# Архитектуры и топологии многопроцессорных вычислительных систем

## 1. Назначение, область применения и способы оценки производительности многопроцессорных вычислительных систем

В настоящее время сфера применения многопроцессорных вычислительных систем (МВС) непрерывно расширяется, охватывая все новые области в различных отраслях науки, бизнеса и производства. Стремительное развитие кластерных систем создает условия для использования многопроцессорной вычислительной техники в реальном секторе экономики.

Если традиционно МВС применялись в основном в научной сфере для решения вычислительных задач, требующих мощных вычислительных ресурсов, то сейчас из-за бурного развития бизнеса резко возросло количество компаний, отводящих использованию компьютерных технологий и электронного документооборота главную роль. В связи с этим непрерывно растет потребность в построении централизованных вычислительных систем для критически важных приложений, связанных с обработкой транзакций, управлением базами данных и обслуживанием телекоммуникаций. Можно выделить две основные сферы применения описываемых систем: обработка транзакций в режиме реального времени (OLTP, on-line transaction processing) и создание хранилищ данных для организации систем поддержки принятия решений (Data Mining, Data Warehousing, Decision Support System). Система для глобальных корпоративных вычислений — это, прежде всего, централизованная система, с которой работают практически все пользователи в корпорации, и, соответственно, она должна все время находиться в рабочем состоянии. Как правило, решения подобного уровня устанавливают в компаниях и корпорациях, где даже кратковременные простои сети могут привести к громадным убыткам. Поэтому для организации такой системы не подойдет обыкновенный сервер со стандартной архитектурой, вполне пригодный там, где нет жестких требований к производительности и времени простоя. Высокопроизводительные системы для глобальных корпоративных вычислений должны отличаться такими характеристиками как повышенная производительность, масштабируемость, минимально допустимое время простоя.

Наряду с расширением области применения по мере совершенствования МВС происходит усложнение и увеличение количества задач в областях, традиционно использующих высокопроизводительную вычислительную технику. В настоящее время выделен круг фундаментальных и прикладных проблем, эффективное решение которых возможно только с использованием сверхмощных вычислительных ресурсов. Этот круг, обозначаемый понятием "Grand challenges", включает следующие задачи:

- предсказания погоды, климата и глобальных изменений в атмосфере;
- науки о материалах;
- построение полупроводниковых приборов;
- сверхпроводимость;
- структурная биология;
- разработка фармацевтических препаратов;
- генетика;
- квантовая хромодинамика;
- астрономия;

- транспортные задачи;
- гидро- и газодинамика;
- управляемый термоядерный синтез;
- эффективность систем сгорания топлива;
- геоинформационные системы;
- разведка недр;
- наука о мировом океане;
- распознавание и синтез речи;
- распознавание изображений.

Многопроцессорные вычислительные системы могут существовать в различных конфигурациях. Наиболее распространенными типами МВС являются:

- системы высокой надежности;
- системы для высокопроизводительных вычислений;
- многопоточные системы.

Отметим, что границы между этими типами МВС до некоторой степени размыты, и часто система может иметь такие свойства или функции, которые выходят за рамки перечисленных типов. Более того, при конфигурировании большой системы, используемой как система общего назначения, приходится выделять блоки, выполняющие все перечисленные функции.

МВС являются идеальной схемой для повышения надежности информационно-вычислительной системы. Благодаря единому представлению, отдельные узлы или компоненты МВС могут незаметно для пользователя заменять неисправные элементы, обеспечивая непрерывность и безотказную работу даже таких сложных приложений как базы данных.

Катастрофоустойчивые решения создаются на основе разнесения узлов многопроцессорной системы на сотни километров и обеспечения механизмов глобальной синхронизации данных между такими узлами.

МВС для высокопроизводительных вычислений предназначены для параллельных расчетов. Имеется много примеров научных расчетов, выполненных на основе параллельной работы нескольких недорогих процессоров, обеспечивающих одновременное проведение большого числа операций.

МВС для высокопроизводительных вычислений обычно собраны из многих компьютеров. Разработка таких систем – процесс сложный, требующий постоянного согласования таких вопросов как инсталляция, эксплуатация и одновременное управление большим числом компьютеров, технических требований параллельного и высокопроизводительного доступа к одному и тому же системному файлу (или файлам), межпроцессорной связи между узлами и координации работы в параллельном режиме. Эти проблемы проще всего решаются при обеспечении единого образа операционной системы для всего кластера. Однако реализовать подобную схему удается далеко не всегда, и обычно она применяется лишь для небольших систем.

Многопоточные системы используются для обеспечения единого интерфейса к ряду ресурсов, которые могут со временем произвольно наращиваться (или сокращаться). Типичным примером может служить группа web-серверов.

Главной отличительной особенностью многопроцессорной вычислительной системы является ее производительность, т.е. количество операций, производимых системой за единицу времени. Различают пиковую и реальную производительность. Под пиковой понимают величину, равную произведению пиковой производительности одного процессора на число таких процессоров в данной машине. При этом предполагается, что все устройства компьютера работают в максимально производительном режиме. Пиковая производительность компьютера вычисляется однозначно, и эта характеристика является базовой, по которой производят сравнение высокопроизводительных вычислительных систем. Чем больше пиковая производительность, тем (теоретически) быстрее пользователь сможет решить свою задачу. Пиковая производительность есть величина теоретическая и, вообще говоря, недостижимая при запуске конкретного приложения. Реальная же производительность, достигаемая на данном приложении, зависит от взаимодействия программной модели, в которой реализовано приложение, с архитектурными особенностями машины, на которой приложение запускается.

Существует два способа оценки пиковой производительности компьютера. Один из них опирается на число команд, выполняемых компьютером за единицу времени. Единицей измерения, как правило, является MIPS (Million Instructions Per Second). Производительность, выраженная в MIPS, говорит о скорости выполнения компьютером своих же инструкций. Но, во-первых, заранее не ясно, в какое количество инструкций отобразится конкретная программа, а во-вторых, каждая программа обладает своей спецификой, и число команд от программы к программе может меняться очень сильно. В связи с этим данная характеристика дает лишь самое общее представление о производительности компьютера.

Другой способ измерения производительности заключается в определении числа вещественных операций, выполняемых компьютером за единицу времени. Единицей измерения является Flops (Floating point operations per second) – число операций с плавающей точкой, производимых компьютером за одну секунду. Такой способ является более приемлемым для пользователя, поскольку ему известна вычислительная сложность программы, и, пользуясь этой характеристикой, пользователь может получить нижнюю оценку времени ее выполнения.

Однако пиковая производительность получается только в идеальных условиях, т.е. при отсутствии конфликтов при обращении к памяти при равномерной загрузке всех устройств. В реальных условиях на выполнение конкретной программы влияют такие аппаратно-программные особенности данного компьютера как: особенности структуры процессора, системы команд, состав функциональных устройств, реализация ввода/вывода, эффективность работы компиляторов.

Одним из определяющих факторов является время взаимодействия с памятью, которое определяется ее строением, объемом и архитектурой подсистем доступа в память. В большинстве современных компьютеров в качестве организации наиболее эффективного доступа к памяти используется так называемая многоуровневая иерархическая память. В качестве уровней используются регистры и регистровая память, основная оперативная память, кэш-память, виртуальные и жесткие диски, ленточные роботы. При этом выдерживается следующий принцип формирования иерархии: при повышении уровня памяти скорость обработки данных должна увеличиваться, а объем уровня памяти – уменьшаться. Эффективность использования такого рода иерархии достигается за счет хранения часто используемых данных в памяти верхнего уровня, время доступа к которой минимально. А поскольку такая память обходится достаточно дорого, ее объем не может

быть большим. Иерархия памяти относится к тем особенностям архитектуры компьютеров, которые имеют огромное значение для повышения их производительности.

Для того чтобы оценить эффективность работы вычислительной системы на реальных задачах, был разработан фиксированный набор тестов. Наиболее известным из них является LINPACK – программа, предназначенная для решения системы линейных алгебраических уравнений с плотной матрицей с выбором главного элемента по строке. LINPACK используется для формирования списка Top500 – пятисот самых мощных компьютеров мира. Однако LINPACK имеет существенный недостаток: программа распараллеливается, поэтому невозможно оценить эффективность работы коммуникационного компонента суперкомпьютера.

В настоящее время большое распространение получили тестовые программы, взятые из разных предметных областей и представляющие собой либо модельные, либо реальные промышленные приложения. Такие тесты позволяют оценить производительность компьютера действительно на реальных задачах и получить наиболее полное представление об эффективности работы компьютера с конкретным приложением.

Наиболее распространенными тестами, построенными по этому принципу, являются: набор из 24 Ливерморских циклов (The Livermore Fortran Kernels, LFK) и пакет NAS Parallel Benchmarks (NPB), в состав которого входят две группы тестов, отражающих различные стороны реальных программ вычислительной гидродинамики. NAS тесты являются альтернативой LINPACK, поскольку они относительно просты и в то же время содержат значительно больше вычислений, чем, например, LINPACK или LFK.

Однако при всем разнообразии тестовые программы не могут дать полного представления о работе компьютера в различных режимах. Поэтому задача определения реальной производительности многопроцессорных вычислительных систем остается пока нерешенной.

## 2. Архитектура вычислительных систем. Классификация архитектур по параллельной обработке данных

Чтобы дать более полное представление о многопроцессорных вычислительных системах, помимо высокой производительности необходимо назвать и другие отличительные особенности. Прежде всего, это необычные архитектурные решения, направленные на повышение производительности (работа с векторными операциями, организация быстрого обмена сообщениями между процессорами или организация глобальной памяти в многопроцессорных системах и др.).

Понятие архитектуры высокопроизводительной системы является достаточно широким, поскольку под архитектурой можно понимать и способ параллельной обработки данных, используемый в системе, и организацию памяти, и топологию связи между процессорами, и способ исполнения системой арифметических операций. Попытки систематизировать все множество архитектур впервые были предприняты в конце 60-х годов и продолжаются по сей день.

В 1966 г. М.Флинном (Flynn) был предложен чрезвычайно удобный подход к классификации архитектур вычислительных систем. В его основу было положено понятие потока, под которым понимается последовательность элементов, команд или данных, обрабатываемая процессором. Соответствующая система классификации основана на рассмотрении числа потоков инструкций и потоков данных и описывает четыре архитектурных класса:

```
SISD = Single Instruction Single Data
MISD = Multiple Instruction Single Data
SIMD = Single Instruction Multiple Data
MIMD = Multiple Instruction Multiple Data
```

SISD (single instruction stream / single data stream) – одиночный поток команд и одиночный поток данных. К этому классу относятся последовательные компьютерные системы, которые имеют один центральный процессор, способный обрабатывать только один поток последовательно исполняемых инструкций. В настоящее время практически все высокопроизводительные системы имеют более одного центрального процессора, однако каждый из них выполняет несвязанные потоки инструкций, что делает такие системы комплексами SISD-систем, действующих на разных пространствах данных. Для увеличения скорости обработки команд и скорости выполнения арифметических операций может применяться конвейерная обработка. В случае векторных систем векторный поток данных следует рассматривать как поток из одиночных неделимых векторов. Примерами компьютеров с архитектурой SISD могут служить большинство рабочих станций Compaq, Hewlett-Packard и Sun Microsystems.

MISD (multiple instruction stream / single data stream) – множественный поток команд и одиночный поток данных. Теоретически в этом типе машин множество инструкций должно выполняться над единственным потоком данных. До сих пор ни одной реальной машины, попадающей в данный класс, создано не было. В качестве аналога работы такой системы, по-видимому, можно рассматривать работу банка. С любого терминала можно подать команду и что-то сделать с имеющимся банком данных. Поскольку база данных одна, а команд много, мы имеем дело с множественным потоком команд и одиночным потоком данных.

SIMD (single instruction stream / multiple data stream) – одиночный поток команд и множественный поток данных. Эти системы обычно имеют большое количество процессоров, от 1024 до 16384, которые могут выполнять одну и ту же инструкцию относительно разных данных в жесткой конфигурации. Единственная инструкция параллельно выполняется над многими элементами данных. Примерами SIMD-машин являются системы CPP DAP, Gamma II и Quadrics Apemille. Другим подклассом SIMD-систем являются векторные компьютеры. Векторные компьютеры манипулируют массивами сходных данных подобно тому, как скалярные машины обрабатывают отдельные элементы таких массивов. Это делается за счет использования специально сконструированных векторных центральных процессоров. Когда данные обрабатываются посредством векторных модулей, результаты могут быть выданы на один, два или три такта частотогенератора (такт частотогенератора является основным временным параметром системы). При работе в векторном режиме векторные процессоры обрабатывают данные практически параллельно, что делает их в несколько раз более быстрыми, чем при работе в скалярном режиме. Примерами систем подобного типа являются, например, компьютеры Hitachi S3600.

MIMD (multiple instruction stream / multiple data stream) – множественный поток команд и множественный поток данных. Эти машины параллельно выполняют несколько потоков инструкций над различными потоками данных. В отличие от упомянутых выше многопроцессорных SISD-машин, команды и данные связаны, потому что они представляют различные части одной и той же задачи. Например, MIMD-системы могут параллельно выполнять множество подзадач с целью сокращения времени выполнения основной задачи. Большое разнообразие попадающих в данный класс систем делает классификацию Флинна не полностью адекватной. Действительно, и четырехпроцессорный SX-5 компании NEC, и тысячепроцессорный Cray T3E попадают в этот класс. Это заставляет использовать другой подход к классификации, иначе описывающий классы компьютерных систем. Основная идея такого подхода может состоять, например, в следующем. Будем считать, что множественный поток команд может быть обработан двумя способами: либо одним конвейерным устройством обработки, работающем в режиме разделения времени для отдельных потоков, либо каждый поток обрабатывается своим собственным устройством. Первая возможность используется в MIMD-компьютерах, которые обычно называют конвейерными или векторными, вторая – в параллельных компьютерах. В основе векторных компьютеров лежит концепция конвейеризации, т.е. явного сегментирования арифметического устройства на отдельные части, каждая из которых выполняет свою подзадачу для пары операндов. В основе параллельного компьютера лежит идея использования для решения одной задачи нескольких процессоров, работающих сообща, причем процессоры могут быть как скалярными, так и векторными.

Классификация архитектур вычислительных систем нужна для того, чтобы понять особенности работы той или иной архитектуры, но она не является достаточно детальной, чтобы на нее можно было опираться при создании МВС, поэтому следует вводить более детальную классификацию, которая связана с различными архитектурами ЭВМ и с используемым оборудованием.

**3. Архитектура вычислительных систем. SMP и MPP-архитектуры. Гибридная архитектура (NUMA). Организация когерентности многоуровневой иерархической памяти**

## SMP-архитектура

SMP (symmetric multiprocessing) – симметричная многопроцессорная архитектура. Главной особенностью систем с архитектурой SMP является наличие общей физической памяти, разделяемой всеми процессорами.


Рис. 3.1. Схематический вид SMP-архитектуры

Память служит, в частности, для передачи сообщений между процессорами, при этом все вычислительные устройства при обращении к ней имеют равные права и одну и ту же адресацию для всех ячеек памяти. Поэтому SMP-архитектура называется симметричной. Последнее обстоятельство позволяет очень эффективно обмениваться данными с другими вычислительными устройствами. SMP-система строится на основе высокоскоростной системной шины (SGI PowerPath, Sun Gigaplane, DEC TurboLaser), к слотам которой подключаются функциональные блоки типов: процессоры (ЦП), подсистема ввода/вывода (I/O) и т. п. Для подсоединения к модулям I/O используются уже более медленные шины (PCI, VME64). Наиболее известными SMP-системами являются SMP-серверы и рабочие станции на базе процессоров Intel (IBM, HP, Compaq, Dell, ALR, Unisys, DG, Fujitsu и др.) Вся система работает под управлением единой ОС (обычно UNIX-подобной, но для Intel-платформ поддерживается Windows NT). ОС автоматически (в процессе работы) распределяет процессы по процессорам, но иногда возможна и явная привязка.

Основные преимущества SMP-систем:

- простота и универсальность для программирования. Архитектура SMP не накладывает ограничений на модель программирования, используемую при создании приложения: обычно используется модель параллельных ветвей, когда все процессоры работают независимо друг от друга. Однако можно реализовать и модели, использующие межпроцессорный обмен. Использование общей памяти увеличивает скорость такого обмена, пользователь также имеет доступ сразу ко всему объему памяти. Для SMP-систем существуют довольно эффективные средства автоматического распараллеливания;
- простота эксплуатации. Как правило, SMP-системы используют систему кондиционирования, основанную на воздушном охлаждении, что облегчает их техническое обслуживание;
- относительно невысокая цена.

Недостатки:

- системы с общей памятью плохо масштабируются.

Этот существенный недостаток SMP-систем не позволяет считать их по-настоящему перспективными. Причиной плохой масштабируемости является то, что в данный момент шина способна обрабатывать только одну транзакцию, вследствие чего возникают проблемы разрешения конфликтов при одновременном обращении нескольких процессоров к одним и тем же областям общей физической памяти. Вычислительные элементы начинают друг другу мешать. Когда произойдет такой конфликт, зависит от скорости связи и от количества вычислительных элементов. В настоящее время конфликты могут происходить при наличии 8-24 процессоров. Кроме того, системная шина имеет ограниченную (хоть и высокую) пропускную способность (ПС) и ограниченное число слотов. Все это очевидно препятствует увеличению производительности при увеличении числа процессоров и числа подключаемых пользователей. В реальных системах можно задействовать не более 32 процессоров. Для построения масштабируемых систем на базе SMP используются кластерные или NUMA-архитектуры. При работе с SMP-системами используют так называемую парадигму программирования с разделяемой памятью (shared memory paradigm).

## MPP-архитектура

MPP (massive parallel processing) – массивно-параллельная архитектура. Главная особенность такой архитектуры состоит в том, что память физически разделена. В этом случае система строится из отдельных модулей, содержащих процессор, локальный банк операционной памяти (ОП), коммуникационные процессоры (рутеры) или сетевые адаптеры, иногда – жесткие диски и/или другие устройства ввода/вывода. По сути, такие модули представляют собой полнофункциональные компьютеры (см. рис.3.2). Доступ к банку ОП из данного модуля имеют только процессоры (ЦП) из этого же модуля. Модули соединяются специальными коммуникационными каналами. Пользователь может определить логический номер процессора, к которому он подключен, и организовать обмен сообщениями с другими процессорами. Используются два варианта работы операционной системы (ОС) на машинах MPP-архитектуры. В одном полноценная операционная система (ОС) работает только на управляющей машине (front-end), на каждом отдельном модуле функционирует сильно урезанный вариант ОС, обеспечивающий работу только расположенной в нем ветви параллельного приложения. Во втором варианте на каждом модуле работает полноценная UNIX-подобная ОС, устанавливаемая отдельно.

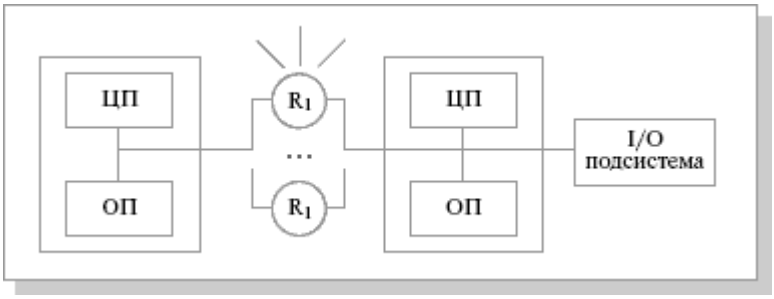


Рис. 3.2. Схематический вид архитектуры с раздельной памятью

Главным преимуществом систем с раздельной памятью является хорошая масштабируемость: в отличие от SMP-систем, в машинах с раздельной памятью каждый

процессор имеет доступ только к своей локальной памяти, в связи с чем не возникает необходимости в потактовой синхронизации процессоров. Практически все рекорды по производительности на сегодня устанавливаются на машинах именно такой архитектуры, состоящих из нескольких тысяч процессоров (ASCI Red, ASCI Blue Pacific).

Недостатки:

- отсутствие общей памяти заметно снижает скорость межпроцессорного обмена, поскольку нет общей среды для хранения данных, предназначенных для обмена между процессорами. Требуется специальная техника программирования для реализации обмена сообщениями между процессорами;
- каждый процессор может использовать только ограниченный объем локального банка памяти;
- вследствие указанных архитектурных недостатков требуются значительные усилия для того, чтобы максимально использовать системные ресурсы. Именно этим определяется высокая цена программного обеспечения для массивно-параллельных систем с раздельной памятью.

Системами с раздельной памятью являются суперкомпьютеры МВС-1000, IBM RS/6000 SP, SGI/CRAY T3E, системы ASCI, Hitachi SR8000, системы Parsytec.

Машины последней серии CRAY T3E от SGI, основанные на базе процессоров Dec Alpha 21164 с пиковой производительностью 1200 Мфлопс/с (CRAY T3E-1200), способны масштабироваться до 2048 процессоров.

При работе с MPP-системами используют так называемую Massive Passing Programming Paradigm – парадигму программирования с передачей данных (MPI, PVM, BSPlib).

## Гибридная архитектура NUMA

Главная особенность гибридной архитектуры NUMA (nonuniform memory access) – неоднородный доступ к памяти.

Гибридная архитектура совмещает достоинства систем с общей памятью и относительную дешевизну систем с раздельной памятью. Суть этой архитектуры – в особой организации памяти, а именно: память физически распределена по различным частям системы, но логически она является общей, так что пользователь видит единое адресное пространство. Система построена из однородных базовых модулей (плат), состоящих из небольшого числа процессоров и блока памяти. Модули объединены с помощью высокоскоростного коммутатора. Поддерживается единое адресное пространство, аппаратно поддерживается доступ к удаленной памяти, т.е. к памяти других модулей. При этом доступ к локальной памяти осуществляется в несколько раз быстрее, чем к удаленной. По существу, архитектура NUMA является MPP (массивно-параллельной) архитектурой, где в качестве отдельных вычислительных элементов берутся SMP (симметричная многопроцессорная архитектура) узлы. Доступ к памяти и обмен данными внутри одного SMP-узла осуществляется через локальную память узла и происходит очень быстро, а к процессорам другого SMP-узла тоже есть доступ, но более медленный и через более сложную систему адресации.

Структурная схема компьютера с гибридной сетью: четыре процессора связываются между собой при помощи кроссбара в рамках одного SMP-узла. Узлы связаны сетью типа "бабочка" (Butterfly):

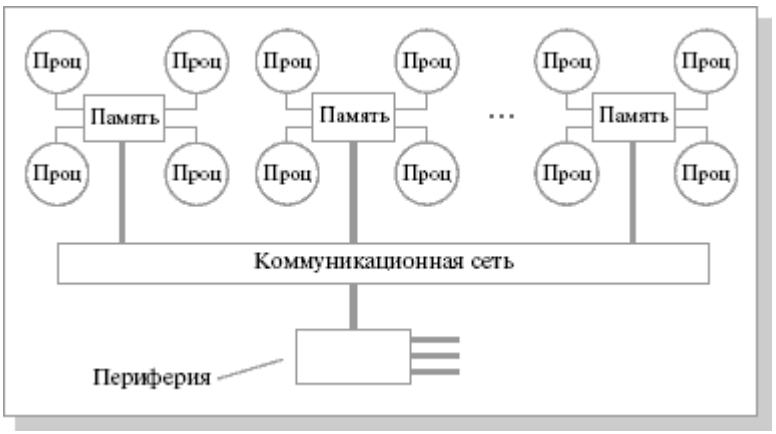


Рис. 3.3. Структурная схема компьютера с гибридной сетью

Впервые идею гибридной архитектуры предложил Стив Воллох, он воплотил ее в системах серии Exemplar. Вариант Воллоха – система, состоящая из восьми SMP-узлов. Фирма HP купила идею и реализовала на суперкомпьютерах серии SPP. Идею подхватил Сеймур Крей (Seymour R.Cray) и добавил новый элемент – когерентный кэш, создав так называемую архитектуру cc-NUMA (Cache Coherent Non-Uniform Memory Access), которая расшифровывается как "неоднородный доступ к памяти с обеспечением когерентности кэшей". Он ее реализовал на системах типа Origin.

## Организация когерентности многоуровневой иерархической памяти

Понятие когерентности кэшей описывает тот факт, что все центральные процессоры получают одинаковые значения одних и тех же переменных в любой момент времени. Действительно, поскольку кэш-память принадлежит отдельному компьютеру, а не всей многопроцессорной системе в целом, данные, попадающие в кэш одного компьютера, могут быть недоступны другому. Чтобы этого избежать, следует провести синхронизацию информации, хранящейся в кэш-памяти процессоров.

Для обеспечения когерентности кэшей существует несколько возможностей:

- использовать механизм отслеживания шинных запросов (snoopy bus protocol), в котором кэши отслеживают переменные, передаваемые к любому из центральных процессоров и при необходимости модифицируют собственные копии таких переменных;
- выделять специальную часть памяти, отвечающую за отслеживание достоверности всех используемых копий переменных.

Наиболее известными системами архитектуры cc-NUMA являются: HP 9000 V-class в SCA-конфигурациях, SGI Origin3000, Sun HPC 15000, IBM/Sequent NUMA-Q 2000. На сегодня максимальное число процессоров в cc-NUMA-системах может превышать 1000 (серия Origin3000). Обычно вся система работает под управлением единой ОС, как в SMP. Возможны также варианты динамического "подразделения" системы, когда отдельные "разделы" системы работают под управлением разных ОС. При работе с NUMA-системами, так же, как с SMP, используют так называемую парадигму программирования с общей памятью (shared memory paradigm).

