

УДК 004.272.43

СОВРЕМЕННАЯ КЛАССИФИКАЦИЯ МНОГОПРОЦЕССОРНЫХ СИСТЕМ

Мартенс-Агюшев Д.С., Мартышкин А.И.

Пензенский государственный технологический университет, Пенза, e-mail: Alexey314@yandex.ru

В статье приведена современная классификация многопроцессорных систем с привязкой к классификации по Флинну. Отмечено, что в классе MIMD имеется подкласс мультипроцессоров, который делится на три типа: UMA, NUMA, COMA. Различие данных типов состоит в механизме доступа к памяти (с общей и разделяемой памятью). Приводится подробное описание указанных типов мультипроцессоров с выделением их достоинств и недостатков, а также с указанием допустимой масштабируемости этих систем. Кроме того, в одной из серии процессоров фирмы AMD используется такая архитектурная организация, которая сочетает преимущества UMA и NUMA систем, что обеспечивает возможность реализации гибких и хорошо масштабируемых мультипроцессорных систем. В такой системе для каждого процессора предусмотрен контроллер памяти. Также разработана специализированная шина с высокой пропускной способностью HyperTransport (HT), которая организует обмен между процессором и памятью без значительных задержек. В результате компания AMD назвала эту архитектуру SUMA. Далее приводится описание контроллера памяти, являющегося, по сути, буфером между памятью и процессором. Этот блок должен обеспечивать равномерную загрузку общей шины, исключая возможности ее перегрузки и перекосов работы системы. В конце работы приводятся основные результаты.

Ключевые слова: мультипроцессор, архитектура памяти, пропускная способность, быстродействие, классификация

MODERN CLASSIFICATION OF MULTIPROCESSOR SYSTEMS

Martens-Atyshev D.S., Martyshkin A.I.

Penza State Technological University, Penza, e-mail: Alexey314@yandex.ru

The article presents a modern classification of multiprocessor systems with reference to Flynn classification. It is noted that the MIMD class has a subclass of multiprocessors, which is divided into three types: UMA, NUMA, COMA. The difference between these types is the mechanism of memory access (shared and shared memory). A detailed description of these types of multiprocessors with the allocation of their advantages and disadvantages, as well as indicating the allowable scalability of these systems. In addition, one of the AMD processor series uses an architectural organization that combines the advantages of UMA and NUMA systems, which provides the ability to implement flexible and highly scalable multiprocessor systems. Such a system provides a memory controller for each processor. A dedicated high-bandwidth HyperTransport (HT) bus has also been developed, which enables the exchange between the processor and memory without significant delays. As a result, AMD named this architecture SUMA. The following is a description of the memory controller, which is essentially a buffer between memory and the processor. This unit should ensure uniform loading of the common bus, eliminating the possibility of its overload and distortion of the system. At the end of the article, the authors give the main results.

Keywords: multiprocessor, memory architecture, bandwidth, performance, classification

Согласно классификации по Флину [1] мультипроцессорные системы (МПС) относятся к классу MIMD (Multiple Instruction-stream Multiple Datastream – несколько потоков команд с несколькими потоками данных). В классе MIMD имеется подкласс мультипроцессоров, который делится на три типа:

1. UMA (Uniform Memory Access) – однородный доступ к памяти, когда все процессоры имеют связь со всеми модулями памяти;

2. NUMA (NonUniform Memory Access) – неоднородный доступ к памяти, у каждого процессора имеется промежуточная память (например кэш);

3. COMA (Cache Only Memory Access) – доступ только к кэш-памяти, любой из имеющихся процессоров содержит в своем составе локальную кэш-память большой емкости.

Различие данных типов состоит в механизме доступа к памяти, машины с полностью доступной памятью (рис. 1) и машины с локальной памятью (рис. 2).

