СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
SIBERIAN FEDERAL UNIVERSITY

ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

Учебное пособие

ИНСТИТУТ КОСМИЧЕСКИХ
И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

УДК 004.272(07)
ББК 32.971.35-02я73
П180

Р е ц е н з е н т ы:

М. Н. Фаворская, доктор технических наук, профессор, зав. кафедрой информатики и вычислительной техники ФГБОУ ВО «Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева»;

В. А. Хабаров, кандидат технических наук, ведущий инженер-конструктор отдела систем управления АО «НПП Радиосвязь»

П180 **Параллельные вычислительные системы** : учеб. пособие / Н. Ю. Сиротинина, О. В. Непомнящий, К. В. Коршун, В. С. Васильев. – Красноярск : Сиб. федер. ун-т, 2019. – 178 с.
ISBN 978-5-7638-4180-0

Рассмотрены архитектуры параллельных вычислительных систем, общие вопросы их применения и проблемы программирования, особенности организации вычислительных процессов и инструментальные средства разработки прикладного параллельного программного обеспечения для мультипроцессорных и мультикомпьютерных параллельных вычислительных систем. Пособие также включает курс практических работ, позволяющих освоить средства параллельного программирования.

Предназначено для бакалавров по направлению подготовки 09.03.01 «Информатика и вычислительная техника» при изучении дисциплины «Параллельные вычислительные системы». Может использоваться в качестве основной и дополнительной литературы при изучении дисциплин «Параллельное программирование», «Современные вычислительные системы» и пр.

Электронный вариант издания см.:
http://catalog.sfu-kras.ru

УДК 004.272(07)
ББК 32.971.35-02я73

ISBN 978-5-7638-4180-0

ОГЛАВЛЕНИЕ

1. ПОНЯТИЕ ПАРАЛЛЕЛИЗМА. АРХИТЕКТУРЫ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Практически в любом учебнике или научной статье, посвященной вычислительной технике, можно встретить слова «интенсивно развивающаяся область», «постоянное совершенствование вычислительной техники» и подобные. Как правило, основной характеристикой вычислительной техники, свидетельствующей о ее развитии, считается производительность современных вычислительных систем. Схожий смысл имеют понятия: быстродействие, вычислительная мощность и пр.

Производительность ЭВМ – характеристика, описывающая скорость выполнения вычислений [1].

Если рассмотреть динамику роста производительности электронных вычислительных машин, можно отметить следующую закономерность: скорость выполнения арифметических операций возрастает приблизительно в 10 раз за каждые 5 лет [2]. За счет чего это происходит?

Первый вариант ответа, который напрашивается сам собой: за счет повышения быстродействия элементной базы.

Действительно, в первом поколении вычислительных машин в качестве коммутирующих элементов использовались электронные лампы с временем задержки на вентиле порядка 1 мкс. В начале 60-х г. XX в. в машинах второго поколения они были заменены на дискретные германиевые транзисторы с временем задержки порядка 0.3 мкс. В середине 60-х г. были внедрены биполярные интегральные схемы с малым уровнем интеграции с временем задержки около 10 нс на вентиль. К середине 70-х г. время задержки снизилось до 1 нс на вентиль за счет совершенствования технологии производства и использования новых полупроводниковых материалов. Таким образом, за период с 1950 по 1975 г. быстродействие элементов, измеряемое как величина, обратная времени задержки вентиля, увеличилась приблизительно в 10^3 раз. В то же время прирост производительности ЭВМ в целом составил 10^5 раз, т. е. на два порядка больше. Объяснить этот разрыв между ростом производительности отдельных компонентов и ростом производительности вычислительной системы в целом можно только совершенствованием способов организации вычислительных систем.

Далее, в середине 70-х г. XX в. появились электронные компоненты, реализованные с применением технологий метал – окисел – проводник (МОП) и комплиментарной МОП (кМОП). Быстродействие этих компонентов почти на порядок ниже, чем у ранее используемых компонентов на базе биполярных транзисторов, но при этом их геометрические размеры существенно меньше. Это позволило разместить на одном кристалле интегральной схемы большее число компонентов, а значит, воплотить более сложные и ресурсоемкие решения по организации ВС.

