

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

**МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ ОБУЧАЮЩИХСЯ ПО
ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Б1.О.06 Архитектура параллельных вычислительных систем

Направление подготовки (специальность)

09.04.01 Информатика и вычислительная техника

Профиль подготовки (специализация)

“Автоматизированные системы обработки информации и управления”

Форма обучения (*очная*)

СОДЕРЖАНИЕ

1.	Тематическое содержание дисциплины	3
----	--	---

1. Тематическое содержание дисциплины

1.1. Тема 1: Основные понятия и характеристики параллельных вычислительных систем (20часов)

1.1.1. Перечень и краткое содержание рассматриваемых вопросов:

1. Основные понятия.

Распараллеливание - естественный жизненный принцип, основанный на утверждении, что коллектив сильнее одиночки. *Параллельные структуры* отображают организацию и взаимодействие исполнителей сложного комплекса взаимосвязанных работ. В вычислительной технике - это проблема параллельного выполнения отдельных программ и целых программных комплексов, отображенная выбором архитектуры процессоров и многопроцессорных вычислительных систем. *Параллельное программирование* объединяет проблемы оптимального статического и динамического планирования выполнения комплексов взаимосвязанных работ при заданном составе и структуре взаимодействия исполнителей.

2. Исторический аспект развития параллельных вычислительных систем.

Развитие элементарно-конструкторской базы, основанное на физических принципах электроники, всегда (и это диалектически правильно) отстает от требований конкретно решаемых задач. Быстродействие одного компьютера недостаточно для, например, массового обслуживания группы перехватчиков, стартующих против налета баллистических целей. Своевременно пришло понимание того, что только *структурными* методами можно добиться необходимой производительности вычислительных средств. Таким основным *структурным* методом является *распараллеливание вычислений*.

Сначала стало ясно, что вместо одной ЭВМ в систему управления необходимо встраивать несколько, решающих единый набор функционально связанных задач. Такой коллектив ЭВМ получил название вычислительного комплекса (ВК). Информационная взаимосвязанность задач потребовала применения средств оперативного обмена данными внутри ВК *исредств синхронизации*. Однако временные *затраты* на организацию взаимодействия ЭВМ в ВК (накладные *расходы*) оказались весьма высоки. Возникла идея использования взамен *разделенной* оперативной памяти (ОП), реализованной в ВК, *общей (разделяемой) ОП*, которая связывает несколько центральных процессоров. Проблема затрат времени на обмен оказалась решенной. Так зародилось понятие *параллельной вычислительной системы (ВС)*.

3. Цель распараллеливания.

Универсальным критерием, используемым при решении задач распараллеливания, является минимум времени выполнения совокупности работ, распределяемых между процессорами. Если несколько процессоров составляют ВС, то важной характеристикой ее эффективности (основные составляющие эффективности — производительность, надежность, стоимость) при специализированном использовании (например, в составе автоматизированной системы управления, АСУ) является коэффициент полезной загрузки k_3 .

4. Организация параллельной обработки процессов.

Основное предположение об эффективности совместной работы нескольких процессоров при решении множества задач (функциональных модулей) управляющей системы основывалось на возможности усреднения нагрузки при динамическом распределении работ между процессорами. Такое усреднение позволяет сгладить пиковую нагрузку на отдельные процессоры, а, следовательно, снизить требования к их производительности. Конечно, это справедливо в том случае, если увеличивающиеся накладные расходы, связанные с организацией динамического управления вычислительным процессом (диспетчированием), оправдывают эти затраты.

5. Научные школы в развитии параллельных технологий.

Отметим существование ряда отечественных школ, послуживших развитию параллельных технологий как важного теоретического и практического научного направления.

Такие школы возглавляли И.В.Прангишвили, С.Я.Виленкин, В.В.Игнатущенко и др. — в области создания микропроцессорных векторных ВС и ВС с перестраиваемой структурой. Новосибирская школа (Ю.Г.Косарев, В.Е.Котов, А.С.Нариньяни и др.) пропагандировала универсальные, многофункциональные структуры ВС и значительно продвинулась в теоретическом обосновании параллельных вычислительных процессов. Плодотворной была Таганрогская школа, возглавляемая А.В. Каляевым, которая защищала идею многопроцессорных ВС с программируемой структурой. С началом "перестройки" закончились очередные попытки "догнать и перегнать" на основе расширенного воспроизведения идей С. Крея в рамках проекта "Электроника ССБИС", осуществляемого В.А. Мельниковым — конструктором замечательной ЭВМ "БЭСМ-6".

1.2. Тема 2 Параллельные структуры вычислительных систем (16 часов)

1.2.1. Перечень и краткое содержание рассматриваемых вопросов:

1. Два уровня распараллеливания.