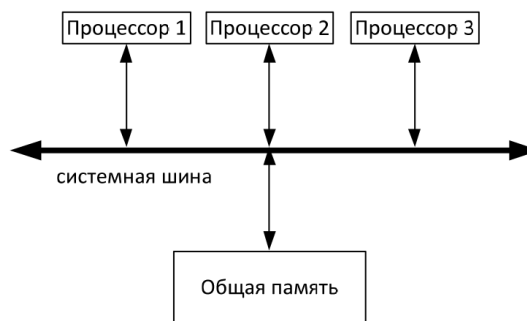


Рис. 1. МПС с общей памятью

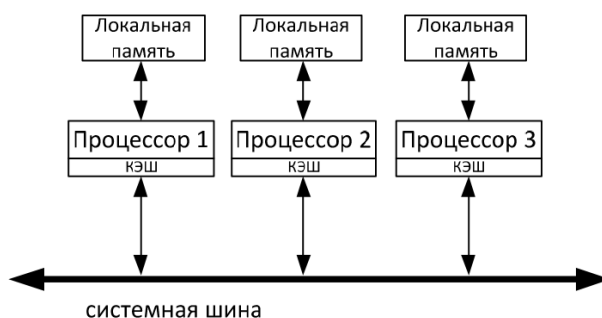


Рис. 2. МПС с распределенной памятью

Подобное разбиение на подклассы весьма удобно, так как в крупных мультимикропроцессорных системах память часто делится на несколько модулей. Для UMA-машин, любой процессор имеет одинаковое время доступа к каждому модулю памяти. То есть, слово извлекается из памяти с той же скоростью, что и любое другое слово. Если подобный процесс невозможен, то быстрые операции замедляются, чтобы соответствовать самым медленным, что соответствует принципу – «однородный» доступ. Подобная однородность позволяет контролировать производительность, что немало важно для создания эффективных программ [2].

В NUMA-машинах, вышеописанная однородность отсутствует. Так как каждый процессор имеет модуль памяти, который размещен рядом, относительно других модулей, соответственно доступ к этому модулю памяти производится намного быстрее, чем к другим. Обращение к СОМА-машинам также является неоднородным.

UMA-мультимикропроцессоры. Наиболее популярный вид UMA-мультимикропроцессоров имеют одну общую шину [3, 4]. От двух и более микропроцессоров и от одного и нескольких модулей памяти используют эту шину для обмена данными. Этот процесс выполняется следующим образом, процессору для считывания слова из памяти, необходимо проверить, что шина не занята. Если на шине не наблюдается обмен данными, процессор отправляет адрес необходимого слова на шину, устанавливает некоторые управляющие сигналы и ожидает реакцию памяти, которая выдает данные на шину по запрошенному адресу. В случае занятости шины, процессор находится в режиме ожидания до тех пор, пока шина не освободится. В данной схеме имеется существенный недостаток. Когда используется от двух до трех процессоров, контролировать доступ к шине не составляет труда, сложности

появляются, при количестве процессоров 32 или 64. Производительность системы в данном варианте в основном характеризуется пропускной способностью шины, и большинство процессоров вынуждены значительное время простаивать.

NUMA-мультимикропроцессоры. Число процессоров в UMA-мультимикропроцессорах с одиночной шиной как правило ограничивается от 16 до 32, а для того чтобы использовать в одной мультимикропроцессорной системе более 100 процессоров, необходимо использовать другое решение. В архитектуре UMA считается, что все модули памяти организуют доступ за одинаковое время. Если изменить подобную организацию доступа, то получаем концепцию мультимикропроцессоров с неоднородным доступом к памяти (NonUniform Memory Access, NUMA).

Как в архитектуре UMA, мультимикропроцессоры NUMA имеют общее адресное пространство для каждого процессора, но с той разницей, что доступ к локальным модулям памяти выполняется быстрее, чем к удаленным [3, 4]. Соответственно, все UMA-программы будут выполняться на NUMA-машинах, но с худшей производительностью, чем на UMA-машине с аналогичной тактовой частотой. В системах NUMA есть три основные характеристики, которые позволяют различать NUMA-мультимикропроцессоры от иных мультимикропроцессоров[2]: имеется общее адресное пространство, доступное для всех процессоров; обращение к удаленной памяти выполняется командами LOAD и STORE; обращение к удаленной памяти производится медленнее, чем обращение к локальной.

Время обращения к удаленной памяти, которое не замаскировано кэшированием (нет кэш), то подобная система имеет название NC-NUMA (No Caching NUMA – NUMA без кэширования). При наличии согласованных кэш, система будет иметь название CC-

NUMA (Coherent Cache NUMA – NUMA с согласованными кэшами). Разработчики вычислительных систем обычно называют подобную систему **аппаратной распределенной общей памятью**. Данный мультипроцессор включает в себя набор процессоров, где каждый имеет свою память, доступ к которой выполнялся по локальной шине. Также, процессоры соединены друг с другом системной шиной. При организации доступа к памяти, запрос направляется к диспетчеру памяти, который определяет, имеется ли необходимое слово в локальной памяти или нет. Если да, то запрос отправляется по локальной шине, если нет, запрос отправляется по системной шине в ту систему, в которой имеется вышеуказанное слово. Исходя из этого, для второго процесса запроса необходимо намного больше времени, чем для первого. Соответственно обработка программы, размещенной, в удаленной памяти, занимало в 10 раз больше времени, чем обработка той же программы, размещенной локально.

Так как в NC-NUMA-машине нет кэш-памяти, то непосредственно обеспечивается согласованность памяти. Если каждое слово памяти имеет возможность храниться только в одном месте, следовательно, отсутствует ситуация наличия копии с устаревшими данными.

CC-NUMA-мультипроцессоры. Один из часто используемых принципов к построению сложных мультипроцессоров, принадлежащих к системам CC-NUMA, применен в мультипроцессоре на основе каталога. Главный смысл заключается в хранении базы данных с информацией о том, где располагается каждая строка кэша и ее статус. При доступе к строке кэша в базу данных отправляется запрос о нахождении строки и является она «чистой» или «грязной» (модифицированной). Т.к. опрос базы данных необходимо выполнять при исполнении любой команды доступа к памяти, следо-

вательно, необходимо специализированное аппаратное обеспечение для поддержки базы данных, которое может обработать запрос за доли шинного цикла.

СОМА-мультипроцессоры. Исходя из того, что доступ к удаленной памяти обрабатывается намного медленнее, чем к локальной в NUMA- и CC-NUMA-машинах, необходимо иное архитектурное решение. Данным решением на сегодня является система, в которой основная память каждого процессора применяется в качестве кэш-памяти. Название этой системы СОМА (Cache Only Memory Access – **доступ только к кэш-памяти**). В СОМА-машине физическое адресное пространство разделяется на строки кэша, которые по запросу свободно передаются в системе. Для каждого блока памяти нет своей машин. Память, которая при необходимости привлекает строки, называется **притягивающей**. Применение основной памяти в качестве емкого кэша увеличивает процент кэш-попаданий, следовательно, и производительность. В СОМА-машинах имеется два проблемных вопроса, каким образом располагаются строки кэша и что делать, когда удаляемая из памяти строка является конечной копией.

SUMA-мультипроцессоры. В одной из серии процессоров фирмы AMD применяется архитектурная организация, сочетающая преимущества UMA и NUMA систем, что дает возможность реализовывать гибкие и масштабируемые мультипроцессорные системы. В подобной системе (рисунок 3) для каждого процессорного узла встраивается контроллер памяти, также разработана специализированная шина с высокой пропускной способностью HyperTransport (HT), которая организует обмен между процессором и памятью без значительных задержек. В результате компания AMD назвала эту архитектуру SUMA (Slightly Uniform Memory Architecture – почти однородная память) [4, 5].

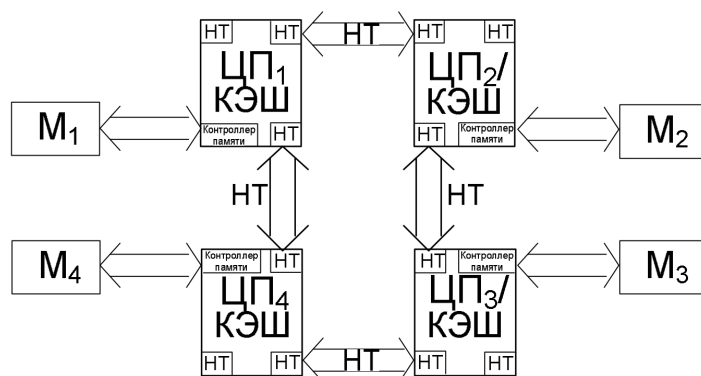


Рис. 3. Структура организации МПС типа SUMA

Главное достоинство архитектуры SUMA это последовательная межпроцессорная шина HyperTransport, для связи процессоров по типу точка-точка применяется одна часть соединений HT, другая часть необходима для связи с периферийными устройствами. Изначально AMD проектировала шину HyperTransport для архитектуры чипов типа AMD64, специализированных процессоров со встроенным контроллером памяти. Шина HyperTransport гарантирует пропускную способность сопоставимую с пропускной способностью оперативной памяти, а также наименьшие задержки при обмене данными по шине.