В настоящее время продолжается развитие элементной базы кМОП в основном в направлении уменьшения размера компонентов. В связи с этим часто упоминается закон Гордона Мура – эмпирическое наблюдение, сделанное в 1965 г. и уточненное 10 лет спустя, согласно которому каждые 2 года количество транзисторов на кристалле интегральной схемы удваивается. В результате повышается быстродействие компонентов за счет сокращения времени переходных процессов, а увеличение плотности размещения позволяет уменьшить время распространения электрических сигналов между отдельными компонентами интегральной схемы. Это развитие можно охарактеризовать таким параметром, как норма технологического процесса – разрешающая способность литографического оборудования, используемого при производстве интегральных микросхем. На рис. 1 приведен график изменения нормы технологического процесса за последние 10 лет. На нем видно, что темп развития по этому пути сокращается. Это связано не только со сложностями технологии производства, но и с фундаментальными физическими ограничениями. Процессы, происходящие в полупроводниках, носят статистический характер. Число атомов, которые формируют полупроводниковый переход, должно составлять несколько сотен. Современные технологии вплотную приблизились к этому рубежу, и, судя по всему, недалек тот день, когда дальнейшее продвижение в этом направлении станет невозможным.

В настоящее время активно ведется работа по поиску новых вариантов элементной базы, отличной от полупроводниковой [2]. Это оптические и квантовые схемы, компоненты на основе биотехнологии, элементы с использованием явлений сверхпроводимости и т. д. Определенные успехи в этом направлении имеются, однако делать прогнозы пока сложно. И даже если альтернатива будет найдена, переход на

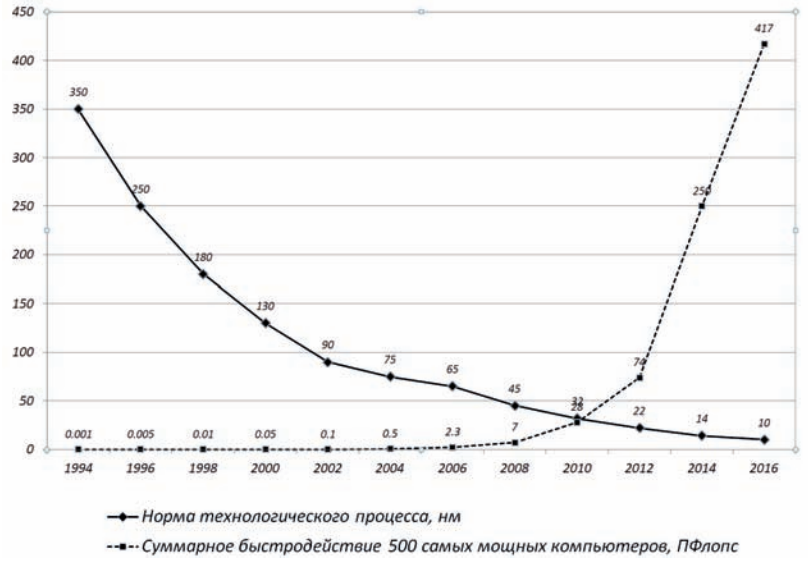

Рис. 1. Динамика изменения нормы технологического процесса

новую технологию, принципиально отличную от полупроводниковой, будет протекать сложно и в течение длительного промежутка времени.

Однако если на ту же временную шкалу наложить график роста производительности современных высокопроизводительных вычислительных систем, то можно отметить, что не только производительность ВС как таковая продолжает расти, но и темпы ее роста постоянно увеличиваются.

Это заставляет предположить, что помимо совершенствования элементной базы у роста производительности вычислительных систем есть и другие причины. Они становятся очевидными, если мы проанализируем исторически изменение архитектур наиболее распространенных вычислительных систем.

Появление первого микропроцессора явилось важным шагом в развитии вычислительной техники и привело к росту производительности как минимум просто вследствие того, что в одном конструктивном модуле были объединены основные узлы вычислительной системы,

что позволило снизить непроизводительные временные затраты на обмен информацией (командами, данными) внутри процессора.

Однако потребности пользователей всегда растут опережающими темпами, и следующим шагом, позволившим поднять производительность процессора ценой минимальных дополнительных аппаратных затрат, явилось введение так называемой опережающей выборки.

Традиционно процесс исполнения команды процессором можно было разделить на следующие основные этапы:

- выборка команды;
- дешифрация команды;
- выборка операндов;
- исполнение команды;
- завершение исполнения команды, включающее формирование слова состояния, опрос прерываний, формирование адреса следующей команды.

Этапы исполнялись последовательно, и каждый этап исполнялся своим аппаратным модулем. Таким образом, на каждом этапе исполнения команды работал только один модуль, остальные простаивали. Вместе с тем один из принципов фон Неймана, положенный в основу архитектуры вычислительных машин, утверждает, что программа хранится в памяти ЭВМ в последовательно расположенных ячейках и естественным порядком исполнения программы является выбор следующей команды из следующей ячейки (за исключением команд, нарушающих этот порядок – это команды переходов, условного и безусловного, вызова и возврата из подпрограмм и т. д.).

Опережающая выборка позволяет совместить этапы исполнения текущей команды, осуществляемой арифметико-логическим устройством процессора, и выборку следующей команды [3]. Это решение не требует добавления новых аппаратных модулей в процессор, а прежде всего изменяет порядок работы уже существующих. Такой вариант реорганизации работы процессора можно рассматривать как зачатки конвейера команд.

Введение опережающей выборки в одном из первых процессоров семейства X86 (Intel 286) позволило поднять его производительность среднем на 30 %. Этот результат оказался настолько эффективным, что уже в следующем поколении процессоров, 386, был реализован полно-

ценный 5-ступенчатый конвейер [4]. Поскольку в конвейерной системе на разных этапах обработки находится несколько команд, этот вариант организации работы можно рассматривать как одну из форм параллелизма.

Далее развитие конвейера команд шло в основном по пути увеличения длины конвейера и сокращения времени, требуемого для отдельной фазы исполнения команды. На последних этапах эволюции конвейерных процессоров в процессорных ядрах реализовывались полностью или частично несколько конвейеров, в результате чего возникали так называемые гипертрейдинговые архитектуры.

Однако конвейер команд представляет собой достаточно сложно управляемую структуру, а разное время исполнения команд разных типов, а также неизбежное наличие команд, нарушающих последовательность исполнения программы, привели ко все большей сложности управления конвейером. Конвейеризация процессора не видна программисту, управление и оптимизация его работы осуществляется аппаратными средствами. С какого-то момента дальнейшее развитие конвейера команд перестало быть эффективным.

Следующим этапом эволюции процессоров стало появление многоядерных процессоров, в котором на одном кристалле располагались несколько полноценных процессорных ядер, часто разделяющих общую кэш-память команд и данных [4].

Этот вид параллелизма допускает программное управление. Загрузка отдельных ядер процессора может осуществляться как средствами системного программного обеспечения, так и прикладными программами, часто позволяющими достичь высокой эффективности в использовании вычислительных ресурсов [5].

Если рассматривать дальнейшее развитие вычислительной техники на примере персональных компьютеров, то одним из важных этапов стало выделение задачи обработки графической информации. Эта задача имеет ряд специфических особенностей, которые затрудняли ее решение средствами процессоров со стандартной архитектурой. Для решения этой задачи в состав вычислительных систем стали вводиться графические сопроцессоры (GPU). Они содержат большое количество специализированных процессорных ядер, которые позволяют выполнять обработку графической информации с высоким параллелизмом [6].

В результате в настоящее время практически любое вычислительное устройство представляет собой параллельную вычислительную систему, в большинстве случаев сложную и неоднородную, в которой одновременно реализуется несколько принципов параллельной обработки информации.

Оба упомянутых направления развития вычислительной техники – совершенствование элементной базы и развитие архитектур вычислительных систем прежде всего в направлении параллелизма – являются взаимосвязанными. Совершенствование элементной базы позволяет не только поднять производительность отдельных компонентов вычислительной системы, но и дает возможность реализовать большую степень параллелизма или более широкий спектр способов параллельной обработки информации.

Важной задачей является также разработка параллельного программного обеспечения. Создание эффективного параллельного ПО является сложной задачей, решение которой невозможно без учета особенностей организации параллельной вычислительной системы.

Прежде чем перейти к детальному изучению параллельных вычислительных систем, необходимо определить следующие понятия: архитектура вычислительной системы (ВС), параллельная вычислительная система (ПВС) и параллелизм. Поскольку полагалось, что эти термины являются интуитивно понятными, долгое время не давалось их строгого определения. Со временем выяснилось, что различные авторы вкладывают в эти понятия различный смысл. Для унификации терминологии в рамках курса введем основные используемые понятия и определения.

Электронная вычислительная машина (ЭВМ) – комплекс технических средств, предназначенных для ввода, хранения, передачи, автоматической обработки и вывода (представления) информации в процессе решения вычислительных и информационных задач [7].

