

## Глава 11 ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

### 11.1. Классификация вычислительных систем

Вычислительные машины за свою полувековую историю прошли стремительный и впечатляющий путь, отмеченный частыми сменами поколений ЭВМ. В этом процессе развития можно выявить целый ряд закономерностей:

- весь период развития средств электронной вычислительной техники отмечен доминирующей ролью классической структуры ЭВМ (структуры фон Неймана), основанной на методах последовательных вычислений;
- основным направлением совершенствования ЭВМ является неуклонный рост производительности (быстродействия) и интеллектуальности вычислительных средств;
- совершенствование ЭВМ осуществлялось в комплексе (элементно-конструкторская база, структурно-аппаратные решения, системно-программный и пользовательский, алгоритмический уровни);
- в настоящее время наметился кризис классической структуры ЭВМ, связанный с исчерпанием всех основных идей последовательного счета. Возможности микроэлектроники также не безграничны, давление пределов ощутимо и здесь.

Дальнейшее поступательное развитие вычислительной техники напрямую связано с переходом к параллельным вычислениям, с идеями построения многопроцессорных систем и сетей, объединяющих большое количество отдельных процессоров и (или) ЭВМ. Здесь появляются огромные возможности совершенствования средств вычислительной техники. Но следует отметить, что при несомненных практических достижениях в области параллельных вычислений до настоящего времени отсутствует их единая теоретическая база.

Термин *вычислительная система* появился в начале — середине 60-х гг. при создании ЭВМ третьего поколения. Это время знаменовалось переходом на новую элементную базу — интегральные схемы. Следствием этого явилось появление новых технических решений: разделение процессов обработки информации и ее ввода-вывода, множественный доступ и коллективное использование вычислительных ресурсов в пространстве и во времени. Появились сложные режимы работы ЭВМ — многопользовательская и многопрограммная обработка.

Отражая эти новшества, и появился термин «вычислительная система». Он не имеет единого толкования в литературе, его иногда даже используют применительно к однопроцессорным ЭВМ. Однако общим здесь является подчеркивание возможности построения параллельных ветвей в вычислениях, что не предусматривалось классической структурой ЭВМ.

Под *вычислительной системой* (ВС) будем понимать совокупность взаимосвязанных и взаимодействующих процессоров или ЭВМ, периферийного оборудования и программного обеспечения, предназначенную для сбора, хранения, обработки и распределения информации. Отличительной особенностью ВС по отношению к ЭВМ является наличие в них нескольких вычислителей, реализующих параллельную обработку. Создание ВС преследует следующие основные цели: повышение производительности системы за счет ускорения процессов обработки данных, повышение надежности и достоверности вычислений, предоставление пользователям дополнительных сервисных услуг и т.д.

Параллелизм в вычислениях в значительной степени усложняет управление вычислительным процессом, использование технических и программных ресурсов. Эти функции выполняет операционная система ВС.

Самыми важными предпосылками появления и развития вычислительных систем служат экономические факторы. Анализ характеристик ЭВМ различных поколений показал, что в пределах интервала времени, характеризующегося относительной стабильностью элементной базы, связь стоимости и производительности ЭВМ выражается квадратичной зависимостью — «законом Гроша»:

$$C_{\text{ЭВМ}} = K_1 \cdot P_{\text{ЭВМ}}^2.$$

Построение же вычислительных систем позволяет значительно сократить затраты, так как для них существует линейная формула:

$$C_{\text{ВС}} = K_2 \cdot \sum_{i=1}^n P_i,$$

где:

$C_{\text{ЭВМ}}, C_{\text{ВС}}$  — соответственно стоимость ЭВМ и ВС;

$K_1$  и  $K_2$  — коэффициенты пропорциональности, зависящие от технического уровня развития вычислительной техники;

$P_{\text{ЭВМ}}, P_i$  — производительность ЭВМ и  $i$ -го из  $n$  комплектующих вычислителей (ЭВМ или процессоров).

На рис. 11.1 представлены графики изменения стоимости вычислений для ЭВМ и ВС. Для каждого поколения ЭВМ и ВС существует критический порог сложности решаемых задач  $P_{\text{кр}}$ , после которого применение автономных ЭВМ становится экономически невыгодным, неэффективным. Критический порог определяется точкой пересечения двух приведенных зависимостей.

248

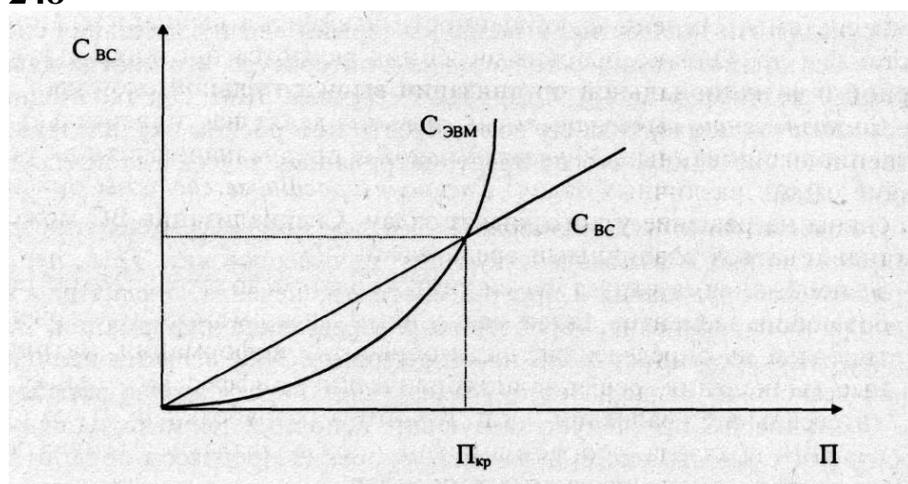


Рис. 11.1. Зависимость стоимости  $C_{\text{к}}$  и  $C_{\text{ЭВМ}}$  от производительности

Кроме выигрыша в стоимости технических средств, следует учитывать и дополнительные преимущества. Наличие нескольких вычислителей в системе

позволяет совершенно по-новому решать проблемы надежности, достоверности результатов обработки, резервирования, централизации хранения и обработки данных, децентрализации управления и т.д.

Основные принципы построения, закладываемые при создании ВС:

- возможность работы в разных режимах;
- модульность структуры технических и программных средств, что позволяет совершенствовать и модернизировать вычислительные системы без коренных их переделок;
- унификация и стандартизация технических и программных решений;
- иерархия в организации управления процессами;
- способность систем к адаптации, самонастройке и самоорганизации;
- обеспечение необходимым сервисом пользователей при выполнении вычислений.

В настоящее время накоплен большой практический опыт в разработке и использовании ВС самого разнообразного применения. Эти системы очень сильно отличаются друг от друга своими возможностями и характеристиками. Различия наблюдаются уже на уровне структуры.