Сложилось представление о двух основных уровнях, на которых в ВС применяются практические методы распараллеливания:

- на уровне программ, процессов, процедур (первый уровень распараллеливания);
- на уровне команд и операций (второй уровень распараллеливания).

Эти уровни обусловили уровни структуризации ВС на пути превращения ее в супер-ЭВМ. Современным практическим воплощением первого уровня структуризации являются однородные многопроцессорные ВС на общей (разделяемой) оперативной памяти. Они получили название симметричных ВС за обеспечение "равноправия" составляющих модулей. Окончательное признание симметричных ВС положило конец поиску "экзотических" архитектур, эффективных лишь при решении определенных классов задач. Универсальность симметричных ВС, возможность реализации на них любых вычислительных процессов с высокой эффективностью оборудования иллюстрируются многими применениями и анализируются ниже.

2. Классификация параллельных ВС.

Общепринята удачная классификация ВС, которую предложил в 1966 г. М.Флинн (США). Основным определяющим архитектурным параметром он выбрал взаимодействие потока команд и потока данных (операндов и результатов).

В ЭВМ классической архитектуры ведется последовательная обработка команд и данных. Команды поступают одна за другой (за исключением точек ветвления

программы), и для них из ОЗУ или регистров так же последовательно поступают операнды. Одной команде (операции) соответствует один необходимый ей набор операндов (как правило, два для бинарных операций). Этот тип архитектуры — "один поток команд — один поток данных", ОКОД (SISD - "SingleInstruction, SingleData"). Тип **ОКМД** — "**один поток команд — много потоков данных**" (**SIMD** — "**SingleInstruction — MultipleData**") охватывает ВС, в которых одной командой обрабатывается набор данных, множество данных, вектор, и вырабатывается множество результатов. К типу **МКОД** — "**много потоков команд — один поток данных**" (**MISD** — "**MultipleInstruction — SingleData**") принято относить векторный конвейер (обычно в составе ВС, чтобы подчеркнуть основной используемый принцип вычислений). Тип **МКМД** — "**много потоков команд — много потоков данных**" (**MIMD** — "**MultipleInstruction — MultipleData**") соответствует более полному и независимому распараллеливанию.

3. "Фон-Неймановские" и "не-Фон-Неймановские" архитектуры.

Основные архитектурно-функциональные принципы построения ЭВМ были разработаны и опубликованы в 1946 г. венгерским математиком и физиком Джоном фон Нейманом и его коллегами Г.Голдстайном и А.Берксом в ставшем классическим отчете "Предварительное обсуждение логического конструирования электронного вычислительного устройства". Основополагающими принципами ЭВМ на основании этого отчета являются: 1) принцип программного управления выполнением программы, и 2) принцип хранимой в памяти программы. Они легли в основу понятия фон-Неймановской архитектуры, широко использующей счетчик команд. Альтернативной архитектурой является "**не-фон-Неймановская**" **архитектура**, допускающая одновременный анализ более одной команды. Поиски ее обусловлены необходимостью распараллеливания выполнения программы между несколькими исполнительными устройствами — процессорами. Счетчик команд при этом не нужен. Порядок выполнения команд определяется наличием исходной информации для выполнения каждой из них. Если несколько команд готовы к выполнению, то принципиально возможно их назначение для выполнения таким же количеством свободных процессоров. Говорят, что такие ВС управляются *потоком данных (dataflow)*.

4. Системы с общей и распределенной памятью.

Системы с общей (разделяемой) оперативной памятью образуют современный класс ВС — многопроцессорных *супер-ЭВМ*. Одинаковый доступ всех процессоров к программам и данным представляет широкие возможности организации параллельного вычислительного процесса (*параллельных вычислений*). Отсутствуют потери реальной производительности на *межпроцессорный* (между задачами, процессами и т.д.) обмен данными. **Системы с распределенной памятью** образуют вычислительные комплексы (ВК) — коллективы ЭВМ с межмашинным обменом для совместного решения задач. В ВК объединяются вычислительные средства систем управления, решающие специальные наборы задач, взаимосвязанных по данным. Принято говорить, что такие ВК выполняют *распределенные вычисления*, а сами ВК называют *распределенными ВК*.

5. Способы межмодульного соединения (комплексирования).

Различают два противоположных способа комплексирования: с общей шиной (шинная архитектура) и с перекрестной (матричной) коммутацией модулей ВС (процессоров, модулей памяти, периферии). *Шина* используется в режиме разделения времени, при котором лишь один модуль в данный момент работает на передачу. Принимать принципиально могут все модули, хотя преимущественно *информация* при

выдаче в нее адресуется. Применяется в микро- и мини-ЭВМ при сравнительно небольшом числе модулей. Практически производится разделение шины на управляющую, адресную и шину данных.

