

## Лекция. КОИУ повышенной производительности

1. Цель и задачи распараллеливания вычислений
2. Уровни параллелизма и классификация комплексов параллельной обработки информации

### 1. Цель и задачи распараллеливания вычислений

История развития автоматизации различных процессов связана, во-первых, с расширением областей применения ЭВМ и, во-вторых, с повышением их качества.

Из всех показателей качества в большинстве случаев основными являются показатели *производительности* и *надежности*. Это в равной мере можно отнести и к машинам общего назначения, и к специализированным ЭВМ. Среди названных показателей доминирующими могут стать либо те, либо другие в зависимости от ориентации ЭВМ, сложности и важности решаемых задач. Так, для ЭВМ, предназначенных для проведения научных исследований, критерием качества, как правило, является производительность, для ЭВМ, функционирующих в экстремальных условиях – надежность. Однако рассматриваемые показатели не являются независимыми. Нетрудно показать, что при прочих равных условиях более быстродействующая ЭВМ обеспечивает и большую надежность.

Действительно такая ЭВМ позволяет уменьшить время решения задачи и, следовательно, его отношение к средней наработке на отказ, что равносильно увеличению последней. В результате возрастает вероятность успешного, своевременного решения задачи. С другой стороны, большая производительность может допустить многократное решение задачи или ее части в случае возникновения сбоя, не выходя за рамки предельного времени получения результатов. Кроме того, избыток производительности позволяет повысить интенсивность проведения тестового контроля, тем самым, увеличивая вероятность выявления потенциальных отказов.

Приведенный перечень механизмов повышения надежности за счет роста производительности может быть продолжен. Но и отмеченного достаточно для иллюстрации тесной взаимосвязи этих показателей и необходимости повышения производительности.

Аналогичным образом можно показать влияние производительности и на другие показатели качества, в частности – объем оперативной памяти. В известном смысле рост производительности можно уподобить увеличению емкости памяти и этим компенсировать ее недостаток.

Помимо рассмотренных мотивов повышения производительности, которые целесообразно отнести к косвенным, следует назвать прямые. Они приводят к безусловной необходимости поиска путей улучшения названного показателя.

В первую очередь здесь следует указать на повышение оперативности решения задач, что приводит к снижению длительности цикла управления и поэтому актуально для компьютерных систем любого типа. Действительно, способность решения задач, являющаяся основным потребительским свойством

вычислительных систем, главным образом определяется показателями производительности.

Так, точность решения некоторых классов задач, существенно зависит от времени их решения. Производительность в этом случае становится принципиальным ограничителем. Она же регламентирует периодичность решения многих задач, ибо недостаток производительности может привести к неудовлетворению требования решения подобных задач всякий раз, когда изменения обстановки оказываются достаточно существенными.

Во-вторых, постоянное совершенствование задач управления неизбежно приводит к усложнению алгоритмов их решения. Это, как правило, связано с увеличением потребного объема вычислений, и для сохранения прежних показателей оперативности необходимо увеличивать производительность ЭВМ.

В-третьих, высокопроизводительные ЭВМ позволяют ослабить требования к качеству программного обеспечения. Выражается это в допустимости разработки программ с меньшей степенью минимизации времени выполнения и объема занимаемой памяти. Учитывая, что стоимость программного обеспечения многократно превосходит стоимость аппаратных средств, снижение требований к программам имеет весьма важное значение. В частности, удастся не только уменьшить затраты на создание заданной совокупности программ, но и сократить сроки их разработки. Кроме того, к созданию программного обеспечения может быть привлечен персонал невысокой квалификации

Таким образом, совершенствование ЭВМ в процессе их эволюции, по существу, можно рассматривать как непрерывный поиск путей повышения их производительности.

В настоящее время известно большое число различных способов повышения производительности. Часть из них широко используется в современных ЭВМ и комплексах, другие носят характер перспективных и найдут применение в будущих образцах. Несмотря на большое многообразие этих способов, каждый из них в общем случае может быть отнесен к одной из трех групп, объединяющих соответственно физические, алгоритмические либо структурные способы.

*Физические способы* связаны с повышением быстродействия элементной базы и обеспечением комфортных условий применения ЭВМ. Совершенствование элементной базы имеет естественные пределы, ограниченные возможностями технологических процессов. Обеспечение комфортных условий не всегда возможно, особенно для ЭВМ, функционирующих в экстремальных ситуациях.