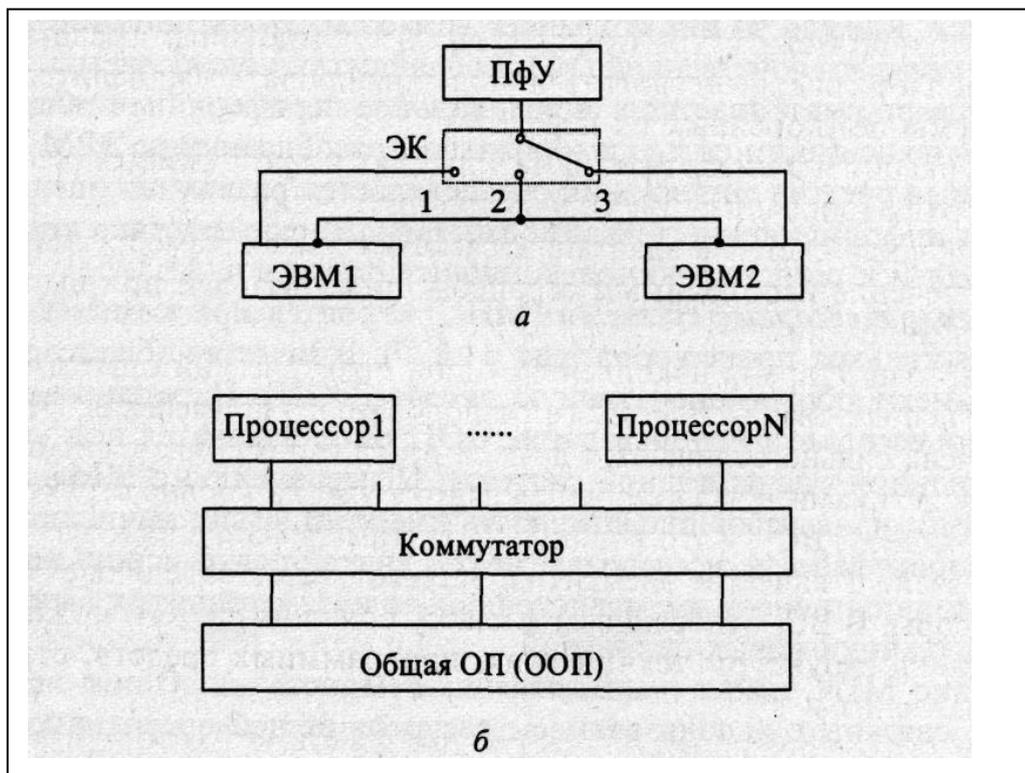
**Структура ВС** — это совокупность комплексированных элементов и их связей. В качестве элементов ВС выступают отдельные ЭВМ и процессоры. В ВС, относящихся к классу больших систем, можно рассматривать структуры технических, программных средств, структуры управления и т.д.

Существует большое количество признаков, по которым классифицируют вычислительные системы: по целевому назначению и выполняемым функциям, по типам и числу ЭВМ или процессоров, по архитектуре системы, режимам работы, методам управления элементами системы, степени разобщенности элементов вычислительной системы и др. Однако основными из них являются признаки структурной и функциональной организации вычислительной системы.

**По назначению вычислительные системы** делят на универсальные и специализированные. **Универсальные ВС** предназначаются для решения самых различных задач. **Специализированные системы** ориентированы на решение узкого класса задач. Специализация ВС может устанавливаться различными средствами:

- во-первых, сама структура системы (количество параллельно работающих элементов, связи между ними и т.д.) может быть ориентирована на определенные виды обработки информации: матричные вычисления, решение алгебраических, дифференциальных и интегральных уравнений и т.п. Практика разработки ВС типа суперЭВМ показала, что, чем выше их производительность, тем уже класс эффективно решаемых ими задач;
- во-вторых, специализация ВС может закладываться включением в их состав специального оборудования и специальных пакетов обслуживания техники.

**По типу вычислительные системы** можно разделить на многомашинные и многопроцессорные ВС. Исторически **многомашинные вычислительные системы** (ММС) появились первыми. Уже при использовании ЭВМ первых поколений возникали задачи повышения производительности, надежности и достоверности вычислений. Для этих целей использовали комплекс машин, схематически показанный на рис. 11.2,а



**Рис. 11.2.** Типы ВС: *а* — многомашинные комплексы; *б* — многопроцессорные системы

Положения 1 и 3 электронного ключа (ЭК) обеспечивали режим повышенной надежности. При этом одна из машин выполняла вычисления, а другая находилась в «горячем» или «холодном» резерве, т.е. в готовности заменить основную ЭВМ. Положение 2 электронного ключа соответствовало случаю, когда обе машины обеспечивали параллельный режим вычислений. Здесь возможны две ситуации:

а) обе машины решают одну и ту же задачу и периодически сверяют результаты решения. Тем самым обеспечивался режим повышенной достоверности, уменьшалась вероятность появления ошибок в результатах вычислений. Примерно по такой же схеме построены управляющие бортовые вычислительные комплексы космических аппаратов, ракет, кораблей. Например, американская космическая система «Шатл» содержала пять вычислительных машин, работающих по такой схеме;

б) обе машины работают параллельно, но обрабатывают собственные потоки заданий. Возможность обмена информацией между машинами сохраняется. Этот вид работы относится к режиму повышенной производительности. Она широко используется в практике организации работ на крупных вычислительных центрах, оснащенных несколькими ЭВМ высокой производительности.

Схема, представленная на рис. 11.2, *а*, была неоднократно повторена в различных модификациях при проектировании разнообразных специализированных ММС. Основные различия ММС заключаются, как правило, в организации связи и обмена информацией между ЭВМ комплекса. Каждая из них сохраняет возможность автономной работы и управляется собственной ОС. Любая другая подключаемая ЭВМ комплекса рассматривается как специальное периферийное оборудование. В зависимости от территориальной разобщенности ЭВМ и используемых средств

сопряжения обеспечивается различная оперативность их информационного взаимодействия. Характеристика возможных уровней и средств взаимодействия изложена в п. 11.3.

**Многопроцессорные системы** (МПС) строятся при комплексировании нескольких процессоров (рис. 11.2, б). В качестве общего ресурса они имеют общую оперативную память (ООП). Параллельная работа процессоров и использование ООП обеспечивается под управлением единой операционной системы. По сравнению с ММС здесь достигается наивысшая оперативность взаимодействия вычислителей-процессоров. Многие исследователи [27] считают, что использование МПС является основным магистральным путем развития вычислительной техники новых поколений.

Однако МПС имеет и существенные недостатки. Они в первую очередь связаны с использованием ресурсов общей оперативной памяти. При большом количестве комплекслируемых процессоров возможно возникновение конфликтных ситуаций, когда несколько процессоров обращаются с операциями типа «чтение» и «запись» к одним и тем же областям памяти. Помимо процессоров к ООП подключаются все каналы (процессоры ввода-вывода), средства измерения времени и т.д. Поэтому вторым серьезным недостатком МПС является проблема коммутации абонентов и доступа их к ООП. От того, насколько удачно решаются эти проблемы, и зависит эффективность применения МПС. Это решение обеспечивается аппаратно-программными средствами. Процедуры взаимодействия очень сильно усложняют структуру ОС МПС. Накопленный опыт построения подобных систем показал, что они эффективны при небольшом числе комплекслируемых процессоров (от 2—4 до 10). В отечественных системах «Эльбрус» обеспечивалась возможность работы до 10 процессоров, до 32 модулей памяти, до 4 процессоров ввода-вывода и до 16 процессоров связи.

Создание подобных коммутаторов представляет сложную техническую задачу, тем более, что они должны быть дополнены буферами для организации очередей запросов. Для разрешения конфликтных ситуаций необходимы схемы приоритетного обслуживания. До настоящего времени в номенклатуре технических средств ЭВТ отсутствуют высокоэффективные коммутаторы общей памяти.

**По типу ЭВМ или процессоров**, используемых для построения ВС, различают однородные и неоднородные системы. **Однородные системы** предполагают комплексирование однотипных ЭВМ (процессоров), неоднородные — разнотипных. В однородных системах значительно упрощается разработка и обслуживание технических и программных (в основном ОС) средств. В них обеспечивается возможность стандартизации и унификации соединений и процедур взаимодействия элементов системы. Упрощается обслуживание систем, облегчается модернизация и их развитие. Вместе с тем существуют и **неоднородные ВС**, в которых комплекслируемые элементы очень сильно отличаются по своим техническим и функциональным характеристикам. Обычно это связано с необходимостью параллельного выполнения многофункциональной обработки. Так, при построении ММС, обслуживающих каналы связи, целесообразно объединять в комплекс связные, коммуникационные машины и машины обработки данных. В таких системах коммуникационные ЭВМ выполняют функции связи, контроля получаемой и передаваемой информации, формирование пакетов задач и т.д. ЭВМ

обработки данных не занимаются не свойственными им работами по обеспечению взаимодействия в сети, а все их ресурсы переключаются на обработку данных. Неоднородные системы находят применение и МПС. Многие ЭВМ, в том числе и ПЭВМ, могут использовать сопроцессоры: десятичной арифметики, матричные и т.п.

*По степени территориальной разобщенности* вычислительных модулей ВС делятся на **системы совмещенного (сосредоточенного) и распределенного (разобщенного)** типов. Обычно такое деление касается только ММС. Многопроцессорные системы относятся к системам совмещенного типа. Более того, учитывая успехи микроэлектроники, это совмещение может быть очень глубоким. При появлении новых СБИС появляется возможность иметь в одном кристалле несколько параллельно работающих процессоров.