6. "Исторические" модели.

Векторная ВС ПС-2000. Разрабатывалась как проблемно-ориентированная ВС для задач обработки геофизической информации, информации со спутников в интересах геологии, картографии, обработки изображений, моделирования поведения среды и т.д. Является типичной иллюстрацией типа ОКМД.

Матричная ВС также является типичным представителем *ОКМД*, расширением принципа векторных ВС.

Классическим примером, прототипом и эталоном стала ВС ILLIAC IV, разработанная в 1971 г. в Иллинойском университете и в начале 1972 г. установленная в Эймском научно-исследовательском центре NASA (Калифорния).

ВС Cray-1, несмотря на последующие разработки, остается эталоном типа МКОД. В России разработан аналог — векторно-конвейерная ЭВМ "Электроника ССБИС", отличающаяся некоторыми увеличенными параметрами комплектации.

МВК "Эльбрус-2". Представляет тип МКМД. Имеет общую оперативную память и перекрестный коммутатор. Является основой сложных автоматизированных систем управления, работающих в реальном времени, использовался в космических исследованиях, а также в области ядерной физики.

1.3 Тема 3 Микропроцессорные системы и способы распараллеливания (22 часов)

1.3.1. Перечень и краткое содержание рассматриваемых вопросов:

1. Мультимикропроцессорные вычислительные системы.

В настоящее время выбор сделан в пользу многопроцессорных симметричных ВС типа *MIMD*, обеспечивающих виртуализацию вычислительных ресурсов. Основу такой ВС составляет *суперскалер*, сосредоточивший в себе все способы достижения максимального быстродействия при выполнении одиночной программы. *Векторные* и векторно-конвейерные процессоры и системы получили своё место. Их эффективность как самостоятельных установок могла быть достаточно высокой только при решении специальных задач и тестов. Поэтому достаточно быстро выяснилось, что эти установки могут выполнять функции интеллектуальных терминалов при решении основной задачи на другом универсальном вычислительном средстве и выполнять лишь отдельные его заявки. Сегодня стало окончательно ясно, что первые эффективны лишь в роли специализированных вычислительных устройств для решения специальных задач. Вторые твердо заняли место в составе многофункциональных арифметическо-логических устройств (*АЛУ суперскалеров*), ибо без конвейеров мы не мыслим себе выполнение всех операций ВС.

2. Направление "мини-супер".

Супер-ЭВМ, предназначенны для решения особо сложных задач в составе систем управления в реальном времени, моделирования сложнейших физических процессов, решения задач исследования операций, задач искусственного интеллекта, выполнения роли майнфреймов и серверов в локальных, корпоративных и глобальных сетях.

Супер-ЭВМ уникальна, мало тиражируема, цена ее высока.

С другой стороны, ничто уже не может остановить "победного шествия" персональных компьютеров. Область применения их стала всеобъемлющей. Они используются и там, где могут справиться с задачами, и там, где уже не справляются, несмотря на применение современных *суперскалеров*.

3. Распределенный и разделяемый вычислительный ресурс второго уровня. Решающие поля.

Второй уровень распараллеливания предполагает распределение команд, инструкций, операций, элементарных функций и других несложных процедур — для выполнения исполнительными устройствами процессоров или в общем вычислительном ресурсе симметричной *ВС*. Здесь существуют свои проблемы, связанные с "элементарным" характером операций, небольшим объемом содержащихся в них *работ*, с их еще большей критичностью *по* отношению к "накладным расходам" на организацию и синхронизацию. Мы предполагаем, что исполнительные устройства *ВС* образуют вычислительный *ресурс* второго уровня (распараллеливания).

Сложилась традиция построения этого ресурса, где основное внимание уделяется построению многофункциональных *АЛУ*. Однако в ряде архитектур пока еще робко пробивает себе дорогу *объединение АЛУ* в единый разделяемый *ресурс* системы — построение решающих полей.

4. Способы распараллеливания.

Различают два основных способа распараллеливания: *по управлению* и *по информации*.

Первый способ — *представление* алгоритма задачи в виде частично-упорядоченной последовательности выполняемых *работ*. Затем в результате диспетчирования реализуется *оптимальный план* выполнения *работ* в *ВС* при ограничениях на *время выполнения* всего алгоритма или за минимальное время.

Основой является *представление* алгоритма *граф-схемой* *G*, отражающей информационные связи между работами (задачами, процессами, процедурами, операторами, макрокомандами и т.д.), на которые разбит *алгоритм*. *Граф G* — взвешенный, ориентированный, без контуров.