**4. Архитектура вычислительных систем. PVP-архитектура. Кластерная архитектура**

## PVP (Parallel Vector Process) – параллельная архитектура с векторными процессорами

Основным признаком PVP-систем является наличие специальных векторно-конвейерных процессоров, в которых предусмотрены команды однотипной обработки векторов независимых данных, эффективно выполняющиеся на конвейерных функциональных устройствах. Как правило, несколько таких процессоров (1-16) работают одновременно с общей памятью (аналогично SMP) в рамках многопроцессорных конфигураций. Несколько узлов могут быть объединены с помощью коммутатора (аналогично MPP). Поскольку передача данных в векторном формате осуществляется намного быстрее, чем в скалярном (максимальная скорость может составлять 64 Гбайт/с, что на 2 порядка быстрее, чем в скалярных машинах), то проблема взаимодействия между потоками данных при распараллеливании становится несущественной. И то, что плохо распараллеливается на скалярных машинах, хорошо распараллеливается на векторных. Таким образом, системы PVP-архитектуры могут являться машинами общего назначения (general purpose systems). Однако, поскольку векторные процессоры весьма дорого стоят, эти машины не могут быть общедоступными.

Наиболее популярны три машины PVP-архитектуры:

1. **CRAY X1**, **SMP**-архитектура. Пиковая производительность системы в стандартной конфигурации может составлять десятки терафлопс.

Рис. 4.1. CRAY SV-2

2. **NEC SX-6**, **NUMA**-архитектура. Пиковая производительность системы может достигать 8 Тфлопс, производительность одного процессора составляет 9,6 Гфлопс. Система масштабируется с единым образом операционной системы до 512 процессоров.
3. **Fujitsu-VPP5000** (**vector parallel processing**), **MPP**-архитектура. Производительность одного процессора составляет 9.6 Гфлопс, пиковая производительность системы может достигать 1249 Гфлопс, максимальная емкость памяти – 8 Тбайт. Система масштабируется до 512

Рис. 4.2. Fujitsu-VPP5000

Парадигма программирования на PVP-системах предусматривает векторизацию циклов (для достижения разумной производительности одного процессора) и их распараллеливание (для одновременной загрузки нескольких процессоров одним приложением).

На практике рекомендуется выполнять следующие процедуры:

- производить векторизацию вручную, чтобы перевести задачу в матричную форму. При этом, в соответствии с длиной вектора, размеры матрицы должны быть кратны 128 или 256;
- работать с векторами в виртуальном пространстве, разлагая искомую функцию в ряд и оставляя число членов ряда, кратное 128 или 256.

За счет большой физической памяти (доли терабайта) даже плохо векторизуемые задачи на PVP-системах решаются быстрее на машинах со скалярными процессорами.

Кластер представляет собой два или более компьютеров (часто называемых узлами), объединяемые при помощи сетевых технологий на базе шинной архитектуры или коммутатора и предстающие перед пользователями в качестве единого информационно-вычислительного ресурса. В качестве узлов кластера могут быть выбраны серверы, рабочие станции и даже обычные персональные компьютеры. Узел характеризуется тем, что на нем работает единственная копия операционной системы. Преимущество кластеризации для повышения работоспособности становится очевидным в случае сбоя какого-либо узла: при этом другой узел кластера может взять на себя нагрузку неисправного узла, и пользователи не заметят прерывания в доступе. Возможности масштабируемости кластеров позволяют многократно увеличивать производительность приложений для большего числа пользователей технологий (Fast/Gigabit Ethernet, Myrinet) на базе шинной архитектуры или коммутатора. Такие суперкомпьютерные системы являются самыми дешевыми, поскольку собираются на базе стандартных комплектующих элементов ("off the shelf"), процессоров, коммутаторов, дисков и внешних устройств.

Кластеризация может осуществляться на разных уровнях компьютерной системы, включая аппаратное обеспечение, операционные системы, программы-утилиты, системы управления и приложения. Чем больше уровней системы объединены кластерной технологией, тем выше надежность, масштабируемость и управляемость кластера.

## Типы кластеров

Условное деление на классы предложено Язеком Радаевским и Дугласом Эдлайном:

- **Класс I**. Класс машин строится целиком из стандартных деталей, которые продают многие поставщики компьютерных компонентов (низкие цены, простое обслуживание, аппаратные компоненты доступны из различных источников).
- **Класс II**. Система имеет эксклюзивные или не слишком широко распространенные детали. Таким образом можно достичь очень хорошей производительности, но при более высокой стоимости.

Как уже отмечалось, кластеры могут существовать в различных конфигурациях. Наиболее распространенными типами кластеров являются:

- системы высокой надежности;
- системы для высокопроизводительных вычислений;
- многопоточные системы.

Отметим, что границы между этими типами кластеров до некоторой степени размыты, и кластер может иметь такие свойства или функции, которые выходят за рамки перечисленных типов. Более того, при конфигурировании большого кластера, используемого как система общего назначения, приходится выделять блоки, выполняющие все перечисленные функции.

Кластеры для высокопроизводительных вычислений предназначены для параллельных расчетов. Эти кластеры обычно собраны из большого числа компьютеров. Разработка таких кластеров является сложным процессом, требующим на каждом шаге согласования таких вопросов как инсталляция, эксплуатация и одновременное управление большим числом компьютеров, технические требования параллельного и высокопроизводительного доступа к одному и тому же системному файлу (или файлам) и межпроцессорная связь между узлами, и координация работы в параллельном режиме. Эти проблемы проще всего решаются при обеспечении единого образа операционной системы для всего кластера. Однако реализовать подобную схему удается далеко не всегда и обычно она применяется лишь для не слишком больших систем.

Многопоточные системы используются для обеспечения единого интерфейса к ряду ресурсов, которые могут со временем произвольно наращиваться (или сокращаться). Типичным примером может служить группа web-серверов.

В 1994 году Томас Стерлинг (Sterling) и Дон Беккер (Becker) создали 16-узловой кластер из процессоров Intel DX4, соединенных сетью 10 Мбит/с Ethernet с дублированием каналов. Они назвали его "Beowulf" по названию старинной эпической поэмы. Кластер возник в центре NASA Goddard Space Flight Center для поддержки необходимыми вычислительными ресурсами проекта Earth and Space Sciences. Проектно-конструкторские работы быстро превратились в то, что известно сейчас как проект Beowulf. Проект стал основой общего подхода к построению параллельных кластерных компьютеров, он описывает многопроцессорную архитектуру, которая может с успехом использоваться для параллельных вычислений. Beowulf-кластер, как правило, является системой, состоящей из одного серверного узла (который обычно называется головным), а также одного или нескольких подчиненных (вычислительных) узлов, соединенных посредством стандартной компьютерной сети. Система строится с использованием стандартных аппаратных компонентов, таких как ПК, запускаемые под Linux, стандартные сетевые

адаптеры (например, Ethernet) и коммутаторы. Нет особого программного пакета, называемого "Beowulf". Вместо этого имеется несколько кусков программного обеспечения, которые многие пользователи нашли пригодными для построения кластеров Beowulf. Beowulf использует такие программные продукты как операционная система Linux, системы передачи сообщений PVM, MPI, системы управления очередями заданий и другие стандартные продукты. Серверный узел контролирует весь кластер и обслуживает файлы, направляемые к клиентским узлам.

## Проблемы выполнения сети связи процессоров в кластерной системе

Архитектура кластерной системы (способ соединения процессоров друг с другом) в большей степени определяет ее производительность, чем тип используемых в ней процессоров. Критическим параметром, влияющим на величину производительности такой системы, является расстояние между процессорами. Так, соединив вместе 10 персональных компьютеров, мы получим систему для проведения высокопроизводительных вычислений. Проблема, однако, будет состоять в поиске наиболее эффективного способа соединения стандартных средств друг с другом, поскольку при увеличении производительности каждого процессора в 10 раз производительность системы в целом в 10 раз не увеличится.

Рассмотрим для примера задачу построения симметричной 16-процессорной системы, в которой все процессоры были бы равноправны. Наиболее естественным представляется соединение в виде плоской решетки, где внешние концы используются для подсоединения внешних устройств.

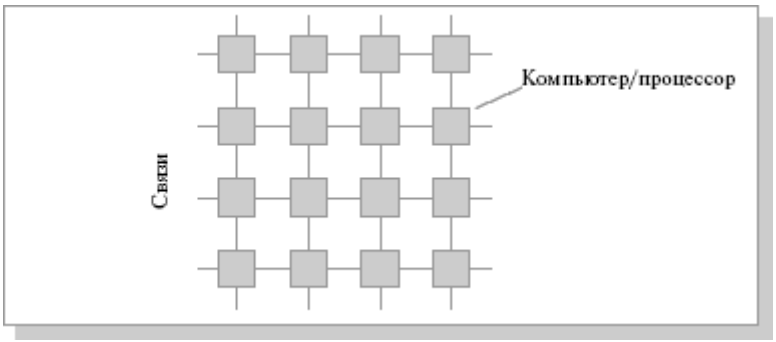


Рис. 4.3. Схема соединения процессоров в виде плоской решетки

При таком типе соединения максимальное расстояние между процессорами окажется равным 6 (количество связей между процессорами, отделяющих самый ближний процессор от самого дальнего). Теория же показывает, что если в системе максимальное расстояние между процессорами больше 4, то такая система не может работать эффективно. Поэтому при соединении 16 процессоров друг с другом плоская схема является нецелесообразной. Для получения более компактной конфигурации необходимо решить задачу о нахождении фигуры, имеющей максимальный объем при минимальной площади поверхности. В трехмерном пространстве таким свойством обладает шар. Но поскольку нам необходимо построить узловую систему, вместо шара приходится использовать куб (если число процессоров равно 8) или гиперкуб, если число процессоров больше 8. Размерность гиперкуба будет определяться в зависимости от числа процессоров, которые необходимо соединить. Так, для соединения 16 процессоров потребуется четырехмерный гиперкуб. Для его построения следует взять обычный трехмерный куб, сдвинуть в нужном направлении и, соединив вершины, получить гиперкуб размером 4.

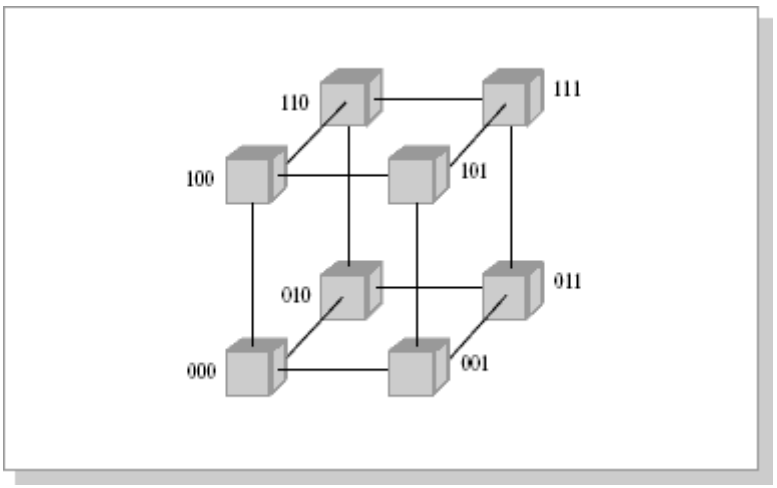


Рис. 4.4. Топология связи, 3-х мерный гиперкуб

Архитектура гиперкуба является второй по эффективности, но самой наглядной. Используются и другие топологии сетей связи: трехмерный тор, "кольцо", "звезда" и другие.

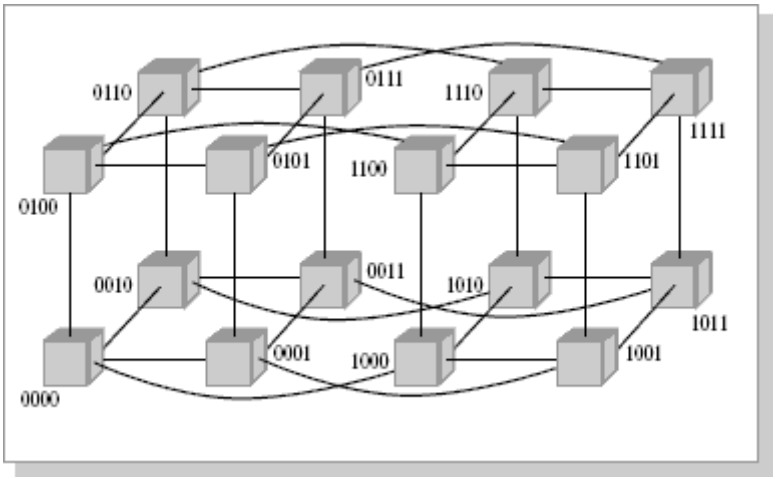


Рис. 4.5. Топология связи, 4-х мерный гиперкуб

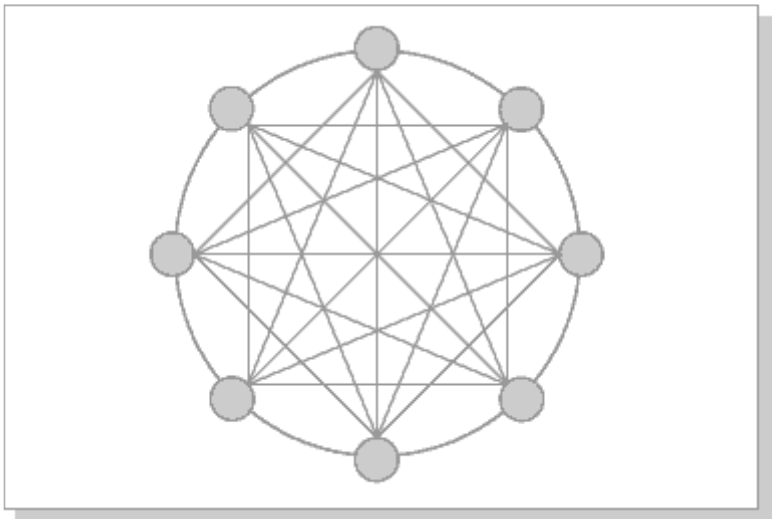

Рис. 4.6. Архитектура кольца с полной связью по хордам (Chordal Ring)

Наиболее эффективной является архитектура с топологией "толстого дерева" (fat-tree). Архитектура "fat-tree" (hypertree) была предложена Лейзерсоном (Charles E. Leiserson) в 1985 году. Процессоры локализованы в листьях дерева, в то время как внутренние узлы дерева скомпонованы во внутреннюю сеть. Поддеревья могут общаться между собой, не затрагивая более высоких уровней сети.

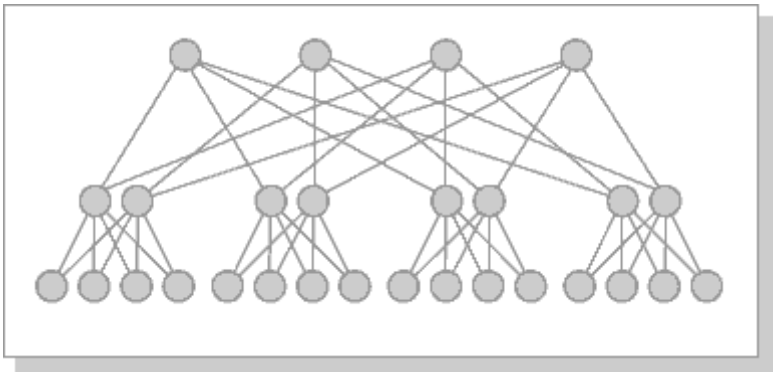

Рис. 4.7. Кластерная архитектура "Fat-tree"

Поскольку способ соединения процессоров друг с другом больше влияет на производительность кластера, чем тип используемых в ней процессоров, то может оказаться более целесообразным создать систему из большего числа дешевых компьютеров, чем из меньшего числа дорогих. В кластерах, как правило, используются операционные системы, стандартные для рабочих станций, чаще всего свободно распространяемые (Linux, FreeBSD), вместе со специальными средствами поддержки параллельного программирования и балансировки нагрузки. При работе с кластерами, так же, как и с MPP-системами, используют так называемую Massive Passing Programming Paradigm – парадигму программирования с передачей данных (чаще всего – MPI). Умеренная цена подобных систем оборачивается большими накладными расходами на взаимодействие параллельных процессов между собой, что сильно сужает потенциальный класс решаемых задач.

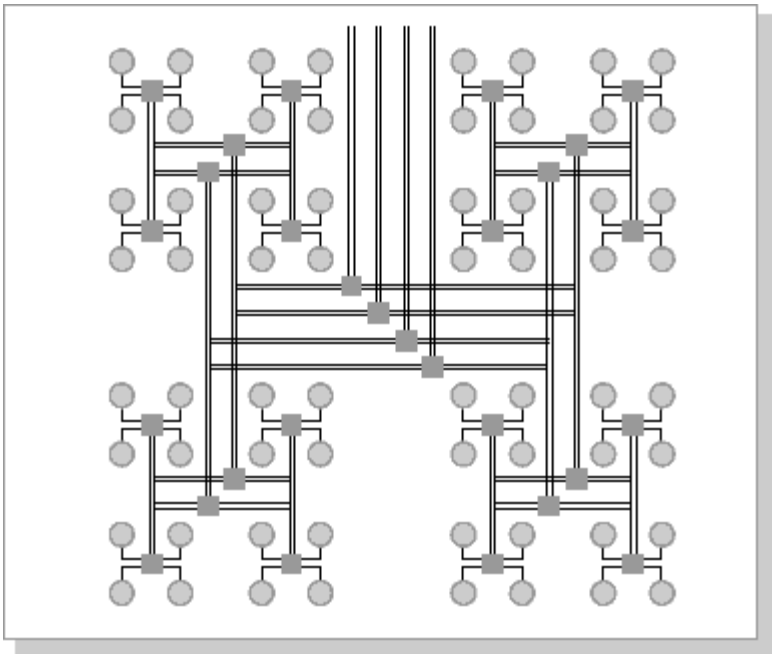

Рис. 4.8. Кластерная архитектура "Fat-tree" (вид сверху на предыдущую схему)

## 5. Принципы построения коммуникационных сред

В самом общем смысле архитектуру компьютера можно определить как способ соединения компьютеров между собой, с памятью и с внешними устройствами. Реализация этого соединения может идти различными путями. Конкретная реализация такого рода соединений называется коммуникационной средой компьютера. Одна из самых простых реализаций – это использование общей шины, к которой подключаются как процессоры, так и память. Сама шина состоит из определенного числа линий связи, необходимых для передачи адресов, данных и управляющих сигналов между процессором и памятью. Этот способ реализован в SMP-системах. Основным недостатком таких систем, как было указано выше, является плохая масштабируемость. Увеличение, даже незначительное, числа устройств на шине вызывает заметные задержки при обмене с памятью и катастрофическое падение производительности системы в целом. Необходимы другие подходы для построения коммуникационной среды, и одним из них является разделение памяти на независимые модули и обеспечение возможности доступа разных процессоров к различным модулям одновременно посредством использования различного рода коммутаторов.

При этом возможны различные конфигурации получающихся систем связи. Так, в компьютерах семейства Cray T3D/T3E все процессоры были объединены специальными высокоскоростными каналами в трехмерный тор, в котором каждый вычислительный узел имел непосредственные связи с шестью соседями. В компьютерах IBM SP/2 взаимодействие процессоров происходит через иерархическую систему коммутаторов, также обеспечивающую возможность соединения каждого процессора с любым другим. Эти оригинальные уникальные решения значительно увеличивают цену компьютеров.

Гораздо более простым и дешевым оказалось использование связей на базе сетей Ethernet – методика, разработанная компанией Xerox. Первоначально использовалась обычная 10-мегабитная сеть, затем стали применять Fast Ethernet, а в последнее время иногда и Gigabit Ethernet. Но для Fast Ethernet характерна большая латентность (задержка в передаче данных), оцениваемая в 160-180 микросекунд, а Gigabit Ethernet отличается высокой стоимостью. Кроме того, он эффективен только при соединении точка–точка, при соединении нескольких узлов его эффективность резко падает, а при соединении более 5–6 узлов она не превосходит по производительности даже Fast Ethernet. Поэтому при создании многопроцессорных вычислительных систем часто предпочтение отдается технологиям SCI, Myrinet или Raceway.

## Примеры построения коммуникационных сред на основе масштабируемого когерентного интерфейса SCI

SCI (Scalable Coherent Interface) принят как стандарт в 1992 г. (ANSI/IEEE Std 1596-1992). Он предназначен для достижения высоких скоростей передачи с малым временем задержки и при этом обеспечивает масштабируемую архитектуру, позволяющую строить системы, состоящие из множества блоков. SCI представляет собой комбинацию шины и локальной сети, обеспечивает реализацию когерентности кэш-памяти, размещаемой в узле SCI, посредством механизма распределенных директорий, который улучшает производительность, скрывая затраты на доступ к удаленным данным в модели с распределенной разделяемой памятью. Производительность передачи данных обычно находится в пределах от 200 Мбайт/с до 1000 Мбайт/с на расстояниях десятков метров с использованием электрических кабелей и километров – с использованием оптоволокна.

SCI уменьшает время межузловых коммуникаций по сравнению с традиционными схемами передачи данных в сетях путем устранения обращений к программным уровням – операционной системе и библиотекам времени выполнения; коммуникации представляются как часть простой операции загрузки данных процессором (командами load или store). Обычно обращение к данным, физически расположенным в памяти другого вычислительного узла и не находящимся в кэше, приводит к формированию запроса к удаленному узлу для получения необходимых данных, которые в течение нескольких микросекунд доставляются в локальный кэш, и выполнение программы продолжается. Прежний подход требовал формирования пакетов на программном уровне с последующей передачей их аппаратному обеспечению. Точно так же происходил и прием, в результате чего задержки были в сотни раз больше, чем у SCI. Однако для совместимости SCI имеет возможность переносить пакеты других протоколов.

Еще одно преимущество SCI – использование простых протоколов типа RISC, которые обеспечивают большую пропускную способность. Узлы с адаптерами SCI могут использовать для соединения коммутаторы или же соединяться в кольцо. Обычно каждый узел оказывается включенным в два кольца (рис. 5.1).

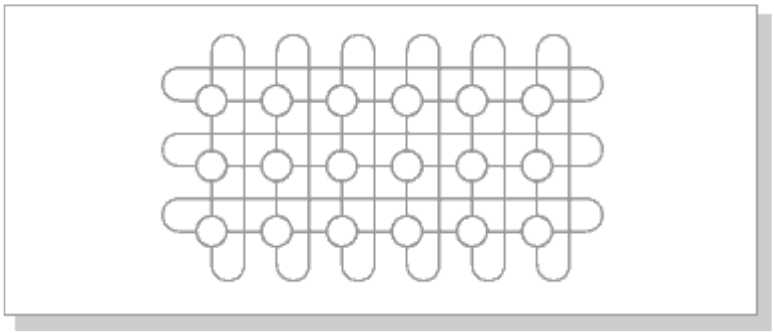

Рис. 5.1. Матрица узлов кластера на основе сети SCI

В отличие от HIPPI, данная технология оптимизирована для работы с динамическим трафиком, однако может быть менее эффективна при работе с большими блоками данных. Протокол передачи данных обеспечивает гарантированную доставку и отсутствие дедлоков. Протокол SCI достаточно сложен, он предусматривает широкие возможности управления трафиком, но использование этих возможностей предполагает наличие развитого программного обеспечения. На коммуникационной технологии SCI основана система связи гиперузлов CTI (Convex Torroidal Interconnect) в системах HP/Convex Exemplar X-class, кроме того, на ней построены кластерные системы SCALI Computer, системы семейства hpcLine компании Siemens, а также cc-NUMA сервера Data General и Sequent.

Традиционная область применения SCI – это коммуникационные среды многопроцессорных систем. На основе этой технологии построены, в частности, компьютеры серии hpcLine от Siemens или модульные серверы NUMA-Q от IBM, ранее известные как Sequent.

Модульные SCI-коммутаторы Dolphin позволяют потребителям строить масштабируемые кластерные решения класса предприятия на платформах Windows NT/2000/XP, Linux, Solaris, VxWorks, LynuxWorks и NetWare с использованием стандартизированного оборудования и программного обеспечения.

| Таблица 5.1. Технология SCI | |
|---|---|
| **Производители оборудования** | Dolphin Interconnect Solutions и др. |
| **Показатели производительности** | Для продуктов Dolphin: пиковая пропускная способность – 667 Мбит/с. Аппаратная латентность – 1.46 мксек |
| **Программная поддержка** | Драйверы для Linux, Windows NT, Solaris. ScaMPI – реализация MPI компании Scali Computer для систем на базе SCI. SISCI API – интерфейс программирования нижнего уровня |
| **Комментарии** | SCI (ANSI/IEEE l596-1992) – стандартизированная технология. Кроме стандартной сетевой среды, SCI поддерживает построение систем с разделяемой памятью и с когерентностью кэшей. На коммуникационной технологии SCI основаны кластерные системы компании SCALI Computer, системы семейства hpcLine компании Siemens, а также cc-NUMA-сервера Data General и Sequent. Технология SCI использовалась для связи гиперузлов в системах HP/Convex Exemplar X-class. |

## Коммуникационная среда MYRINET

Сетевую технологию Myrinet представляет компания Myricom, которая впервые предложила свою коммуникационную технологию в 1994 году, а на сегодня имеет уже более 1000 инсталляций по всему миру. Технология Myrinet основана на использовании многопортовых коммутаторов при ограниченных несколькими метрами длинах связей узлов с портами коммутатора. Узлы в Myrinet соединяются друг с другом через коммутатор (до 128 портов). Максимальная длина линий связи варьируется в зависимости от конкретной реализации.

