Лабораторная работа №49. «Функционирование мультипроцессорных систем.»

III. Расчет учебного времени.

Содержание и порядок проведения занятия	Время, мин.
Вступительная часть. **Основная часть.** Учебные вопросы: 1. Классификация систем параллельной обработки данных. 2. Мультипроцессорные системы с общей памятью. 3. Мультипроцессорные системы с локальной памятью. **Заключительная часть**. **Итого:**	

IV. Литература.

1. Сергей Озеров. Архитектура CPU. "Компьютер", №37, октябрь 2005.
2. Э. Таненбаум. Архитектура компьютера – СПб: Питер, 2003.

V. Учебно-материальное обеспечение.

1. КЛАССИФИКАЦИЯ СИСТЕМ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ

На протяжении всей истории развития вычислительной техники делались попытки найти какую-то общую классификацию, под которую подпадали бы все возможные направления развития компьютерных архитектур. Ни одна из таких классификаций не могла охватить все разнообразие разрабатываемых архитектурных решений и не выдерживала испытания временем. Тем не менее в научный оборот попали и широко используются ряд терминов, которые полезно знать не только разработчикам, но и пользователям компьютеров.

Любая вычислительная система (будь то супер-ЭВМ или персональный компьютер) достигает своей наивысшей производительности благодаря использованию высокоскоростных элементов и параллельному выполнению большого числа операций.

Именно возможность параллельной работы различных устройств системы (работы с перекрытием) является основой ускорения основных операций.

Параллельные ЭВМ часто подразделяются по классификации Флинна на **машины типа SIMD** (Single Instruction Multiple Data - с одним потоком команд при множественном потоке данных) и **MIMD** (Multiple Instruction Multiple Data - с множественным потоком команд при множественном потоке данных).

Как и любая другая, приведенная выше классификация несовершенна: существуют машины прямо в нее не попадающие, имеются также важные признаки, которые в этой классификации не учтены. В частности, к машинам типа SIMD часто относят векторные процессоры, хотя их высокая производительность зависит от другой формы параллелизма - конвейерной организации машины.

Многопроцессорные векторные системы, типа Cray Y-MP, состоят из нескольких векторных процессоров и поэтому могут быть названы MSIMD (Multiple SIMD).

Можно выделить четыре основных типа архитектуры систем параллельной обработки:

1) Конвейерная и векторная обработка.

Основу конвейерной обработки составляет раздельное выполнение некоторой операции в несколько этапов (за несколько ступеней) с передачей данных одного этапа следующему. Производительность при этом возрастает благодаря тому, что одновременно на различных ступенях конвейера выполняются несколько операций.

Конвейеризация эффективна только тогда, когда загрузка конвейера близка к полной, а скорость подачи новых операндов соответствует максимальной производительности конвейера. Если происходит задержка, то параллельно будет выполняться меньше операций и суммарная производительность снизится.

Векторные операции обеспечивают идеальную возможность полной загрузки вычислительного конвейера.

При выполнении векторной команды одна и та же операция применяется ко всем элементам вектора (или чаще всего к соответствующим элементам пары векторов).

Для настройки конвейера на выполнение конкретной операции может потребоваться некоторое установочное время, однако затем операнды

могут поступать в конвейер с максимальной скоростью, допускаемой возможностями памяти. При этом не возникает пауз ни в связи с выборкой новой команды, ни в связи с определением ветви вычислений при условном переходе.

Таким образом, главный принцип вычислений на векторной машине состоит <u>в выполнении некоторой элементарной операции или комбинации из нескольких элементарных операций, которые должны повторно применяться к некоторому блоку данных.</u>

Таким операциям в исходной программе соответствуют небольшие компактные циклы.

2) Машины типа SIMD.

Машины типа SIMD состоят из большого числа идентичных процессорных элементов, имеющих собственную память. Все процессорные элементы в такой машине выполняют одну и ту же программу.

Очевидно, что такая машина, составленная из большого числа процессоров, может обеспечить очень высокую производительность только на тех задачах, при решении которых все процессоры могут делать одну и ту же работу.