В совмещенных и распределенных ММС сильно различается оперативность взаимодействия в зависимости от удаленности ЭВМ. Время передачи информации между соседними ЭВМ, соединенными простым кабелем, может быть много меньше времени передачи данных по каналам связи. Как правило, все выпускаемые в мире ЭВМ имеют средства прямого взаимодействия и средства подключения к сетям ЭВМ. Для ПЭВМ такими средствами являются нуль-модемы, модемы и сетевые карты как элементы техники связи.

*По методам управления элементами ВС* различают **централизованные, децентрализованные и со смешанным управлением**. Помимо параллельных вычислений, производимых элементами системы, необходимо выделять ресурсы на обеспечение управления этими вычислениями. **В централизованных ВС** за это отвечает главная, или диспетчерская, ЭВМ (процессор). Ее задачей является распределение нагрузки между элементами, выделение ресурсов, контроль состояния ресурсов, координация взаимодействия. Централизованный орган управления в системе может быть жестко фиксирован, или эти функции могут передаваться другой ЭВМ (процессору), что способствует повышению надежности системы. Централизованные системы имеют более простые ОС. **В децентрализованных системах** функции управления распределены между ее элементами. Каждая ЭВМ (процессор) системы сохраняет известную автономию, а необходимое взаимодействие между элементами устанавливается по специальным наборам сигналов. С развитием ВС и, в частности, сетей ЭВМ интерес к децентрализованным системам постоянно растет.

В системах **со смешанным управлением** совмещаются процедуры централизованного и децентрализованного управления. Перераспределение функций осуществляется в ходе вычислительного процесса, исходя из сложившейся ситуации.

*По принципу закрепления вычислительных функций* за отдельными ЭВМ (процессорами) различают системы **с жестким и плавающим закреплением функций**. В зависимости от типа ВС следует решать задачи статического или динамического размещения программных модулей и массивов данных, обеспечивая необходимую гибкость системы и надежность ее функционирования.

*По режиму работы ВС* различают системы, работающие **в оперативном и неоперативном временных режимах**. Первые, как правило, используют режим реального масштаба времени. Этот режим характеризуется жесткими

ограничениями на время решения задач в системе и предполагает высокую степень автоматизации процедур ввода-вывода и обработки данных.

Наибольший интерес у исследователей всех рангов (проектировщиков, аналитиков и пользователей) вызывают структурные признаки ВС. От того, насколько структура ВС соответствует структуре решаемых на этой системе задач, зависит эффективность применения ЭВМ в целом. Структурные признаки, в свою очередь, отличаются многообразием: топология управляющих и информационных связей между элементами системы, способность системы к перестройке и перераспределению функций, иерархия уровней взаимодействия элементов. В наибольшей степени структурные характеристики определяются архитектурой системы.

## 11.2. Архитектура вычислительных систем

Большое разнообразие структур ВС затрудняет их изучение. Поэтому вычислительные системы классифицируют с учетом их обобщенных характеристик. С этой целью вводится понятие «архитектура системы».

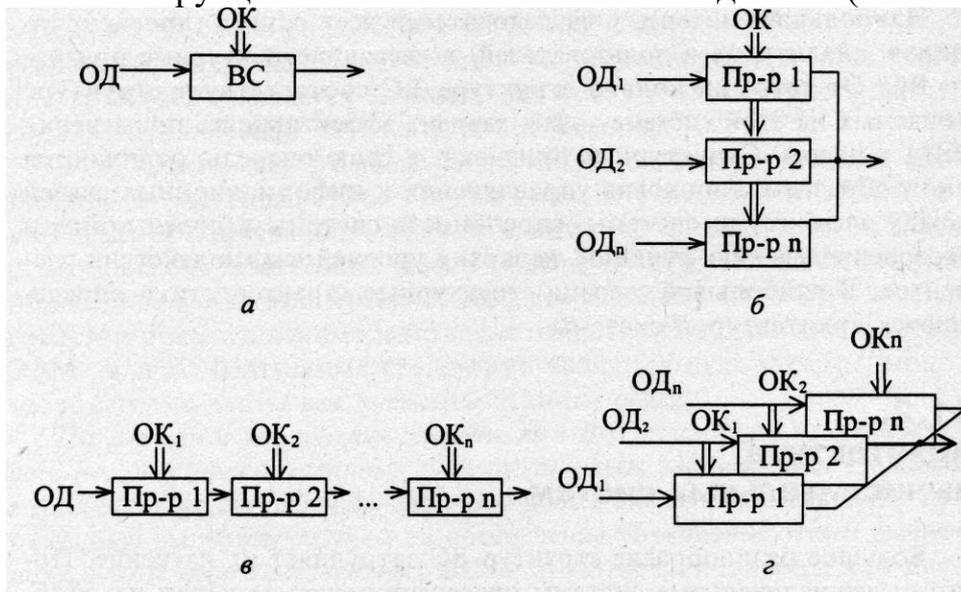
*Архитектура ВС* — совокупность характеристик и параметров, определяющих функционально-логическую и структурную организацию системы. Понятие архитектуры охватывает общие принципы построения и функционирования, наиболее существенные для пользователей, которых больше интересуют возможности систем, а не детали их технического исполнения. Поскольку ВС появились как параллельные системы, то и рассмотрим классификацию архитектур под этой точкой зрения.

Эта классификация архитектур была предложена М. Флинном (M. Flynn) в начале 60-х гг. В ее основу заложено два возможных вида параллелизма: *независимость потоков заданий (команд)*, существующих в системе, и *независимость (несвязанность) данных*, обрабатываемых в каждом потоке. Классификация до настоящего времени еще не потеряла своего значения. Однако подчеркнем, что, как и любая классификация, она носит временный и условный характер. Своим долголетием она обязана тому, что оказалась справедливой для ВС, в которых ЭВМ и процессоры реализуют программные последовательные методы вычислений. С появлением систем, ориентированных на потоки данных и использование ассоциативной обработки, данная классификация может быть некорректной.

Согласно этой классификации существует четыре основных архитектуры ВС, представленных на рис. 11.3:

- одиночный поток команд — одиночный поток данных (ОКОД), в английском варианте — Single Instruction Single Data (SISD) — одиночный поток инструкций — одиночный поток данных;
- одиночный поток команд — множественный поток данных (ОКМД), или Single Instruction Multiple Data (SIMD) — одиночный поток инструкций — одиночный поток данных;
- множественный поток команд — одиночный поток данных (МКОД), или Multiple Instruction Single Data (MISD) — множественный поток инструкций — одиночный поток данных;

- множественный поток команд — множественный поток данных (МКМД), или Multiple Instruction Multiple Data (MIMD) — множественный поток инструкций — множественный поток данных (MIMD).



**Рис. 11.3.** Архитектура ВС: *а* — ОКОД (SISD)-архитектура; *б* — ОКМД (SIMD)-архитектура; *в* — МКОД (MISD)-архитектура; *г* — МКМД (MIMD)-архитектура

Коротко рассмотрим отличительные особенности каждой из архитектур.

**Архитектура ОКОД** охватывает все однопроцессорные и одномашинные варианты систем, т.е. с одним вычислителем. Все ЭВМ классической структуры попадают в этот класс. Здесь параллелизм вычислений обеспечивается путем совмещения выполнения операций отдельными блоками АЛУ, а также параллельной работы устройств ввода-вывода информации и процессора. Закономерности организации вычислительного процесса в этих структурах достаточно хорошо изучены.

**Архитектура ОКМД** предполагает создание структур векторной или матричной обработки. Системы этого типа обычно строятся как однородные, т.е. процессорные, элементы, входящие в систему, идентичны, и все они управляются одной и той же последовательностью команд. Однако каждый процессор обрабатывает свой поток данных. Под эту схему хорошо подходят задачи обработки матриц или векторов (массивов), задачи решения систем линейных и нелинейных, алгебраических и дифференциальных уравнений, задачи теории поля и др. В структурах данной архитектуры желательно обеспечивать соединения между процессорами, соответствующие реализуемым математическим зависимостям. Как правило, эти связи напоминают матрицу, в которой каждый процессорный элемент связан с соседними.

По этой схеме строились системы: первая суперЭВМ — ILLIAC-IV, отечественные параллельные системы — ПС-2000, ПС-3000. Идея векторной обработки широко использовалась в таких известных суперЭВМ, как Cyber-205 и Gray-I, II, III. Узким местом подобных систем является необходимость изменения коммутации между процессорами, когда связь между ними отличается от матричной. Кроме того, задачи, допускающие широкий матричный параллелизм,

составляют достаточно узкий класс задач. Структуры ВС этого типа, по существу, являются структурами специализированных суперЭВМ.

*Третий тип архитектуры МКОД* предполагает построение своеобразного процессорного конвейера, в котором результаты обработки передаются от одного процессора к другому по цепочке. Выгоды такого вида обработки понятны. Прототипом таких вычислений может служить схема любого производственного конвейера. В современных ЭВМ по этому принципу реализована схема совмещения операций, в которой параллельно работают различные функциональные блоки, и каждый из них делает свою часть в общем цикле обработки команды.

В ВС этого типа конвейер должны образовывать группы процессоров. Однако при переходе на системный уровень очень трудно выявить подобный регулярный характер в универсальных вычислениях. Кроме того, на практике нельзя обеспечить и «большую длину» такого конвейера, при которой достигается наивысший эффект. Вместе с тем конвейерная схема нашла применение в так называемых скалярных процессорах суперЭВМ, в которых они применяются как специальные процессоры для поддержки векторной обработки.

*Архитектура МКМД* предполагает, что все процессоры системы работают по своим программам с собственным потоком команд. В простейшем случае они могут быть автономны и независимы. Такая схема использования ВС часто применяется на многих крупных вычислительных центрах для увеличения пропускной способности центра. Большой интерес представляет возможность согласованной работы ЭВМ (процессоров), когда каждый элемент делает часть общей задачи. Общая теоретическая база такого вида работ практически отсутствует. Но можно привести примеры большой эффективности этой модели вычислений. Подобные системы могут быть многомашинными и многопроцессорными. Например, отечественный проект машины динамической архитектуры (МДА) — ЕС-2704, ЕС-2127 — позволял одновременно использовать сотни процессоров.

### **11.3. Комплексование в вычислительных системах**

Для построения вычислительных систем необходимо, чтобы элементы или модули, комплексируемые в систему, были совместимы. Понятие совместимости имеет три аспекта: аппаратный, или технический, программный и информационный. Техническая (HardWare) совместимость предполагает, что еще в процессе разработки аппаратуры обеспечиваются следующие условия:

- подключаемая друг к другу аппаратура должна иметь единые стандартные, унифицированные средства соединения: кабели, число проводов в них, единое назначение проводов, разъемы, заглушки, адаптеры, платы и т.д.;
- параметры электрических сигналов, которыми обмениваются технические устройства, тоже должны соответствовать друг другу: амплитуды импульсов, полярность, длительность и т.д.;
- алгоритмы взаимодействия (последовательности сигналов по отдельным проводам) не должны вступать в противоречие друг с другом.

Последний пункт тесно связан с программной совместимостью. Программная совместимость (SoftWare) требует, чтобы программы, передаваемые из одного технического средства в другое (между ЭВМ, процессорами, между процессорами и

внешними устройствами), были правильно поняты и выполнены другим устройством.

Если обменивающиеся устройства идентичны друг другу, то проблем обычно не возникает. Если взаимодействующие устройства относятся к одному и тому же семейству ЭВМ, но стыкуются разные модели (например, ПК на базе i286 и Pentium), то в таких моделях совместимость обеспечивается «снизу вверх», т.е. ранее созданные программы могут выполняться на более поздних моделях, но не наоборот. Если стыкуемая аппаратура имеет совершенно разную систему команд, то следует обмениваться исходными модулями программ с последующей их трансляцией.

Информационная совместимость комплексируемых средств предполагает, что передаваемые информационные массивы будут одинаково интерпретироваться стыкуемыми модулями ВС. Должны быть стандартизированы алфавиты, разрядность, форматы, структура и разметка файлов, томов и т.д.

В создаваемых ВС стараются обеспечить несколько путей передачи данных, что позволяет достичь необходимой надежности функционирования, гибкости и адаптируемости к конкретным условиям работы. Эффективность обмена информацией определяется скоростью передачи и возможными объемами данных, передаваемыми по каналу взаимодействия. Эти характеристики зависят от средств, обеспечивающих взаимодействие модулей и уровня управления процессами, на котором это взаимодействие осуществляется. Сочетание различных уровней и методов обмена данными между модулями ВС в наиболее полной форме нашло свое выражение в универсальных суперЭВМ и больших ЭВМ, в которых сбалансированно использовались все методы достижения высокой производительности. В этих машинах предусматривались следующие уровни комплексования (рис. 11.4):

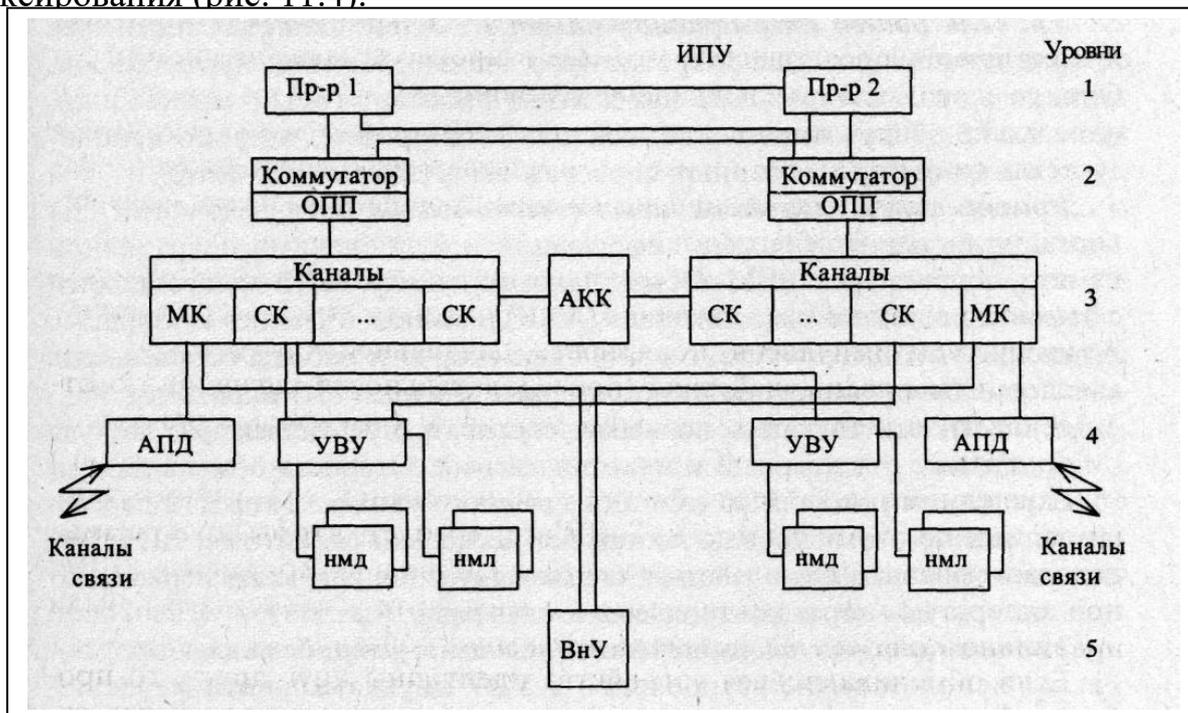


Рис. 11.4. Уровни и средства комплексования

1) прямого управления (процессор — процессор);

- 2) общей оперативной памяти;
- 3) комплекслируемых каналов ввода-вывода;
- 4) устройств управления внешними устройствами (УВУ);
- 5) общих внешних устройств.

На каждом из этих уровней используются специальные технические и программные средства, обеспечивающие обмен информацией.