*Алгоритмические способы* обычно ассоциируют с минимизацией алгоритмической сложности выполняемых программ. К ним следует отнести способы совершенствования не только прикладных и системных процедур. А это, в свою очередь, связано с поиском новых методов и алгоритмов организации вычислительного процесса и решения прикладных задач, что также не всегда возможно.

*Структурные способы* связаны со следующими направлениями. Во-первых, с усложнением схемных решений на построение основных элементов ЭВМ. Во-вторых, с изменением традиционной архитектуры ЭВМ (развитие системы ввода-вывода, организация многоуровневой системы памяти). В-третьих,

с распараллеливанием вычислительных процедур, то есть с выделением в них независимых участков и их одновременным выполнением на различном оборудовании.

В ряде случаев более выгодным и перспективным является улучшение характеристик ЭВМ за счет их комплексирования и архитектурной модификации.

*Целью распараллеливания вычислений* является увеличение производительности ЭВМ за счет совмещения во времени независимых друг от друга процессов решения задачи или нескольких задач пользователей.

При распараллеливании вычислений решают следующие задачи, связанные с организацией вычислительного процесса:

1. Анализ алгоритмов и программ с целью выделения их частей (сегментов), допускающих параллельное выполнение на различных обрабатываемых устройствах (процессорах или ЭВМ).

2. Выбор адекватных средств реализации параллельных вычислений.

3. Распределение сегментов по обрабатываемым устройствам и организация их исполнения.

4. Оценка эффективности параллельного вычислительного процесса с целью его последующего совершенствования за счет изменения метода распределения ресурсов.

## **2. Уровни параллелизма и классификация комплексов параллельной обработки информации**

Уровни параллелизма, вносимого структурными способами в работу ЭВМ или, в общем случае – вычислительных систем, могут быть различными. Так, *физическое представление* вычислительных систем позволяет говорить о распараллеливании вычислительного процесса на уровне отдельных ЭВМ, устройств, блоков и узлов.

*Логическое представление* вычислительных систем позволяет говорить о параллелизме на уровне задач, команд либо данных.

При организации параллельной обработки на этих уровнях можно выделить три основных направления.

Первое направление связано с совмещением во времени различных этапов разных задач. Подобное совмещение возможно даже в однопроцессорных ЭВМ в режиме многопрограммной обработки и широко используется в настоящее время.

Основу второго направления составляет конвейерная обработка. Она также может быть организована в системе с одним обрабатываемым устройством, но разделенным на некоторое число последовательно включенных операционных блоков. Эти блоки должны иметь ориентацию на выполнение строго определенной части операции.

Реализация третьего направления связана с наличием в составе вычислительной системы нескольких обрабатываемых устройств. При этом используются особенности задач, позволяющие осуществить тот или иной параллелизм.

Если в вычислительную систему поступает непрерывный поток не связанных по данным задач, то налицо естественный параллелизм независимых задач.

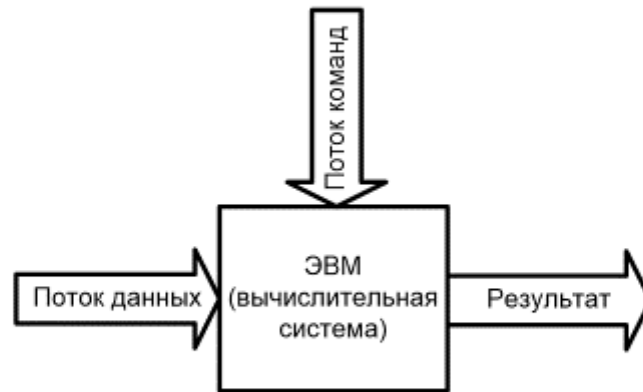
Частным случаем естественного параллелизма выступает параллелизм объектов или данных. Он имеет место тогда, когда по одной и той же программе обрабатывается некоторая совокупность данных о различных, но однотипных объектах, поступающая на обработку одновременно. Такие задачи сводятся к операциям над многомерными векторами, матрицами и аналогичными объектами. Операции над объектами могут выполняться параллельно и независимо друг от друга. В этом случае считается, что задача сама по себе обладает естественным параллелизмом.