Как коммутируемая сеть, аналогичная по структуре сегментам Ethernet, соединенным с помощью коммутаторов, Myrinet может одновременно передавать несколько пакетов, каждый из которых идет со скоростью, близкой к 2 Гбит/с. В отличие от некоммутированных Ethernet и FDDI сетей, которые разделяют общую среду передачи, совокупная пропускная способность сети Myrinet возрастает с увеличением количества машин. На сегодня Myrinet чаще всего используют как локальную сеть (LAN) сравнительно небольшого размера, связывая вместе компьютеры внутри комнаты или здания. Из-за своей высокой скорости, малого времени задержки, прямой коммутации и умеренной стоимости Myrinet часто используется для объединения компьютеров в кластеры. Myrinet также используется как системная сеть (System Area Network, SAN), которая может объединять компьютеры в кластер внутри стойки с той же производительностью, но с более низкой стоимостью, чем Myrinet LAN. Пакеты Myrinet могут иметь любую длину. Таким образом, они могут включать в себя другие типы пакетов, в том числе IP-пакеты. Объединение вычислительных узлов с адаптерами Myrinet в сеть происходит с помощью коммутаторов, которые имеют сейчас 4, 8, 12 или 16 портов. В коммутаторах используется передача пакетов путем установления соединения на время передачи, для маршрутизации сообщений применяется алгоритм прокладки пути (wormhole, "червоточина"). Коммутаторы, как и сетевые адаптеры, построены на специализированных микропроцессорах LANai компании Myricom.

| Таблица 5.2. Технология Myrinet | |
|---|---|
| **Производители оборудования** | Myricom |

| | |
|---|---|
| **Показатели производительности** | Пиковая пропускная способность – 2 Гбит/с, полный дуплекс. Латентность – порядка 4 мксек. |
| **Программная поддержка** | Драйверы для Linux (Alpha, x86, PowerPC, UltraSPARC), Windows NT (x86), Solaris (x86, UltraSPARC) и Tru64 UNIX. GM – интерфейс программирования на нижнем уровне. Пакеты HPVM (включает MPI-FM, реализацию MPI для Myrinet), BIP-MPI и др. |
| **Комментарии** | Myrinet является открытым стандартом. Myricom предлагает широкий выбор сетевого оборудования по сравнительно невысоким ценам. На физическом уровне поддерживаются сетевые среды SAN (System Area Network), LAN (CL-2) и оптоволокно. Технология Myrinet предоставляет широкие возможности масштабирования сети и в настоящее время очень часто используется при построении высокопроизводительных кластеров |

На физическом уровне линки Myrinet состоят из 9 проводников: 8 битов предназначены для передачи информации, интерпретируемой в зависимости от состояния девятого бита как байт данных или управляющий символ; при этом на каждом линке обеспечивается управление потоком и контроль ошибок. Среда Myrinet выгодно отличается от многих других сред передачи, в частности, SCI, простотой концепции и аппаратной реализации протоколов. Она содержит ограниченный набор средств управления трафиком, использующих приливно-отливный буфер, управляющие символы и таймерные интервалы. Myrinet является открытым стандартом, компания Myricom предлагает широкий выбор сетевого оборудования по сравнительно невысоким ценам. Технология Myrinet предоставляет широкие возможности масштабирования сети и часто используется при построении высокопроизводительных вычислительных кластеров.

## 6. Способы организации высокопроизводительных процессоров. Ассоциативные процессоры. Конвейерные процессоры. Матричные процессоры

Существующие в настоящее время алгоритмы прикладных задач, системное программное обеспечение и аппаратные средства преимущественно ориентированы на традиционную адресную обработку данных. Данные должны быть представлены в виде ограниченного количества форматов (например, массивы, списки, записи), должна быть явно создана структура связей между элементами данных посредством указателей на адреса элементов памяти, при обработке этих данных должна быть выполнена совокупность операций, обеспечивающих доступ к данным по указателям. Такой подход обуславливает громоздкость операционных систем и систем программирования, а также служит препятствием к созданию вычислительных средств с архитектурой, ориентированной на более эффективное использование параллелизма обработки данных.

### Ассоциативные процессоры

Ассоциативный способ обработки данных позволяет преодолеть многие ограничения, присущие адресному доступу к памяти, за счет задания некоторого критерия отбора и проведения необходимых преобразований, только над теми данными, которые удовлетворяют этому критерию. Критерием отбора может быть совпадение с любым элементом данных, достаточным для выделения искомых данных из всех имеющихся. Поиск данных может происходить по фрагменту, имеющему большую или меньшую корреляцию с заданным элементом данных.

Исследованы и в разной степени применяются несколько подходов, различающихся полнотой реализации модели ассоциативной обработки. Если реализуется только ассоциативная выборка данных с последующим поочередным использованием найденных данных, то говорят об ассоциативной памяти или памяти, адресуемой по содержимому. При достаточно полной реализации всех свойств ассоциативной обработки используется термин "ассоциативный процессор".

Ассоциативные системы относятся к классу: один поток команд – множество потоков данных (SIMD = Single Instruction Multiple Data). Эти системы включают большое число операционных устройств, способных одновременно по командам управляющего устройства вести обработку нескольких потоков данных. В ассоциативных вычислительных системах информация на обработку поступает от ассоциативных запоминающих устройств (АЗУ), характеризующихся тем, что информация в них выбирается не по определенному адресу, а по ее содержанию.

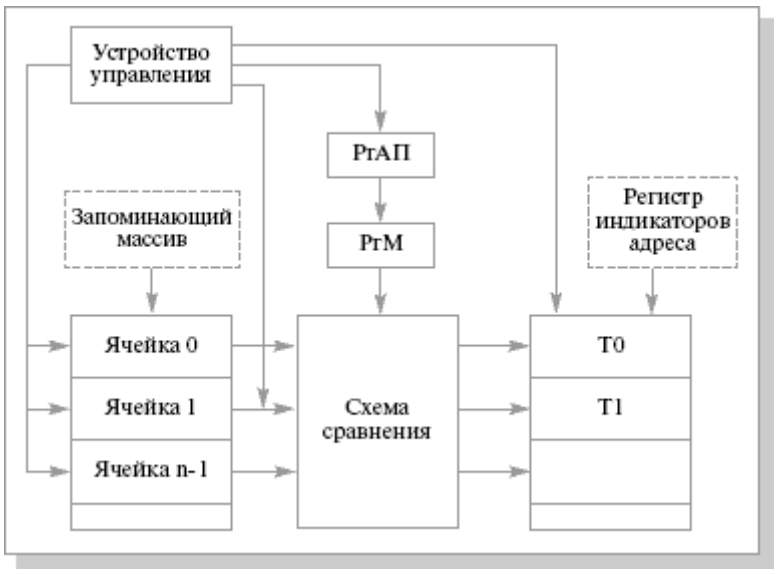


Рис. 6.1. Схема ассоциативной системы

## Конвейерные процессоры

Процессоры современных компьютеров используют особенную технологию – конвейеры, которые позволяют обрабатывать более одной команды одновременно.

Обработка команды может быть разделена на несколько основных этапов, назовем их микрокомандами. Выделим основные пять микрокоманд:

- выборка команды;
- расшифровка команды;
- выборка необходимых операндов;
- выполнение команды;
- сохранение результатов.

Все этапы команды задействуются только один раз и всегда в одном и том же порядке: одна за другой. Это, в частности, означает, что если первая микрокоманда выполнила свою работу и передала результаты второй, то для выполнения текущей команды она больше не понадобится, и, следовательно, может приступить к выполнению следующей команды. Выделим каждую команду в отдельную часть устройства и расположим их в порядке выполнения. В первый момент времени выполняется первая микрокоманда. Она завершает свою работу и начинает выполняться вторая микрокоманда, в то время как первая готова для выполнения следующей инструкции. Первая инструкция может считаться выполненной, когда завершат работу все пять микрокоманд.

Такая технология обработки команд носит название конвейерной обработки. Каждая часть устройства называется ступенью конвейера, а общее число ступеней – длиной конвейера.

Во многих вычислительных системах наряду с конвейером команд используются и конвейеры данных.

Сочетание этих двух конвейеров позволяет достичь очень высокой производительности на определенных классах задач, особенно если используется несколько различных конвейерных процессоров, способных работать одновременно и независимо друг от друга.

Одной из наиболее высокопроизводительных вычислительных конвейерных систем считается CRAY. В этой системе конвейерный принцип обработки используется в максимальной степени. Имеется и конвейер команд, и конвейер арифметических и логических операций. В системе широко применяется совмещенная обработка информации несколькими устройствами. Максимальная пиковая производительность процессора может составлять 12 GFLOPS.

В настоящее время созданы однокристальные векторно-конвейерные процессоры, основными компонентами которых являются скалярный процессор и 8 идентичных векторных устройств, суммарная производительность которых составляет 64 GFLOPS. На их основе построена система SX-6 компании NEC.

## Матричные процессоры

Наиболее распространенными из систем класса один поток команд – множество потоков данных (SIMD) являются матричные системы, которые лучше всего приспособлены для решения задач, характеризующихся параллелизмом независимых объектов или данных. Организация систем подобного типа, на первый взгляд, достаточно проста. Они имеют общее управляющее устройство, генерирующее поток команд и большое число процессорных элементов, работающих параллельно и обрабатывающих каждая свой поток данных. Таким образом, производительность системы оказывается равной сумме производительностей всех процессорных элементов. Однако на практике чтобы обеспечить достаточную эффективность системы при решении широкого круга задач, необходимо организовать связи между процессорными элементами с тем, чтобы наиболее полно загрузить их работой. Именно характер связей между процессорными элементами и определяет разные свойства системы.

Одним из первых матричных процессоров был SOLOMON (60-е годы).

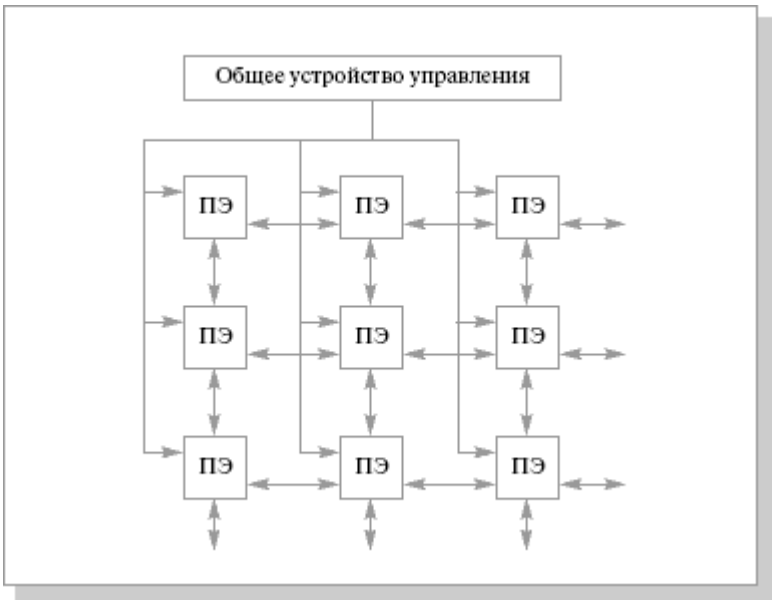


Рис. 6.2. Структура матричной вычислительной системы SOLOMON

Система SOLOMON содержит 1024 процессорных элемента, которые соединены в виде матрицы: 32x32. Каждый процессорный элемент матрицы включает в себя процессор, обеспечивающий выполнение последовательных поразрядных арифметических и логических операций, а также оперативное ЗУ емкостью 16 Кбайт. Длина слова –

переменная от 1 до 128 разрядов. Разрядность слов устанавливается программно. По каналам связи от устройства управления передаются команды и общие константы. В процессорном элементе используется так называемая многомодальная логика, которая позволяет каждому процессорному элементу выполнять или не выполнять общую операцию в зависимости от значений обрабатываемых данных. В каждый момент все активные процессорные элементы выполняют одну и ту же операцию над данными, хранящимися в собственной памяти и имеющими один и тот же адрес.

Идея многомодальности заключается в том, что в каждом процессорном элементе имеется специальный регистр на 4 состояния – регистр моды. Мода (модальность) заносится в этот регистр от устройства управления. При выполнении последовательности команд модальность передается в коде операции и сравнивается с содержимым регистра моды. Если есть совпадения, то операция выполняется. В других случаях процессорный элемент не выполняет операцию, но может, в зависимости от кода, пересылать свои операнды соседнему процессорному элементу. Такой механизм позволяет выделить строку или столбец процессорных элементов, что очень полезно при операциях над матрицами. Взаимодействуют процессорные элементы с периферийным оборудованием через внешний процессор.

Дальнейшим развитием матричных процессоров стала система ILLIAC-4, разработанная фирмой BURROUGHS. Первоначально система должна была включать в себя 256 процессорных элементов, разбитых на группы, каждый из которых должен управляться специальным процессором. Однако по различным причинам была создана система, содержащая одну группу процессорных элементов и управляющий процессор. Если в начале предполагалось достичь быстродействия 1 млрд. операций в секунду, то реальная система работала с быстродействием 200 млн. операций в секунду. Эта система в течение ряда лет считалась одной из самых высокопроизводительных в мире.

В начале 80-х годов в СССР была создана система ПС-2000, которая также является матричной. Основой этой системы является мультипроцессор ПС-2000, состоящий из решающего поля и устройства управления мультипроцессором. Решающее поле строится из одного, двух, четырех или восьми устройств обработки, в каждом из которых 8 процессорных элементов. Мультипроцессор из 64 процессорных элементов обеспечивает быстродействие 200 млн. операций в секунду на коротких операциях.

**7. Способы организации высокопроизводительных процессоров. Клеточные и ДНК-процессоры. Коммуникационные процессоры**

## Клеточные и ДНК-процессоры.

В настоящее время в поисках реальной альтернативы полупроводниковым технологиям создания новых вычислительных систем ученые обращают все большее внимание на биотехнологии, или биокомпьютинг, который представляет собой гибрид информационных, молекулярных технологий, а также биохимии. Биокомпьютинг позволяет решать сложные вычислительные задачи, используя методы, принятые в биохимии и молекулярной биологии, организуя вычисления при помощи живых тканей, клеток, вирусов и биомолекул. Наибольшее распространение получил подход, где в качестве основного элемента (процессора) используются молекулы дезоксирибонуклеиновой кислоты. Центральное место в этом подходе занимает так называемый ДНК-процессор. Кроме ДНК, в качестве биопроцессора могут использоваться также белковые молекулы и биологические мембраны.

## ДНК-процессоры

Так же, как и любой другой процессор, ДНК-процессор характеризуется структурой и набором команд. В нашем случае структура процессора – это структура молекулы ДНК. А набор команд – это перечень биохимических операций с молекулами. Принцип устройства компьютерной ДНК-памяти основан на последовательном соединении четырех нуклеотидов (основных кирпичиков ДНК-цепи). Три нуклеотида, соединяясь в любой последовательности, образуют элементарную ячейку памяти – кодон, совокупность которых формирует затем цепь ДНК. Основная трудность в разработке ДНК-компьютеров связана с проведением избирательных однокодонных реакций (взаимодействий) внутри цепи ДНК. Однако прогресс есть уже и в этом направлении. Существует экспериментальное оборудование, позволяющее работать с одним из 1020 кодонов или молекул ДНК. Другой проблемой является самосборка ДНК, приводящая к потере информации. Ее преодолевают введением в клетку специальных ингибиторов – веществ, предотвращающих химическую реакцию самосшивки.

Использование молекул ДНК для организации вычислений – это не слишком новая идея. Теоретическое обоснование подобной возможности было сделано еще в 50-х годах прошлого века (Р.П. Фейманом). В деталях эта теория была проработана в 70-х годах Ч. Бенеттом и в 80-х М. Конрадом.

Первый компьютер на базе ДНК был создан еще в 1994 г. американским ученым Леонардом Адлеманом. Он смешал в пробирке молекулу ДНК, в которой были закодированы исходные данные, и специальным образом подобранные ферменты. В результате химической реакции структура ДНК изменилась таким образом, что в ней в закодированном виде был представлен ответ задачи. Поскольку вычисления проводились в ходе химической реакции с участием ферментов, на них было затрачено очень мало времени.

Вслед за работой Адлемана последовали другие. Ллойд Смит из Университета Висконсин решил с помощью ДНК задачу доставки четырех сортов пиццы по четырем адресам, которая подразумевала 16 вариантов ответа. Ученые из Принстонского университета

решили комбинаторную шахматную задачу: при помощи РНК нашли правильный ход шахматного коня на доске из девяти клеток (всего их 512 вариантов).

Ричард Липтон из Принстона первым показал, как, используя ДНК, кодировать двоичные числа и решать проблему удовлетворения логического выражения. Суть ее в том, что, имея некоторое логическое выражение, включающее n логических переменных, нужно найти все комбинации значений переменных, делающих выражение истинным. Задачу можно решить только перебором 2n комбинаций. Все эти комбинации легко закодировать с помощью ДНК, а дальше действовать по методике Адлемана. Липтон предложил также способ взлома шифра DES (американский криптографический), трактуемого как своеобразное логическое выражение.

Первую модель биокомпьютера, правда, в виде механизма из пластмассы, в 1999 г. создал Ихуд Шапиро из Вейцмановского института естественных наук. Она имитировала работу "молекулярной машины" в живой клетке, собирающей белковые молекулы по информации с ДНК, используя РНК в качестве посредника между ДНК и белком.

А в 2001 г. Шапиро удалось реализовать вычислительное устройство на основе ДНК, которое может работать почти без вмешательства человека. Система имитирует машину Тьюринга — одну из фундаментальных концепций вычислительной техники. Машина Тьюринга шаг за шагом считывает данные и в зависимости от их значений принимает решения о дальнейших действиях. Теоретически она может решить любую вычислительную задачу. По своей природе молекулы ДНК работают аналогичным образом, распадаясь и рекомбинируясь в соответствии с информацией, закодированной в цепочках химических соединений.

Разработанная в Вейцмановском институте установка кодирует входные данные и программы в состоящих из двух цепей молекулах ДНК и смешивает их с двумя ферментами. Молекулы фермента выполняли роль аппаратного, а молекулы ДНК – программного обеспечения. Один фермент расщепляет молекулу ДНК с входными данными на отрезки разной длины в зависимости от содержащегося в ней кода. А другой рекомбинирует эти отрезки в соответствии с их кодом и кодом молекулы ДНК с программой. Процесс продолжается вдоль входной цепи, и, когда доходит до конца, получается выходная молекула, соответствующая конечному состоянию системы.

Этот механизм может использоваться для решения самых разных задач. Хотя на уровне отдельных молекул обработка ДНК происходит медленно, со скоростью от 500 до 1000 бит/с, что во много миллионов раз медленнее современных кремниевых процессоров, по своей природе она допускает массовый параллелизм. По оценкам Шапиро и его коллег, в одной пробирке может одновременно происходить триллион процессов, так что при потребляемой мощности в единицы нановатт может выполняться миллиард операций в секунду.

В конце февраля 2002 г. появилось сообщение, что фирма Olympus Optical претендует на первенство в создании коммерческой версии ДНК-компьютера, предназначенного для генетического анализа. Машина была создана в сотрудничестве с доцентом Токийского университета Акирой Тояма.

Компьютер, построенный Olympus Optical, имеет молекулярную и электронную составляющие. Первая осуществляет химические реакции между молекулами ДНК, обеспечивает поиск и выделение результата вычислений. Вторая – обрабатывает информацию и анализирует полученные результаты.

Возможностями биокомпьютеров заинтересовались и военные. Американское агентство по исследованиям в области обороны DARPA выполняет проект, получивший название Bio-Comp (Biological Computations, биологические вычисления). Его цель – создание мощных вычислительных систем на основе ДНК.

Пока до практического применения компьютеров на базе ДНК еще очень далеко. Однако в будущем их смогут использовать не только для вычислений, но и как своеобразные нанофабрики лекарств. Поместив подобное "устройство" в клетку, врачи смогут влиять на ее состояние, исцеляя человека от самых опасных недугов.

Клеточные компьютеры представляют собой самоорганизующиеся колонии различных "умных" микроорганизмов, в геном которых удалось включить некую логическую схему, которая могла бы активизироваться в присутствии определенного вещества. Для этой цели идеально подошли бы бактерии, стакан с которыми и представлял бы собой компьютер. Такие компьютеры очень дешевы в производстве. Им не нужна стерильная атмосфера, как при производстве полупроводников.

Главное свойство такого компьютера состоит в том, что каждая его клетка представляет собой миниатюрную химическую лабораторию. Если биоорганизм запрограммирован, то он просто производит нужные вещества. Достаточно вырастить одну клетку, обладающую заданными качествами, и можно легко и быстро вырастить тысячи клеток с такой же программой.

Основная проблема, с которой сталкиваются создатели клеточных биокомпьютеров, – организация всех клеток в единую работающую систему. На сегодня практические достижения в области клеточных компьютеров напоминают достижения 20-х годов в области ламповых и полупроводниковых компьютеров. Сейчас в Лаборатории искусственного интеллекта Массачусетского технологического университета создана клетка, способная хранить на генетическом уровне 1 бит информации. Также разрабатываются технологии, позволяющие единичной бактерии отыскивать своих соседей, образовывать с ними упорядоченную структуру и осуществлять массив параллельных операций.

В 2001 г. американские ученые создали трансгенные микроорганизмы (т. е. микроорганизмы с искусственно измененными генами), клетки которых могут выполнять логические операции И и ИЛИ.

Специалисты лаборатории Оук-Ридж, штат Теннесси, использовали способность генов синтезировать тот или иной белок под воздействием определенной группы химических раздражителей. Ученые изменили генетический код бактерий Pseudomonas putida таким образом, что их клетки обрели способность выполнять простые логические операции. Например, при выполнении операции И в клетку подаются два вещества (по сути – входные операнды), под влиянием которых ген вырабатывает определенный белок. Теперь ученые пытаются создать на базе этих клеток более сложные логические элементы, а также подумывают о возможности создания клетки, выполняющей параллельно несколько логических операций.

Потенциал биокомпьютеров очень велик. К достоинствам, выгодно отличающим их от компьютеров, основанных на кремниевых технологиях, относятся:

- более простая технология изготовления, не требующая для своей реализации столь жестких условий, как при производстве полупроводников;

- использование не бинарного, а тернарного кода (информация кодируется тройками нуклеотидов), что позволит за меньшее количество шагов перебрать большее число вариантов при анализе сложных систем;
- потенциально исключительно высокая производительность, которая может составлять до $10^{14}$ операций в секунду за счет одновременного вступления в реакцию триллионов молекул ДНК;
- возможность хранить данные с плотностью, в триллионы раз превышающей показатели оптических дисков;
- исключительно низкое энергопотребление.

Однако, наряду с очевидными достоинствами, биокомпьютеры имеют и существенные недостатки, такие как:

- сложность со считыванием результатов – современные способы определения кодирующей последовательности несовершенны, сложны, трудоемки и дороги;
- низкая точность вычислений, связанная с возникновением мутаций, прилипанием молекул к стенкам сосудов и т.д.;
- невозможность длительного хранения результатов вычислений в связи с распадом ДНК в течение времени.

Хотя до практического использования биокомпьютеров еще очень далеко, и они вряд ли будут рассчитаны на широкие массы пользователей, предполагается, что они найдут достойное применение в медицине и фармакологии, а также с их помощью станет возможным объединение информационных и биотехнологий.

## Коммуникационные процессоры

Коммуникационные процессоры – это микрочипы, представляющие собой нечто среднее между жесткими специализированными интегральными микросхемами и гибкими процессорами общего назначения. Коммуникационные процессоры программируются, как и привычные для нас ПК-процессоры, но построены с учетом сетевых задач, оптимизированы для сетевой работы и на их основе производители – как процессоров, так и оборудования – пишут программное обеспечение для специфических приложений. Коммуникационный процессор имеет собственную память и оснащен высокоскоростными внешними каналами для соединения с другими процессорными узлами. Его присутствие позволяет в значительной мере освободить вычислительный процессор от нагрузки, связанной с передачей сообщений между процессорными узлами. Скоростной коммуникационный процессор с RISC-ядром позволяет управлять обменом данными по нескольким независимым каналам, поддерживать практически все распространенные протоколы обмена, гибко и эффективно распределять и обрабатывать последовательные потоки данных с временным разделением каналов.

Сама идея создания процессоров, предназначенных для оптимизации сетевой работы и при этом достаточно универсальных для программной модификации, родилась в связи с необходимостью устранить различия в подходах к созданию локальных сетей (различные подходы к архитектуре сети, классификации потоков и т.д.). Несомненно, истинной причиной бума сетевых процессоров стало ускорение темпов развития рынка. Когда рынок движется на "Internet-скорости", поставщики оборудования уже не могут тратить по два года на разработку специализированных микросхем для реализации конкретных сетевых функций. Эти два года (и вложенные деньги) будут потрачены зря, если рынок за это время уйдет в другом направлении. Выход один – разрабатывать процессоры, которые поставщики оборудования могут внедрить и выпустить в новом продукте в течение

нескольких месяцев. Бум сетевых процессоров, окончательно оформившийся в середине 1999 г., не был кратким, и в последующие годы индустрия развивалась крайне бурно.

По прогнозам одних аналитиков, очень скоро специальные микросхемы будут вытеснены стандартными сетевыми процессорами. Другие аналитики считают, что у сетевых процессоров, без сомнения, есть будущее, но они смогут преобладать только на некоторых сегментах рынка, где необходимы укороченные циклы разработки, быстрота и гибкость.

Предполагается, что на этом рынке не будет преобладать какая-либо одна компания, как, например, Intel на рынке ПК. Однако считается, что Intel останется одним из ключевых игроков, разделив $2,9 млрд. с IBM, Motorola и дюжиной других компаний.

Новая серия коммуникационных процессоров INTEL IXP4xx построена на базе распределенной архитектуры XScale и включает мощные мультимедийные возможности, а также развитые сетевые интерфейсы Ethernet. Сочетание высокой производительности и низкого энергопотребления позволяет эффективно применять коммуникационные процессоры INTEL не только в классических сетевых приложениях, но и для построения Internet-ориентированных встраиваемых систем промышленного назначения.

Эффективность работы промышленных предприятий сегодня напрямую зависит от гибкости применяемых систем автоматизированного управления. Крупные производственные установки требуют использования нескольких децентрализованных систем управления, связанных друг с другом мощной информационной сетью, способной работать в сложных промышленных условиях. Зачастую эти средства промышленной коммуникации призваны обеспечить возможность гибкого управления, программирования и контроля работы распределенных систем управления из удаленных диспетчерских пунктов. Достижение этих целей возможно с помощью коммуникационных процессоров, предназначенных для подключения персональных компьютеров к промышленным информационным сетям. Дополнительные возможности, обеспечиваемые коммуникационными процессорами, должны быть интересны, прежде всего, тем пользователям, которым необходимо осуществлять сложные транзакции или наладить прямую голосовую и видеосвязь в рамках сетевой инфраструктуры.