**Уровень прямого управления** служит для передачи коротких однобайтовых приказов-сообщений. Последовательность взаимодействия процессоров сводится к следующему. Процессор-инициатор обмена по интерфейсу прямого управления (ИПУ) передает в блок прямого управления байт-сообщение и подает команду «Прямая запись». У другого процессора эта команда вызывает прерывание, относящееся к классу внешних. В ответ он вырабатывает команду «Прямое чтение» и записывает передаваемый байт в свою память. Затем принятая информация расшифровывается и по ней принимается решение. После завершения передачи прерывания снимаются, и оба процессора продолжают вычисления по собственным программам. Видно, что уровень прямого управления не может использоваться для передачи больших массивов данных, однако оперативное взаимодействие отдельными сигналами широко используется в управлении вычислениями. У ПЭВМ типа IBM PC этому уровню соответствует комплексирование процессоров, подключаемых к системной шине.

**Уровень общей оперативной памяти** (ООП) является наиболее предпочтительным для оперативного взаимодействия процессоров. Однако в этом случае ООП эффективно работает только при небольшом числе обслуживаемых абонентов. Этот уровень широко используется в многопроцессорных серверах вычислительных сетей.

**Уровень комплекслируемых каналов ввода-вывода** предназначается для передачи больших объектов информации между блоками оперативной памяти, сопрягаемых ЭВМ. Обмен данными между ЭВМ осуществляется с помощью адаптера «канал-канал» (АКК) и команд «Чтение» и «Запись». Адаптер — это устройство, согласующее скорости работы сопрягаемых каналов. Обычно сопрягаются селекторные каналы (СК) машин как наиболее быстродействующие, но можно сопрягать мультиплексные каналы (МК), а также селекторный и мультиплексный. Скорость обмена данными определяется скоростью самого медленного канала. Скорость передачи данных по этому уровню составляет несколько Мбайтов/с. В ПЭВМ данному уровню взаимодействия соответствует подключение периферийной аппаратуры через контроллеры и адаптеры.

**Уровень устройств управления внешними устройствами** предполагает использование встроенного в УВУ двухканального переключателя и команд «Зарезервировать» и «Освободить». Двухканальный переключатель позволяет подключать УВУ одной машины к селекторным каналам различных ЭВМ. По команде «Зарезервировать» канал-инициатор обмена имеет доступ через УВУ к любым накопителям на дисках НМД или на магнитных лентах НМЛ. На рис. 11.4 схематически показано, что они управляются одним УВУ. На самом деле УВУ магнитных дисков и лент — совершенно различные устройства. Обмен канала с накопителями продолжается до полного завершения работ и получения команды «Освободить». Лишь после этого УВУ может подключиться к конкурирующему

каналу. Только такая дисциплина обслуживания требований позволяет избежать конфликтных ситуаций. Этот уровень целесообразно использовать в вычислительных сетях при построении больших банков данных.

Пятый *уровень* предполагает использование *общих внешних устройств*. Для подключения отдельных устройств используется автономный двухканальный переключатель.

Пять уровней комплексирования получили название *логических* потому, что они объединяют на каждом уровне разнотипную аппаратуру, имеющую сходные методы управления. Каждое из устройств может иметь логическое имя, используемое в прикладных программах. Этим достигается независимость программ пользователей от конкретной физической конфигурации системы. Связь логической структуры программы и конкретной физической структуры ВС обеспечивается операционной системой по указаниям-директивам пользователя, при генерации ОС и по указаниям диспетчера-оператора вычислительного центра. Различные уровни комплексирования позволяют создавать самые различные структуры ВС.

Второй логический уровень позволяет создавать многопроцессорные ВС. Обычно он дополняется и первым уровнем, что повышает оперативность взаимодействия процессоров. Вычислительные системы сверхвысокой производительности должны строиться как многопроцессорные. Центральным блоком такой системы является быстродействующий коммутатор, обеспечивающий необходимые подключения абонентов (процессоров и каналов) к общей оперативной памяти.

Уровни 1, 3, 4, 5 обеспечивают построение разнообразных машинных комплексов. Особенно часто используется третий в комбинации с четвертым. Целесообразно их дополнять и первым уровнем.

Пятый уровень комплексирования используется в редких специальных случаях, когда в качестве внешнего объекта используется какое-то дорогое уникальное устройство. В противном случае этот уровень малоэффективен. Любое внешнее устройство — это недостаточно надежное устройство точной механики, а значит, выгоднее использовать четвертый уровень комплексирования, когда можно сразу управлять не одним, а несколькими внешними устройствами, включая и резервные.

Сочетание уровней и методов взаимодействия позволяет создавать самые различные многомашинные и многопроцессорные системы.

#### **11.4. Типовые структуры вычислительных систем**

С момента появления первых систем было опробовано большое количество разнообразных структур систем, отличающихся друг от друга различными техническими решениями. Практика показала, что каждая структура вычислительной системы эффективно обрабатывает лишь задачи определенного класса. При этом необходимо, чтобы структура вычислительной системы максимально соответствовала структуре решаемых задач. Только в этом случае система обеспечивает максимальную производительность.

Универсальной структуры вычислительной системы, одинаково хорошо обрабатывающей задачи любого типа, не существует. Интересные результаты

исследований по этим вопросам даны в работах [14, 19, 20]. В них приведены сопоставления различных видов программного параллелизма и соответствующих им структур вычислительных систем.

**Классификация уровней программного параллелизма включает шесть позиций:** независимые задания, отдельные части заданий, программы и подпрограммы, циклы и итерации, операторы и команды, фазы отдельных команд. Для каждой из них имеются специфические свойства параллельной обработки, апробированные в различных структурах вычислительных систем. Заметим, что данный перечень совершенно не затрагивает этапы выбора алгоритмов решения, на которых могут анализироваться альтернативные алгоритмы (а значит, и программы), дающие различные результаты.

Для каждого вида параллельных работ имеются структуры вычислительных средств, используемые в различных вычислительных системах. Верхние три уровня, включающие независимые задания, шаги или части заданий и отдельные программы, имеют единое средство параллельной обработки — мультипроцессирование, т.е. многопроцессорные вычислительные системы, относящиеся к архитектуре МКМД. Программные циклы и итерации требуют использования векторной обработки (архитектура ОКМД). Операторы и команды, выполняемые ЭВМ, ориентированы на многофункциональную обработку. Параллельная обработка фаз последовательно выполняемых команд приводит к организации конвейера команд.

Рассмотрим возможные структуры вычислительных систем, которые обеспечивают перечисленные виды программного параллелизма.

**ОКОД-структуры.** Два нижних вида параллелизма реализуются в любых современных ЭВМ, включая и персональные ЭВМ. Данный тип архитектуры объединяет любые системы в однопроцессорном (одномашинном) варианте.

За 50 лет развития электронной вычислительной техники классическая структура ЭВМ претерпела значительные усовершенствования, однако основной принцип программного управления не был нарушен. Данная структура оказалась сосредоточенной вокруг оперативной памяти, так как именно цепь «процессор — оперативная память» во многом определяет эффективную работу компьютера. При выполнении каждой команды необходимо неоднократное обращение к оперативной памяти: выбор команды, операндов, отсылка результатов и т.д.

Можно перечислить большое число приведенных улучшений классической структуры ЭВМ, ставших в настоящее время определенными стандартами при построении новых ЭВМ: иерархическое построение памяти ЭВМ, появление сверхоперативной памяти и кэш-памяти, относительная и косвенная адресация памяти, разделение процессоров ввода-вывода и обработки задач, появление систем прерывания и приоритетов и т.д.

В этом ряду следует рассматривать и организацию конвейера последовательно выполняемых команд: формирование адреса команды, выбор команды, формирование адресов и выбор операндов, выполнение команды, запись результата. Однако примитивная организация памяти (память линейна и одномерна) не позволяет организовать длинный и эффективный конвейер. Линейные участки современных программ редко превышают десяток, полтора последовательно выпол-

няемых команд. Поэтому конвейер часто перезапускается, что снижает производительность ЭВМ в целом.

Многофункциональная обработка также нашла свое место при построении ЭВМ. Например, даже в персональных ЭВМ, построенных на микропроцессорах i486 и Pentium, в состав ЭВМ могут включаться и другие специализированные средства обработки: умножители, делители, сопроцессоры или блоки десятичной арифметики, сопроцессоры обработки графической информации и др. Все они совместно с центральным процессором ЭВМ позволяют создавать своеобразные микроконвейеры, целью которых является повышение скорости вычислений.