Модель вычислений для машины SIMD очень похожа на модель вычислений для векторного процессора: <u>одиночная операция выполняется над большим блоком данных.</u>

В отличие от ограниченного конвейерного функционирования векторного процессора, матричный процессор (синоним для большинства SIMD-машин) может быть значительно более гибким. Обрабатывающие элементы таких процессоров - это универсальные программируемые ЭВМ, так что задача, решаемая параллельно, может быть достаточно сложной и содержать ветвления.

Обычное проявление этой вычислительной модели в исходной программе примерно такое же, как и в случае векторных операций: циклы на элементах массива, в которых значения, вырабатываемые на одной итерации цикла, не используются на другой итерации цикла.

Модели вычислений на векторных и матричных ЭВМ настолько схожи, что эти ЭВМ часто обсуждаются как эквивалентные.

3) Машины типа MIMD.

Термин "мультипроцессор" покрывает большинство машин типа MIMD и (подобно тому, как термин "матричный процессор" применяется к машинам типа SIMD) часто используется в качестве синонима для машин типа MIMD.

В мультипроцессорной системе каждый процессорный элемент (ПЭ) выполняет свою программу достаточно независимо от других процессорных элементов.

Процессорные элементы, конечно, должны как-то связываться друг с другом, что делает необходимым более подробную классификацию машин типа MIMD.

В мультипроцессорах с общей памятью (сильносвязанных мультипроцессорах) имеется память данных и команд, доступная всем ПЭ. С общей памятью ПЭ связываются с помощью общей шины или сети обмена.

В противоположность этому варианту в слабосвязанных многопроцессорных системах (машинах с локальной памятью) вся память делится между процессорными элементами и каждый блок памяти доступен только связанному с ним процессору. Сеть обмена связывает процессорные элементы друг с другом.

Базовой моделью вычислений на MIMD-мультипроцессоре является совокупность независимых процессоров, эпизодически обращающихся к разделяемым данным.

Существует большое количество вариантов этой модели. На одном конце спектра - модель распределенных вычислений, в которой программа делится на довольно большое число параллельных задач, состоящих из множества подпрограмм. На другом конце спектра - модель потоковых вычислений, в которых каждая операция в программе может рассматриваться как отдельный процесс. Такая операция ждет своих входных данных (операндов), которые должны быть переданы ей другими процессами. По их получении операция выполняется, и полученное значение передается тем процессам, которые в нем нуждаются. В потоковых моделях вычислений с большим и средним уровнем гранулярности, процессы содержат большое число операций и выполняются в потоковой манере.

Следует отметить, что многие современные супер-ЭВМ представляют собой многопроцессорные системы, в которых в качестве процессоров используются векторные процессоры или процессоры типа SIMD. **Такие машины относятся к машинам класса MSIMD**.

Машины типа MSIMD, как можно себе представить, дают возможность использовать лучший из этих двух принципов декомпозиции: векторные операции для тех частей программы, которые подходят для этого, и гибкие возможности MIMD-архитектуры для других частей программы.

Многопроцессорные системы за годы развития вычислительной техники претерпели ряд этапов своего развития. Исторически первой стала осваиваться технология SIMD. Однако в настоящее время наметился устойчивый интерес к архитектурам MIMD. Этот интерес главным образом определяется двумя факторами:

1). Архитектура MIMD дает большую гибкость: при наличии адекватной поддержки со стороны аппаратных средств и программного обеспечения MIMD может работать как однопользовательская система, обеспечивая высокопроизводительную обработку данных для одной прикладной задачи, как многопрограммная машина, выполняющая множество задач параллельно, и как некоторая комбинация этих возможностей.

2). Архитектура MIMD может использовать все преимущества современной микропроцессорной технологии на основе строгого учета соотношения стоимость/производительность. В действительности, практически все современные многопроцессорные системы строятся на тех же микропроцессорах, которые можно найти в персональных компьютерах, рабочих станциях и небольших однопроцессорных серверах.

Одной из отличительных особенностей многопроцессорной вычислительной системы является **сеть обмена**, с помощью которой процессоры соединяются друг с другом или с памятью.

Модель обмена настолько важна для многопроцессорной системы, что многие характеристики производительности и другие оценки выражаются отношением времени обработки к времени обмена, соответствующим решаемым задачам.

Существуют две основные модели межпроцессорного обмена: одна основана на передаче сообщений, другая - на использовании общей памяти.