**8. Способы организации высокопроизводительных процессоров. Процессоры баз данных. Потоковые процессоры. Нейронные процессоры. Процессоры с многозначной (нечеткой) логикой**

## Процессоры баз данных

Процессорами (машинами) баз данных в настоящее время принято называть программно-аппаратные комплексы, предназначенные для выполнения всех или некоторых функций систем управления базами данных (СУБД). Если в свое время системы управления базами данных предназначались в основном для хранения текстовой и числовой информации, то теперь они рассчитаны на различные форматы данных, в том числе графические, звуковые и видео. Процессоры баз данных выполняют функции управления и распространения, обеспечивают дистанционный доступ к информации через шлюзы, а также репликацию обновленных данных с помощью различных механизмов тиражирования. В больших информационных системах наметился переход от тривиальной архитектуры "клиент – сервер" к трехуровневой архитектуре с распределенными базами данных (клиент, сервер с СУБД и серверы собственно с данными).

Современные процессоры баз данных должны обеспечивать естественную связь накапливаемой в базах данных информации со средствами оперативной обработки транзакций и Internet-приложениями. Это должны быть системы, которые дают пользователям возможность в любой момент обратиться к корпоративным данным и проанализировать их, вне зависимости от того, где эти данные размещаются.

Решение таких задач требует существенного увеличения производительности систем управления базами данных. Однако традиционная программная реализация многочисленных функций современных СУБД на ЭВМ общего назначения приводит к появлению громоздких и непроизводительных систем с недостаточно высокой надежностью. Необходим поиск новых архитектурных и аппаратных решений. Интенсивные исследования, проводимые в этой области в настоящее время, привели к пониманию необходимости использования в качестве процессоров баз данных специализированных параллельных вычислительных систем. Создание такого рода систем связывается с реализацией параллелизма при выполнении последовательности операций и транзакций, а также конвейерной потоковой обработки данных.

## Потоковые процессоры

Потоковыми называют процессоры, в основе работы которых лежит принцип обработки многих данных с помощью одной команды. Согласно классификации Флинна, они принадлежат к SIMD (single instruction stream / multiple data stream) архитектуре. Технология SIMD позволяет выполнять одно и то же действие, например, вычитание и сложение, над несколькими наборами чисел одновременно. SIMD-операции для чисел двойной точности с плавающей запятой ускоряют работу ресурсоемких приложений для создания контента, трехмерного рендеринга, финансовых расчетов и научных задач. Кроме того, усовершенствованы возможности 64-разрядной технологии MMX (целочисленных SIMD-команд); эта технология распространена на 128-разрядные числа, что позволяет ускорить обработку видео, речи, шифрование, обработку изображений и фотографий. Потоковый процессор повышает общую производительность, что особенно важно при работе с 3D-графическими объектами.

Может быть отдельный потоковый процессор (Single-streaming processor — SSP) и многопотоковый процессор (Multi-Streaming Processor – MSP).

Ярким представителем потоковых процессоров является семейство процессоров Intel, начиная с Pentium III, в основе работы которых лежит технология Streaming SIMD Extensions (SSE, потоковая обработка по принципу "одна команда – много данных"). Эта технология позволяет выполнять такие сложные и необходимые в век Internet задачи как обработка речи, кодирование и декодирование видео- и аудиоданных, разработка трехмерной графики и обработка изображений.

Представителями класса SIMD считаются матрицы процессоров: ILLIAC IV, ICL DAP, Goodyear Aerospace MPP, Connection Machine 1 и т.п. В таких системах единое управляющее устройство контролирует множество процессорных элементов. Каждый процессорный элемент получает от устройства управления в каждый фиксированный момент времени одинаковую команду и выполняет ее над своими локальными данными.

Другими представителями SIMD-класса являются векторные процессоры, в основе которых лежит векторная обработка данных. Векторная обработка увеличивает производительность процессора за счет того, что обработка целого набора данных (вектора) производится одной командой. Векторные компьютеры манипулируют массивами сходных данных подобно тому, как скалярные машины обрабатывают отдельные элементы таких массивов. В этом случае каждый элемент вектора надо рассматривать как отдельный элемент потока данных. При работе в векторном режиме векторные процессоры обрабатывают данные практически параллельно, что делает их в несколько раз более быстрыми, чем при работе в скалярном режиме. Максимальная скорость передачи данных в векторном формате может составлять 64 Гбайт/с, что на 2 порядка быстрее, чем в скалярных машинах. Примерами систем подобного типа являются, например, процессоры фирм NEC и Hitachi.

## Нейронные процессоры

Одно из наиболее перспективных направлений разработки принципиально новых архитектур вычислительных систем тесно связано с созданием компьютеров нового поколения на основе принципов обработки информации, заложенных в искусственных нейронных сетях (НС).

Первые практические работы по искусственным нейросетям и нейрокомпьютерам начались еще в 40-50-е годы. Под нейронной сетью обычно понимают совокупность элементарных преобразователей информации, называемых "нейронами", которые определенным образом соединены друг с другом каналами обмена информации – "синаптическими связями".

Нейрон, по сути, представляет собой элементарный процессор, характеризующийся входным и выходным состоянием, передаточной функцией (функция активации) и локальной памятью.

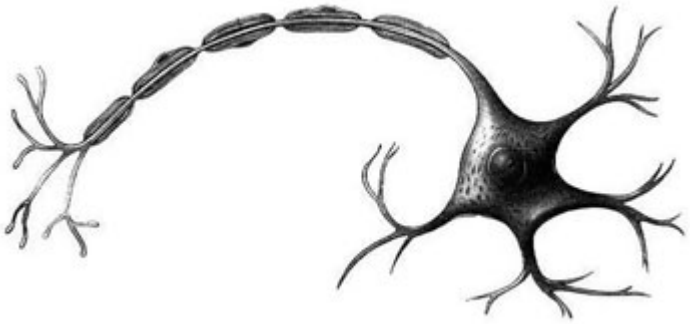

Рис. 8.1.

Состояния нейронов изменяются в процессе функционирования и составляют кратковременную память нейросети. Каждый нейрон вычисляет взвешенную сумму пришедших к нему по синапсам сигналов и производит над ней нелинейное преобразование. При пересылке по синапсам сигналы умножаются на некоторый весовой коэффициент. В распределении весовых коэффициентов заключается информация, хранящаяся в ассоциативной памяти НС. Основным элементом проектирования сети является ее обучение. При обучении и переобучении НС ее весовые коэффициенты изменяются. Однако они остаются постоянными при функционировании нейросети, формируя долговременную память.

НС может состоять из одного слоя, из двух, из трех и большего числа слоев, однако, как правило, для решения практических задач более трех слоев в НС не требуется.

Число входов НС определяет размерность гиперпространства, в котором входные сигналы могут быть представлены точками или гиперобластями из близко расположенных точек. Количество нейронов в слое сети определяет число гиперплоскостей в гиперпространстве. Вычисление взвешенных сумм и выполнение нелинейного преобразования позволяют определить, с какой стороны от той или иной гиперплоскости находится точка входного сигнала в гиперпространстве.

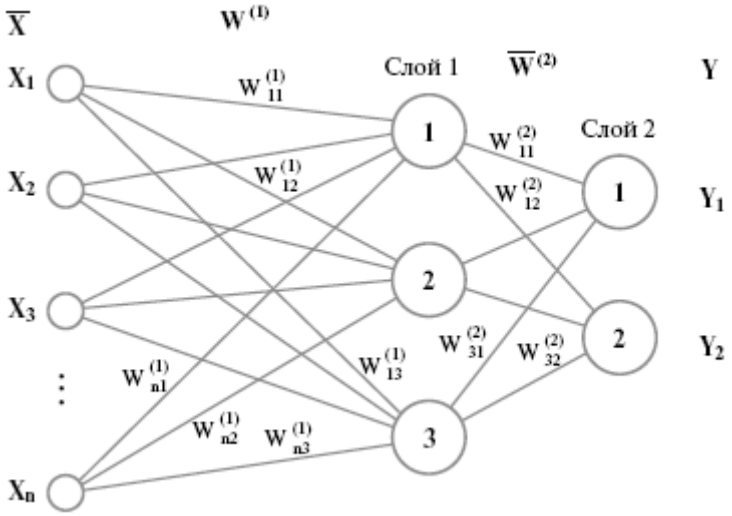


Рис. 8.2.

Возьмем классическую задачу распознавания образов: определение принадлежности точки одному из двух классов. Такая задача естественным образом решается с помощью одного нейрона. Он позволит разделить гиперпространство на две непересекающиеся и

невложенные гипербласти. Входные сигналы в задачах, решаемых с помощью нейросетей, образуют в гиперпространстве сильно вложенные или пересекающиеся области, разделить которые с помощью одного нейрона невозможно. Это можно сделать, только проведя нелинейную гиперповерхность между областями. Ее можно описать с помощью полинома n-го порядка. Однако степенная функция слишком медленно считается и поэтому очень неудобна для вычислительной техники. Альтернативным вариантом является аппроксимация гиперповерхности линейными гиперплоскостями. Понятно, что при этом точность аппроксимации зависит от числа используемых гиперплоскостей, которое, в свою очередь, зависит от числа нейронов в сети. Отсюда возникает потребность в аппаратной реализации как можно большего числа нейронов в сети. Количество нейронов в одном слое сети определяет ее разрешающую способность. Однослойная НС не может разделить линейно зависимые образы. Поэтому важно уметь аппаратно реализовывать многослойные НС.

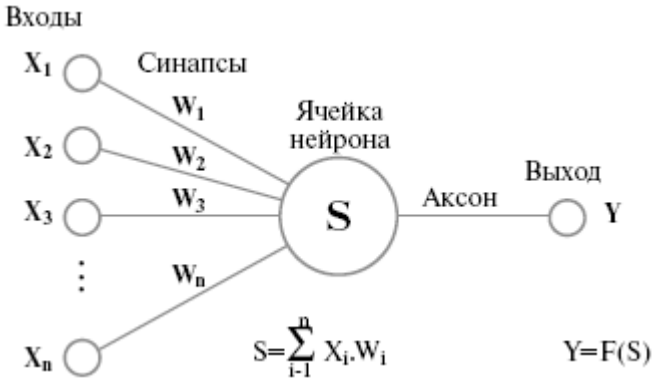


Рис. 8.3.

Искусственные нейронные сети отличаются удивительными свойствами. Они не требуют детализированной разработки программного обеспечения и открывают возможности решения задач, для которых отсутствуют теоретические модели или эвристические правила, определяющие алгоритм решения. Такие сети обладают способностью адаптироваться к изменениям условий функционирования, в том числе к возникновению заранее непредусмотренных факторов. По своей природе НС являются системами с очень высоким уровнем параллелизма.

В нейрокомпьютерах используются принципы обработки информации, осуществляемые в реальных нейронных сетях. Эти принципиально новые вычислительные средства с нетрадиционной архитектурой позволяют выполнять высокопроизводительную обработку информационных массивов большой размерности. В отличие от традиционных вычислительных систем, нейросетевые вычислители, аналогично нейронным сетям, дают возможность с большей скоростью обрабатывать информационные потоки дискретных и непрерывных сигналов, содержат простые вычислительные элементы и с высокой степенью надежности позволяют решать информационные задачи обработки данных, обеспечивая при этом режим самоперестройки вычислительной среды в зависимости от полученных решений.

Вообще говоря, под термином "нейрокомпьютер" в настоящее время подразумевается довольно широкий класс вычислителей. Это происходит по той простой причине, что формально нейрокомпьютером можно считать любую аппаратную реализацию нейросетевого алгоритма, от простой модели биологического нейрона до системы распознавания символов или движущихся целей. Нейрокомпьютеры не являются компьютерами в общепринятом смысле этого слова. В настоящее время технология еще

не достигла того уровня развития, при котором можно было бы говорить о нейрокомпьютере общего назначения (который являлся бы одновременно искусственным интеллектом). Системы с фиксированными значениями весовых коэффициентов – вообще самые узкоспециализированные из нейросетевого семейства. Обучающиеся сети более адаптированы к разнообразию решаемых задач. Обучающиеся сети более гибки и способны к решению разнообразных задач. Таким образом, построение нейрокомпьютера – это каждый раз широчайшее поле для исследовательской деятельности в области аппаратной реализации практически всех элементов НС.

В начале 21 века, в отличие от 40-50-х годов прошлого столетия, существует объективная практическая потребность научиться создавать нейрокомпьютеры, т.е. необходимо аппаратно реализовать довольно много параллельно действующих нейронов, с миллионами фиксированных или параллельно адаптивно модифицируемых связей-синапсов, с несколькими полносвязными слоями нейронов.

В то же время физические возможности технологии интегральной электроники не безграничны. Геометрические размеры транзисторов больше нельзя физически уменьшать: при технологически достижимых размерах порядка 1 мкм и меньше проявляются физические явления, незаметные при больших размерах активных элементов – начинают сильно сказываться квантовые размерные эффекты. Транзисторы перестают работать как транзисторы.

Для аппаратной реализации НС необходим новый носитель информации. Таким новым носителем информации может быть свет, который позволит резко, на несколько порядков, повысить производительность вычислений.

Единственной технологией аппаратной реализации НС, способной в будущем прийти на смену оптике и оптоэлектронике, является нанотехнология, способная обеспечить не только физически предельно возможную степень интеграции субмолекулярных квантовых элементов с физически предельно возможным быстродействием, но и столь необходимую для аппаратной реализации НС трехмерную архитектуру.

Длительное время считалось, что нейрокомпьютеры эффективны для решения так называемых неформализуемых и плохо формализуемых задач, связанных с необходимостью включения в алгоритм решения задачи процесса обучения на реальном экспериментальном материале. В первую очередь к таким задачам относилась задача аппроксимации частного вида функций, принимающих дискретное множество значений, т. е. задача распознавания образов.

В настоящее время к этому классу задач добавляется класс задач, иногда не требующий обучения на экспериментальном материале, но хорошо представимый в нейросетевом логическом базисе. К ним относятся задачи с ярко выраженным естественным параллелизмом обработки сигналов, обработка изображений и др. Подтверждением точки зрения, что в будущем нейрокомпьютеры будут более эффективными, чем прочие архитектуры, может, в частности, служить резкое расширение в последние годы класса общематематических задач, решаемых в нейросетевом логическом базисе. К ним, кроме перечисленных выше, можно отнести задачи решения линейных и нелинейных алгебраических уравнений и неравенств большой размерности; систем нелинейных дифференциальных уравнений; уравнений в частных производных; задач оптимизации и других задач.

## Процессоры с многозначной (нечеткой) логикой

Идея построения процессоров с нечеткой логикой (fuzzy logic) основывается на нечеткой математике. Математическая теория нечетких множеств, предложенная проф. Л.А. Заде, являясь предметом интенсивных исследований, открывает все более широкие возможности перед системными аналитиками. Основанные на этой теории различные компьютерные системы, в свою очередь, существенно расширяют область применения нечеткой логики.

Подходы нечеткой математики позволяют оперировать входными данными, непрерывно меняющимися во времени, и значениями, которые невозможно задать однозначно, такими, например, как результаты статистических опросов. В отличие от традиционной формальной логики, известной со времен Аристотеля и оперирующей точными и четкими понятиями типа истина и ложь, да и нет, ноль и единица, нечеткая логика имеет дело со значениями, лежащими в некотором (непрерывном или дискретном) диапазоне.

Функция принадлежности элементов к заданному множеству также представляет собой не жесткий порог "принадлежит – не принадлежит", а плавную сигмоиду, проходящую все значения от нуля до единицы. Теория нечеткой логики позволяет выполнять над такими величинами весь спектр логических операций – объединение, пересечение, отрицание и др.

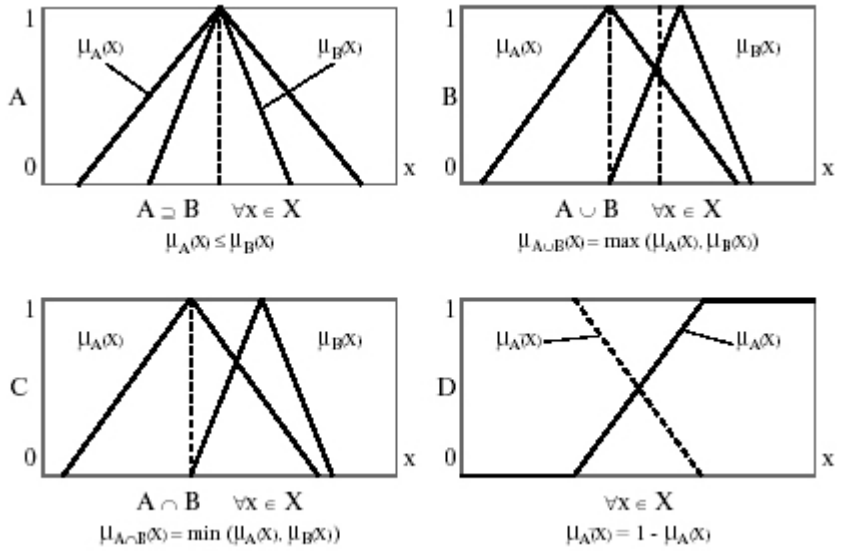


Рис. 8.4. Операции включения (A), объединения (B), пересечения (C) и дополнения (D) НМ

Согласно знаменитой теореме FAT (Fuzzy Approximation Theorem), доказанной Коско, любая математическая система может быть аппроксимирована системой, основанной на нечеткой логике. Свое второе рождение теория нечеткой логики пережила в начале восьмидесятых годов, когда сразу несколько групп исследователей (в основном в США и Японии) занялись созданием электронных систем различного применения, использующих нечеткие управляющие алгоритмы. Используя преимущества нечеткой логики, заключающиеся в простоте содержательного представления, можно упростить проблему, представить ее в более доступном виде и повысить производительность системы.

Задачи с помощью нечеткой логики решаются по следующему принципу:

1. численные данные (показания измерительных приборов, результаты анкетирования) фаззируются (переводятся в нечеткий формат);

2. обрабатываются по определенным правилам;
3. дефаззируются и в виде привычной информации подаются на выход.

Оказалось возможным создание нечеткого процессора, позволяющего выполнять различные нечеткие операции и приближенные рассуждения (нечеткий вывод) в соответствии с правилами логического вывода. В 1986 году в AT and T Bell Labs создавались процессоры с "прошитой" нечеткой логикой обработки информации. В начале 90-х компания Adaptive Logic из США выпустила кристалл, сделанный по аналогово-цифровой технологии. Он позволит сократить сроки конструирования многих встроенных систем управления реального времени, заменив собой традиционные схемы нечетких микроконтроллеров. Аппаратный процессор нечеткой логики второго поколения принимает аналоговые сигналы, переводит их в нечеткий формат, затем, применяя соответствующие правила, преобразует результаты в формат обычной логики и далее – в аналоговый сигнал. Все это осуществляется без внешних запоминающих устройств, преобразователей и какого бы то ни было программного обеспечения нечеткой логики. Этот микропроцессор относительно прост по сравнению с громоздкими программными обеспечениями. Но так как его основу составляет комбинированный цифровой/аналоговый кристалл, он функционирует на очень высоких скоростях (частота отсчетов входного сигнала – 10 кГц, а скорость расчета – 500 тыс. правил/с), что во многих случаях приводит к лучшим результатам в системах управления по сравнению с более сложными, но медлительными программами.


Рис. 8.5. Архитектура нечеткого компьютера (МНВ-механизм нечеткого ввода)

В Европе и США ведутся интенсивные работы по интеграции fuzzy-команд в ассемблеры промышленных контроллеров встроенных устройств (чипы Motorola 68HC11.12.21). Такие аппаратные средства позволяют в несколько раз увеличить скорость выполнения приложений и компактность кода по сравнению с реализацией на обычном ядре. Кроме того, разрабатываются различные варианты fuzzy-сопроцессоров, которые контактируют с центральным процессором через общую шину данных, концентрируют свои усилия на размывании/уплотнении информации и оптимизации использования правил (продукты Siemens Nixdorf).

Идеи нечеткой логики не являются панацеей и не смогут совершить переворот в компьютерном мире. Нечеткая логика не решит тех задач, которые не решаются на основе логики двоичной, но во многих случаях она удобнее, производительнее и дешевле. Разработанные на ее основе специализированные аппаратные решения (fuzzy-вычислители) позволят получить реальные преимущества в быстродействии. Если каскадировать fuzzy-вычислители, мы получим один из вариантов нейропроцессора или нейронной сети. Во многих случаях эти понятия просто объединяют, называя общим термином "neuro-fuzzy logic".

В настоящее время перспективой использовать процессоры, основанные на нечеткой логике, всерьез заинтересовались военные. Известно, что NASA рассматривает возможность применения (если еще не применяет) нечетких систем для управления процессами стыковки космических аппаратов.

## 9. Коммутаторы для многопроцессорных вычислительных систем. Простые коммутаторы

Коммуникационные среды вычислительных систем (ВС) состоят из адаптеров вычислительных модулей (ВМ) и коммутаторов, обеспечивающих соединения между ними. Используются как простые коммутаторы, так и составные, компонуемые из набора простых. Простые коммутаторы могут соединять лишь малое число ВМ в силу физических ограничений, однако обеспечивают при этом минимальную задержку при установлении соединения. Составные коммутаторы, обычно строящиеся из простых в виде многокаскадных схем с помощью линий "точка-точка", преодолевают ограничение на малое количество соединений, однако увеличивают и задержки.

### Простые коммутаторы

Типы простых коммутаторов:

- с временным разделением;
- с пространственным разделением.

Достоинства: простота управления и высокое быстродействие.

Недостатки: малое количество входов и выходов.

Примеры использования:

- простые коммутаторы с временным разделением используются в системах SMP Power Challenge от SGI,
- простые коммутаторы с пространственным разделением (Gigaplane) используются в семействе Sun Ultra Enterprise.

### Простые коммутаторы с временным разделением

Простые коммутаторы с временным разделением называются также шинами или шинными структурами. Все устройства подключаются к общей информационной магистрали, используемой для передачи информации между ними (рис. 9.1). Обычно шина является пассивным элементом, управление передачами осуществляется передающими и принимающими устройствами.

Процесс передачи выглядит следующим образом.

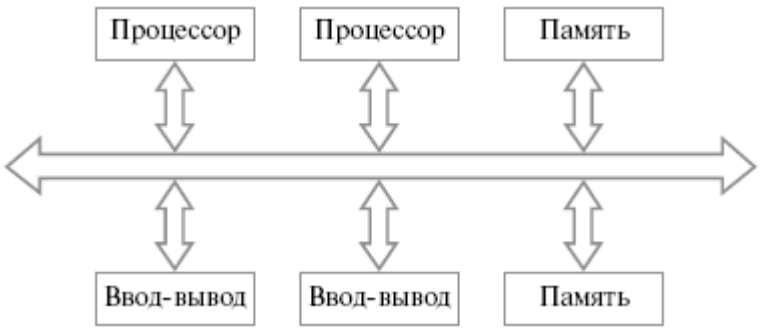

Рис. 9.1. Общая схема шинной структуры

Передающее устройство сначала получает доступ к шине, далее пытается установить контакт с устройством-адресатом и определить его способность к приему данных. Принимающее устройство распознает свой адрес на шине и отвечает на запрос передающего. Далее передающее устройство сообщает, какие действия должно произвести принимающее устройство в ходе взаимодействия. После этого происходит передача данных.

Так как шина является общим ресурсом, за доступ к которому соревнуются подключенные к ней устройства, необходимы методы управления предоставлением доступа устройств к шине. Возможно использование центрального устройства для управления доступом к шине, однако это уменьшает масштабируемость и гибкость системы.

Для разрешения конфликтов, возникающих при одновременном запросе устройств на доступ к шине, используются различные приемы, в частности:

- назначение каждому устройству уникального приоритета (статического или динамического);
- использование очереди запросов FIFO;
- выделение фиксированных временных интервалов каждому устройству.

## Алгоритмы арбитража

## Статические приоритеты

Каждое устройство в системе получает уникальный приоритет, – при одновременном запросе нескольких устройств на передачу доступ к шине предоставляется устройству с наивысшим приоритетом. На практике часто используется соединение устройств в цепь, при котором приоритет устройства определяется местом его подключения к шине. Для контроля доступа к шине используется отдельный блок управления.

## Динамические приоритеты

Так же, как и в предыдущем алгоритме, устройства получают уникальные приоритеты, однако в отличие от него эти приоритеты непостоянны во времени. Приоритеты динамически изменяются, предоставляя устройствам более или менее равные шансы получения доступа к шине. Чаще всего применяются следующие способы изменения приоритетов: предоставление наивысшего приоритета устройству, наиболее долго не использовавшему шину, и циклическая смена приоритетов. Контроль доступа к шине осуществляет устройство, получившее доступ к шине в предыдущем цикле арбитража.

## Фиксированные временные интервалы

Все устройства по порядку получают одинаковые временные интервалы для осуществления передачи. Если устройство не имеет данных для передачи, то интервал тем не менее следующему устройству не предоставляется.

## Очередь FIFO

Создается очередь запросов "первый пришел – первый ушел", однако сохраняется проблема арбитража между почти одновременными запросами, а также возникает необходимость поддержания очереди запросов достаточной длины. Преимуществом данного алгоритма является возможность достижения максимальной пропускной способности шины.

## Особенности реализации шин

Внутри микросхем шины используются для объединения функциональных блоков микропроцессоров, микросхем памяти, микроконтроллеров. Шины используются для объединения устройств на печатных платах и печатных плат в блоках. В последнее время широко применяются шины следующих стандартов:

- ISA – Industry Standard Architecture
- EISA – Extended ISA
- VESA – Video Electronics Standards Association
- PCI – Peripheral Computer Interconnect
- I2C – Inter Integrated Circuit
- AGP – Accelerated Graphic Port

Шины используются также в мезонинной технологии, где на большой плате устанавливается один или несколько шинных разъемов для установки меньших плат, так называемых мезонинов.

