1. Принципы построения параллельных вычислительных систем

1.1. Пути достижения параллелизма

В общем плане под *параллельными вычислениями* понимаются процессы обработки данных, в которых одновременно могут выполняться несколько операций компьютерной системы. Достижение параллелизма возможно только при выполнимости следующих требований к архитектурным принципам построения вычислительной среды:

- **независимость функционирования отдельных устройств ЭВМ** – данное требование относится в равной степени ко всем основным компонентам вычислительной системы – к устройствам ввода-вывода, к обрабатывающим процессорам и к устройствам памяти;
- **избыточность элементов вычислительной системы** – организация избыточности может осуществляться в следующих основных формах:
  - *использование специализированных устройств*, таких, например, как отдельные процессоры для целочисленной и вещественной арифметики, устройства многоуровневой памяти (регистры, кэш);
  - *дублирование устройств ЭВМ* путем использования, например, нескольких однотипных обрабатывающих процессоров или нескольких устройств оперативной памяти.

Дополнительной формой обеспечения параллелизма может служить *конвейерная* реализация обрабатывающих устройств, при которой выполнение операций в устройствах представляется в виде исполнения последовательности составляющих операцию подкоманд. Как результат, при вычислениях на таких устройствах на разных стадиях обработки могут находиться одновременно несколько различных элементов данных.

Возможные пути достижения параллелизма детально рассматриваются в Hockney and Jesshope (1988), Patterson and Hennessy (1996), Culler and Singh (1998), Корнеев (1999), Воеводин В.В. и Воеводин Вл.В. (2002), Таненбаум (2002); в этих же работах описывается история развития параллельных вычислений и приводятся примеры конкретных параллельных ЭВМ (см. также Xu and Hwang (1998), Culler, Singh and Gupta (1998) Buyya (1999)).

При рассмотрении проблемы организации параллельных вычислений следует различать следующие возможные режимы выполнения независимых частей программы:

– *многозадачный режим* (*режим разделения времени*), при котором для выполнения нескольких процессов используется единственный процессор; данный режим является псевдопараллельным, когда активным (исполняемым) может быть один единственный процесс, а все остальные процессы находятся в состоянии ожидания своей очереди на использование процессора; использование режима разделения времени может повысить эффективность организации вычислений (например, если один из процессов не

может выполняться из-за ожидания вводимых данных, процессор может быть задействован для выполнения другого, готового к исполнению процесса – см. Tanenbaum (2001)), кроме того, в данном режиме проявляются многие эффекты параллельных вычислений (необходимость взаимоисключения и синхронизации процессов и др.) и, как результат, этот режим может быть использован при начальной подготовке параллельных программ;

– *параллельное выполнение*, когда в один и тот же момент времени может выполняться несколько команд обработки данных; данный режим вычислений может быть обеспечен не только при наличии нескольких процессоров, но и реализован при помощи конвейерных и векторных обрабатывающих устройств;

– *распределенные вычисления*; данный термин обычно используют для указания параллельной обработки данных, при которой используется несколько обрабатывающих устройств, достаточно удаленных друг от друга, и в которых передача данных по линиям связи приводит к существенным временным задержкам; как результат, эффективная обработка данных при данном способе организации вычислений возможна только для параллельных алгоритмов с низкой интенсивностью потоков межпроцессорных передач данных; перечисленные условия являются характерными, например, при организации вычислений в *многомашинных вычислительных комплексах*, образуемых объединением нескольких отдельных ЭВМ с помощью каналов связи локальных или глобальных информационных сетей.

В рамках данного учебного материала основное внимание будет уделяться второму типу организации параллелизма, реализуемому на многопроцессорных вычислительных системах.

1.2. Примеры параллельных вычислительных систем

Разнообразие параллельных вычислительных систем поистине огромно. В каком-то смысле каждая такая система уникальна. В них устанавливаются различные аппаратные составляющие: процессоры (Intel, IBM, AMD, HP, NEC, Cray, …), сетевые карты (Ethernet, Myrinet, Infiniband, SCI, …). Они функционируют под управлением различных операционных систем (версии Unix/Linux, версии Windows, …)и используют различное прикладное программное обеспечение. Кажется, что найти между ними что-то общее практически невозможно. Конечно же, это не так, и ниже мы попытаемся с общих позиций сформулировать некоторые известные варианты классификаций параллельных вычислительных систем, но прежде рассмотрим несколько примеров.

1.2.1. Суперкомпьютеры

Началом эры суперкомпьютеров с полным правом может считаться 1976 год – год появления первой векторной системы Cray 1. Результаты, показанные ею, пусть и на ограниченном в то время наборе приложений, были столь впечатляющи в сравнении с остальными, что система заслуженно получила название “суперкомпьютер” и в течение длительного времени определяла развитие всей индустрии высокопроизводительных вычислений. Однако в результате совместной эволюции архитектур и программного обеспечения на рынке стали появляться системы с весьма кардинально различающимися характеристиками, потому само понятие “суперкомпьютер” стало многозначным, и пересматривать его пришлось неоднократно.

Попытки дать определение термину *суперкомпьютер*, опираясь только на производительность, неизбежно приводят к необходимости постоянно поднимать планку, отделяющую его от рабочей станции или даже обычного настольного компьютера. Так по определению Оксфордского словаря вычислительной техники 1986 года, для того, чтобы получить это гордое название, нужно было иметь производительность в 10 MFlops[1)]. Сегодня, как известно, производительность настольных систем на два порядка выше.

Из альтернативных определений наиболее интересны два: экономическое и философское. Первое из них гласит, что суперкомпьютер – это система, цена которой выше 1-2 млн. долларов. Второе, что суперкомпьютер – это компьютер, мощность которого всего на порядок меньше необходимой для решения современных задач. В некотором общем плане, под суперкомпьютером можно понимать вычислительную систему, которая обладает предельными характеристиками по производительности среди имеющихся в каждый конкретный момент времени компьютеров.

[1)] MFlops – million of floating point operations per second – миллион операций над числами с плавающей запятой в секунду, GFlops – миллиард, TFlops – триллион соответственно.

1.2.1.1 Программа ASCI

Программа **ASCI** (http://www.llnl.gov/asci/) – Accelerated Strategic Computing Initiative, поддерживаемая Министерством энергетики США, в качестве одной из основных целей имеет создание суперкомпьютера с производительностью в 100 TFlops.

Первая система серии ASCI – **ASCI Red** построенная в 1996 г. компанией Intel стала и первым в мире компьютером с производительностью в 1 TFlops (в дальнейшем была доведена до 3 TFlops).

Тремя годами спустя появились **ASCI Blue Pacific** от IBM и **ASCI Blue Mountain** от SGI, ставшие первыми на тот момент суперкомпьютерами с быстродействием 3 TFlops.

Наконец, в июне 2000 г. была введена в действие система **ASCI White** (http://www.llnl.gov/asci/platforms/white/), с пиковой производительностью свыше 12 TFlops (реально показанная производительность на тесте LINPACK составила на тот момент 4938 GFlops, позднее была доведена до 7304 GFlops).

Аппаратно ASCI White представляет собой систему IBM RS/6000 SP с 512-ю симметричными мультипроцессорными (SMP) узлами. Каждый узел имеет 16 процессоров, система в целом – 8192 процессора. Оперативная память системы – 4 TB, емкость дискового пространства 180 TB.

Все узлы системы являются симметричными мультипроцессорами IBM RS/6000 POWER3 с 64-х разрядной архитектурой. Каждый узел автономен, обладает собственной памятью, операционной системой, локальным диском и 16 процессорами.

Процессоры POWER3 являются суперскалярными 64-х 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 четырёх операций с плавающей запятой за такт. Тактовая частота каждого процессора 375 MHz.

Программное обеспечение ASCI White поддерживает смешанную модель программирования – передача сообщений между узлами и многопотоковость внутри SMP-узла.

Операционная система представляет собой версию UNIX – IBM AIX. AIX поддерживает как 32-х, так и 64-х разрядные системы RS/6000.

Поддержка параллельного кода на ASCI White включает параллельные библиотеки, отладчики (в частности TotalView), профилировщики, утилиты IBM и сервисные программы по анализу эффективности выполнения. Поддерживаются библиотеки MPI, OpenMP, потоки POSIX и транслятор директив IBM. Имеется параллельный отладчик IBM.

1.2.1.2 Система BlueGene

Самый мощный на данный момент суперкомпьютер в мире создан IBM. Точнее говоря, работы над ним еще не закончены. В настоящий момент система имеет полное название “BlueGene/L DD2 beta-System” и представляет собой “первую очередь” полной вычислительной системы. Согласно прогнозам к моменту ввода в строй ее пиковая производительность достигнет 360 TFlops.

В качестве основных областей применения разработчики называют гидродинамику, квантовую химию, моделирование климата и др.

Текущий вариант системы имеет следующие характеристики:

- 32 стойки по 1024 двухядерных 32-битных процессора PowerPC 440 0.7 GHz в каждой,
- пиковая производительность – порядка 180 TFlops,
- максимальная показанная производительность (на тесте LINPACK) – 135 TFlops.

1.2.1.3 Система МВС-1000

Один из самых известных в России суперкомпьютеров МВС-1000М (Многопроцессорная Вычислительная Система) установлен в Межведомственном Суперкомпьютерном Центре Российской Академии Наук.

Работы по созданию МВС-1000М проводились с апреля 2000 года по август 2001 года.

Согласно официальным данным (http://www.jscc.ru) состав системы:

- 384 двухпроцессорных модуля на базе Alpha 21264 667 MHz (кэш L2 4 Mb), собранные в виде 6 базовых блоков, по 64 модуля в каждом,
- управляющий сервер,
- файл-сервер NetApp F840,

- сеть Myrinet 2000,
- сети Fast/Gigabit Ethernet,
- сетевой монитор,
- система бесперебойного электропитания.

Каждый вычислительный модуль имеет по 2 Gb оперативной памяти, HDD 20 Gb, сетевые карты Myrinet (2000 Mbit) и Fast Ethernet (100 Mbit).

При обмене данными между модулями с использованием протоколов MPI на сети Myrinet пропускная способность в МВС-1000М составляет 110 - 150 Mb в секунду.

Программное обеспечение системы составляют:

- операционные системы управляющего и резервного управляющего сервера – ОС Linux RedHat 6.2 с поддержкой SMP,
- операционная система вычислительных модулей – ОС Linux RedHat 6.2 с поддержкой SMP,
- операционная среда параллельного программирования – пакет “MPICH GM,
- программные средства коммуникационных сетей (Myrinet, Fast Ethernet),
- оптимизированные компиляторы языков программирования C, C++, FORTRAN фирмы Compaq,
- отладчик параллельных программ TotalView,
- средства профилирования параллельных программ,
- средства параллельного администрирования.

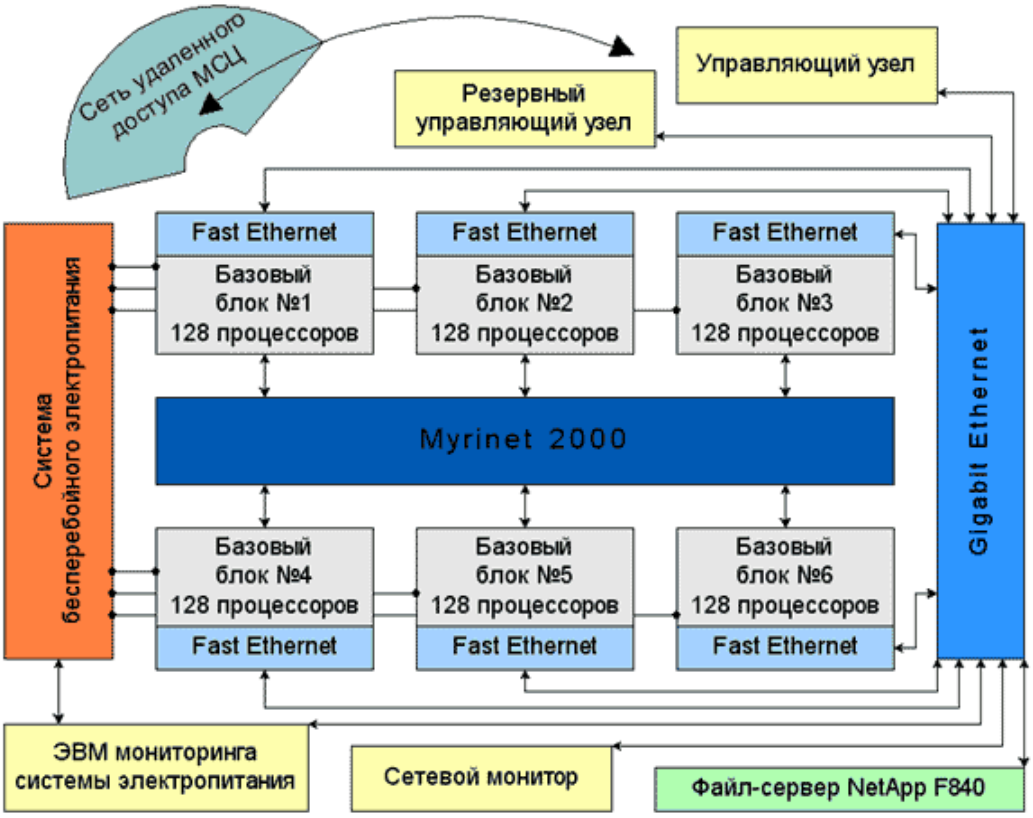


Рис. 1.1. Структура суперкомпьютера МВС-1000М

Обслуживается МВС-1000М двумя основными компонентами:

- подсистемой удаленного управления и непрерывного мониторинга,
- подсистемой коллективного доступа.

В летнем списке Top500 2004 года с пиковой производительностью 1024 GFlops и максимально показанной на тесте LINPACK 734 GFlops МВС-1000М занял 391 позицию. В 2005 г. начаты работы по выводу системы из эксплуатации вследствие установки более производительной системы МВС-15000.

1.2.1.4 Система МВС-15000

В настоящий момент в МСЦ РАН вводится в строй уже сейчас самый мощный суперкомпьютер России (согласно последней редакции списка Top50 стран СНГ – http://supercomputers.ru/index.php).

Аппаратная конфигурация вычислительных узлов МВС-15000 включает в себя:

- 2 процессора IBM PowerPC 970 с тактовой частотой 2.2 GHz, кэш L1 96 Kb и кэш L2 512 Kb,
- 4 Gb оперативной памяти на узел,
- 40 Gb жесткий диск IDE,
- 2 встроенных адаптера Gigabit Ethernet,
- адаптер Myrinet типа M3S-PCIXD-2-I.

Состав программного обеспечения МВС-15000:

- операционные системы SuSe Linux Enterprise Server версии 8 для платформ x86 и PowerPC,
- пакет GM 2.0.12 в качестве коммуникационной среды Myrinet,
- пакет MPICH-GM в качестве среды параллельного программирования,
- система управления прохождением задач и их пакетной обработки.

В настоящий момент (начало 2005) система МВС-15000 имеет общее количество узлов 276 (552 процессора), пиковую производительность 4857.6 GFlops и максимально показанную на тесте LINPACK 3052 GFlops.

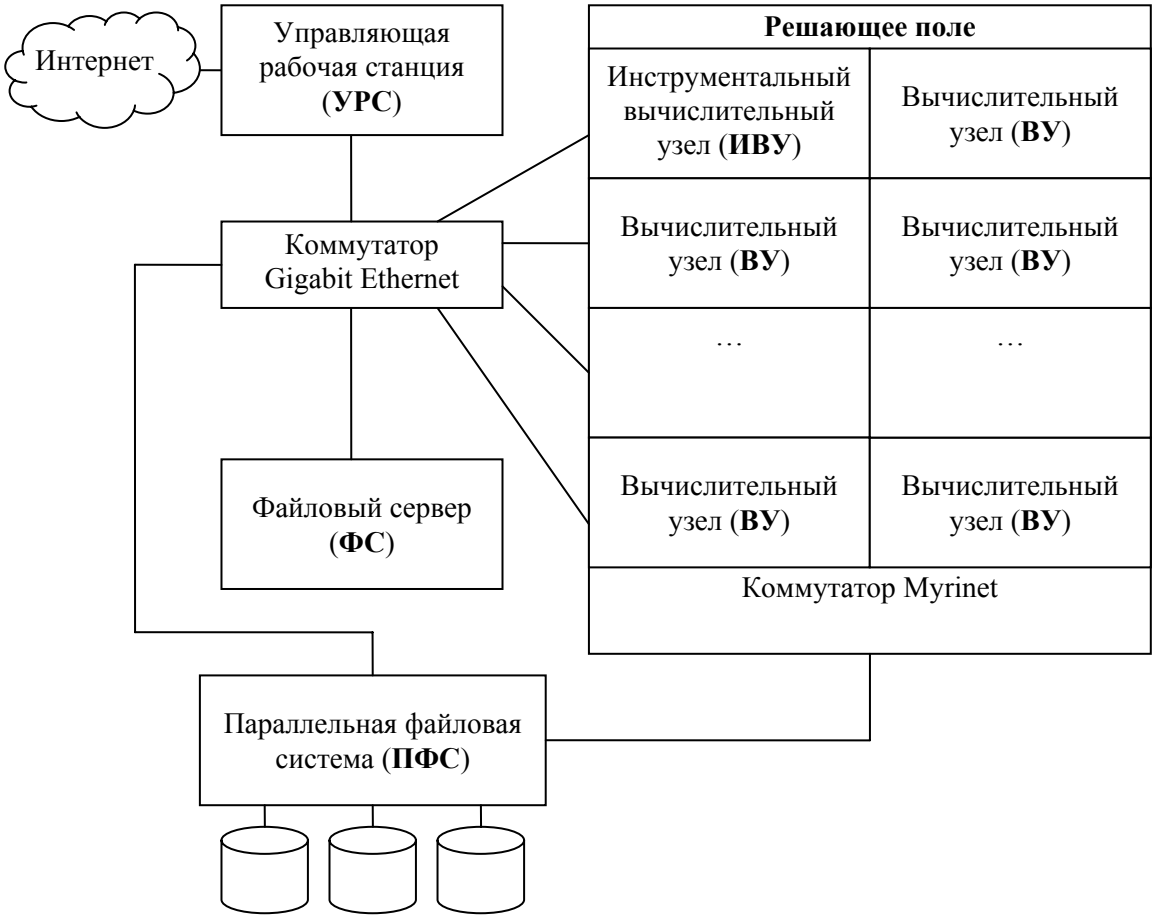


Рис. 1.2. Структурная схема СК МВС-15000

1.2.2. Кластеры

Кластер – группа компьютеров, объединенных в локальную вычислительную сеть (ЛВС) и способных работать в качестве единого вычислительного ресурса. Предполагает более высокую надежность и эффективность, нежели ЛВС, и существенно более низкую стоимость в сравнении с

другими типами параллельных вычислительных систем (за счет использования типовых аппаратных и программных решений).

Исчисление истории кластеров можно начать от первого проекта, в котором одной из основных целей являлось установление связи между компьютерами, – проекта ARPANET[2]. Именно тогда были заложены первые, оказавшиеся фундаментальными, принципы, приведшие впоследствии к созданию локальных и глобальных вычислительных сетей, и, конечно же, всемирной глобальной компьютерной сети Интернет. Правда, с момента ввода в действие сети ARPANET до появления первого кластера должно было пройти более двадцати лет.

Эти годы вместили в себя гигантский скачок в развитии аппаратной базы, появление и воцарение на рынке микропроцессоров и персональных компьютеров, накопление критической массы идей и методов параллельного программирования, что привело, в конечном счете, к решению извечной проблемы уникальности каждой параллельной вычислительной установки – разработке стандартов на создание параллельных программ для систем с общей и распределенной памятью. Добавим к этому дороговизну имеющихся на тот момент решений в области высокопроизводительных систем, предполагавших использование быстродействующих, а потому специфических компонент. Также учтем непрерывное улучшение соотношения “цена/производительность” персональных компьютеров. В свете всех этих обстоятельств появление кластеров было неизбежным.

Преимущества нового подхода создания вычислительных систем большой мощности, получившие признание практически сразу после первого представления такой системы, со временем только возрастали, поддерживаемые непрерывным ростом производительности типовых компонент.

В настоящее время в списке TOP500 самых высокопроизводительных систем кластеры составляют большую часть – 294 установки.

1.2.2.1 Кластер Beowulf

Первым в мире кластером, по-видимому, является кластер, созданный под руководством Томаса Стерлинга и Дона Бекера в научно-космическом центре NASA – Goddard Space Flight Center – летом 1994 года. Названный в честь героя скандинавской саги, обладавшего по преданию силой тридцати человек, кластер состоял из 16 компьютеров на базе процессоров 486DX4 с тактовой частотой 100 MHz. Каждый узел имел 16 Mb оперативной памяти. Связь узлов обеспечивалась тремя параллельно работавшими 10Mbit/s сетевыми адаптерами. Кластер функционировал под управлением операционной системы Linux, использовал GNU компилятор и поддерживал параллельные программы на основе MPI. Процессоры узлов кластера были слишком быстрыми по сравнению с пропускной способностью обычной сети Ethernet, поэтому для балансировки системы Дон Бекер переписал драйверы Ethernet под Linux для создания дублированных каналов и распределения сетевого трафика.

Идея “собери суперкомпьютер своими руками” быстро пришлась по вкусу в первую очередь академическому сообществу. Использование типовых массово выпускающихся компонент, как аппаратных, так и программных вело к значительному уменьшению стоимости разработки и внедрения системы. Вместе с тем производительность получающегося вычислительного комплекса была вполне достаточной для решения существенного количества задач, требовавших большого объема вычислений. Системы класса “кластер Beowulf” стали появляться по всему миру.

Четыре годя спустя в Лос-Аламосской национальной лаборатории (США) астрофизик Майкл Уоррен и другие ученые из группы теоретической астрофизики построили суперкомпьютер Avalon, который представлял собой Linux-кластер на базе процессоров Alpha 21164A с тактовой частотой 533 MHz. Первоначально включавший 68 процессоров, позднее Avalon был расширен до 140. Каждый узел содержал 256 Mb RAM, 3 Gb HDD, Fast Ethernet card. Общая стоимость проекта Avalon составила чуть более 300 тыс. долл.

На момент ввода в строй полной версии (осень 1998 года) с пиковой производительностью в 149 GFlops и показанной на тесте LINPACK 48.6 GFlops кластер занял 113 место в списке Top500.

В том же году на самой престижной конференции в области высокопроизводительных вычислений Supercomputing'98 создатели Avalon представили доклад “Avalon: An Alpha/Linux Cluster Achieves 10 GFlops for $150k”, получивший первую премию в номинации “наилучшее отношение цена/производительность”.

В настоящее время под кластером типа “Beowulf” понимается система, состоящая из одного серверного узла и одного или более клиентских узлов, соединенных при помощи Ethernet или некоторой

[2] ARPANET – проект Агентства передовых исследовательских проектов (Advanced Research Projects Agency, DARPA) Министерства обороны США по созданию компьютерной сети с целью проведения экспериментов в области компьютерных коммуникаций, поддержания связи в условиях ядерного нападения, разработки концепции децентрализованного управления (1966 – 1969 гг.).

другой сети. Это система, построенная из готовых серийно выпускающихся промышленных компонент, на которых может работать ОС Linux или ОС Windows, стандартных адаптеров Ethernet и коммутаторов. Она не содержит специфических аппаратных компонент и легко воспроизводима. Серверный узел управляет всем кластером и является файл-сервером для клиентских узлов. Он также является консолью кластера и шлюзом во внешнюю сеть. Большие системы Beowulf могут иметь более одного серверного узла, а также, возможно, специализированные узлы, например, консоли или станции мониторинга. В большинстве случаев клиентские узлы в Beowulf пассивны. Они конфигурируются и управляются серверными узлами и выполняют только то, что предписано серверным узлом.

1.2.2.2 Кластер AC3 Velocity Cluster

Кластер AC3 Velocity Cluster, установленный в Корнельском университете (США) (http://www.tc.cornell.edu) стал результатом совместной деятельности университета и консорциума AC3 (Advanced Cluster Computing Consortium), образованного компаниями Dell, Intel, Microsoft, Giganet и еще 15 производителями ПО с целью интеграции различных технологий для создания кластерных архитектур для учебных и государственных учреждений.

Состав кластера:

- 64 четырехпроцессорных сервера Dell PowerEdge 6350 на базе Intel Pentium III Xeon 500 MHz, 4 GB RAM, 54 GB HDD, 100 Mbit Ethernet card;
- 1 восьмипроцессорный сервер Dell PowerEdge 6350 на базе Intel Pentium III Xeon 550 MHz, 8 GB RAM, 36 GB HDD, 100 Mbit Ethernet card.

Четырехпроцессорные сервера смонтированы по восемь штук на стойку и работают под управлением ОС Microsoft Windows NT 4.0 Server Enterprise Edition. Между серверами установлено соединение на скорости 100 Мбайт/с через Cluster Switch компании Giganet.

Задания в кластере управляются с помощью Cluster ConNTroller, созданном в Корнельском университете. Пиковая производительность AC3 Velocity составляет 122 GFlops при стоимости в 4-5 раз меньше, чем у суперкомпьютеров с аналогичными показателями.

На момент ввода в строй (лето 2000 года) кластер с показателем производительности на тесте LINPACK в 47 GFlops занимал 381 строку списка TOP500.

1.2.2.3 Кластер NCSA NT Supercluster

В 2000 году в Национальном центре суперкомпьютерных технологий (NCSA – National Center for Supercomputing Applications) на основе рабочих станций Hewlett-Packard Kayak XU PC workstation (http://www.hp.com/desktops/kayak/) был собран еще один кластер, для которого в качестве операционной системы была выбрана ОС Microsoft Windows. Недолго думая, разработчики окрестили его “NT Supercluster” (http://archive.ncsa.uiuc.edu/SCD/Hardware/NTCluster/).

На момент ввода в строй кластер с показателем производительности на тесте LINPACK в 62 GFlops и пиковой производительностью в 140 GFlops занимал 207 строку списка TOP500.

Кластер построен из 38 двупроцессорных систем на базе Intel Pentium III Xeon 550 MHz, 1 Gb RAM, 7.5 Gb HDD, 100 Mbit Ethernet card.

Связь между узлами основана на сети Myrinet (http://www.myri.com/myrinet/index.html).

Программное обеспечение кластера:

- операционная система – Microsoft Windows NT 4.0,
- компиляторы – с языков Fortran77, C/C++,
- уровень передачи сообщений основан на HPVM (http://www-csag.ucsd.edu/projects/clusters.html).

1.2.2.4 Кластер Thunder

В настоящий момент число систем, собранных на основе процессоров корпорации Intel и представленных в списке Top500, составляет 318 штук. Самый мощный суперкомпьютер, представляющий собой кластер на основе Intel Itanium2, установлен в Ливерморской Национальной Лаборатории (США).

Аппаратная конфигурация кластера Thunder (http://www.llnl.gov/linux/thunder/):

- 1024 сервера, по 4 процессора Intel Itanium 1.4 GHz в каждом,
- 8 Gb оперативной памяти на узел,
- общая емкость дисковой системы 150 Tb.

Программное обеспечение:

- операционная система CHAOS 2.0,
- библиотека передачи сообщений MPICH2,
- отладчик TotalView,
- компиляторы Intel и GNU Fortran, C/C++.

В настоящий момент с пиковой производительностью 22938 GFlops и максимально показанной на тесте LINPACK 19940 GFlops кластер Thunder занимает 5-ю позицию списка Top500 (на момент установки – лето 2004 года – занимал 2-ю строку).

1.2.3. Высокопроизводительный вычислительный кластер ННГУ

В качестве следующего примера рассмотрим вычислительный кластер Нижегородского университета, оборудование для которого было передано в рамках Академической программы Интел в 2001 г. В состав кластера входит (см. рис. 1.3):

– 2 вычислительных сервера, каждый из которых имеет 4 процессора Intel Pentium III 700 Мгц, 512 MB RAM, 10 GB HDD, 1 Гбит Ethernet card;

– 12 вычислительных серверов, каждый из которых имеет 2 процессора Intel Pentium III 1000 Мгц, 256 MB RAM, 10 GB HDD, 1 Гбит Ethernet card;

– 12 рабочих станций на базе процессора Intel Pentium 4 1300 Мгц, 256 MB RAM, 10 GB HDD, 10/100 Fast Etherrnet card.

Следует отметить, что в результате передачи подобного оборудования Нижегородский госуниверситет оказался первым вузом в Восточной Европе, оснащенным ПК на базе новейшего процессора INTEL®PENTIUM®4. Подобное достижение является дополнительным подтверждением складывающегося плодотворного сотрудничества ННГУ и корпорации Интел.

Важной отличительной особенностью кластера является его неоднородность (*гетерогенность*). В состав кластера входят рабочие места, оснащенные процессорами Intel Pentium 4 и соединенные относительно медленной сетью (100 Мбит), а также вычислительные 2- и 4- процессорные сервера, обмен данными между которыми выполняется при помощи быстрых каналов передачи данных (1000 Мбит). В результате кластер может использоваться не только для решения сложных вычислительно-трудоемких задач, но также и для проведения различных экспериментов по исследованию многопроцессорных кластерных систем и параллельных методов решения научно-технических задач.

В качестве системной платформы для построения кластера выбраны современные операционные системы семейства Microsoft Windows (для проведения отдельных экспериментов имеется возможность использования ОС Unix). Выбор такого решения определяется рядом причин, в числе которых основными являются следующие моменты:

– операционные системы семейства Microsoft Windows (так же как и ОС Unix) широко используются для построения кластеров; причем, если раньше применение ОС Unix для этих целей было преобладающим системным решением, в настоящее время тенденцией является увеличение числа создаваемых кластеров под управлением ОС Microsoft Windows (см., например, www.tc.cornell.edu/ac3/, www.windowclusters.org и др.),

– разработка прикладного программного обеспечения выполняется преимущественно с использованием ОС Microsoft Windows,

– корпорация Microsoft проявила заинтересованность в создании подобного кластера и передала в ННГУ для поддержки работ все необходимое программное обеспечение (ОС MS Windows 2000 Professional, ОС MS Windows 2000 Advanced Server и др.).

В результате принятых решений программное обеспечение кластера является следующим:

– вычислительные сервера работают под управлением ОС Microsoft® Windows® 2000 Advanced Server; на рабочих местах разработчиков установлена ОС Microsoft® Windows® 2000 Professional,

– в качестве сред разработки используются Microsoft® Visual Studio 6.0; для выполнения исследовательских экспериментов возможно использование компилятора Intel® C++ Compiler 5.0, встраиваемого в среду Microsoft® Visual Studio,

– на рабочих местах разработчиков установлены библиотеки:

- Plapack 3.0 (см. www.cs.utexas.edu/users/plapack),
- MKL (см. developer.intel.com/software/products/mkl/index.htm),

– в качестве средств передачи данных между процессорами установлены две реализации стандарта MPI:

- Argonne MPICH (www.unix.mcs.anl.gov/mpi/MPICH/),

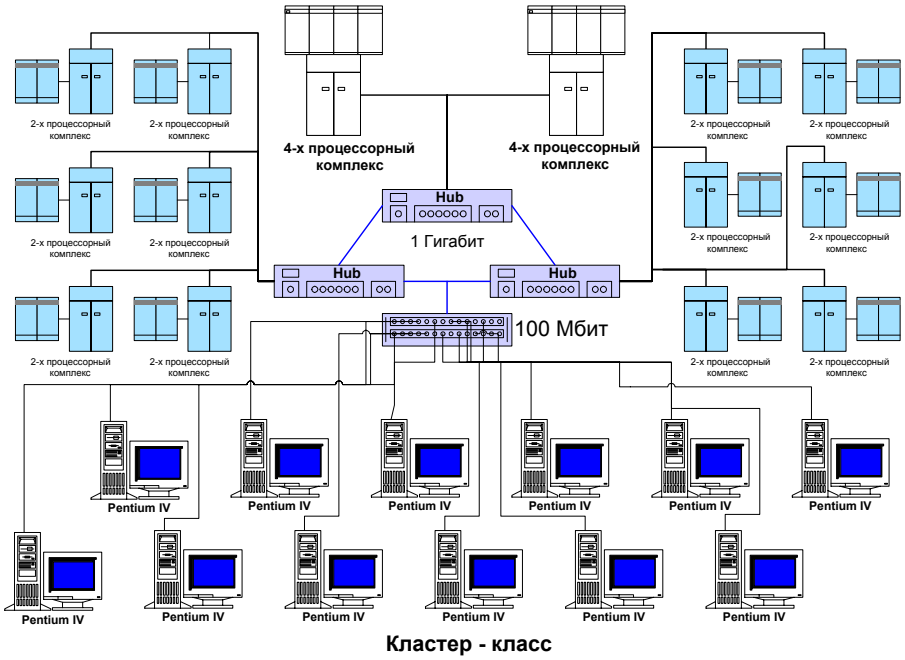


Рис. 1.3. Структура вычислительного кластера Нижегородского университета

- MP-MPICH (www.lfbs.rwth-aachen.de/~joachim/MP-MPICH.html),

– в опытной эксплуатации находится система разработки параллельных программ DVM (см. www.keldysh.ru/dvm/).

1.3. Классификация вычислительных систем

Одним из наиболее распространенных способов классификации ЭВМ является *систематика Флинна* (Flynn), в рамках которой основное внимание при анализе архитектуры вычислительных систем уделяется способам взаимодействия последовательностей (*потоков*) выполняемых команд и обрабатываемых данных. В результате такого подхода различают следующие основные типы систем (см. Flynn (1996), