

захисту інформації, отриманої в процесі проведення досліджень та збереження конфіденційності цих даних протягом кожного етапу взаємодії з ними.

Список літератури

1. МАКС - Теоретичні відомості [Електронний ресурс] // Режим доступу: http://intranet.tdmu.edu.ua/data/kafedra/internal/informatika/classes_stud.htm
2. Ж. Бланшет, М. Саммерфилд Qt 4: Программирование GUI на C++. — М.: «КУДИЦ-ПРЕСС», 2007. — С. 648.

УДК 004.738.5

М.В. Кузьменко

Науковий керівник – Сидоренко В.В., ст. викладач
Кіровоградський національний технічний університет

Порівняльна характеристика багатопроцесорних архітектур

Часто буває, що відносно великі накладні витрати, пов’язані з обробкою переривань, незрідка роблять доцільним включення в систему додаткових процесорів. Одним з аргументів на користь створення багатопроцесорних обчислювальних систем є підвищення надійності обчислювальної системи за допомогою багатократного резервування. Якщо один з процесорів багатопроцесорної системи відмовляє, система може перерозподілити завантаження між тими, що залишилися. Для комп’ютерів перших поколінь підвищення надійності в такий спосіб часто виявлялося доцільним, особливо в додатках, що вимагали цілодобової доступності.

Іншим аргументом на користь включення в систему додаткових процесорів є той факт, що алгоритми, які використовують для вирішення багатьох прикладних завдань, незрідка піддаються “розпаралелюванню”: розділенню роботи між декількома більш менш незалежно працюючими процесорами. Залежно від алгоритму рівень досяжного паралелізму може сильно розрізнятися. Відношення продуктивності системи до кількості процесорів і продуктивності однопроцесорної машини називають “коефіцієнтом масштабування”. Для різних завдань, алгоритмів, ОС і апаратної архітектури цей коефіцієнт різний, але завжди менше одиниці і завжди зменшується в міру збільшення кількості процесорів.

Порядок доступу до пам’яті в SPARC. Сучасні процесори надають можливість управляти порядком доступу команд до пам’яті. Наприклад, біля мікропроцесорів SPARCvQ визначено три режими роботи з пам’яттю, які перемикаються бітами в статусному регістрі процесора: вільний доступ до пам’яті (RMO), частково впорядкований доступ (PSO) і повністю впорядкований доступ (TSO).

Кожен наступний режим підвищує упевненість програміста в тому, що його програма прочитає з пам’яті саме те, що туди записав інший процесор, але одночасно наводить і до падіння продуктивності. Найбільший програш забезпечує реалізація режиму TSO, коли ми просто вимикаємо і динамічне переупорядковування команд, і кешування даних.

Іншим вузьким місцем багатопроцесорних систем є системна шина. Системи шинної архітектури прості в проектуванні і реалізації, до них легко підключати нові пристрої, тому така архітектура набула широкого поширення. Проте, особливо в багатопроцесорних системах, шина часто є одним з основних обмежувачів

продуктивності. Підвищення пропускної спроможності шини частенько можливо, але наводить до підвищення загальної вартості системи.

Втім, при великій кількості вузлів проблеми виникають і біля систем з настільки високошвидкісною шиною, як FinePane. Крім того, у міру зростання фізичних розмірів системи, стає необхідно брати до уваги фізичну швидкість передачі сигналів – як сигналів самої магістралі, так і запитів до арбітра шини і його відповідей. Тому шинна топологія з'єднань при багатьох десятках і сотнях вузлів виявляється недопустима, і застосовуються складніші топології.

Системи NUMA-Q. Багатопроцесорні сервери IBM NUMA-Q складаються з окремих процесорних модулів, кожен з яких має власну оперативну пам'ять і чотири процесори x86. Модулі сполучені високошвидкісними каналами IQ-Link з центральним комутатором. Заміна загальної шини на зіркоподібну топологію з центральним комутатором дозволяє вирішити проблеми арбітражу доступу до шини, зокрема, усунути затримки при запиті до арбітра шини і чеканні його відповіді пристрою, який подає запит.

При більшому числі модулів застосовується гіперкубічна топологія, коли кожен вузол зазвичай також містить декілька процесорів і власну оперативну пам'ять.

Завдяки множинності доріг, маршрутизатори можуть вибирати для кожного повідомлення найменш завантажену в даний момент дорогу або обходити вузли, що відмовили.

Масивно-паралельні системи Cray/SGI Origin. Вузли суперкомп'ютерів сімейства Cray/SGI Origin сполучені в гіперкуб каналами з пропускною спроможністю 1 Гбайт/с. Адаптери з'єднань забезпечують не просто обмін даними, а прозорий доступ процесорів кожного з вузлів до оперативної пам'яті інших вузлів і забезпечення когерентності процесорних кешів.

Відмінність в швидкості доступу до локальної пам'яті процесорного модуля і інших модулів є проблемою і при невдалому розподілі завантаження між модулями приведе до значного зниження продуктивності системи. Відомо два основних рішення цієї проблеми:

1. COMA (Cache Only Memory Architecture) – архітектура пам'яті, при якій робота з нею відбувається як з кешем. Система переносить сторінки пам'яті, з якою даний процесорний модуль працює частіше за інших, в його локальну пам'ять.

2. CC-NUMA (Cache-Coherent Non-Uniform Memory Access). У цій архітектурі адаптери між модульних з'єднань забезпечуються власною кеш-пам'яттю, яка використовується при зверненнях до ОЗУ інших модулів. Основна діяльність центрального комутатора і каналів зв'язку полягає в підтримці когерентності цих кешів.

Зрозуміло, що ці архітектури не вирішують в корені проблеми неоднорідності доступу: для обох можливо побудувати таку послідовність міжпроцесорних взаємодій, яка “проміє” всі кеші і перезавантажить міжмодульні зв'язки, а в разі COMA приведе до постійного перекачування сторінок пам'яті між модулями. Те ж саме справедливо і для симетричних багатопроцесорних систем із загальною шиною.

Можна лише підкреслити, що масштабованість багатопроцесорних систем визначається в першу чергу природою завдання і рівнем паралелізму, закладеним у використаний для вирішення цього завдання алгоритм. Різні типи багатопроцесорних систем і різні топології міжпроцесорних з'єднань придатні і оптимальні для різних завдань.

Список літератури

1. Жаркова А.И. Многопроцессорные системы. – М.: 2004 – 173 с.
2. Богданов А.В. Архитектура и топологии многопроцессорных вычислительных систем. Уч. изд. Интернет-Университет Информационных Технологий: 2005. – 176 стр.