Шины, объединяющие устройства, из которых состоит вычислительная система, являются критическим ресурсом, отказ которого может привести к отказу всей системы. Шины обладают также рядом принципиальных ограничений. Возможность масштабируемости шинных структур ограничивается временем, затрачиваемым на арбитраж, и количеством устройств, подключенных к шине. При этом чем больше подключенных устройств, тем больше времени затрачивается на арбитраж. Время арбитража ограничивает и пропускную способность шины. Кроме того, в каждый момент времени шина используется для передачи только одним устройством, что становится узким местом при увеличении количества подключенных устройств. пропускная способность шины ограничивается ее шириной – количеством проводников, используемых для передачи данных, – и тактовой частотой ее работы. Данные величины имеют физические ограничения.

## Простые коммутаторы с пространственным разделением

Простые коммутаторы с пространственным разделением позволяют одновременно соединять любой вход с любым одним выходом (ординарные) или несколькими выходами (неординарные). Такие коммутаторы представляют собой совокупность мультиплексоров, количество которых соответствует количеству выходов коммутатора, при этом каждый вход коммутатора должен быть заведен на все мультиплексоры. Структура этих коммутаторов показана на рис. 9.2.

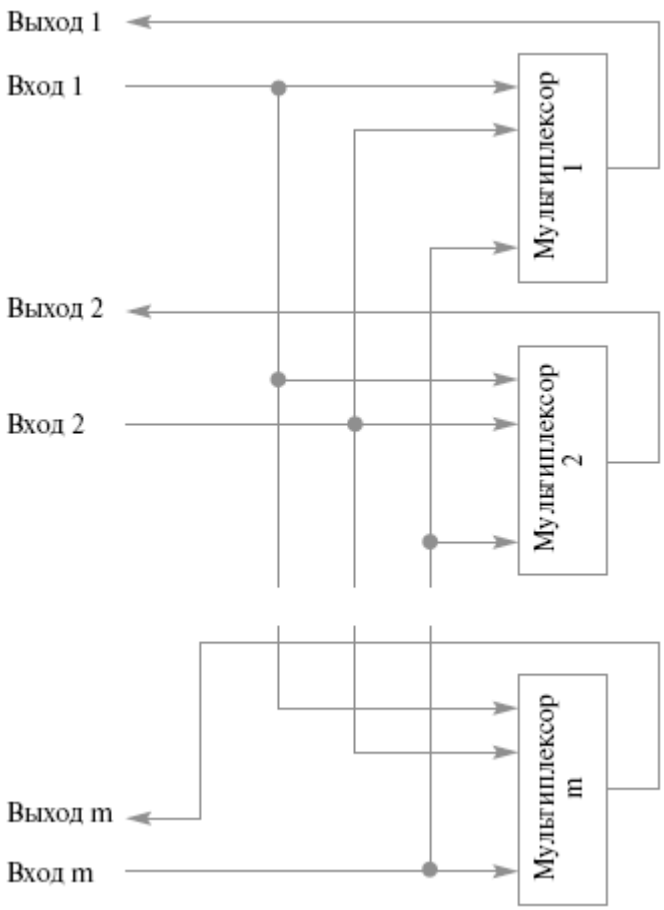


Рис. 9.2. Простой коммуникатор с пространственным разделением

Достоинства:

- возможность одновременного контакта со всеми устройствами;
- минимальная задержка;

Недостатки:

- высокая сложность порядка n x m, где n – количество входов, m – количество выходов;
- сложность обеспечения надежности.

**10. Коммутаторы для многопроцессорных вычислительных систем. Составные коммутаторы. Распределенные составные коммутаторы**

## Составные коммутаторы

Простые коммутаторы имеют ограничения на число входов и выходов, а также могут требовать большого количества оборудования при увеличении этого числа (в случае пространственных коммутаторов). Поэтому для построения коммутаторов с большим количеством входов и выходов используют совокупность простых коммутаторов, объединенных с помощью линий "точка-точка".

Составные коммутаторы имеют задержку, пропорциональную количеству простых коммутаторов, через которые проходит сигнал от входа до выхода, т.е. числу каскадов. Однако объем оборудования составного коммутатора меньше, чем простого с тем же количеством входов и выходов.

Чаще всего составные коммутаторы строятся из прямоугольных коммутаторов 2 x 2 с двумя входами и выходами. Они имеют два состояния: прямое пропускание входов на соответствующие выходы и перекрестное пропускание. Коммутатор 2 x 2 состоит из собственно блока коммутации данных и блока управления. Блок управления в зависимости от поступающих на него управляющих сигналов определяет, какой тип соединения следует осуществить в блоке коммутации - прямой или перекрестный. При этом если оба входа хотят соединиться с одним выходом, то коммутатор разрешает конфликт и связывает с данным выходом только один вход, а запрос на соединение со стороны второго блокируется или отвергается.

## Коммутатор Клоза

Коммутатор Клоза может быть построен в качестве альтернативы для прямоугольного коммутатора с (m x d) входами и (m x d) выходами. Он формируется из трех каскадов коммутаторов: m коммутаторов (d x d) во входном каскаде, m коммутаторов (d x d) в выходном и d промежуточных коммутаторов (m x m).

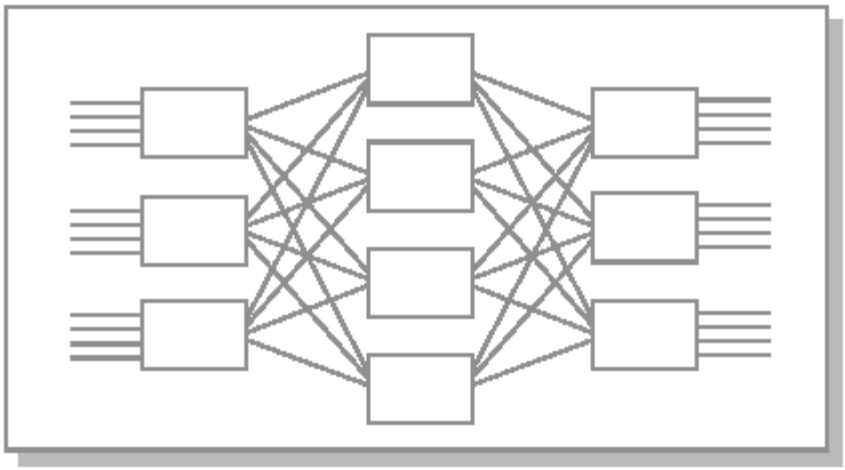

Рис. 10.1. Коммутатор Клоза 3 x 4

Соединения внутри коммутатора устроены следующим образом:

- j-й выход i-ого коммутатора входного каскада соединен с i-ым входом j-ого промежуточного коммутатора;
- j-й вход k-ого коммутатора выходного каскада соединен с k-ым выходом j-ого промежуточного коммутатора.

Данный тип составных коммутаторов позволяет соединять любой вход с любым выходом, однако при установленных соединениях добавление нового соединения может потребовать разрыва и переустановления всех соединений.

## Баньян-сети

Коммутаторы этого типа строятся на базе прямоугольных коммутаторов таким образом, что существует только один путь от каждого входа к каждому выходу.

Наиболее важной разновидностью баньян-сетей является дельта-сеть. Она формируется из прямоугольных коммутаторов (a x b) и представляет собой n-каскадный коммутатор с an входами и bn выходами. Составляющие коммутаторы соединены так, что для соединения любого входа и выхода образуется единственный путь одинаковой для всех пар входов и выходов длины.

## Распределенные составные коммутаторы

В распределенных вычислительных системах ресурсы разделяются между задачами, каждая из которых исполняется на своем подмножестве процессоров. В связи с этим возникает понятие близости процессоров, которая является важной для активно взаимодействующих процессоров. Обычно близость процессоров выражается в различной каскадности соединений, различных расстояниях между ними.

Один из вариантов создания составных коммутаторов заключается в объединении прямоугольных коммутаторов (v+1 x v+1), v > 1 таким образом, что один вход и один выход каждого составляющего коммутатора служат входом и выходом составного коммутатора. К каждому внутреннему коммутатору подсоединяются процессор и память, образуя вычислительный модуль с v-каналами для соединения с другими вычислительными модулями. Свободные v-входов и v-выходов каждого вычислительного модуля соединяются линиями "точка-точка" с входами и выходами других коммутаторов, образуя граф межмодульных связей.

Наиболее эффективным графом межмодульных связей с точки зрения организации обмена данными между вычислительными модулями является полный граф. В этом случае между каждой парой вычислительных модулей существует прямое соединение. При этом возможны одновременные соединения между произвольными вычислительными модулями.

Однако обычно создать полный граф межмодульных связей невозможно по ряду причин. Обмен данными приходится производить через цепочки транзитных модулей. Из-за этого увеличиваются задержки, и ограничивается возможность установления одновременных соединений. Таким образом, эффективный граф межмодульных связей должен минимизировать время межмодульных обменов и максимально увеличить количество одновременно активизированных соединений. Кроме того, на выбор графа межмодульных связей влияет учет отказов и восстановлений вычислительных модулей и линий связи.

## Граф межмодульных связей Convex Exemplar SPP1000

В качестве примера реального графа межмодульных связей рассмотрим структуру системы Convex Exemplar SPP1000. В основе каждого составного блока системы лежит прямоугольный коммутатор (5 x 5), до 16 подобных блоков объединяются каналами "точка-точка" в кольцо (одномерный тор), состоящее из четырех независимых подканалов.

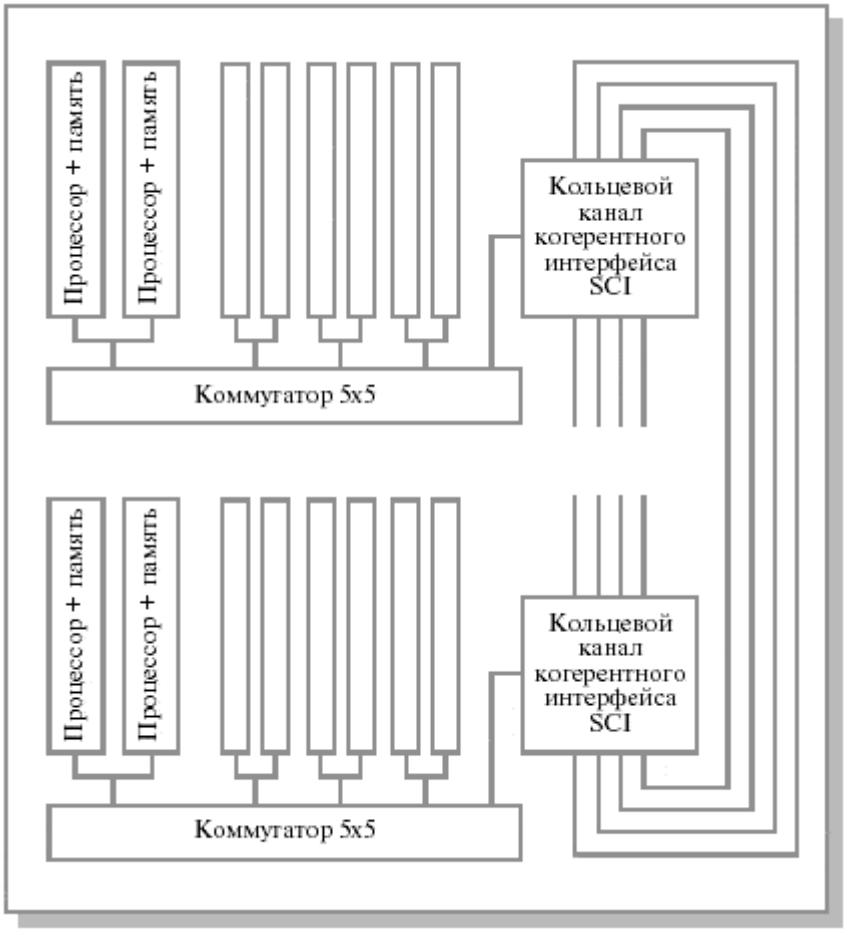


Рис. 10.2. Граф межмодульных связей Convex Exemplar SPP1000

Внутри каждого блока четыре входа и выхода прямоугольного коммутатора (5 x 5) используются для взаимодействия устройств внутри блока (при этом в каждом блоке располагается по два процессора), пятые вход и выход используются для объединения блоков в кольцо. При этом каждый из четырех кольцевых каналов рассматривается как независимый ресурс, и система сохраняет работоспособность до тех пор, пока существует хотя бы один функционирующий кольцевой канал.

## Граф межмодульных связей МВС-100

Система МВС-100 предлагает блочный подход к построению архитектуры параллельной вычислительной системы. Структурный модуль системы состоит из 16 вычислительных узлов, образующих матрицу 4x4 (рис. 10.3). Угловые узлы соединяются попарно по диагонали, таким образом, максимальная длина пути между любой парой элементов равна трем. В исходной же матрице 4 x 4 эта длина равна шести. Каждый блок имеет 12 выходов, что позволяет объединять их в более сложные структуры.

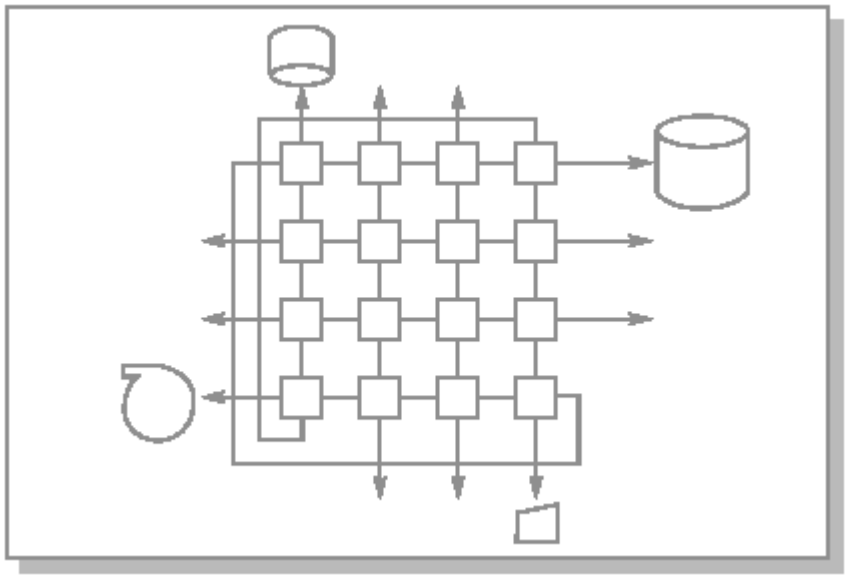

Рис. 10.3. Структурный модуль МВС-100

Для МВС-100 базовый вычислительный блок содержит 32 узла. Такой блок строится из двух структурных модулей в соответствии со схемой, приведенной на рис. 10.4. В этом случае максимальная длина пути между любой парой вычислительных узлов равна пяти. При этом остается 16 свободных связей, что позволяет продолжить объединение. При объединении двух базовых блоков по схеме, приведенной на рис. 10.4 (64 вычислительных узла) максимальная длина пути составит 6, как и в гиперкубе, а количество свободных связей будет равно 16.

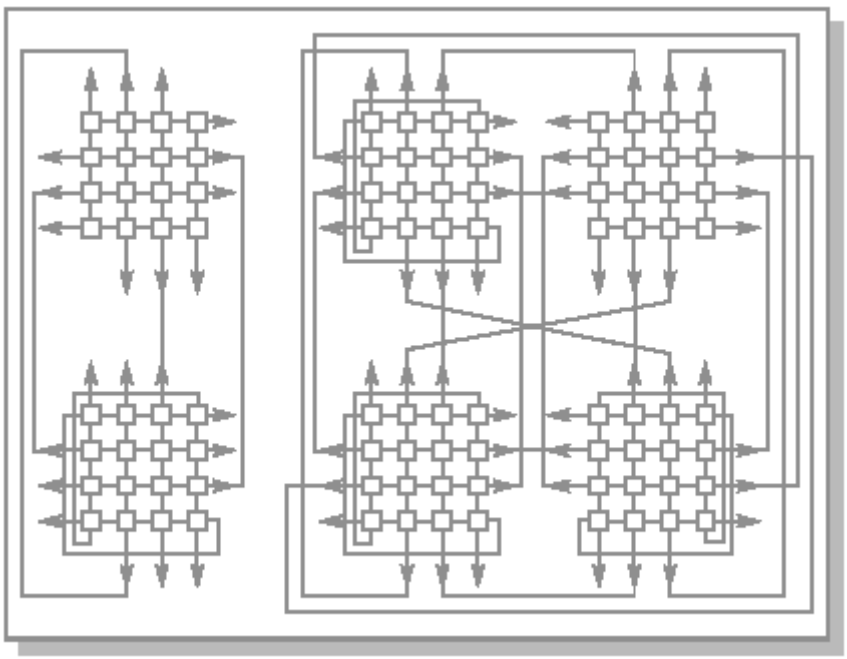

Рис. 10.4. Варианты объединения структурных модулей МВС-100

## Граф межмодульных связей МВС-1000

Архитектура системы МВС-1000 аналогична архитектуре МВС-100. Основой системы является масштабируемый массив процессорных узлов. Каждый узел содержит

вычислительный микропроцессор Alpha 21164 с производительностью 2 GFLOPS при тактовой частоте 500 MHz и оперативную память объемом 128 MB, с возможностью расширения. Процессорные узлы взаимодействуют через коммуникационные процессоры TMS320C44 производства Texas Instruments, имеющие по 4 внешних канала (линка) с общей пропускной способностью 80 Мбайт/с (20 Мбайт/с каждый). Также разрабатывается вариант системы с использованием коммуникационных процессоров SHARC (ADSP 21060) компании Analog Devices, имеющих по 6 каналов с общей пропускной способностью до 240 Мбайт/с (40 Мбайт/с каждый).

Процессорные узлы связаны между собой по оригинальной схеме, сходной с топологией двухмерного тора (для 4-линковых узлов). Аналогично МВС-100, структурный модуль МВС-1000 состоит из 16 вычислительных модулей, образующих матрицу 4 х 4, в которой четыре угловых элемента соединяются через транспьютерные линки по диагонали попарно. Оставшиеся 12 линков предназначаются для подсоединения внешних устройств (4 линка угловых ВМ) и соединений с подобными ВМ.

Конструктивным образованием МВС-1000 является базовый вычислительный блок, содержащий 32 вычислительных модуля. Максимальная длина пути между любыми из 32 вычислительных модулей равна пяти, при этом число свободных линков после комплектации блока составляет 16, что позволяет продолжить процедуру объединения. Возможная схема объединения четырех базовых блоков в 128-процессорную систему приведена на рис. 10.5.

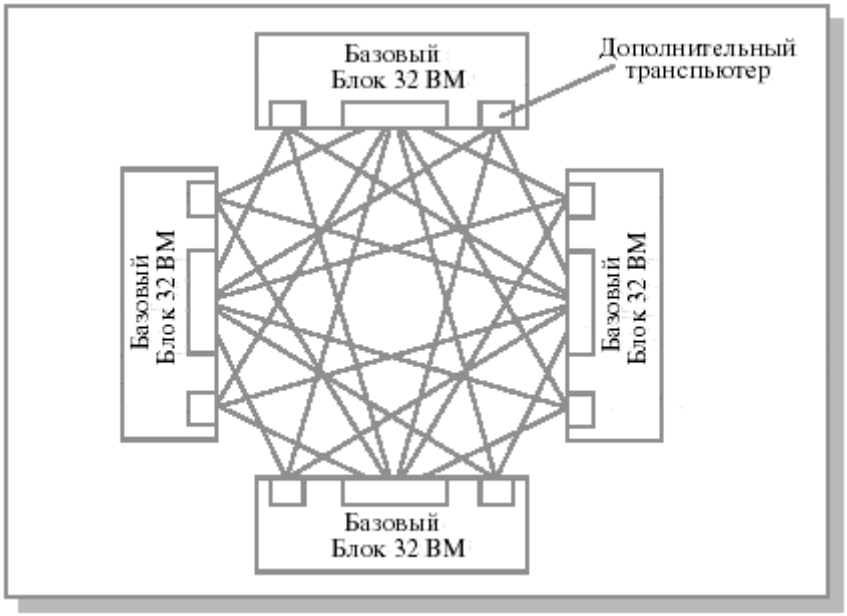


Рис. 10.5. Структура 128-процессорной системы МВС-1000, 4 базовых блока

## 11. Требования к компонентам МВС

Заголовок лекции нужно понимать в более широком смысле, чем просто набор требований к техническим характеристикам компонентов вычислительной системы: процессору, дисковым массивам, памяти, коммутаторам и тому подобным аппаратным средствам. В какой-то степени надежная работа компонентов систем подразумевается априори: компоненты должны работать настолько долго, насколько это необходимо и поддерживать при этом заданные значения параметров системы (ясно, что такое положение является идеализацией). Достигается такая надежность путем улучшения технологий создания компонентов, сборки систем и их эксплуатации и т.п. приемами. Большое значение имеют научно-технические исследования в области создания принципиально новых подходов в разработке и создании как известных компонентов, так и принципиально новых приборов. Но не меньшее, если не большее значение имеют требования, предъявляемые к вычислительной системе, которую планируется построить для реализации конкретных целей, как единому целому: для решения задач определенного круга (научных, экономических и т.п.), как базовой основы для обработки больших потоков данных (информационные системы), оптимальной реализации модели программирования и т.д. Отсюда, в результате проведенного анализа, вытекает выбор архитектуры МВС.

Разработчикам систем необходимо, прежде всего, проанализировать следующие связанные между собой вопросы:

- отношение "стоимость/производительность";
- надежность и отказоустойчивость системы;
- масштабируемость системы;
- совместимость программного обеспечения.

Требования к надежности и отказоустойчивости системы рассматриваются в другой лекции.

### Отношение "стоимость/производительность"

Добиться дополнительного повышения производительности в МВС сложнее, чем произвести масштабирование внутри узла. Основным барьером является трудность организации эффективных межузловых связей. Коммуникации, которые существуют между узлами, должны быть устойчивы к задержкам программно поддерживаемой когерентности. Приложения с большим количеством взаимодействующих процессов работают лучше на основе SMP-узлов, в которых коммуникационные связи более быстрые. В кластерах, как и в MPP-системах, масштабирование приложений более эффективно при уменьшении объема коммуникаций между процессами, работающими в разных узлах. Это обычно достигается путем разбиения данных.

Именно такой подход используется в наиболее известном приложении на основе кластеров OPS (Oracle Parallel Server).

Появление любого нового направления в вычислительной технике определяется требованиями компьютерного рынка. Поэтому у разработчиков компьютеров нет единственной цели. Большая универсальная вычислительная машина (мейнфрейм) или суперкомпьютер стоят дорого. Для достижения поставленных целей при проектировании

высокопроизводительных конструкций приходится игнорировать стоимостные характеристики.

Суперкомпьютеры фирм Cray Inc., NEC и высокопроизводительные мэйнфреймы компании IBM, суперкластеры фирмы SGI относятся именно к этой категории компьютеров. Другим противоположным примером может служить сравнительно недорогая конструкция, где производительность принесена в жертву для достижения низкой стоимости. К этому направлению относятся персональные компьютеры IBM PC. Между этими двумя крайними направлениями находятся конструкции, основанные на отношении "стоимость/производительность", в которых разработчики находят баланс между стоимостью и производительностью. Типичными примерами такого рода компьютеров являются миникомпьютеры и рабочие станции.

Для сравнения различных компьютеров между собой обычно используются стандартные методики измерения производительности. Эти методики позволяют разработчикам и пользователям задействовать полученные в результате испытаний количественные показатели для оценки тех или иных технических решений, и, в конце концов, именно производительность и стоимость дают пользователю рациональную основу для решения вопроса, какой компьютер выбрать.

Например, в качестве критерия измерения производительности используется тест LINPACK. Данный тест был выбран из-за его доступности почти для всех рассматриваемых систем. Тест LINPACK был введен Джеком Донгаррой (Jack Dongarra) в 1976 г. Данный тест основан на решении плотной системы линейных уравнений. Как один из вариантов LINPACK используется версия теста, которая позволяет пользователю менять размерность задачи и оптимизировать программное обеспечение для достижения наилучшей производительности для данной машины. Такая производительность не отражает общую производительность этой системы. Однако она отражает ее производительность при решении плотной системы линейных уравнений.

Для оценки производительности вычислительных систем используются также тесты SPECfp_rate_base2000: SPEC, SPECfp и SPECrate, которые являются зарегистрированными торговыми марками Standard Performance Evaluation Corporation. Для оценки скорости работы памяти системы используется тест STREAM Triad.

## Масштабируемость

Масштабируемость представляет собой возможность наращивания числа и мощности процессоров, объемов оперативной и внешней памяти и других ресурсов вычислительной системы. Масштабируемость должна обеспечиваться архитектурой и конструкцией компьютера, а также соответствующими средствами программного обеспечения.

Так, например, возможность масштабирования кластера ограничена значением отношения скорости процессора к скорости связи, которое не должно быть слишком большим (реально это отношение для больших систем не может быть более 3-4, в противном случае не удается даже реализовать режим единого образа операционной системы). С другой стороны, последние 10 лет истории развития процессоров и коммуникаторов показывают, что разрыв в скорости между ними все увеличивается. Добавление каждого нового процессора в действительно масштабируемой системе должно давать прогнозируемое увеличение производительности и пропускной способности при приемлемых затратах. Одной из основных задач при построении масштабируемых систем является минимизация стоимости расширения компьютера и упрощение планирования. В идеале добавление

процессоров к системе должно приводить к линейному росту ее производительности. Однако это не всегда так. Потери производительности могут возникать, например, при недостаточной пропускной способности шин из-за возрастания трафика между процессорами и основной памятью, а также между памятью и устройствами ввода/вывода. В действительности реальное увеличение производительности трудно оценить заранее, поскольку оно в значительной степени зависит от динамики поведения прикладных задач.

