press.- New York, 1997.
2. McClean B. The McClean Report [2001 Edition] / B. McClean, B. Matas, and T. Yancey.- IC Insights, 2001.
3. McDonald A. Architectural Semantics for Practical Transactional Memory / A. McDonald, J. Chung, B. Carlstrom, C. C. Minh, H. Chafi, C. Kozyrakis, and K. Olukotun// Proceedings of the International Symposium on Computer Architecture.- June 2006.
4. Мельник А.А. Процессоры обработки сигналов / А.А. Мельник. - Львов :ИППМ НАН Украины, 1983.
5. Ліщина Н. М. Застосування елементів інформаційних технологій до розв'язування математичних завдань / Н. М. Ліщина, В.О. Ліщина // Збірник наукових праць: проблеми педагогічних технологій. - Випуск 3.- Луцьк: Луцький інститут розвитку людини, 2011.- с. 201-205.
6. Мельник А.О. Порівняльний аналіз типів пам'яті комп'ютера / А.О. Мельник, Д.Х. Аль Равашдех // Вісник Національного університету "Львівська політехніка: комп'ютерні системи та мережі. - Львів : Львівська політехніка, 2007. - № 603.-с.81-86.