Контроллер памяти. Для предотвращения конфликтов на общей шине и организации эффективной работы подсистемы «процессор-память» применяют различные аппаратные решения, называемые контроллерами памяти или буферными устройствами. Во время непрерывного выполнения операции (транзакции) запись или чтения один из процессоров захватывает шину до тех пор, пока обработка транзакции не будет завершена. Следовательно, шина и процессоры переходят в режим ожидания, пока память не выполнит физические операции чтения или записи. Таким образом, циклы шины, которые могли быть обработаны другими ЦП, теряются. Чтобы повысить пропускную способность общей шины и минимизировать временные потери необходимо поддерживать для шины режимы расщепления транзакций чтения и буферизации транзакций записи. Операция физического чтения выполняется в памяти автономно под управлением контроллера памяти, по окончании операции физического чтения контроллер памяти сигнализирует запрашивающий процессор о готовности данных. В ответ процессор снова запрашивает общую шину и читает слово данных из контроллера памяти. Буферизация операций записи происходит следующим образом: процессор выставляет на общую шину адрес ячейки памяти и данные которые необходимо записать. Они записываются в регистрах контроллера памяти, после этого процессор отключается от шины, так как ответ от памяти в данном случае не нужен. Операция физической записи в память выполняется под управлением контроллера памяти.

Некоторыми из основных задач, при проектировании многопроцессорных вычислительных систем с общей шиной, являются разрешение конфликтных ситуаций во время доступа процессоров к памяти, уменьшение задержек на шине при выполнении операций обмена данными, а также масштабируемость системы. При построе-

нии МПС применяются различные методы доступа в подсистему «процессор-память». Анализ применяемых методов доступа к памяти обозначил ряд имеющихся проблем. Методы, использующие общую память, реализуются в виде архитектур типа UMA, а те системы, которые используют принцип разделения памяти, находят применения в архитектурах типа NUMA и ее модификаций. Первый метод позволяет процессорам за равный промежуток времени предоставить доступ к каждому модулю памяти. Таким образом, возможен контроль над производительностью системы, но если процессоров более трех, то возникает проблемы урегулирования доступа к общей шине. Второй метод, позволяет регулировать масштабируемость системы, но из-за неоднородности памяти, доступ к удаленной памяти намного медленнее, чем к локальной, что понижает производительность системы. Исходя из этого, могут возникать значительные задержки при обращении к памяти, простои или отказы в работе процессоров при предельных нагрузках и/или ограниченном времени ожидания. Однако, практическое решение проблем доступа в подсистему «процессор-память» является достаточно сложным и имеет неоднозначный ответ. В данной области необходимо учитывать, что в реальных системах, имеются ограничения на количество процессоров и систем коммутации в шине обмена процессора с памятью, связанную с дороговизной реализации. Задача разработки методов доступа в подсистему «процессор-память», особенно как для архитектур с общей памятью, так и с разделяемой памятью связанных со сбором, обработкой и передачей информации с многочисленных источников является чрезвычайно актуальной и важной.

Список литературы

1. Воеводин В.В., Капитонова А.П. Методы описания и классификации архитектур вычислительных систем. – М.: Изд-во Моск. университета, 1994. – 79 с.
2. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. – СПб.: Питер, 2013. – 816 с.
3. Мартышкин А.И. Разработка и исследование разомкнутых моделей подсистемы «процессор-память» многопроцессорных вычислительных систем архитектур UMA и NUMA // Вестник Рязанского государственного радиотехнического университета. – 2015. – № 54-1. – С. 121-126.
4. Martyshkin A.I. Development and Research of open-loop Models the subsystem «processor-memory» of Multiprocessor systems architectures UMA, NUMA and SUMA // ARPN Journal of Engineering and Applied Sciences. – 2016. – Т. 11. – № 23. – PP. 13526-13535.
5. Мартышкин А.И. Разработка и исследование математических моделей подсистемы «Процессор-память» многопроцессорных вычислительных систем архитектур NUMA и SUMA на разомкнутых сетях массового обслуживания // Вестник Рязанского государственного радиотехнического университета. – 2016. – № 55. – С. 55-63.