Очевидно, что концепция комплекса задач не позволяет ориентироваться только на естественный параллелизм независимых задач при распараллеливании вычислительного процесса. В свою очередь, параллелизм объектов присущ довольно ограниченному кругу задач. Поэтому альтернативой естественному параллелизму выступает параллелизм независимых ветвей. Его сущность состоит в следующем. В исходной задаче стремятся выделить независимые части (ветви), которые при наличии нескольких обрабатывающих устройств могут выполняться независимо друг от друга. Независимость ветвей определяется отсутствием функциональных связей между ними, отсутствием связи по использованию памяти, независимостью по управлению и программной независимостью. Отсутствие функциональных связей предполагает, что ни одна из входных для одной ветви величин не является результирующей величиной для другой. Отсутствие связи по используемой памяти означает, что ни одна из ветвей не модифицирует общие данные. Независимость по управлению означает, что условия выполнения одной ветви не зависят от результатов или признаков выполнения другой ветви. Программная независимость предписывает необходимость выполнения обеих ветвей по разным блокам программы.

Практика исследований в области распараллеливания вычислений показывает, что к настоящему времени сложилось два подхода к организации параллельного вычислительного процесса. Первый связан с распараллеливанием обычных последовательных алгоритмов для их исполнения в вычислительных системах с числом обрабатывающих устройств (машин, процессоров), не превышающим нескольких единиц. Реализация второго подхода связана с построением специальных алгоритмов для вычислительных систем, содержащих десятки процессоров.

Описание вычислительной системы при распараллеливании вычислений только с физической или логической стороны не может считаться полным. В то же время описание на обоих уровнях делает его довольно громоздким. Поэтому на практике описанные признаки не нашли применения. Распространение получила классификация Флинна, различающая ЭВМ по типу потоков команд и данных и характеризующая любую ЭВМ только с точки зрения реализованного в ней параллелизма.

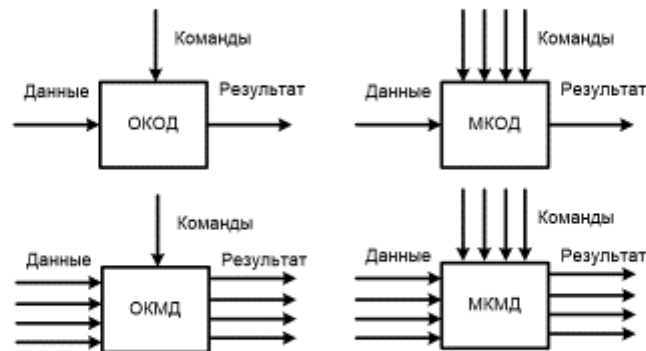
## Принцип классификации Флинна



Поток команд – одиночный (ОК) и множественный (МК)  
Поток данных – одиночный (ОД) и множественный (МД)

В этой системе введено по два типа потоков команд и данных – *одиночный* и *множественный* (ОК и МК для потоков команд, ОД и МД для потоков данных), позволяющих сформировать всего четыре класса ЭВМ.

## Классы параллельных вычислительных систем по Флинну



**ОКОД** – обычные ЭВМ  
**МКОД** и **ОКМД** – частичный параллелизм  
**МКМД** – полный параллелизм

Из них простейшим является класс ОКОД, соответствующий обычным ЭВМ. Удобство рассматриваемой классификации состоит в том, что никакой вид параллелизма, реализованный в такой машине, не способен перевести ее в разряд параллельных. Система оказывается чувствительной только к таким видам параллелизма, которые предусматривают одновременное функционирование различных операционных устройств.

Остальные три класса – МКОД, ОКМД и МКМД – идентифицируют так называемые параллельные ЭВМ. Причем первые два класса из трех названных

реализуют частичный или неполный параллелизм – либо на уровне команд, либо на уровне данных соответственно. Поэтому эффект от его организации достигается лишь на специальных задачах, что и послужило поводом считать ЭВМ классов МКОД и ОКМД специализированными.

Альтернативой рассмотренным являются ЭВМ класса МКМД, реализующие полный параллелизм и поэтому одинаково пригодные для любых задач и их комплексов и называемые супермашинами общего назначения или универсальными.