В последние годы широко используются еще несколько модификаций классической структуры. В связи с успехами микроэлектроники появилась возможность построения RISC-компьютеров (Reduced Instruction Set Computing), т.е. ЭВМ с сокращенным набором команд.

Большие ЭВМ предыдущих поколений не имели большой сверхоперативной памяти, поэтому они имели достаточно сложную систему CISC-команд (Complete Instruction Set Computing — вычисления с полной системой команд). В этих машинах большую долю команд составляли команды типа «Память-память», в которых операнды и результаты операций находились в оперативной памяти. Время обращения к памяти и время вычислений соотносились примерно, как 5:1. В RISC-машинах с большой сверхоперативной памятью большой удельный вес составляют операции «регистр-регистр» и отношение времени обращения к памяти и времени вычислений составляет 2:1 [27].

Поэтому в RISC-ЭВМ основу системы команд составляют наиболее употребительные, «короткие» операции типа алгебраического сложения. Сложные операции выполняются как подпрограммы, состоящие из простых операций. Это позволяет значительно упростить внутреннюю структуру процессора, уменьшить фазы дробления конвейерной обработки и увеличить частоту работы конвейера. Но здесь необходимо отметить, что за эффект приходится расплачиваться усложнением процедур обмена данными между регистрами сверхоперативной памяти и кэш-памяти с оперативной памятью.

Другой модификацией классической структуры ЭВМ является VLIW (Very Large Instruction Word) — ЭВМ с «очень длинным командным словом». ЭВМ этого типа выбирает из памяти суперкоманду, включающую несколько команд. Здесь возможны варианты.

В самом простом случае это приводит к появлению буфера команд (кэш-команд) с целью ускорения конвейера операций. В более сложных случаях в состав суперкоманд стараются включать параллельные команды, не связанные общими данными. Если процессор ЭВМ при этом построен из функционально независимых устройств (устройства алгебраического сложения, умножения, сопроцессоры), то в этом случае обеспечивается максимальный эффект работы ЭВМ. Но это направление связано с кардинальной перестройкой процессов трансляции и исполнения программ. Здесь значительно усложняются средства автоматизации программирования.

VLIW-компьютеры могут выполнять суперскалярную обработку, т.е. одновременно выполнять две или более команды. В целом ряде структур суперЭВМ использовалась эта идея. Отметим, что и в ПЭВМ последних выпусков имеется

возможность выполнения двух команд одновременно. Эта реализация имеет две цели:

- уменьшение отрицательного влияния команд ветвления вычислительного процесса путем выполнения независимых команд двух различных ветвей программы. При этом в какой-то степени исключаются срывы конвейера в обработке команд программы;
- одновременное выполнение двух команд (независимых по данным и регистрам их хранения), например команды пересылки и арифметические операции.

**ОКМД-структуры.** Для реализации программного параллелизма, включающего циклы и итерации, используются матричные или векторные структуры. В них эффективно решаются задачи матричного исчисления, задачи решения систем алгебраических и дифференциальных уравнений, задачи теории поля, геодезические задачи, задачи аэродинамики. Теоретические проработки подобных структур относятся к концу 50-х — 60-м гг. Данные структуры очень хорошо зарекомендовали себя при решении перечисленных задач, но они получились очень дорогими по стоимости и эксплуатации. Кроме того, в тех случаях, когда структура параллелизма отличалась от матричной, возникает необходимость передачи данных между процессорами через коммутаторы. При этом эффективность вычислений резко снижается. Подобные структуры могут использоваться как сопроцессоры в системах будущих поколений.

**МКОД-структуры** большой практической реализации не получили. Задачи, в которых несколько процессоров могли бы эффективно обрабатывать один поток данных, в науке и технике неизвестны. С некоторой натяжкой к этому классу можно отнести фрагменты многофункциональной обработки, например обработку на разных процессорах команд с фиксированной и плавающей точкой.

Так же как фрагмент такой структуры, можно рассматривать локальную сеть персональных компьютеров, работающих с единой базой данных, но скорее всего это — частный случай использования МКМД-структуры.

**МКМД-структуры** являются наиболее интересным классом структур вычислительных систем. После разочарований в структурах суперЭВМ, основанных на различном сочетании векторной и конвейерной обработки, усилия теоретиков и практиков сосредоточены в этом направлении.

Уже из названия МКМД-структур видно, что в данных системах можно найти все перечисленные виды параллелизма. Этот класс дает большое разнообразие структур, сильно отличающихся друг от друга своими характеристиками (рис. 11.5).

Важную роль здесь играют способы взаимодействия ЭВМ или процессоров в системе. **В сильносвязанных системах** достигается высокая оперативность взаимодействия процессоров посредством общей оперативной памяти. При этом пользователь имеет дело с многопроцессорными вычислительными системами. Наиболее простыми по строению и организации функционирования являются **однородные, симметричные структуры**. Они обеспечивают простоту подключения процессоров и не требуют очень сложных централизованных операционных систем, размещаемых на одном из процессоров.

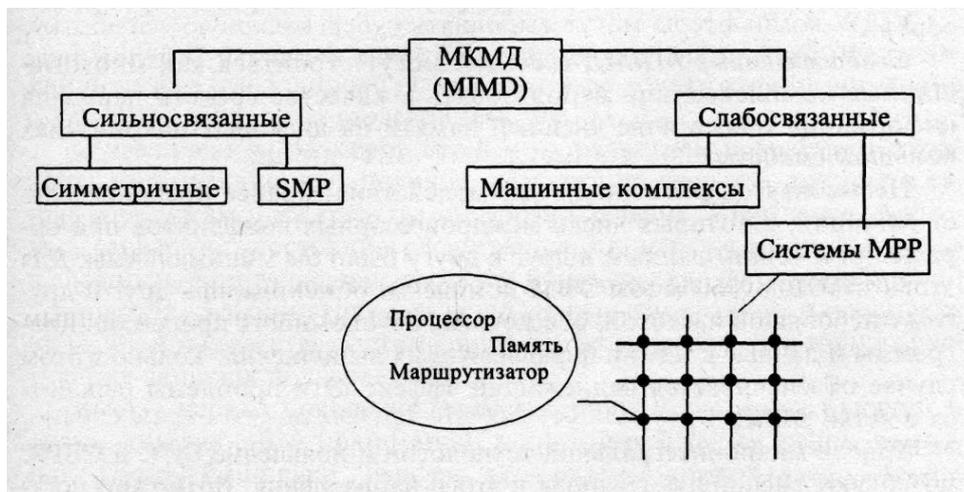


Рис. 11.5. Типовые структуры ВС в МКМД (MIMD)-классе

Однако при построении таких систем возникает много проблем с использованием общей оперативной памяти. Число комплексируемых процессоров не должно превышать 16. Для уменьшения числа обращений к памяти и конфликтных ситуаций может использоваться многоблочное построение ОП, функциональное закрепление отдельных блоков за процессорами, снабжение комплексируемых процессоров собственной памятью типа кэш. Но все эти методы не решают проблемы повышения производительности ВС в целом. Аппаратные затраты при этом существенно возрастают, а производительность систем увеличивается незначительно.

Появление мощных микропроцессоров типа Pentium привело к экспериментам по созданию многопроцессорных систем на их основе. Так, для включения мощных серверов в локальные сети персональных компьютеров была предложена несколько измененная структура использования ООП — *мультипроцессорирование с разделением памяти* (Shared Memory multiprocessing, SMP). На общей шине оперативной памяти можно комплексовать до четырех микропроцессоров.

**Слабосвязанные МКМД-системы** могут строиться как многомашинные комплексы или использовать в качестве средств передачи информации общее поле внешней памяти на дисковых накопителях большой емкости.

Невысокая оперативность взаимодействия заранее предопределяет ситуации, в которых число межпроцессорных конфликтов при обращении к общим данным и друг к другу было бы минимальным. Для этого необходимо, чтобы ЭВМ комплекса обменивались друг с другом с небольшой частотой, обеспечивая автономность процессов (программы и данные к ним) и параллелизм их выполнения. Только в этом случае обеспечивается надлежащий эффект. Эти проблемы решаются в сетях ЭВМ.