В многопроцессорной системе с общей памятью один процессор осуществляет запись в конкретную ячейку, а другой процессор производит считывание из этой ячейки памяти. Чтобы обеспечить согласованность данных и синхронизацию процессов, обмен часто реализуется по принципу взаимно исключающего доступа к общей памяти методом "почтового ящика".

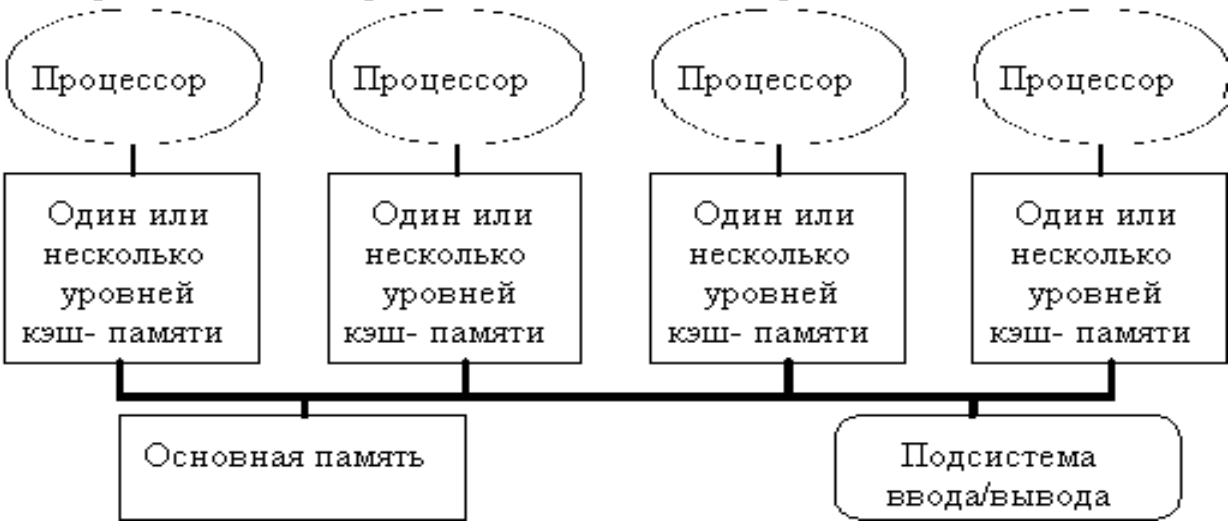
В архитектурах с локальной памятью непосредственное разделение памяти невозможно. Вместо этого процессоры получают доступ к совместно используемым данным посредством передачи сообщений по сети обмена. Эффективность схемы коммуникаций зависит от протоколов обмена, основных сетей обмена и пропускной способности памяти и каналов обмена.

Таким образом, существующие MIMD-машины распадаются на два основных класса в зависимости от количества объединяемых процессоров, которое определяет и способ организации памяти, и методику их межсоединений.

2. МУЛЬТИРПОЦЕССОРНЫЕ СИСТЕМЫ С ОБЩЕЙ ПАМЯТЬЮ

Системы с общей (разделяемой) основной памятью объединяют до нескольких десятков (обычно менее 32) процессоров. Сравнительно небольшое количество процессоров в таких машинах позволяет иметь одну централизованную общую память и объединить процессоры и память с помощью одной шины.

При наличии у процессоров КЭШ-памяти достаточного объема высокопроизводительная шина и общая память могут удовлетворить обращения к памяти, поступающие от нескольких процессоров. Поскольку имеется единственная память с одним и тем же временем доступа, эти машины иногда называются UMA (Uniform Memory Access). Такой способ организации со сравнительно небольшой разделяемой памятью в



настоящее время является наиболее популярным. Структура подобной системы представлена на рис. 1.

Рис. 1. Типовая архитектура мультипроцессорной системы с общей памятью.

Требования, предъявляемые современными процессорами к полосе пропускания памяти можно существенно сократить путем применения больших многоуровневых КЭШей. Тогда, если эти требования снижаются, несколько процессоров смогут разделять доступ к одной и той же памяти.

Начиная с 1980 года эта идея, подкрепленная широким распространением микропроцессоров, стимулировала многих разработчиков на создание небольших мультипроцессоров, в которых несколько процессоров разделяют одну физическую память, соединенную с ними с помощью разделяемой шины. Из-за малого размера процессоров и заметного сокращения требуемой полосы пропускания шины, достигнутого за счет возможности реализации достаточно большой кэш-памяти, такие машины стали исключительно эффективными по стоимости.

Современные конструкции предполагают размещение до четырех процессоров на одной плате. В такой машине кэши могут содержать как **разделяемые**, так и **частные данные**.

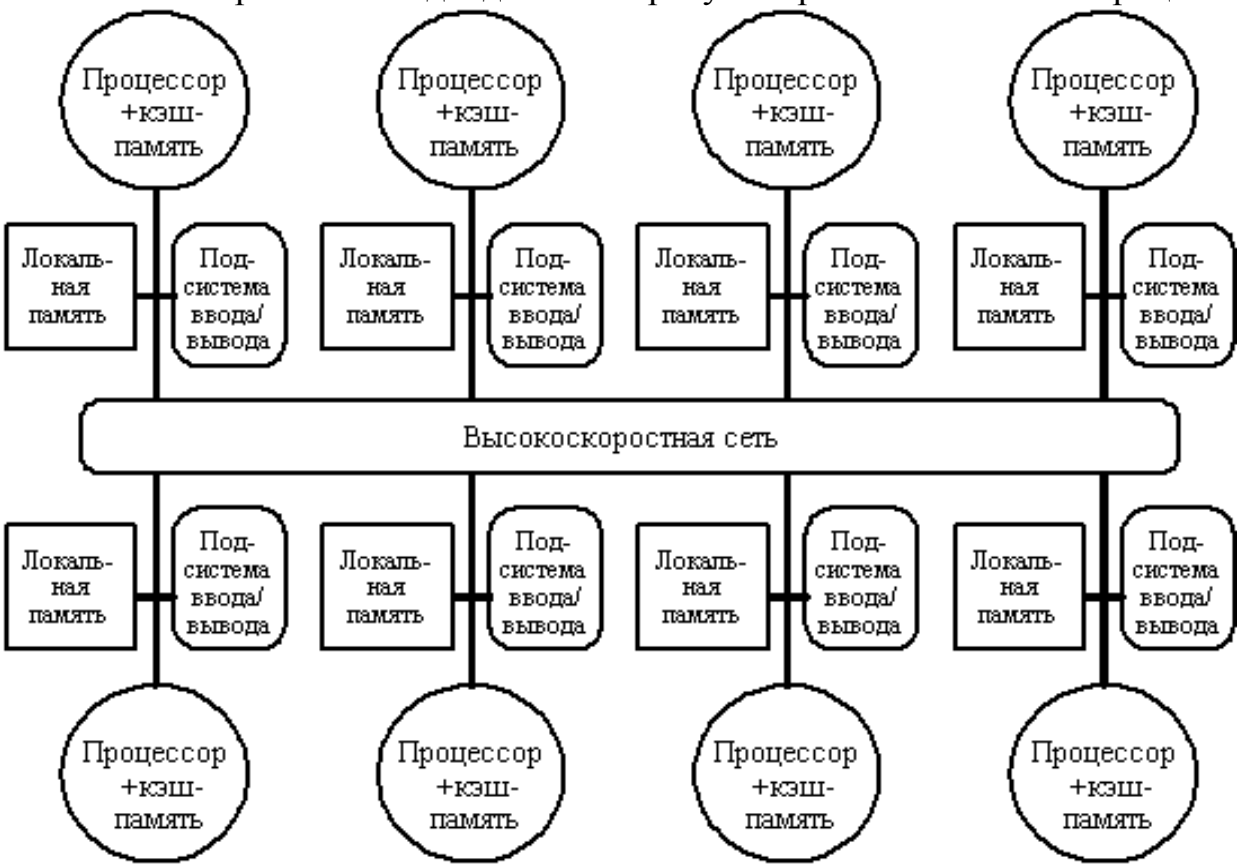
Частные данные - это данные, которые используются одним процессором, в то время как разделяемые данные используются многими процессорами, по существу обеспечивая обмен между ними.

Когда кэшируется элемент частных данных, их значение переносится в кэш для сокращения среднего времени доступа, а также требуемой полосы пропускания. Поскольку никакой другой процессор не использует эти данные, этот процесс идентичен процессу для однопроцессорной машины с КЭШ-памятью.

Если кэшируются разделяемые данные, то разделяемое значение реплицируется и может содержаться в нескольких КЭШах. Кроме сокращения задержки доступа и требуемой полосы пропускания такая репликация данных способствует также общему сокращению количества обменов. Однако кэширование разделяемых данных вызывает новую проблему: когерентность КЭШ-памяти.

3. МУЛЬТИПРОЦЕССОРНЫЕ СИСТЕМЫ С ЛОКАЛЬНОЙ ПАМЯТЬЮ

Второй класс образуют крупномасштабные системы с распределенной памятью. Для того, чтобы поддерживать большое количество процессоров, приходится распределять основную память между ними, в противном случае полосы пропускания памяти просто может не хватить для удовлетворения запросов, поступающих от очень большого числа процессоров. Естественно при таком подходе также требуется реализовать связь процес-



соров между собой.

На рис.2 представлена типовая архитектура системы с распределенной памятью.

С ростом числа процессоров просто невозможно обойти необходимость реализации модели распределенной памяти с высокоскоростной сетью для связи процессоров. С быстрым ростом производительности процессоров и связанным с этим ужесточением требования увеличения полосы пропускания памяти, масштаб систем (т.е. число процессоров в системе), для которых требуется организация распределенной памяти, уменьшается, так же как и уменьшается число процессоров, которые удается поддерживать на одной разделяемой шине и общей памяти.

Распределение памяти между отдельными узлами системы имеет два главных преимущества:

Во-первых, это эффективный с точки зрения стоимости способ увеличения полосы пропускания памяти, поскольку большинство обращений могут выполняться параллельно к локальной памяти в каждом узле.

Во-вторых, это уменьшает задержку обращения (время доступа) к локальной памяти. Эти два преимущества еще больше сокращают количество процессоров, для которых архитектура с распределенной памятью имеет смысл.

Как уже было отмечено, любая крупномасштабная многопроцессорная система должна использовать множество устройств памяти, которые физически распределяются вместе с процессорами. Имеется две альтернативных организации адресации этих устройств памяти и связанных с этим два альтернативных метода для передачи данных между процессорами.

1). Физически отдельные устройства памяти могут адресоваться как логически **единое адресное пространство**, что означает, что любой процессор может выполнять обращения к любым ячейкам памяти, предполагая, что он имеет соответствующие права доступа. Такие машины называются машинами с распределенной разделяемой (общей) памятью (DSM - distributed shared memory), масштабируемые архитектуры с разделяемой памятью, а иногда NUMA's - Non-Uniform Memory Access, поскольку время доступа зависит от расположения ячейки в памяти.

2). В альтернативном случае, адресное пространство состоит **из отдельных адресных пространств**, которые логически не связаны и доступ к которым не может быть осуществлен аппаратно другим процессором. В таком примере каждый модуль процессор-память представляет собой отдельный компьютер, поэтому такие системы называются многомашинными (multicomputers).

С каждой из этих организаций адресного пространства связан свой механизм обмена.

Для машины с единым адресным пространством это адресное пространство может быть использовано для обмена данными посредством операций загрузки и записи. Поэтому эти машины и получили название машин с разделяемой (общей) памятью.

Для машин с множеством адресных пространств обмен данными должен использовать другой механизм: передачу сообщений между процессорами; поэтому эти машины часто называют машинами с передачей сообщений.

Каждый из этих механизмов обмена имеет свои преимущества. Для обмена в **общей памяти** это включает:

- совместимость с хорошо понятными используемыми как в однопроцессорных, так и маломасштабных многопроцессорных системах, механизмами, которые используют для обмена общую память.

- простота программирования, когда модели обмена между процессорами сложные или динамически меняются во время выполнения. Подобные преимущества упрощают конструирование компилятора.

- более низкая задержка обмена и лучшее использование полосы пропускания при обмене малыми порциями данных.

- возможность использования аппаратно управляемого кэширования для снижения частоты удаленного обмена, допускающая кэширование всех данных, как разделяемых, так и неразделяемых.

Основные преимущества обмена с **помощью передачи сообщений** являются:

- аппаратура может быть более простой, особенно по сравнению с моделью разделяемой памяти, которая поддерживает масштабируемую когерентность кэш-памяти.

- модели обмена достаточно просты.