Термин *вычислительная система* является более широким и подразумевает возможность объединения нескольких ЭВМ (процессоров), разнообразных периферийных и запоминающих устройств, а также программного обеспечения, как системного, так и прикладного.

Архитектура ЭВМ – это наиболее общие принципы построения ЭВМ, реализующие программное управление работой и взаимодействием основных ее функциональных узлов [8].

К архитектуре относятся:

- структура памяти ЭВМ;
- способы доступа к памяти и внешним устройствам;
- возможность изменения конфигурации компьютера;
- система команд;
- форматы данных;
- организация интерфейса.

Основы учения об архитектуре вычислительных машин заложил выдающийся американский математик Джон фон Нейман. Он подключился к созданию первой в мире ламповой ЭВМ ENIAC в 1944 г., когда ее конструкция была уже выбрана. В процессе работы над проектом фон Нейман высказал идею принципиально новой ЭВМ. В 1946 г. ученые изложили свои принципы построения вычислительных машин в ставшей классической статье «Предварительное рассмотрение логической конструкции электронно-вычислительного устройства» [9].

Были сформулированы следующие основные принципы фон Неймана [3, 9]:

- использование двоичной системы счисления для представления данных и команд;
- однородность памяти: как программы (команды), так и данные хранятся в одной и той же памяти; над командами можно выполнять такие же действия, как и над данными;
- адресуемость памяти: основная память состоит из пронумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка;
- последовательное программное управление: программа состоит из набора команд, которые располагаются в памяти и выполняются процессором друг за другом в определенной последовательности, одна после завершения другой;
- принцип условного перехода: в определенных ситуациях последовательное исполнение команд может быть нарушено.

В течение значительного периода времени создаваемые ЭВМ базировались преимущественно на принципах фон Неймана, которые сыграли, безусловно, важную роль в развитии вычислительной техники, определив его основное направление на длительный срок. В то же время они неоднократно подвергались критике. В частности, принцип

последовательного выполнения команд ограничивал возможности вычислительной машины, сводя их к возможности одного процессора. С появлением параллельных вычислительных систем этот принцип утратил свое значение. Создание и повсеместное распространение параллельных вычислительных систем ознаменовало наступление постфон-неймановской эпохи.

Предполагают, что первое упоминание о принципе параллелизма в проекте вычислительного устройства было обнаружено в публикации американского генерала Менебриа под названием «Набросок аналитической машины, изобретенной Чарльзом Бэббиджем», датируемой октябрем 1842 г., где прозвучала следующая мысль: «В случае выполнения серии идентичных вычислений, подобных операции умножения и необходимых для формирования цифровых таблиц, машина может быть введена в действие с целью выдачи нескольких результатов одновременно, что очень существенно сократит весь объем процессов» [10]. Способность выполнять параллельные вычисления не была реализована в окончательном варианте машины Бэббиджа. Идея параллельной организации вычислений возникла почти на 100 лет раньше, чем техника достигла такого состояния, когда ее реализация стала возможной.

Для рассмотрения и анализа предлагаются несколько вариантов определений, которые встречаются в работах различных авторов.

Параллелизм – воспроизведение в нескольких копиях некоторой аппаратной структуры, что позволяет достигнуть повышения производительности за счет одновременной работы всех элементов структуры, осуществляющих решение различных частей этой задачи [11].

Параллелизм – способность к частичному совмещению или одновременному выполнению операций [12].

Параллелизм – одновременное выполнение двух или более частей программы двумя или более процессорными модулями ВС [13].

Приведенный список определений параллелизма, безусловно, не является исчерпывающим. Ряд специалистов (например Коуги [11]), помимо параллелизма выделяют *конвейеризацию*, которая представляет собой расщепление выполняемой функции на более мелкие части и их выполнение специализированными аппаратными модулями – ступенями конвейера. Несмотря на специфическую организацию, обработка данных в конвейере выполняется также параллельно.

1.1. Классификация Флинна для параллельных вычислительных систем

Активное развитие параллельных вычислительных систем, постоянный поиск новых способов их организации породили чрезвычайно широкий спектр параллельных архитектур. Для того, чтобы ориентироваться во всем этом многообразии и иметь возможность кратко охарактеризовать новую или уже существующую вычислительную систему, необходима классификация ПВС. Всего существует более 10 различных классификаций, более или менее детальных [14]. Наиболее часто упоминаемой, интуитивно понятной является систематика, предлагаемая Флинном [3, 14]. Данная классификация основана на том, как в машине увязываются потоки команд и обрабатываемых данных.

Поток – последовательность элементов (команд или данных), выполняемая или обрабатываемая процессором.

Флинн выделяет 4 класса вычислительных систем.

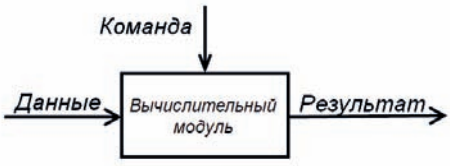

Рис. 2. Вычислительная система класса ОКОД (SISD)

ОКОД (single instruction – single data, SISD) – обычная последовательная ЭВМ фон Неймана, в которой имеется один поток команд (и практически одно устройство обработки команд), и каждая арифметическая команда инициирует одну операцию с одним потоком данных (рис. 2) [3, 14, 15].

Один поток команд – много потоков данных

ОКМД (single instruction – multiple data, SIMD) – в таких системах сохраняется один поток команд, но уже векторных, которые инициируют многочисленные операции. Каждый элемент вектора рассматривается как член отдельного потока данных (рис. 3) [3, 14].

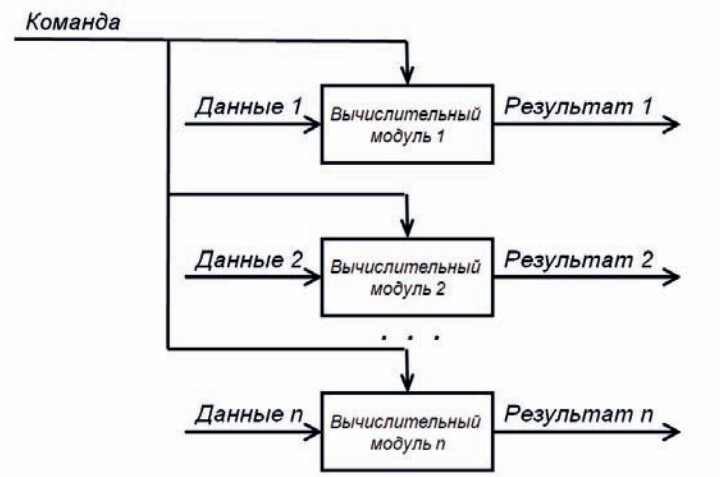

Рис. 3. Вычислительная система класса ОКМД (SIMD)

Много потоков команд – один поток данных

МКОД (multiple instruction – single data, MISD) – этот класс в настоящее время мало развит. Предполагается, что в машинах такого типа несколько потоков команд одновременно работают с одним элементом данных. В ряде работ, посвященных архитектурам ПВС, этот класс рассматривается как синоним конвейерной ЭВМ (рис. 4) [3, 14]. В данном курсе мы будем рассматривать именно такую интерпретацию этого класса.

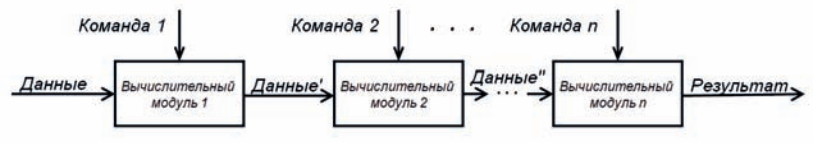

Рис. 4. Вычислительная система класса МКОД (MISD)

МКМД (multiple instruction – multiple data, MIMD) – много потоков команд означают существование нескольких устройств обработки команд, и, следовательно, и нескольких потоков данных (рис. 5) [3, 14].

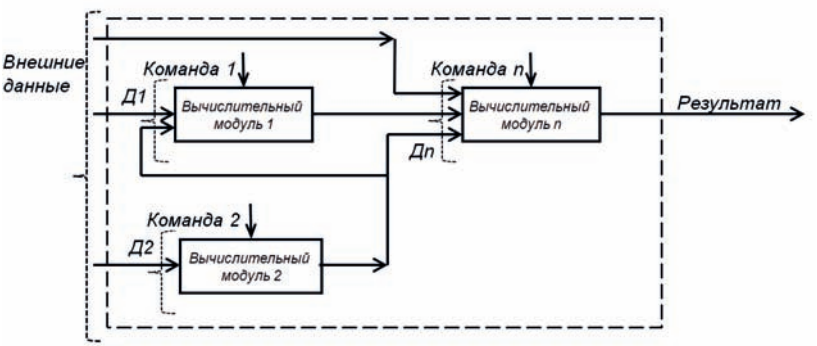

Рис. 5. Вычислительная система класса МКМД (MIMD)

Недостаток систематики Флинна состоит в том, что распределение систем по классам затруднено вследствие неопределенности в отношении потока данных.

Это приводит к тому, что конвейерная векторная ЭВМ может быть отнесена:

- к классу ОКОД, поскольку она обрабатывает один поток векторных данных;
- к классу ОКМД, как объяснялось выше;
- к классу МКОД, так как в ней имеется конвейерное векторное устройство, которое можно рассматривать как эквивалент для параллельного выполнения множества векторных команд над одиночным векторным потоком данных или над множественным скалярным потоком [63, 64].

Кроме того, она излишне широка. Так, большинство современных параллельных ВС относятся к классу МКМД, поэтому данный класс требует более детальной классификации [14, 16].

Для более детальной классификации вычислительных систем, относящихся к классу МКМД, предложена практически общепризнанная схема, в которой дальнейшее разделение типов многопроцессорных систем основывается на используемых способах организации оперативной памяти в этих системах (рис. 6) [3, 16, 17].

Данный поход позволяет различать два важных типа многопроцессорных систем: мультипроцессоры, или системы с общей разделя-

Рис. 6. Структура класса МКМД

емой памятью, и мультикомпьютеры, или системы с распределенной памятью [18, 19].

Основной особенностью мультипроцессорной архитектуры (рис. 7) является наличие набора вычислительных модулей и логически (а зачастую и физически) общей памяти данных, через которую выполняется обмен данными.

Среди мультипроцессорных систем выделяют системы с физически общей памятью и системы с распределенной памятью.

Системы с общей памятью, или с однородным доступом к памяти (Uniform Memory Access, UMA) используют единую (централизованную) общую память, к которой обеспечивается однородный (равноправный) доступ всех вычислительных модулей системы. Этот принцип служит основой для построения параллельно-векторных суперкомпьютеров (Parallel Vector Processor, PVP) и симметричных мультипроцессоров (Symmetric MultiProcessor, или SMP) [17, 20].

Симметричные мультипроцессоры состоят из совокупности процессоров, обладающих одинаковыми возможностями доступа к памяти

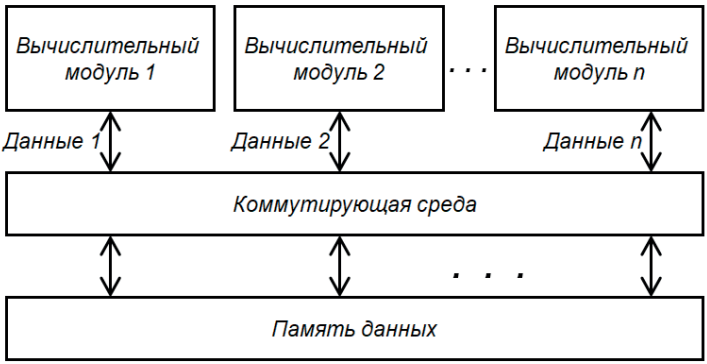

Рис. 7. Обобщенная архитектура мультипроцессорной системы

и внешним устройством и функционирующих под управлением единой ОС. Частным случаем SMP являются однопроцессорные компьютеры. Все процессоры этого типа имеют разделяемую общую память с единым адресным пространством. Использование SMP обеспечивает следующие возможности:

- программирование в модели общей памяти (POSIX threads, OpenMP);
- масштабирование приложений при низких начальных затратах путем применения без преобразования приложений на новых более производительных аппаратных средствах;
- одинаковое время доступа ко всей памяти;
- возможность пересылки сообщений с большой пропускной способностью;
- неделимые операции синхронизации и блокировки.

Преимуществом этих систем является существование сравнительно эффективных средств автоматического распараллеливания.

Основной недостаток SMP-систем – низкая масштабируемость, степень которой ограничена возможностью реализации одинакового для всех процессоров времени доступа в память с достаточной скоростью. Как правило, число процессоров в подобных системах не превышает 32. Для построения масштабируемых систем на базе SMP используются кластерные или NUMA-архитектуры.