Успехи микроинтегральной технологии и появление БИС и СБИС позволяют расширить границы и этого направления. Возможно построение систем с десятками, сотнями и даже тысячами процессорных элементов, с размещением их в непосредственной близости друг от друга. Если каждый процессор системы имеет собственную память, то он также будет сохранять известную автономию в вычислениях. Считается, что именно такие системы займут доминирующее положение в мире компьютеров в ближайшие десять — пятнадцать лет. Подобные ВС

получили название *систем с массовым параллелизмом* (Mass-Parallel Processing, MPP).

Все процессорные элементы в таких системах должны быть связаны единой коммутационной средой. Нетрудно видеть, что здесь возникают проблемы, аналогичные ОКМД-системам, но на новой технологической основе.

Передача данных в MPP-системах предполагает обмен не отдельными данными под централизованным управлением, а подготовленными процессами (программами вместе с данными). Этот принцип построения вычислений уже не соответствует принципам программного управления классической ЭВМ. Передача данных процесса по его готовности больше соответствует принципам построения «поточковых машин» (машин, управляемых потоками данных). Подобный подход позволяет строить системы с громадной производительностью и реализовывать проекты с любыми видами параллелизма, например перейти к «систолическим вычислениям» с произвольным параллелизмом. Однако для этого необходимо решить целый ряд проблем, связанных с описанием и программированием коммутаций процессов и управления ими. Математическая база этой науки в настоящее время практически отсутствует.

### 11.5. Кластеры

Вычислительные системы как мощные средства обработки заданий пользователей широко используются не только автономно, но и в сетях ЭВМ в качестве серверов (см. рис. 11.5).

С увеличением размеров сетей и их развитием возрастают плотность информационных потоков, нагрузка на средства доступа к сетевым ресурсам и на средства обработки заданий. Круг задач, решаемых серверами, постоянно расширяется, становится многообразным и сложным. Чем выше ранг сети, тем более специализированными они становятся. Администраторы сетей должны постоянно наращивать их мощь и количество, оптимизируя характеристики сети под возрастающие запросы пользователей.

В сетях первых поколений серверы строились на основе больших и очень дорогих ЭВМ (mainframe), выпускаемых целым рядом компаний: Digital Equipment, Tandem, влившихся в корпорацию Compaq, IBM, Hewlett-Packard. Все они работали под управлением ОС Unix и способны были объединяться для совместной работы.

Как и во всякой развивающейся технологии, сложные универсальные серверы различных фирм-изготовителей должны были уступить место стандартным массовым решениям. Успехи микроэлектроники, повсеместное применение ПЭВМ, широкое распространение Internet/ Intranet-технологий позволили перейти к более простым и дешевым системам, например, на основе платформы Wintel. Опыт создания серверов на основе SMP- и MPP-структур показал, что они не обеспечивают хорошей адаптации к конкретным условиям функционирования, остаются дорогими и сложными в эксплуатации.

Одним из перспективных направлений здесь является *кластеризация*, т.е. технология, с помощью которой несколько серверов, сами являющиеся вычислительными системами, объединяются в единую систему более высокого ранга для повышения эффективности функционирования системы в целом.

Целями построения кластеров могут служить:

- улучшение масштабируемости (способность к наращиванию мощности);
- повышение надежности и готовности системы в целом;
- увеличение суммарной производительности;
- эффективное перераспределение нагрузок между компьютерами кластера;
- эффективное управление и контроль работы системы и т.п.

**Улучшение масштабируемости** или способность к наращиванию мощности предусматривает, что все элементы кластера имеют аппаратную, программную и информационную совместимость. В сочетании с простым и эффективным управлением изменение оборудования в идеальном кластере должно обеспечивать соответствующее изменение значений основных характеристик, т.е. добавление новых процессоров, дисковых систем должно сопровождаться пропорциональным ростом производительности, надежности и т.п. В реальных системах эта зависимость имеет нелинейный характер.

Масштабируемость SMP- и MPP-структур достаточно ограничена. При большом числе процессоров в SMP-структурах возрастает число конфликтов при обращении к общей памяти, а в MPP-структурах плохо решаются задачи преобразования и разбиения приложений на отдельные задания процессорам. В кластерах же администраторы сетей получают возможность увеличивать пропускную способность сети за счет включения в них дополнительных серверов, даже уже из числа работающих, с учетом того, что балансировка и оптимизация нагрузки будут выполняться автоматически.

Следующей важной целью создания кластера является **повышение надежности и готовности системы в целом**. Именно эти качества способствуют популярности и развитию кластерных структур. Избыточность, изначально заложенная в кластеры, способна их обеспечить. Основой этого служит возможность каждого сервера кластера работать автономно, но в любой момент он может переключиться на выполнение работ другого сервера в случае его отказа.

Коэффициент готовности систем рассчитывается по формуле

$$K_z = T_p / (T_p + T_o),$$

где:  $T_p$  — полезное время работы системы;

$T_o$  — время отказа и восстановления системы, в течение которого она не могла выполнять свои функции.

Большинство современных серверов имеет 99%-ную готовность. Это означает, что около четырех дней в году они не работают. Подчеркнем, что готовность 99,9%, достигаемая обычно спаркой серверов — основного и резервного, означает годовой простой около 500 мин., 99,999% — 5 мин., а 99,9999% — 30 с.

Появление критически важных приложений в областях бизнеса, финансов, телекоммуникаций, здравоохранения и др. требует обеспечения коэффициента готовности не менее чем «заветные пять девяток» и даже выше.

**Повышение суммарной производительности** кластера, объединяющего несколько серверов, обычно не является самоцелью, а обеспечивается автоматически. Ведь каждый сервер кластера сам является достаточно мощной вычислительной системой, рассчитанной на выполнение им всех необходимых функций в части управления соответствующими сетевыми ресурсами. С развитием сетей все большее значение приобретают и распределенные вычисления. При этом многие компьютеры, в том числе и серверы, могут иметь не очень большую

нагрузку. Свободные ресурсы домашних компьютеров, рабочих станций локальных вычислительных сетей, да и самих серверов можно использовать для выполнения каких-либо трудоемких вычислений. При этом стоимость создания подобных вычислительных кластеров очень мала, так как все их составные части работают в сети и только при необходимости образуют виртуальный (временный) вычислительный комплекс.

Совокупные вычислительные мощности кластеров могут быть сравнимы с мощностями суперЭВМ и даже превышать их при неизмеримо меньшей стоимости. Такие технологии применительно к отдельным классам задач хорошо отработаны. Например, существует задача анализа сигналов, принимаемых радиотелескопами, с целью поиска внеземных цивилизаций; имеется проект distributed.net, реализующий алгоритм дешифрования, и др. Круг подобных задач не очень широк, но число одновременно привлекаемых компьютеров для этих целей может быть громадным — десятки, сотни и даже тысячи.

Работа кластера под управлением единой операционной системы позволяет оперативно контролировать процесс вычислений и *эффективно перераспределять нагрузки между компьютерами кластера.*

Управление такими проектами требует создания специального клиентского и серверного программного обеспечения, работающего в фоновом режиме. Компьютеры при этом периодически получают задания от сервера, включаются в работу и возвращают результаты обработки. Последние версии браузеров (browser) еще более упрощают процесс взаимодействия, так как на клиентской машине можно активизировать выполнение различных программ-сценариев (скриптов).

*Эффективное управление и контроль работы системы* подразумевает возможность работы отдельно с каждым узлом, вручную или программно отключать его для модернизации либо ремонта с последующим возвращением его в работающий кластер. Эти операции скрыты от пользователей, они просто не замечают их. Кластерное ПО, интегрированное в операционные системы серверов, позволяет работать с узлами как с единым пулом ресурсов (Single System Image, SSI), внося необходимые общие изменения с помощью одной операции для всех узлов.

Какие же средства имеются для построения кластеров?

Существуют различные методы и средства построения надежных систем с резервированием. Они рассматриваются как по отношению к средствам обработки, так и по отношению к средствам хранения данных. Например, есть источники бесперебойного питания и резервные блоки питания, способные устранять влияние сбоев и отказов в электроснабжении. Имеются также дисковые массивы RAID, обеспечивающие непрерывную обработку запросов к информации, хранящейся на дисках, даже в случае выхода из строя одного или нескольких из них. Но нам более интересны средства обработки.

Кластеры объединяют несколько серверов под единым управлением. Все новые серверы, как Правило, являются многопроцессорными и относятся к SMP-структурам, что обеспечивает многоступенчатую возможность переключения нагрузки отказавшего элемента как внутри кластера, так и внутри сервера. Существуют серверы с различным количеством процессоров (от 2 до 16). Правда, фирма Sun работает над созданием 64-процессорной SMP-модели сервера. IBM

предполагает с появлением микропроцессора 1А-64 Merced (новое название его — Itanium) выпустить SMP-систему, рассчитанную на 16 процессоров. Напротив, фирма Dell считает, что применение более восьми процессоров в SMP-структуре нецелесообразно из-за трудностей преодоления конфликтов при обращении их к общей оперативной памяти.

Большой интерес к построению кластеров стала проявлять фирма Microsoft. В связи с широкой популярностью операционной системы Windows NT, предназначенной для управления сетями крупных предприятий, появились различные варианты кластерного обеспечения. Сама фирма Microsoft предлагает бесплатную версию своего кластерного ПО, встроенного в Windows NT и поддерживающего Microsoft Cluster Server (MSCS). Этот кластерный продукт, известный под названием Wolfpack («волчья стая»), еще достаточно слаб, но уверенно прогрессирует. В настоящее время он обеспечивает разделение нагрузки между двумя узлами-серверами и то только путем замены одного сервера другим, а не путем ее перераспределения.

Достаточно трудно решается и проблема масштабирования, так как четырехузловая схема эквивалентна лишь двойному увеличению производительности по сравнению с одним узлом. Предполагается, что в будущем Wolfpack будет поддерживать до 16 узлов в кластере.

Унификация инженерно-технических решений предполагает соответственно и стандартизацию аппаратных и программных процедур обмена данными между серверами. Для передачи управляющей информации в кластере используются специальные магистрали, имеющие более высокие скорости обмена данными. В качестве такого стандарта предлагается интеллектуальный ввод-вывод (Intelligent Input/ Output — I2O). Спецификация I2O определяет унифицированный интерфейс между операционной системой и устройствами ввода-вывода, освобождая процессоры и их системные шины от обслуживания периферии.

Как и у любой новой технологии, у кластеризации имеются свои недостатки:

- задержки разработки и принятия общих стандартов;
- большая доля нестандартных и закрытых разработок различных фирм, затрудняющих их совместное использование;
- трудности управления одновременным доступом к файлам;
- сложности с управлением конфигурацией, настройкой, развертыванием, оповещениями серверов о сбоях и т.п.

## **11.6. Организация функционирования вычислительных систем**

Управление вычислительными процессами в ВС осуществляют операционные системы, которые являются частью общего программного обеспечения. В состав ОС включают как программы централизованного управления ресурсами системы, так и программы автономного использования вычислительных модулей. Последнее условие необходимо, поскольку в ВС обычно предусматривается более высокая надежность функционирования, например требование сохранения работоспособности при наличии в ней хотя бы одного исправного модуля. Требование увеличения производительности также предполагает возможность параллельной и даже автономной работы модулей при обработке отдельных заданий или пакетов заданий.

В зависимости от структурной организации ВС можно выявить некоторые особенности построения их операционных систем.

**Операционные системы многомашиных ВС** являются более простыми. Обычно они создаются как надстройка автономных ОС отдельных ЭВМ, поскольку здесь каждая ЭВМ имеет большую автономию в использовании ресурсов (своя оперативная и внешняя память, свой обособленный состав внешних устройств и т.д.). В них широко используются программные методы локального (в пределах вычислительного центра) и дистанционного (сетевая обработка) комплексирования.

Общим для построения ОС многомашиных комплексов служит тот факт, что для каждой машины ВС другие машины играют роль некоторых внешних устройств, и их взаимодействие осуществляется по интерфейсам, имеющим унифицированное программное обеспечение. Все обмены данными между ЭВМ должны предусматриваться пользователями путем включения в программы специальных операторов распараллеливания вычислений. По этим обращениям ОС ВС включает особые программы управления обменом. При этом ОС должна обеспечивать распределение и последующую пересылку заданий или их частей, оформляя их в виде самостоятельных заданий. Такие ОС, организуя обмен, должны формировать и устанавливать связи, контролировать процессы обмена, строить очереди запросов, решать конфликтные ситуации.

В многомашиных ВС диспетчерские функции могут решаться на централизованной или децентрализованной основе. Связь машин обычно устанавливается в порядке подчиненности: «главная ЭВМ — вспомогательная ЭВМ». Например, в пакете Norton Commander имеется возможность установить подобную связь: Master — Slave.

**Программное обеспечение многопроцессорных ВС** отличается большей сложностью. Это объясняется глубиной сложностью всестороннего анализа процессов, формируемых в ВС, а также сложностью принятия решения в каждой конкретной ситуации. Здесь все операции планирования и диспетчеризации связаны с динамическим распределением ресурсов (оперативной и внешней памяти, процессоров, данных системных таблиц, программ, периферийного оборудования и т.п.). Центральное место в этом отводится степени использования и методам управления общей оперативной памятью. Здесь очень часто могут формироваться множественные конфликты, требующие сложных процедур решения, что приводит к задержкам в вычислениях. Как таковые автономные ОС отдельных процессоров отсутствуют.

Для обеспечения эффективной работы многопроцессорных систем их операционные системы специализируют на следующие типовые методы взаимодействия процессоров:

- «ведущий — ведомый»;
- симметричная или однородная обработка во всех процессорах;
- раздельная независимая работа процессоров по обработке заданий.

Выбор метода «ведущий — ведомый» в наибольшей степени соответствует ВС с централизованным управлением. Здесь имеется определенная аналогия с многомашиными системами, организованными по принципу «главная ЭВМ — вспомогательная ЭВМ». Диспетчерские функции выполняются только одним

процессором системы. Закрепление этих функций может быть фиксированным и плавающим.

Для этого выделяется специализированный процессор или обычный процессор универсального типа, переключающийся и на выполнение вычислений. Системы типа «ведущий — ведомый» отличаются довольно простым аппаратным и программным обеспечением. Они должны получить распространение в MPP-структурах, но следует иметь в виду, что длительное время планирования может быть причиной простоев ведомых вычислителей.

Симметричная или однородная обработка в матрице процессоров возможна при использовании однотипных процессорных элементов, каждый из которых имеет непосредственные связи по передаче данных с другими элементами. В отличие от ОКМД-структур ранних выпусков, в которых синхронизировалось выполнение отдельных команд, в MPP-структурах симметричная обработка должна обеспечивать синхронизацию выполнения целых процессов. К сожалению, ни один из существующих языков программирования не содержит эффективных средств управления параллельными вычислениями. Такая система имеет большие преимущества. Она обладает повышенной живучестью и сохраняет работоспособность при выходе из строя даже нескольких процессоров матрицы, так как здесь имеется более высокий уровень резервирования. В ней обеспечивается более полная загрузка процессоров с лучшим использованием их процессорного времени. Расход других общесистемных ресурсов также выше.

В связи с успехами микроэлектроники появилась возможность реализовывать эти структуры в виде сверхбольших интегральных схем (СБИС), что позволяет получить дополнительные преимущества:

- короткие соединительные линии между процессорными элементами. Это приводит к расширению полосы пропускания и уменьшению задержек;
- регулярность структуры, позволяющая увеличивать плотность упаковки СБИС и упрощать ее разработку;
- высокая степень распараллеливания вычислений, что позволяет обеспечить высокую производительность.

Для управления процессом вычислений из однородной среды процессорных элементов выделяется один, играющий роль ведущего. Эти функции при необходимости могут передаваться от одного процессора к другому.

Раздельная или независимая работа вычислителей в многопроцессорных ВС осуществляется при параллельной обработке независимых заданий. Это позволяет получить максимальную производительность системы. Процедуры управления ею достаточно просты и уже опробованы на практике.