Возможность масштабирования системы определяется не только архитектурой аппаратных средств, но зависит от свойств программного обеспечения. Масштабируемость программного обеспечения затрагивает все его уровни, от простых механизмов передачи сообщений до работы с такими сложными объектами как мониторы транзакций и вся среда прикладной системы. В частности, программное обеспечение должно минимизировать трафик межпроцессорного обмена, который может препятствовать линейному росту производительности системы. Аппаратные средства (процессоры, шины и устройства ввода/вывода) являются только частью масштабируемой архитектуры, на которой программное обеспечение может обеспечить предсказуемый рост производительности. Важно понимать, что, например, простой переход на более мощный процессор может привести к перегрузке других компонентов системы. Это означает, что действительно масштабируемая система должна быть сбалансирована по всем параметрам.

## Совместимость и мобильность программного обеспечения

Концепция программной совместимости впервые в широких масштабах была применена разработчиками системы IBM/360. Основная задача при проектировании всего ряда моделей этой системы заключалась в создании такой архитектуры, которая была бы одинаковой с точки зрения пользователя для всех моделей системы, независимо от цены и производительности каждой из них. Большие преимущества такого подхода, позволяющего сохранять существующий задел программного обеспечения при переходе на новые (как правило, более производительные) модели, были быстро оценены как производителями компьютеров, так и пользователям, и начиная с этого времени практически все фирмы-поставщики компьютерного оборудования взяли на вооружение эти принципы, поставляя серии совместимых компьютеров. Следует заметить, однако, что со временем даже самая передовая архитектура неизбежно устаревает и возникает потребность внесения радикальных изменений и в архитектуру, и в способы организации вычислительных систем.

В настоящее время одним из наиболее важных факторов, определяющих современные тенденции в развитии информационных технологий, является ориентация компаний-поставщиков компьютерного оборудования на рынок прикладных программных средств. Это объясняется, прежде всего, тем, что для конечного пользователя, в конце концов, важно программное обеспечение, позволяющее решить его задачи, а не выбор той или иной аппаратной платформы. Переход от однородных сетей программно совместимых компьютеров к построению неоднородных сетей, включающих компьютеры разных производителей, в корне изменил и точку зрения на саму сеть: из сравнительно простого средства обмена информацией она превратилась в средство интеграции отдельных ресурсов - мощную распределенную вычислительную систему, каждый элемент которой (сервер или рабочая станция) лучше всего соответствует требованиям конкретной прикладной задачи.

Этот переход выдвинул ряд новых требований. Прежде всего, такая вычислительная среда должна позволять гибко менять количество и состав аппаратных средств и программного

обеспечения в соответствии с меняющимися требованиями решаемых задач. Во-вторых, она должна обеспечивать возможность запуска одних и тех же программных систем на различных аппаратных платформах, т.е. обеспечивать мобильность программного обеспечения. В третьих, эта среда должна гарантировать возможность применения одних и тех же человеко-машинных интерфейсов на всех компьютерах, входящих в неоднородную сеть. В условиях жесткой конкуренции производителей аппаратных платформ и программного обеспечения сформировалась концепция открытых систем, представляющая собой совокупность стандартов на различные компоненты вычислительной среды, предназначенных для обеспечения мобильности программных средств в рамках неоднородной, распределенной вычислительной системы.

Одним из вариантов моделей открытой среды является модель OSE (Open System Environment), предложенная комитетом IEEE POSIX. На основе этой модели национальный институт стандартов и технологии США выпустил документ "Application Portability Profile (APP). The U.S. Government's Open System Environment Profile OSE/1 Version 2.0", который определяет рекомендуемые для федеральных учреждений США спецификации в области информационных технологий, обеспечивающие мобильность системного и прикладного программного обеспечения. Все ведущие производители компьютеров и программного обеспечения в США в настоящее время придерживаются требований этого документа.

## 12. Надежность и отказоустойчивость МВС

Одной из основных проблем построения вычислительных систем остается задача обеспечения их продолжительного функционирования.

Важнейшей характеристикой вычислительных систем является надежность, т.е. работа системы без сбоев в определенных условиях в течение определенного времени. Повышение надежности основано на принципе предотвращения неисправностей путем снижения интенсивности отказов и сбоев за счет применения электронных схем и компонентов с высокой и сверхвысокой степенью интеграции, снижения уровня помех, облегченных режимов работы схем, обеспечения тепловых режимов их работы, а также за счет совершенствования методов сборки аппаратуры.

Понятие надежности включает не только аппаратные средства, но и программное обеспечение, которое используется, в частности, для анализа производительности систем и управления конфигурациями. Главной целью повышения надежности систем является целостность хранящихся в них данных. Единицей измерения надежности является среднее время наработки на отказ (MTBF - Mean Time Between Failure), иначе - среднее время безотказной работы.

Отказоустойчивость - это способность вычислительной системы продолжать действия, заданные программой, после возникновения неисправностей. Введение отказоустойчивости требует избыточного аппаратного и программного обеспечения. Направления, связанные с предотвращением неисправностей и с отказоустойчивостью, - основные для обеспечения надежности. Концепции параллельности и отказоустойчивости вычислительных систем естественным образом связаны между собой, поскольку в обоих случаях требуются дополнительные функциональные компоненты. Поэтому на параллельных вычислительных системах достигается как наиболее высокая производительность, так и, во многих случаях, очень высокая надежность. Имеющиеся ресурсы избыточности в параллельных системах могут гибко использоваться как для повышения производительности, так и для повышения надежности. Структура многопроцессорных и многомашинных систем приспособлена к автоматической реконфигурации и обеспечивает возможность продолжения работы системы после возникновения неисправностей.

В настоящее время эти два понятия - надежности и отказоустойчивости - при описании компьютерных систем часто смешивают. Во многом это объяснятся тем, что пользователя (не обязательно индивидуального) интересует главное: вычислительная система должна работать необходимое время и предоставлять определенный набор услуг. Для достижения безотказной работы используются разные приемы, часть из которых мы здесь рассматриваем, не акцентируя внимания на том, к какому из вышеуказанных понятий эти приемы относятся.

Для повышения надежности информационно-вычислительной системы идеальной схемой являются кластерные системы. Благодаря единому представлению, отдельные неисправные узлы или компоненты кластера могут быть без остановки работы и незаметно для пользователя заменены, что обеспечивает непрерывность и безотказную работу вычислительной системы даже в таких сложных приложениях как базы данных.

Основа надежности кластера - это некоторое избыточное количество отказоустойчивых серверов (узлов), в зависимости от конфигурации кластера и его задач.

Кластерная конфигурация узлов, коммуникационного оборудования и памяти может обеспечить зеркалирование данных, резервирование компонентов самоконтроля и предупреждения, а также совместное использование ресурсов для минимизации потерь при отказе отдельных компонентов.

Решение, обеспечивающее повышенную отказоустойчивость сервера, должно включать:

- компоненты с "горячей" заменой;
- диски, вентиляторы, внешние накопители, устройства PCI, источники питания;
- избыточные источники питания и вентиляторы;
- автоматический перезапуск и восстановление системы;
- память с коррекцией ошибок;
- функции проверки состояния системы;
- превентивное обнаружение и анализ неисправностей;
- средства удаленного администрирования системы.

Во многих случаях кластер, как типичный представитель МВС, представляется пользователю и администратору как единая система. Наблюдение за системой включает сбор, хранение и извлечение таких показателей как использование центрального процессора и памяти, температура системы и процессора, скорость вращения вентиляторов; эти и другие параметры помогают пользователям и администраторам понимать общее состояние системы и эффективность ее использования.

Единое управление системами кластера позволяет максимально увеличить период безотказной работы, контроль и управление приложениями, операционными системами и аппаратными средствами. При этом все узлы кластера управляются из единого центра контроля.

Программы-утилиты обеспечивают улучшение защиты и возможности восстановления данных, а также сглаживают последствия сбоев в работе оборудования для конечного пользователя. Операционная система кластера служит для управления всеми функциями кластера.

Программное обеспечение дает возможность организовать эффективную службу сопровождения и мониторинга решения, позволяя собирать данные на уровне узла, используя плату управления. Важным направлением является совершенствование и развитие библиотеки MPI и развитие системы отладки параллельных программ, работающих на МВС. К ней относятся отладчики, профилировщики, обеспечивающие контроль над прохождением задач.

В операционной системе HP-UX11i, созданной компанией Hewlett-Packard и предназначенной для обслуживания критически важных задач в Internet, для повышения надежности предусмотрена возможность подключения дополнительных процессоров без перезагрузки ОС. Применение файловой системы Veritas дает возможность резервного копирования в режиме online и дефрагментации дисков без выключения системы. Операционная система может отключать неработоспособные процессоры и блоки памяти без выключения системы.

Системы хранения должны быть представлены RAID-системами высокой готовности. Избыточные соединения должны обеспечивать доступность данных даже в случае выхода из строя узлов, контроллеров или кабелей. Соединение с системами хранения данных в

кластере может быть реализовано как с использованием интерфейсов SCSI, так и на основе Fibre Channel технологии.

Для синхронизации и совместной работы серверов в качестве кластера необходимы избыточные соединения между серверами, называемые "системным соединением" (private interconnect). Системное соединение используется для передачи сигналов о состоянии серверов, а также применяется параллельными базами данных для передачи данных.

Катастрофоустойчивые решения создаются на основе разнесения узлов кластерной системы на сотни километров и обеспечения механизмов глобальной синхронизации данных между такими узлами.

В качестве примера повышения надежности кластерной системы приведем решения фирмы Hewlett-Packard. В этих решениях, в зависимости от нужного уровня отказоустойчивости, серверные узлы кластера размещаются следующим образом:

- централизованно (локальный кластер);
- по соседним зданиям (кампусный кластер);
- по нескольким территориям в пределах города (метро кластер);
- в разных городах, странах или континентах (два связанных кластера - континентальный кластер).

В дополнение к дублированному центральному коммутатору, все аппаратные компоненты: системный контроллер, источники питания, системы охлаждения, часы - полностью дублированы. Система не имеет единичной точки сбоя. Для сравнения - если такой простой элемент как системные часы выйдет из строя в дорогостоящем сервере HP Superdome или IBM p680, вся система прекратит работу.

В систему должны быть заранее установлены или сконфигурированы запасные модули, так что при отказе одного из модулей запасной модуль может заменить его практически немедленно. Отказавший модуль может ремонтироваться автономно, в то время как система продолжает работать.

Принцип быстрого проявления неисправности обычно реализуется с помощью двух методов - самоконтроля и сравнения. Средства самоконтроля предполагают, что при выполнении некоторой операции модуль делает и некоторую дополнительную работу, позволяющую подтвердить правильность полученного состояния. Примерами этого метода являются коды обнаружения неисправности при хранении данных и передаче сообщений. Метод сравнения основывается на выполнении одной и той же операции двумя или большим числом модулей и сопоставлении результатов компаратором. В случае обнаружения несовпадения результатов работа приостанавливается.

Методы самоконтроля были основой построения отказоустойчивых систем в течение многих лет. Они требуют реализации дополнительных схем и времени разработки и, вероятно, будут доминировать в устройствах памяти и устройствах связи благодаря простоте и ясности логики. Однако для сложных устройств обработки данных экономические соображения, связанные с применением стандартных массовых компонентов, навязывают использование методов сравнения. Поскольку компараторы сравнительно просты, их применение дает некоторое увеличение логических схем при существенном сокращении времени разработки. Следует отметить, что в более ранних отказоустойчивых конструкциях 30% логических схем процессоров и 30% времени разработки уходило на реализацию средств самоконтроля. С этой точки зрения схемы

сравнения добавляют лишь универсальные схемы с простой логикой. В результате сокращаются общие расходы на разработку и логику.

Еще одним средством построения отказоустойчивой архитектуры является принцип дублирования дуплексных модулей, который предполагает создание некоторой комбинации двух модулей ("супермодуля"), построенных на принципах быстрого проявления неисправности. Такой "супермодуль" продолжает работать, даже когда отказывает один из субмодулей.

Дублирование дуплексных модулей требует большего объема оборудования, но позволяет делать выбор одного из режимов работы: организацию либо двух независимых вычислений на принципах быстрого проявления неисправности, выполняющихся на двух парах модулей, либо одного высоконадежного вычисления, выполняющегося на всех четырех модулях.

Необходимо помнить, что сама по себе избыточность только снижает надежность в случае дублирования и троирования. Для существенного увеличения уровня готовности избыточная конструкция должна обеспечивать возможность ремонта и замены отказавших модулей.

**13. Кластеры и массивно-параллельные системы различных производителей. Примеры кластерных решений IBM. Примеры кластерных решений HP. Примеры кластерных решений SGI**

Развитие сетевых технологий привело к появлению недорогих, но эффективных коммуникационных решений. Это и предопределило появление кластерных вычислительных систем, фактически являющихся одним из направлений развития компьютеров с массовым параллелизмом. Классические суперкомпьютеры, использующие специализированные процессоры таких производителей как Cray или NEC (векторно-параллельные или массивно-параллельные), недешевы, поэтому и стоимость подобных систем несравнима со стоимостью систем, находящихся в массовом производстве.

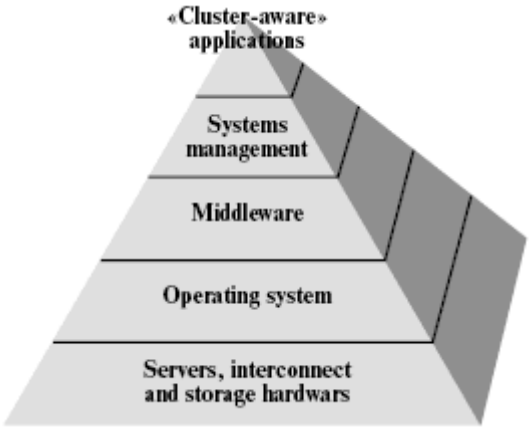


Рис. 13.1. Пирамида уровней кластерной системы

Вычислительные системы (ВС), создаваемые из массово выпускаемых компонентов, стали альтернативой традиционным суперкомпьютерным системам. При выполнении многих прикладных задач такие ВС, даже с небольшим или средним (до 128–256) числом вычислительных модулей, показывают производительность, не уступающую или даже превосходящую производительность традиционных суперкомпьютеров как с распределенной, так и с разделяемой памятью. При этом такие ВС обладают рядом преимуществ, среди которых: более низкая стоимость, короткий цикл разработки и возможность оперативно использовать наиболее эффективные вычислительные и коммуникационные компоненты из имеющихся на рынке во время создания системы. Поэтому неудивительно, что ведущие разработчики высокопроизводительной техники приступили к созданию кластерных систем.

## Примеры кластерных решений IBM

В начале 2000 г. специалисты IBM создали Linux-кластер из установленных в стойке серверов IBM xSeries, объединив их с соответствующими сетями, системами управления (аппаратное и программное обеспечение) и необходимыми услугами. После выпуска в 2001 г. кластера 1300 компания IBM представила кластер 1350 на процессорах Intel Xeon.

Схема Linux-кластера или суперкластера нетривиальна. В ней имеется несколько логических слоев, и уровень сложности возрастает при увеличении размера системы. Стоит отметить, что на больших системах простое воспроизведение в большом количестве малых кластеров почти никогда не приводит к успеху.

Хотя число узлов, необходимых для решения задачи, довольно легко оценивается для любого приложения, требуемое число узлов в действительности оказывается больше из-за необходимости иметь сервисные узлы, обслуживающие инфраструктуру кластера. Так, для каждых 32-64 узлов, в зависимости от компоновки, необходим центральный узел. Если такой узел используется как вычислительный, у него должна быть соответствующая конфигурация.

В любой системе должен быть управляющий узел, который, в частности, может быть и одним из главных узлов. Для организации ввода/вывода также необходимы отдельные узлы, которые работают либо с устройствами хранения информации, либо с сетевыми рутерами.

Основой Linux-кластера являются плотно упакованные системы с процессорами Intel, установленные в стойке. Наиболее часто используемым модулем является стандартная 19" стойка. Внутри стоек устанавливаются узлы, аппаратура для эффективного соединения компонентов, такая как коммутаторы или хабы, аппаратура управления внутренней сетью системы, терминальные серверы и т.п.

Узлы могут быть функционально объединены в две группы:

1. Вычислительные узлы, выполняющие основные вычислительные задачи, для которых спроектирована система.
2. Узлы инфраструктуры, такие как узлы ввода-вывода, узлы управления и узлы запоминающих устройств. Узлы инфраструктуры обеспечивают управление системами и заданными функциями, необходимыми для объединения компьютерных узлов в систему.

Упаковка вычислительных узлов, насколько это возможно, должна быть плотной и иметь достаточные возможности для эффективного соединения компонентов. Существенно включение сервисного процессора для функций управления системами. Стандартным вычислительным узлом для кластера 1350 является IBM xSeries 335. Это позволяет один или два процессора Intel Pentium 4 (Xeon) с быстрой динамической памятью и диском размещать в стандартном корпусе размером "1U". Символ 1U обозначает 1,75 дюймов высоты в стандартном 19-дюймовом корпусе. X335 имеет встроенный сервисный процессор и два слота для соединения с другими компонентами системы.

Головные узлы, узлы управления и узлы запоминающих устройств обеспечивают особые функции для управления кластером (обеспечение загрузки, управление устройствами, внешний ввод/вывод и т.д). Сервер 2U IBM xSeries 345, основанный на процессорах Xeon, в кластере 1350 используется как узел управления и хранения данных и может также применяться как вычислительный узел. Коммутаторы используются для межпроцессорного соединения в параллельном программировании и для различных функций управления.

Для параллельного программирования в качестве межпроцессорного соединения обычно используется коммутатор Myrinet фирмы Myricom. Пропускная способность канала составляет приблизительно 200 Мбайт/с в каждом направлении со временем задержки 6-8 мкс.

Если параллельное программное окружение требует меньше межпроцессорных соединений, то высокоскоростные соединения можно заменить на более дешевые

продукты типа Ethernet. Для заказчика могут быть выбраны GigaNet, Quadratics, SCI или ServerNet. В дальнейшем, после доработки, можно выбрать InfiniBand.

Коммутаторы используются для построения внутренних сетей для систем управления и интерфейса внешних сетей. В качестве альтернативных решений заказчику предлагаются различные коммутаторы от Cisco.

Терминальные серверы обеспечивают удаленный доступ к консолям ОС узлов через последовательную сеть. Дополнительные функциональные возможности добавляются посредством клавиатуры, мыши, монитора.

Внешние устройства ввода/вывода, такие как SCSI Raid, должны быть в стойках с узлами, коммутаторами и т.д.

Основное отличие состоит в сборке (интеграции). В то время как кластер для заказчиков может быть собран в любом учреждении или даже на полу у заказчика (неудачная идея), то аппаратная часть кластера 1350 производится (т.е. интегрируется) и тестируется на заводе IBM. Процесс сборки кластера в заводских условиях обладает рядом преимуществ.

Пример конфигурации кластера 1350 приведен в таблице 13.1.

Программное обеспечение (ПО) для кластера 1350 существенно зависит от требований заказчика. Дополнительные технические условия могут потребовать особых программных пакетов. Коммерческий программный пакет, например, может включать в себя WebSphere, DB2, MySQL и т.д. HPC пакет может включать MPICH, PVM, Maui Scheduler, математические библиотеки, трансляторы, профилировщики и т.д.

Оперативная система Linux инсталлирована на каждом узле кластера. Кластер 1350 запускается под Red Hat Linux. В дальнейшем планируется ставить ОС SuSE (4Q02).

Управление системами кластера для Linux (CSM) — это лицензионный программный продукт IBM. Он обеспечивает функции управления системами, сходными по форме с программами поддержки параллельных систем (Parallel System Support Programs — PSSP) для AIX-систем уровня поддержки на RS/6000 SP. CSM — это стандартный программный продукт для кластера 1350.

CSM для Linux включает технологию, извлеченную из PSSP, и сейчас доступную на AIX для управления кластерами, собранными из серверов xSeries и запускаемыми под Linux, серверами IBM pSeries, управляемыми AIX, или комбинацией обеих операционных систем.

Таблица 13.1. Конфигурация кластера 1350.

| **Класс** | **Число узлов кластера** | **Скорость процессора, ГГц** | **Память системы, Гбайт** | **Внутренняя память, Гбайт** | **Соединение кластера, Мбит/с** |
|---|---|---|---|---|---|
| **Начальный** | 8 | 2,0 | 0,512 | 18 | 10/100 Ethernet |
| **Средний** | 32 | 2,4 | 1 | 18 | 10/100 Ethernet |
| **Профессиональный** | 128 | 2,8 | 1 | 36 | Gigabit |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | Ethernet |
| **Высоко-производительный** | 64 | 2,8 | 1 | 36 | Myrinet-2000 |

## Примеры кластерных решений HP

Слияние в 2002 г. компаний Hewlett Packard и Compaq обеспечило HP прочное положение лидера по продаже Linux-систем, соответствующих индустриальным стандартам на базе архитектур IA-32 и IA-64. Данная технология дополнена мощной поддержкой разработок ядра Linux на базе семейства Itanium, а также разработок с открытым кодом в целом.

Кластеры HP строятся путем объединения компьютеров в группы, которые называются "кластерами предприятия". Каждый узел кластера имеет по крайней мере один процессор, оперативную память и образ операционной системы. Для связи между узлами используются специальные протоколы связи и системные процессы.

Поддержка ОС Linux со стороны HP охватывает все семейство серверов HP, основанных на архитектуре Intel (IA-32 и IA-64), включая все серверы промышленного стандарта HP ProLiant, сверхплотную (blade) архитектуру, рабочие станции HP, настольные компьютеры Evo, отдельные портативные компьютеры, серверы ProLiant для применения в качестве межсетевых экранов и даже портативные устройства iPAQ. HP также продолжает поддерживать технологию ОС Linux для архитектуры AlphaServer, разработанную компанией Compaq. ОС Linux работает на Alpha системах, начиная с 1994 г. Это был первый пример 64-разрядной системы с поддержкой Linux. Именно с него начались современные разработки ОС Linux на базе семейства Itanium. HP поддерживает на своих серверах дистрибутивы Red Hat и SuSE, планируя осуществлять поддержку дистрибутивов операционной системы UnitedLinux после ее выпуска. HP предлагает заказчикам возможность предварительно установить любую ОС Linux на выбранные серверы ProLiant и рабочие станции Evo.

HP поддержала лабораторию Sandia в ее планах по развертыванию кластерной системы Cplant на базе ОС Linux с самой высокой на сегодня производительностью. Сейчас HP совместно с Pacific Northwest National Laboratories работает над созданием вычислительной системы, основанной на 1400 процессорах Itanium 2 с оптоволоконными межсоединениями на основе решений Quadrics.

Программное обеспечение HP может поддерживать большинство современных средств разработки и настройки производительности для кластерных решений на базе системы Linux. При выборе этих средств действуют ограничения, связанные с типами процессоров и межузловых соединений. В число программных продуктов входят:

- Транслятор Intel C++ Compiler для Linux;
- Транслятор Intel Fortran Compiler для Linux;
- Библиотека Intel Math Kernel Library;
- Intel Vtune Performance Analyzer — средство оптимизации программного кода.

## Примеры кластерных решений SGI

Седьмого января 2003 г. компания SGI представила новое семейство 64-разрядных Linux-серверов и суперкластеров, названных SGI Altix 3000 (серверы SGI Altix 3300 и

суперкластеры SGI Altix 3700). Система SGI Altix 3000 использует процессоры Intel Itanium 2 и основана на архитектуре глобальной разделяемой памяти SGI NUMAflex, которая является реализацией архитектуры неоднородного доступа к памяти (NUMA). NUMAflex появилась в 1996 г. и с тех пор использовалась в известной серии серверов и суперкомпьютеров SGI Origin, основанных на процессорах MIPS и 64-разрядной операционной системе IRIX. Дизайн NUMAflex позволяет помещать процессор, память, систему ввода/вывода, соединительные провода, графическую подсистему в модульные компоненты, иначе называемые блоками или кирпичиками. Эти кирпичики могут комбинироваться и конфигурироваться с большой гибкостью, чтобы удовлетворять потребности клиента в ресурсах и рабочей нагрузке. Используя этот дизайн третьего поколения, компания SGI смогла создать систему SGI Altix 3000 на основе традиционных блоков ввода/вывода (IX- и PX-блоки), хранения данных (D-блоки) и соединительных компонентов (маршрутизирующие блоки/R-блоки). Основным отличием этой новой системы является процессорный блок (C-блок), который содержит процессоры Itanium 2.

Ключевой особенностью системы является использование каскадируемых коммутаторов в маршрутизирующих элементах. Каскадируемые коммутаторы обеспечивают системе малое время задержки, или замедление доступа к памяти, несмотря на модульную конструкцию. Это критично для машин, использующих архитектуру неоднородного доступа к памяти (NUMA). Задержки всегда были проблемой в архитектуре NUMA, так как память распределяется между узлами, а не сосредоточена в одном месте. Каскадируемые коммутаторы используют каталогизируемую схему памяти для отслеживания данных, находящихся в разных кэшах. В результате меньшие объемы данных пересылаются между частями памяти, что выливается в снижение задержек по сравнению с традиционными системами, основанными на шинах.

Системное ПО для SGI Altix 3000 состоит из стандартного дистрибутива Linux для процессоров Itanium и SGI ProPack – продукта, добавляющего особые свойства Linux. SGI ProPack включает ядро 2.4, HPC-библиотеки, настроенные для использования особенностей архитектуры SGI, утилиты для работы с NUMA и драйверы. Также SGI ProPack включает дополнительные инструменты и библиотеки для улучшения работы больших NUMA-систем, особенно при одновременном выполнении нескольких ресурсоемких приложений. Это позволяет эффективно использовать системные ресурсы и доставлять результаты в разумное время: характеристики, критичные для сред высокопроизводительных вычислений.

Утилиты работы с NUMA, библиотеки HPC и дополнительное ПО, установленные на стандартный дистрибутив Linux, создают программное окружение для высокопроизводительных вычислений, эффективное при больших вычислительных нагрузках и нагрузках, связанных с передачей данных. SGI ProPack создает промежуточный слой ПО, которое позволяет на основе Linux создавать блоки для построения больших сред высокопроизводительных вычислений.

Высокопроизводительные программы требуют баланса между процессором и подсистемой памяти для поддержания постоянного уровня потока данных. Кластеры SGI Altix 3000 были протестированы с помощью тестов STREAM Triad, которые измеряют скорость работы памяти. 64-процессорная система достигла уровня производительности памяти в 125 Гбайт/с на едином образе операционной системы: превосходство в 460% над 64-процессорной системой HP Superdome, которая показала производительность 27 Гбайт/с. По сравнению с 32-процессорным сервером IBM eServer p690, система SGI Altix показывает удвоенную производительность, а при вдвое меньшей стоимости —

улучшение показателя цена/производительность на 640 %. Результаты также показывают, что Linux может хорошо масштабироваться за рамками ограничения в 8 процессоров.

Семейство SGI Altix 3000 демонстрирует пропускную способность системы ввода/вывода более чем 2 Гбайт/с при использовании единого образа Linux — лучший результат для Linux-систем. При постоянном увеличении объемов обрабатываемых данных возможности перемещения информации с диска в память играют все более важную роль в общей производительности системы. Это достижение позволяет приложениям Linux решить проблемы оперирования большими объемами данных в средах высокопроизводительных вычислений.

**14. Кластеры и массивно-параллельные системы различных производителей. SMP Power Challenge фирмы Silicon Graphics. Семейство SUN Ultra Enterprise компании SUN**

## SMP Power Challenge фирмы Silicon Graphics

Компания Silicon Graphics (SGI) была создана в 1981 г. Основным направлением ее работы в течение многих лет было создание высокопроизводительных графических рабочих станций. В настоящее время интересы SGI распространяются на рынок высокопроизводительных вычислений как для технических, так и для коммерческих приложений. В частности, она концентрирует усилия на разработке и внедрении современных технологий визуализации вычислений, трехмерной графики, обработки звука и мультимедиа.

Рис. 14.1. SMP Power Challenge

В свое время разработка процессора R10000 позволила компании перейти к объединению своих серверов Challenge (на базе процессора R4000) и PowerChallenge (на базе процессора R8000) в единую линию продуктов. Благодаря повышенной производительности этого процессора на целочисленных операциях и плавающей точке, обе линии продуктов могут быть объединены без потери производительности.

Серверы Silicon Graphics работают под управлением операционной системы IRIX (ОС UNIX реального времени), построенной в соответствии с требованиями стандартов SVID (System V Interface Definition) и XPG4. Она поддерживает возможность работы нескольких машин на одном шлейфе SCSI (multi-hosted SCSI), четырехкратное зеркалирование и 128-кратное расщепление дисковых накопителей. На платформе поддерживаются многие продукты компаний Oracle, Informix и Sybase.

Компьютеры CHALLENGE DM/L/XL ориентированы в первую очередь на коммерческие применения, а POWER CHALLENGE L/XL — на работу с плавающей запятой. Системы CHALLENGE DM относятся к среднему классу.

POWER CHALLENGE относится к классу симметричных мультипроцессорных ЭВМ (SMP-системы), базирующихся на поколении суперскалярных процессоров MIPS R8000 фирмы Silicon Graphics.

Отличительными особенностями этих систем являются:

- масштабируемость суперкомпьютинга;
- использование большой динамической памяти (адресация у POWER CHALLENGE до 16 Гбайт — в два раза выше, чем у Cray T90/C90/J90);
- 64-разрядная архитектура;
- двоичная совместимость со всем семейством компьютеров SGI, включая рабочие станции Indy.

Таблица 14.1. Производительность компьютеров POWER CHALLENGE

| **Компьютеры** | **Число проц.** | **Пиковая произв-ть, Гфлоп** | **Произв-ть I/O, Гбайт/с** | **RAM, Гбайт** | **Диски, Тбайт** |
|---|---|---|---|---|---|
| **POWER CHALLENGE XL** | 1-18 | 0.36-6,5 | до 1,2 | 0,064-16 | до 6,3 |
| **POWER CHALLENGE array** | до 144 | до 51,8 | до 4 | до 128 | до 63 |

По пиковой производительности POWER CHALLENGE уступает векторным компьютерам Cray (старым, естественно, моделям). Однако процессор R8000 на 90 МГц оказался быстрее, чем процессоры в Cray Y-MP (333 МФлоп) и компьютере Cray J90 (200 МФлоп). Данные тестов LINPACK показывают, что при равном числе процессоров SMP-компьютеры от SGI опережают Cray J90 и уступают Cray C90 как на средних (N=100), так и на длинных (N=1000) векторах всего в 3 раза. Уровень распараллеливания (отношение производительности n процессоров к производительности одного процессора) в серверах SGI немного выше, чем в DEC AlphaServer 8400, и, в свою очередь, несколько ниже, чем у Cray. Известно, что с ростом числа процессоров эффективность распараллеливания сильно уменьшается. Задачи, которые хорошо распараллеливаются при большом числе процессоров, часто могут эффективно выполняться и в рамках модели распределенной памяти, в том числе в кластерных системах. Такие приложения бывают, например, в динамике жидкости, обработке сейсмоданных и т.д. Silicon Graphics предлагает пользователям кластер POWER CHALLENGEarray, который может содержать до 8 SMP-серверов POWER CHALLENGE L/XL. Он имеет до 144 процессоров с пиковой производительностью 52 GFLOPS и до 128 Гбайт RAM, и, таким образом, превосходит по ряду показателей машины Cray. Серверы связываются в кластер через FDDI или HiPPI.

POWER CHALLENGE XL превосходит старшие модели Cray по размеру оперативной памяти, уступая по производительности ввода-вывода и максимальному размеру дискового пространства. Больший возможный размер памяти в системах SGI связан с использованием дешевой DRAM-технологии по сравнению с дорогой высокопроизводительной SRAM-памятью в Cray C90/T90.

## Семейство SUN Ultra Enterprise компании SUN

Sun Ultra Enterprise — это серия масштабируемых, удобных в управлении и надежных серверов. Ее можно разделить на следующие группы:

- серверы рабочих групп: Sun Enterprise 10s, 250, 220R, 450 и 420R ;
- серверы отдела предприятия: Sun Enterprise 3500 и 4500;
- серверы масштаба предприятия: Sun Enterprise 5500, 6500 и 10000.

Sun Enterprise 10s — сервер Sun начального уровня, взаимодействующий со всеми клиентами сети: персональными компьютерами (ПК), системами Macintosh, рабочими станциями UNIX, сетевыми компьютерами. Благодаря процессору UltraSPARC-IIi,

работающему с тактовой частотой 440 МГц, поддерживающему до 1 Гб памяти, 4 PCI слота расширения и до 18,2 Гб на внешнем носителе, эта система предоставляет недорогую платформу для работы критических сетевых приложений. Сервер Sun Enterprise 10s предназначен для Web, электронной почты, файловых служб и служб печати. Кроме того, Solaris для Intranet — расширение операционной системы Sun Solaris — обеспечивает предоставление Web-услуг.

Рис. 14.2. Sun Enterprise 10s

Sun Microsystems представляет двухпроцессорный SMP-сервер масштаба рабочей группы. В линейке продуктов Sun сервер Sun Enterprise 250 позиционируется между системами Sun Enterprise Ultra 5s и 10s и сервером Sun Enterprise 450 и замещает собой сервер Sun Enterprise 150. Сервер Sun Enterprise 250 поставляется в виде напольного решения или стойки.

Рис. 14.3. Сервер Sun Enterprise 250

Сервер Sun Enterprise 250 удобен для небольших, возможно, удаленных офисов, рассчитанных на 50-100 пользователей. Сервер Sun Enterprise 250 поддерживает каждодневную работу офиса: выполнение приложений баз данных, финансовых приложений; поддержка обмена сообщениями, предоставление сервиса приложений, а также поддержка стандартных приложений рабочих групп (документооборот, распределение и планирование товаров и услуг и т.д.). Он также подходит для рынка провайдеров Internet-услуг и офисных приложений.

Сервер Sun Enterprise 250 поддерживает от 128 Мбайт до 2Гбайт оперативной памяти, при этом скорость передачи данных достигает 1,6 Гбайт/с. Он также вмещает в себя до шести 1-дюймовых или 1,6-дюймовых дисков UltraSCSI со скоростью внешнего интерфейса 40

Мбайт/с, допускающих работу в режиме "горячей замены". Используются диски на 4,2 Гбайт, 9 Гбайт или 18 Гбайт, таким образом, максимальный объем хранимых данных равен, соответственно, 25 Гбайт, 54 Гбайт или 108 Гбайт.

Сервер Sun Enterprise 220R — двухпроцессорный сервер масштаба рабочей группы, предназначенный для проведения сетевых вычислений и основанный на технологии UltraSPARC. Этот сервер следующего поколения предоставляет заказчику возможности мультипроцессорной системы, диски UltraSCSI, шину ввода/вывода PCI, являющуюся стандартом в отрасли, в компактном корпусе, предназначенном для монтажа в промышленные стойки. Содержит до двух процессоров UltraSPARC-II с тактовой частотой 450 МГц и 2 Гбайт памяти.

Рис. 14.4. Сервер Sun Enterprise 220R

Сервер Sun Enterprise 450 предназначен для организации электронной почты, работы с базами данных, создания кластерных комплексов, предоставления Web-сервиса, обеспечения управления ресурсами предприятия.

Рис. 14.5. Сервер Sun Enterprise 450

В составе сервера работают до четырех процессоров UltraSPARC с тактовой частотой 300 МГц или 400 МГц, внутренняя шина UPA со скоростью 1,6Гбайт/с и подсистема ввода-вывода на базе шины PCI, обеспечивающая пропускную способность 600 Мбайт/с. Оперативная память расширяется до 4Гбайт, встроенные диски Ultra SCSI емкостью до 84 ГБ с режимом быстрой "горячей замены" и возможности подключения внешних накопителей общей емкостью до 6 Тбайт.

Сервер может взаимодействовать с любыми рабочими станциями, Intel и Macintosh. Надежность заложена в сервер Sun Enterprise 450 изначально: такие возможности как коррекция ошибок ECC на внутренней шине и в памяти, автоматическое восстановление системы после сбоя и резервируемые с режимом "горячей замены" блоки питания и дисковые накопители присутствуют здесь в стандартной конфигурации.

Сервер Sun Enterprise 420R — четырехпроцессорный сервер масштаба рабочей группы, предназначенный для проведения сетевых вычислений и основанный на технологии UltraSPARC. Сервер предоставляет возможности мультипроцессорной системы, диски UltraSCSI, шину ввода/вывода PCI, являющуюся стандартом в отрасли, в компактном корпусе, предназначенном для монтажа в промышленные стойки.

Рис. 14.6. Сервер Sun Enterprise 420R

Сервер Sun Enterprise 420R предназначен для Internet-провайдеров и провайдеров услуг, финансовых учреждений, для организации высокопроизводительных вычислений и для любых отраслей, где требуется мощный сервер обработки данных, занимающий немного места. Чрезвычайно важно, что сервер Sun Enterprise 420R обеспечивает высокий уровень производительности при полной двоичной совместимости вверх и вниз по всей линейке серверов и рабочих станций.

Сервер Sun Enterprise 3500 дает возможность исполнять сложные деловые приложения и предоставлять Internet/Intranet услуги с той же производительностью и доступностью, что и весьма дорогостоящие крупномасштабные системы. Время простоев значительно снижено благодаря устойчивой архитектуре сервера, средствам системного управления и новым программным возможностям, таким как динамическая реконфигурация и выбор альтернативного маршрута (Dynamic Reconfiguration and Alternate Pathing).

Рис. 14.7. Сервер Sun Enterprise 3500

Сочетание масштабируемой операционной системы Sun Solaris, которая устанавливается на всех серверах Sun, и модульных аппаратных компонентов позволяет легко наращивать

производительность и расширять возможности системы. В качестве сервера приложений Enterprise 3500 демонстрирует высокую производительность, объединяя возможности восьми UltraSPARC процессоров, связанных высокопроизводительной системой Gigaplane. Общая архитектура семейства серверов Sun Enterprise 3500-6500 позволяет производить локальное наращивание вычислительных мощностей путем подключения до 30 процессоров. Программное обеспечение Sun Enterprise SyMON облегчает управление системой, благодаря простому в использовании интерфейсу и системе предупреждения сбоев аппаратного обеспечения.

Сервер Sun Enterprise 4500 — это компактный сервер среднего уровня с вычислительными возможностями, практически в два раза превосходящий возможности наращиваемости сервера Sun Enterprise 3500. Сервер обеспечивает доступность для критически важных приложений и удобные средства управления системой для деловых приложений баз данных и электронной коммерции. Общая модульная архитектура семейства серверов от Sun Enterprise 3500 до Sun Enterprise 6500 облегчает модификацию. Благодаря использованию в серверах Sun операционной системы Solaris можно работать с более чем 12000 приложений.

Рис. 14.8. Сервер Sun Enterprise 4500

В сервере Sun Enterprise 4500 возможна установка до 14 процессоров. До 4 серверов Sun Enterprise 4500 можно установить в стойку центра данных. Программное обеспечение Sun Enterprise management Center облегчает управление системой, благодаря простому в использовании интерфейсу и системе предупреждения сбоев аппаратного обеспечения. Общая архитектура семейства серверов от Sun Enterprise 3500–6500 позволяет производить локальное наращивание вычислительных возможностей путем подключения до 30 процессоров.

Сервер Sun Enterprise 5500 предназначен для больших систем центров данных. Число процессоров меняется от 1 до 14. Основная память расширяется от 256 Мбайт до 14 Гбайт. Возможные конфигурации наполнения одного банка от 256 Мбайт до 1 Гбайт (группы из 8 модулей SIMM). Имеется внешний массив хранения данных. Поддерживается более 6 Тбайт данных.

Рис. 14.9. Сервер Sun Enterprise 5500

Сервер Sun Enterprise 6500 создан для выполнения критически важных промышленных приложений, таких как хранилища данных и ERP-системы. Модульные компоненты этого высокомасштабируемого сервера позволяют увеличивать производительность и возможности ввода/вывода. Такие возможности как динамическая реконфигурация и выбор альтернативного маршрута (Dynamic Reconfiguration and Alternate Pathing), введенные в операционную систему Solaris, позволяют добавлять, удалять, модифицировать и обслуживать системные компоненты, не прекращая работы сервера. Использование в сервере Sun Enterprise 6500 операционной системы Solaris гарантирует ему масштабирование до максимальных возможностей. Количество процессоров меняется от 1 до 30. Основная память масштабируется от 256 Мбайт до 30 Гбайт. Возможные конфигурации наполнения одного банка от 256 Мбайт до 1 Гбайт (группы из 8 модулей SIMM). Поддерживается более 10 Тбайт данных.

Рис. 14.10. Сервер Sun Enterprise 6500

Сервер Enterprise 10000 нацелен на работу с важными приложениями: информационными хранилищами, системами принятия решений, консолидированных ЛВС (LAN) или высокообъемных приложений с онлайновой обработкой транзакций (OLTP).

Рис. 14.11. Сервер Enterprise 10000

Enterprise 10000 — это единственная UNIX-система, обеспечивающая работу с независимыми разделами, как на мэйнфрейме, что весьма важно для эффективного использования вычислительного центра. Система обеспечивает масштабирование в том, что касается производительности, числа пользователей, емкости приложений, расширяясь до 64 процессоров UltraSPARC с тактовой частотой 400 МГц, и включает дисковую подсистему в том же едином корпусе. Свойства обеспечения постоянной работоспособности SunTrust делают систему Enterprise 10000 наиболее надежной в своем классе.

Система Enterprise 10000, вмещающая до 64 Гбайт разделяемой памяти, с шириной пропускания внутрисистемной магистрали до 12 Гбайт/с для быстрой пересылки данных и фиксированного времени задержки, а также с поддержкой RAID 0, RAID 1 и RAID 5, превосходит по производительности все другие масштабируемые системы. Кроме того, обеспечивается поддержка до 20 Тбайт дискового пространства, что существенно для проектов крупнейших вычислительных центров. Возможности "горячей замены" позволяют легко производить обновление и замену компонентов в существующих системах без необходимости их перезагрузки или выключения питания.

**15. Кластеры и массивно-параллельные системы различных производителей. Семейство массивно-параллельных машин ВС МВС-100 и МВС-1000. ВС с распределенной памятью производства Sequent и DATA GENERAL. Кластеры DIGITAL**

## Семейство массивно-параллельных машин ВС МВС-100 и МВС-1000

Массивно-параллельные масштабируемые системы МВС предназначены для решения прикладных задач, требующих большого объема вычислений и обработки данных. Суперкомпьютерная установка системы МВС представляет собой мультипроцессорный массив, объединенный с внешней дисковой памятью и устройствами ввода-вывода информации под общим управлением персонального компьютера или рабочей станции.

Основные области фактического применения суперкомпьютеров МВС-100/1000:

1. решение задач расчета аэродинамики летательных аппаратов, в том числе явления интерференции при групповом движении;
2. расчет трехмерных нестационарных течений вязкого сжимаемого газа;
3. расчеты течений с локальными тепловыми неоднородностями в потоке;
4. разработка квантовой статистики моделей поведения вещества при экстремальных условиях;
5. расчеты структурообразования биологических макромолекул;
6. моделирование динамики молекулярных и биомолекулярных систем;
7. решение задач линейных дифференциальных игр. Динамические задачи конфликтов управления;
8. решение задач механики деформируемых твердых тел, в том числе с учетом процессов разрушения.

Программное обеспечение, установленное на вычислительных системах, по сути, минимально по своему объему: трансляторы с языков FORTRAN и C (C++); дополнительные средства описания параллельных процессов; программные средства PVM и MPI; средства реализации многопользовательских режимов и удаленного доступа.

Межведомственный суперкомпьютерный центр (МСЦ) был открыт 5 ноября 1999 г. В нем в качестве основной машины была установлена 16-процессорная система фирмы HP V2250 производительностью 15 Гопер/с. Параллельно с этой системой в МСЦ работал также 96-процессорный вариант отечественной системы МВС-1000. Суммарная производительность всех систем центра достигала 230 Гопер/с (2,3x1011).

## МВС-100

Подход, который используется при создании отечественных суперкомпьютеров, состоит в закупке новейших комплектующих изделий, создания на этой основе суперкомпьютерных систем, их интеграции в информационно-вычислительные сети и необходимых усилий в области применения, т.е. в разработке прикладных программ и методов математического моделирования. Такая концепция реализована в мультипроцессорной вычислительной системе МВС-100.

МВС-100 - это отечественная мультипроцессорная система второго поколения. В настоящее время она заменяется на МВС-1000. Система поставляется в виде типовых конструктивных модулей по 32, 64 и 128 процессоров. Число процессоров в модулях и

число модулей может варьироваться. Для основной обработки применяются микропроцессоры Intel 860 с производительностью до 100 МФлоп (64 разряда при двойной точности) и присоединенной оперативной памятью, изменяемой от 8 до 32 Мбайт. Для межпроцессорного обмена в каждом узле используется транспьютер, работающий с той же оперативной памятью, а также памятью внешнего обмена объемом 2-8 Мбайт. Общая пропускная способность 4 транспьютерных каналов для внешнего обмена - 20 Мбайт/с. Система МВС-100 эксплуатируется в ряде институтов РАН и промышленности. Установки МВС-100 с суммарной производительностью более 50 Гопер/с эксплуатируются в вычислительных центрах РАН (в Москве, Екатеринбурге, Новосибирске, Владивостоке) и в отраслевых ВЦ. Показана возможность эффективного распараллеливания вычислений и обработки данных.

## МВС-1000

МВС-1000 - система третьего поколения, основанная на использовании микропроцессоров Alpha 21164 (разработка компании DEC-Compaq; выпускается также заводами фирм Intel и Samsung) с производительностью до 1-2 Гопер/с и присоединенной оперативной памятью объемом 0,1-2 Гбайт. Система МВС-1000 с производительностью до 1 Тфлоп состоит из 8 стоек (512 узлов).

В основном исполнении системы межпроцессорный обмен структурно аналогичен используемому в системе МВС-100 и реализуется в двух модификациях: на базе <транспьютероподобного> связного микропроцессора TMS320C44 (фирма Texas Instruments), имеющего 4 канала с пропускной способностью каждого в 20 Мбайт/с, либо на базе связного микропроцессора SHARC ADSP 21060 (фирма Analog Devices), имеющего 6 внешних каналов с пропускной способностью каждого 40 Мбайт/с.

Исполнение МВС-1000К отличается использованием для межпроцессорного обмена коммутационной сети MYRINET (фирма Myricom, США) с пропускной способностью канала в дуплексном режиме 2х160 Мбайт/с. Кроме того, предусмотрено подключение к каждому процессору памяти на жестком диске с объемом 2-9 Гбайт. В стандартной стойке располагается до 64 процессоров системы МВС-1000 или 24 процессора системы МВС-1000К. Предусмотрены средства системного объединения стоек для установок с большим числом процессоров.

Процессорные узлы связаны между собой по схеме, сходной с топологией двухмерного тора (для 4-линковых узлов). Конструктивным образованием МВС-1000 является базовый вычислительный блок, содержащий 32 вычислительных модуля.

Для управления массивом процессоров и внешними устройствами, а также для доступа к системе извне используется так называемый хост-компьютер (управляющая машина). Обычно это рабочая станция AlphaStation с процессором Alpha и операционной системой Digital Unix (Tru64 Unix) или ПК на базе Intel с операционной системой Linux.

Начиная с 1999 г., все вновь выпускаемые МВС-1000 строятся как кластеры выделенных рабочих станций. Это означает, что, в отличие от ранних версий МВС-1000, в качестве вычислительного модуля используются не специализированные ЭВМ, предназначенные только для применения в качестве деталей суперкомпьютерной установки, а обычные персональные компьютеры. В качестве коммуникационной аппаратуры используются обычные сетевые платы и коммутаторы, применяемые для построения офисных локальных сетей.

В качестве базовой ОС узла используется Linux, что фактически является общепринятым мировым стандартом для построения систем такого класса. Это позволило многократно расширить и упростить, по сравнению с ранними версиями МВС-1000, адаптацию разнообразного программного обеспечения, как свободно распространяемого, так и коммерческого.

Разработчики МВС-1000 предлагают пользователям набор единых архитектурных решений, охватывающий три класса параллельных вычислительных систем:

- большие и сверхбольшие системы, состоящие из сотен узлов. Примером такой системы может служить МВС-1000М, эксплуатирующаяся в настоящее время в Межведомственном суперкомпьютерном центре РФ;
- средние и малые системы, поставляемые по заказу. В настоящее время серийно выпускаются МВС-1000/16 и МВС-1000/32, из 16 и 32 процессоров, соответственно. В ИПМ РАН в настоящее время в регулярной эксплуатации находится одна установка МВС-1000/16. Кроме того, аналогичные системы успешно эксплуатируются в целом ряде научных центров страны;
- виртуальные параллельные системы на базе офисной локальной сети. Для освоения и изучения параллельных технологий, а также для постоянной эксплуатации в условиях острого дефицита финансовых средств предлагается программное обеспечение, позволяющее организовать полноценную параллельную систему на базе оборудования вузовского компьютерного класса, или аналогичной офисной локальной сети. При этом с точки зрения пользователя система не только не отличается от <настоящей> по способу работы, но и позволяет получать очевидный выигрыш в скорости при выполнении реальных программ. В отличие от реализованных аппаратно суперкомпьютеров МВС-1000, такой полностью виртуальный суперкомпьютер называется МВС-900.

## ВС с распределенной памятью компании Sequent

## Sequent

В 1999 г. компьютерные корпорации IBM и Sequent объявили о своем слиянии, и в июле того же года Sequent фактически стала подразделением IBM. После этого IBM остановила значительное обновление линии серверов NUMA Q, выпускавшихся Sequent.

Sequent являлся поставщиком масштабируемых NUMA-серверов (серии NUMA-Q 1000 и NUMA-Q 2000), включающих до 64 процессоров Intel и предназначенных в основном для коммерческих систем онлайновой обработки транзакций и поддержки СУБД.

Рис. 15.1. Сервер Sequent

Кроме того, NUMA-технологии, разработанные специалистами Sequent, используются и в серверах от IBM. Разработчики IBM утверждают, что NUMA станет определяющей технологией для UNIX- и NT-серверов уже в начале 21 века.

Sequent была, по-видимому, второй после IBM компанией, осуществившей поставки UNIX-кластеров баз данных в середине 1993 г. Она предлагала решения, соответствующие среднему и высокому уровню готовности своих систем. Первоначально Sequent Hi-Av Systems обеспечивали дублирование систем, которые разделяли общие диски. Пользователи могли выбирать ручной или автоматический режим переключения на резерв в случае отказа. Hi-Av Systems обеспечивает также горячее резервирование IP-адресов и позволяет кластеру, в состав которого входит до четырех узлов, иметь единственный сетевой адрес.

Компания Sequent одной из первых освоила технологию Fast-Wide SCSI, что позволило ей добиться значительного увеличения производительности систем при обработке транзакций. Компания поддерживает дисковые подсистемы RAID уровней 1, 3 и 5. Кроме того, она предлагает в качестве разделяемого ресурса ленточные накопители SCSI. Модель SE90 поддерживает кластеры, в состав которых могут входить два, три или четыре узла, представляющих собой многопроцессорные системы Symmetry 2000 или Symmetry 5000 в любой комбинации. Это достаточно мощные системы. Например, Sequent Symmetry 5000 Series 790 может иметь от 2 до 30 процессоров Pentium 66 МГц, оперативную память емкостью до 2 Гбайт и дисковую память емкостью до 840 Гбайт.

При работе с Oracle Parallel Server все узлы кластера работают с единственной копией базы данных, расположенной на общих разделяемых дисках.

## Система Sequent NUMA-Q

Среди разработок архитектура cc-NUMA выделяется принципиально. Это архитектура симметричного мультипроцессирования (SMP), обладающая множеством достоинств: простая модель программирования, отличная масштабируемость, возможность работы с количеством процессоров более восьми, переносимость приложений и т. д. Но есть и недостатки в виде высокой стоимости и специального программного обеспечения. У истоков создания архитектуры cc-NUMA стояла компания Sequent, реализовавшая собственную версию NUMA-Q.

| Таблица 15.1. NUMA-Q 2000 | |
|---|---|
| **Производитель** | IBM (ранее Sequent) |
| **Класс архитектуры** | Многопроцессорная система с общей памятью (cc-NUMA). Используется для организации сложных информационных систем. |
| **Модификации** | Model E410/E330/E320/E300/E200 |
| **Процессоры** | Intel Pentium III Xeon (700 MHz в модели E410) |
| **Узел** | От 4 до 64 процессоров, до 64 GB оперативной памяти; узел состоит из базовых плат по 4 процессора (quads), соединенных между собой коммутатором IQ-Link |
| **Масштабируемость** | Возможна организация кластеров, включающих до 4 узлов |
| **Системное ПО** | Используется операционная система DYNIX/ptx: версия UNIX от Sequent.Внутри одной системы могут одновременно исполняться UNIX и Windows NT |

Первой NUMA-системой была машина Butterfly, разработанная BBN в 1981 г. Первой действующей системой с архитектурой cc-NUMA стала Stanford DASH. Группа, работавшая с ней, использовала возможность изучить функционирование операционной системы SGI IRIX на 32-процессорном узле cc-NUMA в 1992 г.

## ВС с распределенной памятью фирмы DATA GENERAL

Компания Data General Corp. была куплена в 1999 г. EMC Corporation и стала ее подразделением.

До этого Data General поставляла многопроцессорные SMP-серверы серий AV 5500, AV 8500 и AV 9500. Эти серверы поддерживают работу с отказоустойчивыми дисковыми и ленточными подсистемами CLARiiON, средства автоматической диагностики AV/Alert, инициируемые оператором или автоматические средства переключения на резервную систему, управление внешней памятью в режиме on-line, управление вводом/выводом и быстрое восстановление файлов.

В случае отказа процессора, памяти или компонента ввода/вывода система автоматически начинает процесс выключения и затем осуществляет собственную перезагрузку с исключением отказавших компонентов. Стандартным средством указанных систем является наличие избыточных источников питания.

Максимальная степень готовности достигается при подключении двух серверов к высоконадежному дисковому массиву CLARiiON. Дисковые массивы CLARiiON Series C2000 Disk Array обеспечивают RAID уровней 0, 1, 3 и 5 в любых сочетаниях, до 20 накопителей в одном шасси общей емкостью до 80 Гбайт и возможность замены накопителя без выключения питания. В конструкции дискового массива используются избыточные интеллектуальные контроллеры с дублированными связями, обеспечивающие отказоустойчивость. Ленточный массив CLARiiON Series 4000 поддерживает отказоустойчивое резервное копирование и восстановление. В составе массива используется специальный процессор, реализующий схему расщепления данных, подобную RAID уровня 5. Ленточный массив обеспечивает не только высокую пропускную способность, но и реализует защиту от отказов носителя и накопителя. В действительности, даже при отказе накопителя или картриджа, операции резервного копирования или восстановления данных продолжаются без потери данных. В массив можно устанавливать 3, 5 или 7 накопителей. При двухкратной компрессии данных общая емкость ленточного массива может достигать 48 Гбайт.

## Кластеры DIGITAL

Как и Sequent и Data General, фирма Digital (DEC) прекратила свое самостоятельное существование, сначала став отделением компании Compaq, а затем вместе с ней войдя в состав Hewlett-Packard. Компания DEC известна тем, что она являлась разработчиком серверов AlphaServer, на основе которых строились Alpha-кластеры. Перейдя в другую компанию, коллектив Digital под маркой HP продолжает работу по обновлению этой линии серверов.

Архитектура систем на основе процессора Alpha была разработана в 1988-1991 гг. с перспективой развития на 20-25 лет. Выпускаемое в настоящее время четвертое поколение процессоров содержит четырехканальное суперскалярное ядро, 80 регистров для целочисленных операций, 72 регистра с плавающей запятой, причем в процессе обработки одновременно может находиться до 80 инструкций. Для архитектуры процессора Alpha характерно внеочередное исполнение команд, логика предсказания ветвлений, полностью интегрированная на кристалле кэш-память первого уровня, многоканальные устройства доступа к оперативной памяти. В семейство Alpha-сервер входит четыре серии серверов: AlphaServer DS, AlphaServer ES, AlphaServer GS и AlphaServer SC. Если в моделях AlphaServer DS используется не более 64 процессоров, то в серверах AlphaServer SC их может быть несколько сотен (до 512 и более). Максимальная тактовая частота процессоров в настоящее время составляет 1,25 ГГц. Все компьютеры семейства построены по коммутируемой технологии, что позволяет избежать недостатков, присущих системной шине.

С каждой системой поставляется пакет HP Insight Manager - инструментальное средство, обеспечивающее мониторинг и управление серверами платформы Alpha. Системы AlphaServer GS разработаны специально для поддержки самых мощных бизнес-приложений. Модульная архитектура систем обеспечивает высокую масштабируемость.

В настоящее время на системах AlphaServer начального уровня поддерживаются операционные системы Tru64 Unix, OpenVMS и Linux, а в серверах среднего уровня и высокопроизводительных серверах - Tru64 Unix и OpenVMS.

Одним из преимуществ платформы Alpha стали развитые решения в области построения кластеров. Операционная система OpenVMS Cluster позволяет объединять до 95 узлов с практически неограниченным территориальным распределением, с полностью

кооперативным совместным использованием ресурсов - систем хранения данных, файлов и даже отдельных записей. При этом каждая система имеет отдельное оборудование, может запускаться и останавливаться независимо. Приложения и программные продукты допускают динамическое обновление, новые устройства хранения данных также можно добавлять и заменять в <горячем> режиме, а обновление ОС OpenVMS развертывается во всем кластере поэтапно, останавливать его работу для этого не требуется. OpenVMS Galaxy позволяет выполнять в одной системе несколько экземпляров OpenVMS. Каждый экземпляр OpenVMS Galaxy можно запускать и останавливать отдельно. Вычислительная среда OpenVMS Galaxy предлагает широкие возможности масштабирования в SMP-конфигурациях. Серверы AlphaServer нового поколения имеют высокую степень масштабируемости, позволяя строить системы с числом процессоров от 2 до 128 с использованием всего двух типов <блоков>: 2-процессорных и 8-процессорных.

Таблица 15.2. Серии DS, ES, GS80, GS160, GS320

| | **Серия DS** | **Серия ES** | **GS80** | **GS160** | **GS320** |
|---|---|---|---|---|---|
| Количество моделей | 6 | 3 | 1 | 1 | 1 |
| Поддерживаемые процессоры | EV67 600 МГц, EV68 833 МГц, EV68 1.25 ГГц | EV68 833 МГц, EV68 1 ГГц, EV68 1.25 ГГц | EV68 1.224 ГГц | EV68 1.25 ГГц | EV68 1.25 ГГц |
| Количество процессоров | 1-2 | 1-4 | 1-8 | 1-16 | 1-32 |
| Кэш-память 2-го уровня на один процессор, Мбайт | 2-4-8 | 8-16 | 16 | 16 | 16 |
| Макс. объем оперативной памяти, Гбайт | 4 | 32 | 64 | 128 | 256 |
| Макс. объем внутренних дисков, Гбайт | 436 | 2900 | 252 | 504 | 504 |
| Макс. кол-во PCI слотов в/в | 5 | 10 | 56 | 112 | 224 |
| Пропускная способность шины в/в, Гбайт/с | 0,532 | 1,85 | 3,2 | 6,4 | 12,8 |
| Макс. число аппаратных разделов | 1 | 1 | 2 | 4 | 8 |
| Макс. число программных разделов (Open VMS) | - | - | 8 | 16 | 32 |

**16. Кластеры и массивно-параллельные системы различных производителей. Современные суперкомпьютеры: Hitachi SR8000, Серия Fujitsu VPP5000, Cray T3E-1200, ASCI White**

## Серия Hitachi SR8000

Серия SR8000, или Супертехнический сервер, была разработана для численного моделирования сложных научно-технических задач (структурный анализ, динамика жидкости, предсказание погоды и т.п.). Серия объединяет возможности как векторного суперкомпьютера S-3000, так и параллельного компьютера SR2201.

Рис. 16.1. SR8000

Высокопроизводительный 64-разрядный RISC-микропроцессор разработан и создан Hitachi с использованием CMOS-технологии 0,14 микрометровой длины логических элементов. Для максимальной эффективности микропроцессоров на крупномасштабных задачах используются возможности псевдовекторной обработки. Это позволяет данным выбираться из оперативной памяти конвейерным способом без задержки сменяемых процессов. В результате данные подаются из памяти в арифметические устройства также эффективно, как в суперкомпьютере векторного типа.

Выпускаются модели SR8000 и SR8000 E1/F1/G1.

| Таблица 16.1. Конфигурация узла | | | | |
|---|---|---|---|---|
| **Модель** | **SR8000** | **SR8000 E1** | **SR8000 F1** | **SR8000 G1** |
| Пиковая прозв-ть, Гфлоп | 8 | 9,6 | 12 | 14,4 |
| Память | 2/4/8 | 2/4/8/16 | 2/4/8/16 | 2/4/8/16 |

Для 144-узловой конфигурации модели G1 (450 МГц) при решении полной системы линейных уравнений размерностью 141000 была достигнута скорость в 1709 Гфлоп/ (теоретически возможная - 2074 Гфлоп/с), что дало эффективность 63%. На 112-узловой модели F1 (375 МГц) достигнута скорость в 1035 Гфлоп/с из 1344 Гфлоп/с

(эффективность - 77%). На отдельном узле при решении полной линейной системы и симметричной задачи на собственные значения (порядок 5000) процессорные скорости были выше 6,2 и 4,1 Гфлоп/с, соответственно.

| Таблица 16.2. Конфигурация системы | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Число узлов** | | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 |
| **Произв-ть, Гфлоп** | **SR8000** | 32 | 64 | 128 | 256 | 512 | 1024 | - | - |
| | **SR8000 E1** | 38,4 | 76,8 | 153,6 | 307,2 | 614,4 | 1228,8 | 2457,6 | 4915,2 |
| | **SR8000 F1** | 48 | 96 | 192 | 384 | 768 | 1536 | 3072 | 6144 |
| | **SR8000 G1** | 57,6 | 115,2 | 230,4 | 460,8 | 921,6 | 1843,2 | 3686,4 | 7372,8 |
| **Максимальный объем общей памяти, Гбайт** | **SR8000 E1/F1/G1** | 64 | 128 | 256 | 512 | 1024 | 2048 | 4096 | 8192 |
| **Внешний интерфейс** | | Ultra SCSI, Ethernet/Fast Ethernet, Gigabit Ethernet, ATM, HIPPI, Fibre Channel | | | | | | | |

## Серия Fujitsu VPP5000

Серия VPP5000 является преемником прежних систем VPP700/VPP700E (последняя система имеет тактовый цикл 6,6 нс вместо 7 нс). Глобальные изменения в архитектуре относительно серий VPP700 малы. Тактовый цикл был уменьшен наполовину. Архитектура узлов VPP5000 почти идентична узлам VPP700. Каждый узел в системе, называемый процессорным элементом (ПЭ), является мощным векторным процессором (9,6 Гфлоп/с пиковой скорости и тактовый цикл 3,3 нс). Векторный процессор дополнен RISC-скалярным процессором с пиковой скоростью 1,2 Гфлоп/с. Формат скалярных команд имеет 64 разряда и может выполнять до 4 операций параллельно. Каждый ПЭ имеет память до 16 Гбайт и каждый ПЭ непосредственно соединяется с другими ПЭ со скоростью передачи 1,6 Гбайт/с.

Рис. 16.2. Система VPP5000

VPP5000U - это однопроцессорная машина без сети и расширений передачи данных, которые требуются для VPP5000.

Скалярное устройство поддерживает RISC-архитектуру <очень длинного командного слова> (VLIW - Very Long Instruction Word), одновременно выполняя до 4 команд за один тактовый цикл. Высокая скалярная производительность достигается посредством как первого и второго кэшей, так и асинхронного выполнения обращения к памяти, команд с плавающей запятой и векторных команд.

Векторное устройство состоит из 4 конвейеров, векторного регистра и регистра маски (mask register) со скоростью векторных операций до 9,6 Гфлоп/ПЭ. Конвейер квадратного корня увеличивает производительность в операциях, включая квадратные корни. Векторные операции выполняются со скоростью 2,4 Гфлоп.

Все ПЭ соединяются через высокоскоростную сеть с поперечной коммутацией. Особое устройство связи между ПЭ, называемое DTU (Data transfer unit), делает возможным одновременное выполнение соединений между процессорами и вычисления. Это позволяет выполнять передачу и прием данных со скоростью 615 Мбайт/с в каждом направлении, в то время как ПЭ выполняют вычисления.

Система VPP5000 имеет дополнительные возможности для операций с плавающей запятой расширенной точности и непрямого доступа к памяти, возникающего в различных вычислительных алгоритмах.

Компоненты ПЭ являются высокопроизводительными энергосберегающими CMOS (complementary metal oxide semiconductor) LSI-микросхемами, произведенными по 0,22 мк технологии и содержащими до 33 миллионов транзисторов каждая, со временем вентильной задержки (gate delay time) в 24 пикосекунды. Для оперативной памяти используется 128-разрядная SDRAM (synchronous dynamic RAM) со временем произвольного доступа в 45 наносекунд.

Проведенные тесты показали, что для системы из 32-х процессоров при решении полной линейной системы порядка 170 880 скорость составила 296,1 Гфлоп/с (эффективность - 96%). Для отдельного процессора скорость в 6,04 Гфлоп/с была достигнута при решении системы порядка 2 000. При вычислении многочлена 10-го порядка была определена скорость в 8,68 Гфлоп/с (эффективность - более 90%).

Основные технические характеристики:

- год выпуска - ноябрь 1999;
- 9,6 Гфлоп векторной производительности на ПЭ;
- 1,2 Гфлоп скалярной производительности;
- масштабируется от 1 до 128 ПЭ (512 ПЭ для особого размещения) и достигает пиковой производительности 1,228 Тфлоп;
- 4, 8 или 16 Гбайт оперативной памяти SDRAM на ПЭ (максимум 2 Тбайта на систему);
- 76,8 Гбайт/с пропускная способность памяти (memory transfer bandwidth) на ПЭ;
- 64-разрядная архитектура;
- операционная система UXP/V Unix System V Release 4.

Спецификации системы VPP5000U:

- число процессоров - 1;
- теоретическая пиковая производительность - 9,6 Гфлоп.;
- оперативная память - 4-16 Гбайт;

Спецификации системы VPP5000:

- число процессоров - от 4 до 128 (512 ПЭ для особого размещения);
- теоретическая пиковая производительность - от 38,4 Гфлоп до 1,229 Тфлоп (4,915 Тфлоп для 512 ПЭ);
- оперативная память - от 16 Гбайт до 2 048 Тбайт (8 192 Тбайт для 512 ПЭ);
- пропускная способность шины - 1,6 Гбайт/с/ПЭ.

## Современные суперкомпьютеры - Cray T3E-1200

Системы Cray T3E - это масштабируемые параллельные системы, которые используют DECchip 21164 (DEC Alpha EV5) RISC-процессоры с пиковой производительностью 600 Мфлоп и 21164A для машин Cray T3E-900 и Cray T3E-1200. Каждый процессорный элемент (ПЭ) Cray T3E имеет свою собственную DRAM-память объемом от 64 Мбайт до 2 Гбайт. В отличие от системы CRAY T3D, в которой исполняемая задача запрашивает фиксированное количество процессоров на все время выполнения, в CRAY T3E свободные процессоры могут использоваться другими задачами. Модели T3E, T3E-900, T3E-1200, T3E-1350.

Рис. 16.3. Gray T3E

Каждый узел в системе содержит один процессорный элемент (ПЭ), включающий процессор, память и средство коммутации, которое осуществляет связь между ПЭ. Система конфигурируется до 2048 процессоров. Пиковая производительность составляет 2,4 Тфлоп. Разделяемая, высокопроизводительная, глобально адресуемая подсистема памяти делает возможным обращение к локальной памяти каждого ПЭ в Cray T3E. Процессорные элементы в системе Cray T3E связаны в трехмерный тор двунаправленной высокоскоростной сетью с малым временем задержки, которая в шесть раз превосходит по скорости аналогичную сеть в Cray T3D. Также добавлена адаптивная маршрутизация, при которой возможен обход участков с высокой эффективностью передачи.

Системы Cray T3E выполняют операции ввода/вывода через многочисленные порты на один или более каналов GigaRing. Каналы ввода/вывода интегрированы в трехмерную межузловую сеть и пропорциональны размеру системы. При этом при добавлении ПЭ пропускная способность каналов ввода/вывода увеличивается, и масштабируемые

приложения могут выполняться на системах с большим числом процессоров так же эффективно, как на системах с меньшим числом процессоров.

Для Cray T3E была создана масштабируемая версия операционной системы ОС UNICOS - ОС UNICOS/mk. Операционная система UNICOS/mk разделена на программы-серверы, распределенные среди процессоров Cray T3E. Это позволяет управлять набором ресурсов системы как единым целым. Локальные серверы обрабатывают запросы ОС, специфичные для каждого ПЭ. Глобальные серверы обеспечивают общесистемные возможности, такие как управление процессами и файловые операции.

В добавлении к пользовательским ПЭ, которые выполняют приложения и команды, системы Cray T3E включают специальные системные ПЭ, которые выполняют глобальные серверы UNICOS/mk. Так как глобальные серверы расположены на системных ПЭ и не дублируются по всей системе, UNICOS/mk эффективно масштабируема, полнофункциональна и обслуживает от десятков до тысячи ПЭ с минимальной перегрузкой.

UNICOS/mk обеспечивает следующие программные функции:

- распределение серверов управления файлами. Функции файлового сервера распределяются, используя локальные файловые программы-сервера, для обеспечения максимальной производительности и эффективности;
- ПЭ может генерировать не только последовательную, но и параллельную передачу данных, используя некоторые или даже все ПЭ данной программы;
- множество глобальных файловых серверов: система управления файлами распределена на множество системных ПЭ, которые позволяют полностью использовать параллельные дисковые каналы, поддерживаемые на Cray T3E.

## Система T3E-1200

Быстродействие серии Cray T3E-1200 в два раза превышает производительность систем Cray T3E при уменьшенной вдвое стоимости за Мфлоп. Конфигурации в воздушно-жидкостном охлаждении имеют от 6 процессоров, а в жидкостном - от 32 процессоров. Каждый процессор имеет производительность в 1,2 Гфлоп; для всей системы пиковая производительность меняется от 7,2 Гфлоп до 2,5 Тфлоп. Масштабируется до тысяч процессоров. Серия выпущена в 1997 г.

Система предназначена для наиболее важных научных и технических задач в аэрокосмической, автомобильной, финансовой, химико-фармацевтической, нефтяной и т.д. отраслях промышленности, а также в широких областях прикладных исследований, включая химию, гидродинамику, предсказание погоды и сейсмические процессы.

Для поддержки масштабируемости используется операционная система UNICOS/mk - масштабируемая версия UNICOSR. Системы T3E-1200 поддерживают как явное распараллеливание распределенной памяти посредством CF90 и C/C++ с передачей сообщений (MPI, MPI-2 и PVM) и передачу данных, так и неявное распараллеливание посредством возможностей HPF и Cray CRAFT.

На системах T3E каждый интерфейс GigaRing имеет максимальную пропускную способность в 500 Мбайт/с.

В дополнение к высокой производительности и пропускной способности процессорных элементов и высокой масштабируемости, системы Cray T3E-1200 имеют две уникальные особенности: STREAMS и E-Регистры. STREAMS доводят до максимума пропускную способность локальной памяти, позволяя микропроцессору запускать при полной скорости для ссылки на векторoподобные данные. E-Регистры предоставляют операции gather/scatter (соединение/вразброс) для ссылок на локальную и удаленную память и используют полную пропускную способность внутреннего соединения для удаленного чтения и записи отдельного слова.

Оценка производительности вычислительной системы производилась при решении плотной линейной системы уравнений порядка 148800 на машине T3E-1200 с 1200 процессорами. Была достигнута скорость в 1,127 Тфлоп/с, что составляет 63% эффективности.

| Таблица 16.3. Оценка производительности | |
|---|---|
| **Число процессоров** | 6 - 128 32 - 2048 |
| **Тактовая частота процессора, МГц** | 600 |
| **Пиковая производительность, Тфлоп** | 2,4+ |
| **Размер памяти на процессор, Гбайт** | 0,256 - 2 |
| **Топология внутреннего соединения** | 3D двухнаправленный тор |
| **Максимальная двоичная пропускная способность, Гбайт/с** | 122 |
| **Максимальное число каналов GIGARING** | 128 |
| **Пиковая пропускная способность ввода/вывода, Гбайт** | 128 |

## ASCI White

Проект ASCI (Accelerated Strategic Computing Initiative - ускоренная стратегическая вычислительная инициатива) инициирован оборонными программами Министерства энергетики США в сотрудничестве с лабораториями Lawrence Livermore и Los Alamos (США) для перехода от ядерных испытаний к методам, основанным на численном моделировании создания ядерного оружия, оценки его производительности и т.п. Инициатива ASCI является ключевым элементом программы обслуживания арсеналов Stockpile Stewardship, направленной на обеспечение безопасности и надежности ядерных арсеналов страны при отсутствии испытаний ядерного оружия.

В конце июня 2000 г. компания IBM сообщила, что она построила самый быстрый суперкомпьютер в мире (на тот момент), выполняющий до 12 триллионов вычислений в секунду, что в тысячу раз быстрее, чем производительность <Deep Blue>. Суперкомпьютер RS/6000 SP, известный как ASCI White, занимающий площадь размером в два баскетбольных поля, используется Министерством энергетики США в программе по обеспечению безопасности и надежности запасов ядерного оружия без проведения натурных испытаний.

Система ASCI White является третьим шагом в плане Министерства энергетики США, согласно которому производительность суперкомпьютерной системы в 2004 г. должна составлять 100 Топер/с. В рамках ASCI-проекта в течение нескольких лет предполагается создать серию суперкомпьютеров производительностью в 1, 3, 10, 30 и 100 Тфлоп.

При проверке возможностей суперкомпьютера ASCI White показал вычислительную производительность в 12,28 Тфлоп, превысив требования контракта в этом пункте на 23%. Система инсталлирована в калифорнийской национальной Ливерморской лаборатории.

Система состоит из 8 192 микропроцессоров, имеет оперативную память объемом в 6 Тбайт и дисковую память в 160 Тбайт, что достаточно для шестикратного хранения всех книг библиотеки Конгресса США.

Аппаратное окружение ASCI White включает в себя систему IBM RS/6000 SP с 512 симметричными мультипроцессорными машинами (SMP-узлами). Каждый узел имеет 16 процессоров, а для системы в целом - 8192 процессора, обеспечиваемая пиковая производительность составляет не менее 12 Топер/с. Система имеет общую память 4 Тбайт и дисковую память 150 Тбайт.

Дополнительно система IBM SP оснащена внешней дисковой памятью, параллельной файловой системой GPFS, архивной памятью и средствами визуализации. Специализированная высокоскоростная сеть образует магистраль и соединяет все компоненты системы ASCI White.

Система IBM SP, которая формирует ядро ASCI White, образована из многих пакетов, в большинстве своем содержащих четыре узла. Все узлы являются симметричными мультипроцессорами IBM RS/6000 POWER3 с 64-разрядной архитектурой. Каждый узел является автономной машиной, обладающей собственной памятью, операционной системой, локальным диском и 16 процессорами. IBM производит несколько разновидностей узлов POWER3. Узлы ASCI White известны как узлы Nighthawk-2 (NH-2).

Процессоры POWER3 являются суперскалярными (одновременное выполнение многих команд) 64-разрядными чипами конвейерной организации с двумя устройствами по обработке команд с плавающей запятой и тремя устройствами по обработке целочисленных команд. Они способны выполнять до восьми команд за тактовый цикл и до четырех операций с плавающей запятой за такт. Все узлы соединены внутренней коммутационной сетью SP.

Общая параллельная файловая система IBM GPFS (General Parallel File System) обеспечивает обслуживание файловой системы для параллельных и последовательных приложений, запускаемых в окружении RS/6000 SP. GPFS разработана аналогично файловой системе UNIX: почти все приложения запускаются под GPFS так же, как они запускаются в других файловых системах. Это означает, что пользователи могут продолжать применять обычные команды UNIX для простых операций над файлами.

GPFS предоставляет совместный доступ к файлам, который может охватывать много дисководов на многих узлах SP. Отдельные файлы хранятся как ряд <блоков>, распределенных через диски на различных узлах памяти. Также поддерживается одновременное чтение и запись различных файлов.

Для защиты вычислительных средств (Secure Computing Facility - SCF) используется архивная система хранения данных HPSS (High Performance Storage System).

Система ASCI White построена таким образом, чтобы поддерживать смешанные моды программирования кластерной распределенной памяти с SMP общей памяти. MPI обычно используется для соединения распределенной памяти от узла к узлу.

Операционная система, как и на машине ASCI Blue-Pacific, представляет собой версию UNIX IBM AIX. AIX поддерживает как 32-разрядные, так и 64-разрядные системы RS/6000. Номер текущей версии - AIX 4.3.

Поддержка параллельного кода на ASCI White включает параллельные библиотеки, отладчики, профилировщики, утилиты IBM и сервисные программы, которые производят анализ эффективности выполнения. Поддерживаются MPI, OpenMP, потоки POSIX и транслятор директив IBM. Доступны: параллельный отладчик IBM, средства профилирования и TotalView.