

МЕТОДЫ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ОБМЕНА ДАННЫМИ В МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМАХ

Предлагаются методы ускорения обмена данными в мультипроцессорных системах с общей магистралью, основанные на множественном оконном доступе к локальной памяти процессоров. Рассматривается возможность синхронизации процедур обмена по шине данных и шине управления.

Methods of data transaction acceleration in global-bus multi-processor systems based on multiple window processor local memory access are proposed. Possibility of transaction synchronization through data and control buses is considered.

Введение

Широкое внедрение автоматических систем управления техническими комплексами и технологическими процессами требует ускоренного проектирования вычислительных средств для решения различных по сложности задач обработки информации в режиме реального времени.

Перспективным направлением создания промышленного ряда систем для управления различными объектами и технологическими процессами является использование масштабируемых параллельных вычислительных систем, построенных на однотипных процессорных модулях.

Модульный принцип построения программных и аппаратных средств позволяет ускорить разработку систем и снизить их стоимость. При таком подходе к организации систем путем масштабирования может быть обеспечена необходимая производительность систем, которая определяется конкретными условиями применения.

Возможность масштабирования наиболее эффективно реализуется на основе магистральной топологии, когда в качестве коммуникационной среды для обмена информацией между процессорными модулями используется общая магистраль. Изменение числа процессорных модулей в этом случае не требует модификации коммуникационной среды системы.

Время обработки информации в параллельных системах зависит не только от эффективности распараллеливания задачи, скорости обработки параллельных ветвей, но и от эффективности процедур обмена данными между процессорными модулями.

Следовательно, важной проблемой повышения эффективности параллельной обработки информации является создание методов и средства ускорения обмена данными между процессорными модулями в мультипроцессорных системах с общей магистралью.

Архитектура систем с общей магистралью

Наиболее простыми с точки зрения технической реализации являются мультипроцессорные системы с общим адресным пространством [1-7]. Системы с такой архитектурой, построенные на базе одинаковых процессоров, то есть являются симметричными мультипроцессорами (symmetric multiprocessor) или SMP-системами. Общая память используется не только для хранения системной информации и обмена между процессорами, но и для хранения программ процессоров. Процессоры обращаются к памяти за командами, что приводит к непроизводительным затратам времени на ожидание доступа и разрешение конфликтов. Наличие в каждом процессорном модуле кэш-памяти уменьшает число конфликтов, однако в этом случае возникают трудности с обеспечением когерентности данных, что усложняет взаимодействие процессоров.

В системах с такой организацией памяти используется небольшое число процессоров (например, SMP-серверы обычно строятся на основе 2, 4 или 8 процессоров), то есть возможность масштабирования весьма ограничена.

Уменьшение числа конфликтных ситуаций может быть достигнуто за счет использования в каждом процессорном модуле собственной локальной памяти, в которой хра-

нится программа данного процессора [2, 5, 8]. Процессор взаимодействует со своей памятью через локальную магистраль, а с общей памятью через системную (общую) магистраль. Такие системы относятся к классу систем с неоднородным доступом к памяти (NUMA-системы).

Для ускорения обмена данными между процессорами в системах может быть предусмотрен непосредственный доступ каждого процессора к определенной области адресного пространства локальной памяти других процессоров, называемой коммуникационной памятью [8]. За счет организации интерфейса, доступ к такой памяти возможен как со стороны локальной, так и со стороны системной магистрали.

К недостаткам систем с коммуникационной памятью следует отнести то, что с увеличением числа процессорных модулей все больший объем адресного пространства локальной памяти каждого процессора выделяется на коммуникационную память. Сокращение объема индивидуальной локальной памяти в каждом процессорном модуле является одним из важных факторов, ограничивающих число процессоров в системе.

Для устранения указанного недостатка может быть использован оконный механизм доступа к локальной памяти каждого процессора со стороны системной магистрали [2]. Устройство доступа, реализующее механизм окна, подключается между системной и локальной магистралями. В адресном пространстве общей памяти для каждой локальной памяти выделяется адрес для регистра адреса и регистра данных устройства доступа. В регистр адреса записывается начальный адрес локальной памяти, к которой осуществляется доступ со стороны системной магистрали. Передача данных (чтение или запись) осуществляется через регистр данных. При этом адрес в регистре адреса модифицируется (например, прибавляется или вычитается константа).

Для сравнительной оценки систем с разной архитектурой на функциональном уровне будем учитывать коэффициент эффективности использования системной магистрали, который определяется по формуле

$$K_{\text{CM}} = \frac{N_S + N_D}{M}, \quad (1)$$

где N_S – число обращений к системной магистрали для инициализации и синхронизации процедур обмена (непроизводительные затраты времени); N_D – число обращений к системной магистрали для передачи непосредственно данных; M – число передаваемых слов.

Коэффициент K_{CM} определяет среднее число обращений к системной магистрали для передачи одного слова между компонентами системы. При равной скорости обработки параллельных ветвей алгоритмов система с меньшим значением K_{CM} будет затрачивать меньше времени на решение задач за счет более быстрого межпроцессорного обмена.

Определение K_{CM} в общем случае является нетривиальной задачей и требует разработки программ (если известна система команд) или, по крайней мере, алгоритмов синхронизации и пересылки данных на этапе проектирования системы.

Системы с оконным доступом позволяют эффективно использовать адресное пространство при обмене данными между процессорными модулями, но требуют непроизводительных затрат времени на подготовку обмена. К ним относятся затраты времени на синхронизацию обмена, захвата и инициализацию окна. Для синхронизации применяют флажки в локальной памяти процессоров, а для захвата окна (ограждения от других процессоров на время пересылки данных) используют семафоры в общей памяти.

Исследуя алгоритм обмена данными для систем с оконным доступом в соответствии с (1) можно получить

$$K_{\text{CM}} = 1 + \frac{N_{\text{OK}} + 4}{M}, \quad (2)$$

где N_{OK} – суммарное число обращений к магистрали для проверки и захвата семафоров каждым из процессоров.

При большом числе процессоров и большой связности графа обмена данными N_{OK} может достигать больших значений и, следовательно, существенно влиять на время решения задач.

Ниже показана возможность сокращения непроизводительных затрат времени обмена

данными для систем с оконным доступом к памяти.

Методы ускорения обмена данными

Для повышения эффективности межпроцессорного обмена в децентрализованных системах можно применить механизм множественного оконного доступа к локальной памяти.

Для описания систем будем использовать MSBC-модель [9], которая учитывает распределение памяти и особенности доступа к адресному пространству процессоров.

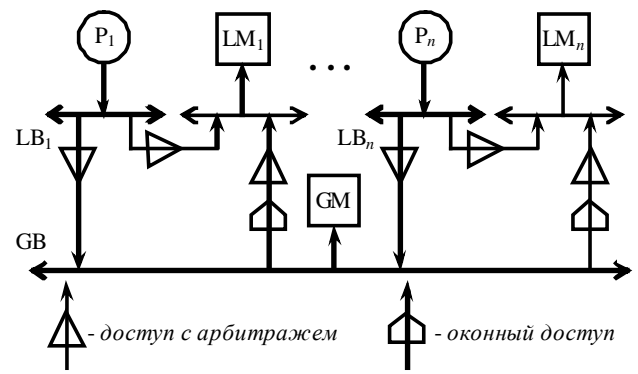
Организация системы на функциональном уровне показана на рис. 1. В адресном пространстве общей памяти GM каждому процессору P_i выделено по два адреса для обращения к каждой локальной памяти: WA_{ij} – адрес регистра начального адреса массива, WD_{ij} – адрес регистра данных (j – номер локальной памяти LM_j , к которой осуществляется обращение).

К локальной памяти LM_i подключено одно окно с множественным доступом, на которое выделяется $W=2(n-1)$ адресов в общей памяти системы. Число окон с множественным доступом в системе равно числу n процессоров. Укрупненная структурная схема окна с множественным доступом показана на рис. 2.

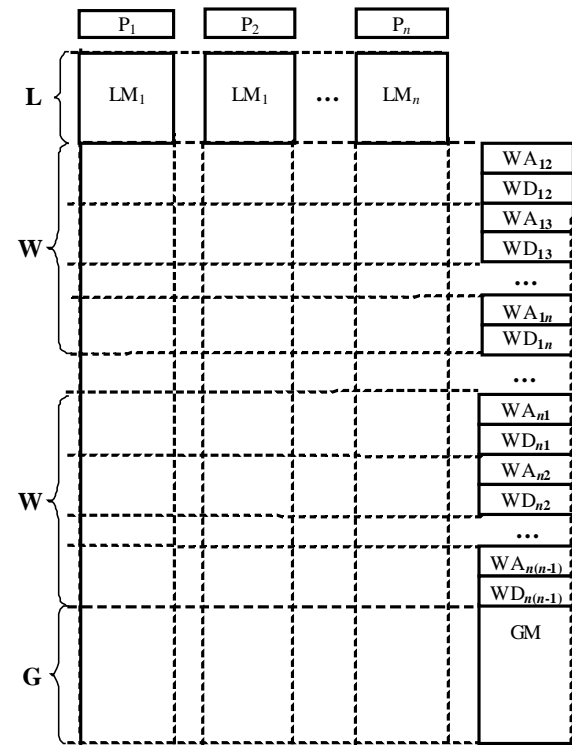
Регистры начальных адресов образуют блок памяти начальных адресов (ПНА), выполненный в виде СОЗУ.

Перед началом обмена с процессором P_j процессор P_i записывает по адресу WA_{ji} начальный адрес локальной памяти LM_j , с которого начинается пересылка массива. Обмен выполняется посредством стандартных циклов обращения процессора P_i к WD_{ij} .

Данные при этом пересылаются через блок данных БД. При каждом обращении соответствующий адрес в ПНА увеличивается или уменьшается, что автоматически делает доступным очередной адрес LM_j . Аналогичным образом с LM_j могут одновременно и без предварительного захвата окна работать и любые другие процессоры, так как возникающие конфликты обращения к системной магистрали разрешаются стандартными аппаратными средствами арбитража.



а)



б)

Рис. 1. Граф управляющих потоков (а) и карта распределения памяти (б) системы с множественным оконным доступом

Доступ к локальной памяти процессоров не требует предварительного захвата окон. Синхронизацию можно осуществить с помощью флажков, находящихся в локальной памяти процессоров. Процессор, который подготовил в своей локальной памяти массив данных для пересылки настраивает окно процессора, который должен принять данные, захватывая магистраль на один цикл. В следующем цикле он производит установку флажка. Проверка флажка осуществляется процессором в своей локальной памяти, то есть системная магистраль для этого не используется.

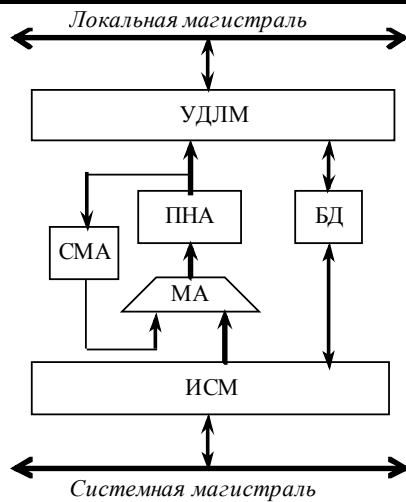


Рис. 2. Устройство множественного доступа: УДЛМ – устройство доступа к локальной магистрали; ИСМ – интерфейс системной магистрали; БД – буфер данных; ПНА – память начальных адресов; СМА – схема модификации адресов; МА – мультиплексор адреса.

Если флажок установлен, процессор настраивает необходимое окно (за один цикл), а затем пересылает массив данных в свою локальную память. Для пересылки одного слова системная магистраль захватывается один раз. Завершив пересылку массива, процессор устанавливает флажок в памяти другого процессора. Если флажок записывается по адресу, который автоматически установился в окне, то для этого нет необходимости лишней раз выполнять инициализацию окна. При такой организации обмена $N_S=4$. Действительно, магистраль два раза используется для настройки окна и два раза для установки флажков. Для пересылки данных к магистрали необходимо обратиться $N_D=M$ раз.

С учетом (1) коэффициент эффективности использования магистрали можно записать как

$$K_{CM}(1) = 1 + \frac{4}{M}. \quad (3)$$

Сравнивая выражения (2) и (3), видно, что предложенный метод повышает скорость межпроцессорного обмена по сравнению с простым оконным доступом, поскольку отпадает необходимость захвата окон. С увеличением длины передаваемого массива среднее число обращений к системной магистрали для пересылки одного слова стремится к минимальному значению – одному обращению. Однако при пересылке небольших

массивов информации непроизводительные расходы в большей степени сказываются на скорости обмена.

Уменьшить непроизводительные расходы можно с помощью следующего метода синхронизации обмена по шине управления. Организация системы поясняется рис. 3.

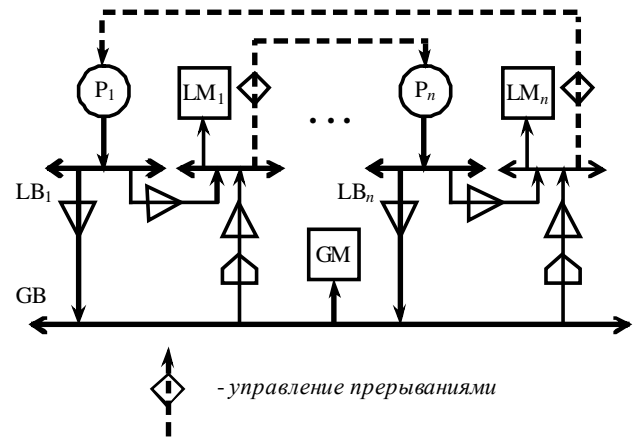


Рис. 3. Система с синхронизацией обмена по шине управления

Для установки флажков синхронизации системная магистраль в данном случае не используется. Процессор P_i для установки флажка синхронизации в локальной памяти процессора P_j формирует сигнал требования прерывания для этого процессора. Подпрограмма обработки соответствующего прерывания в процессоре P_j устанавливает (сбрасывает) флажок в своей локальной памяти. Состояние флажка проверяется в нужный момент времени основной программой обмена данными. Аналогично процессор P_j изменяет состояние флажка в локальной памяти процессора P_i .

Для формирования сигнала требования прерывания выделяются адреса в локальной памяти процессоров. Запись слова по данному адресу через соответствующий интерфейс запускает контроллер прерывания, который реализует процедуру инициализации прерывания.

Таким образом, для пересылки массива только один раз используется системная магистраль для инициализации окна, после чего непосредственно пересылается массив данных. Следовательно, для данного метода получим

$$K_{CM}(2) = 1 + \frac{1}{M}. \quad (4)$$

Очевидно, что количество обращений к системной магистрали в данном случае меньше, чем для предыдущего метода.

Сравнение эффективности методов

При сравнении выражений (2), (3) и (4), очевидно, что предложенные методы выигрывают в эффективности обмена по сравнению с известным, поскольку при большой связности графа межпроцессорного обмена $N_{ок}$ в выражении (2) может достигать больших значений.

Сравнивая предложенные методы (см. табл. 1), можно сделать вывод, что синхронизация по шине управления более эффективна при передаче между процессорами небольших массивов данных (примерно до 32 слов). Действительно, для передачи массива из 8 слов системная магистраль используется меньше приблизительно на 30%, а для 16 слов – на 18%.

С увеличением длины передаваемых массивов различия между методами относитель-

но использования системной магистрали становятся незначительными. В этом случае следует отдать предпочтение первому из предложенных методов, поскольку его реализация требует меньших аппаратных затрат.

Табл. 1. Сравнение методов

M	4	8	16	32
$K_{см}(1)/K_{см}(2)$	1,6	1,33	1,18	1,1

Выводы

Предложенные методы обмена данными, как показано выше, обеспечивают уменьшение числа обращений к системной магистрали. Это создает предпосылки для ускорения вычислений и, как следствие, расширения области применения мультипроцессорных систем, повышения качества управления и моделирования в режиме реального времени.

Список литературы

1. Белецкий В.Н. Многопроцессорные и параллельные структуры с организацией асинхронных вычислений. – К.: Наукова думка, 1988. – 240 с.
2. Валях Е. Последовательно-параллельные вычисления. – М.: Мир, 1985. – 456 с.
3. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2002. – 608 с.
4. Головкин Б.А. Параллельные вычислительные системы. – М: Наука, 1980. – 519 с.
5. Корнеев В.В. Параллельные вычислительные системы. – М.: “Нолидж”, 1999. – 320 с.
6. Майерс Г. Архитектура современных ЭВМ: В 2-кн. Кн. 1. Пер. с англ. – М.: Мир, 1985. – 364 с.
7. Мультипроцессорные системы и параллельные вычисления./ Под. ред. Ф.Г.Энслоу. – М.: Мир, 1976. – 383 с.
8. Siewiorek D. P. A Case Study of C. mmp, Cm* and C.vmp, Part I: Experience with Fault-Tolerance in Microprocessor Systems // Proceedings of the IEEE. – 1978. – Vol. 66, No. 10. – P. 1178-1199.
9. Жабин В.И. Графическое описание архитектуры вычислительных систем // Вісник Національного технічного університету України “Київський політехнічний інститут”. Інформатика, управління та обчислювальна техніка. – К: “БЕК+”, 2001. – № 36. – С. 80–88.

Поступила в редакцию 8.12.2009