Второй способ распараллеливания — *по информации* — используется тогда, когда можно распределить обрабатываемую информацию между процессорами для обработки *по* идентичным алгоритмам (*по* одному алгоритму).

1.4 Тема 4 Распараллеливание в ВС на уровне исполнительных устройств (18 часов)

1.4.1. Перечень и краткое содержание рассматриваемых вопросов:

1 Конвейеры операций.

Выполнение любой *операции* складывается из нескольких последовательных этапов, каждый из которых может выполняться своим функциональным узлом. Это легко показать на операциях сложения и умножения. Выполнение деления мантисс (порядки вычитаются) чаще всего производится с помощью вычитания из делимого делителя, сдвига влево полученного остатка, нового вычитания делителя из результата сдвига и т.д. В некоторых *ВС* находится обратная величина делителя с помощью аппроксимирующих полиномов. Затем *делимое* умножается на эту величину.

2 Векторные конвейеры. "Зацепление" векторов.

Наряду с использованием конвейеров для обработки единичных (скалярных) данных, используют так называемые *векторные конвейеры*, единичной информацией для которых являются вектора — массивы данных. Применение *векторных конвейеров* определило класс *ВС* — *векторно-конвейерных ВС*, сегодня ещё являющихся основой построения некоторых *супер-ЭВМ—ВС* сверхвысокой производительности.

Для эффективности *векторно-конвейерных ВС* (например, для подготовки алгоритмов решения задач на *ВС* "Электроника-ССБИС") необходима *векторизация* задач. Это — такое преобразование алгоритма, при котором максимально выделяются (если не вся задача сводится к этому) элементы обработки массивов данных одинаковыми операциями. Сюда входят все задачи, основанные на матричных преобразованиях, обработка изображений, сигналов, *моделирование* поведения среды и т.д.

3 Обобщенная процедура динамического распараллеливания в многофункциональном АЛУ.

В многопроцессорном вычислительном комплексе "Эльбрус-2" используется динамическое распределение *работ* между исполнительными устройствами *АЛУ*. При этом успешно разрешается противоречие между безадресной системой команд, предполагающей их выполнение на стеке, и возможностью одновременной загрузки многих исполнительных устройств.

При ближайшем рассмотрении *стек* представляется сугубо последовательной структурой. Работа со стеком, как через "узкое горло", производится с использованием его вершины, и в случае многофункциональных *АЛУ* (т.е. *АЛУ*, состоящих из набора нескольких специализированных по операциям исполнительных устройств) представляется проблематичной. Казалось бы, *стек* по самой своей природе не предназначен для распараллеливания. Однако это не так, что и было продемонстрировано в МВК "Эльбрус-2".

4 Адресный способ распараллеливания.

Демонстрируется общий подход при реализации *адресного распараллеливания*, используемого после формирования трехадресных команд.

1.5 Тема 5 Параллельная обработка стека и статическое распараллеливание в решающем поле (28 часов)

1.5.1. Перечень и краткое содержание рассматриваемых вопросов:

1 Подстеки и их взаимодействие.

В процессорах *супер-ЭВМ* используются многофункциональные *АЛУ*, состоящие из специализированных по операциям исполнительных устройств. Широкое применение микропроцессоров позволяет реализовать в составе *АЛУ решающие поля* на основе универсальных исполнительных устройств, которые могут выполнять последовательности команд или целые процедуры. Рассмотрим возможность динамического распараллеливания в таких *АЛУ* при выполнении арифметических операторов программы в безадресной системе команд процессора, воспроизводящей выполнение *работ* на стеке.

Существуют пути обобщения такой структуры *ВС* на основе комплектации многопроцессорных *ВС* на общем вычислительном ресурсе, - *решающем поле*. Такая

структура в наибольшей степени адекватна концепции двух основных уровней распараллеливания: уровня программ и уровня команд.

2 Статическое распараллеливание в АЛУ. VLIW- и EPIC-архитектуры.

С развитием архитектурной сложности *ВС* и с проектированием больших интегральных схем появилось новое направление в развитии архитектур — *использование принципа программного управления каждым тактом машины*. Это значит, что *программа* состоит из команд, задающих в такте выполнения инструкции каждому ИУ АЛУ. Команда изображается "длинным командным словом" (отсюда название *VLIW-архитектура, very long information word*). В нем предусмотрены позиции, соответствующие каждому ИУ. Они указывают, какую работу должно начать выполнять каждое ИУ. При этом учитывается состояние данного ИУ, временные соотношения для ранее инициированного поступления на него информации и т.д. Т.е. состояние всех устройств процессора планируется программно, статически, не оставляя элементов динамического планирования, как рассматривалось выше.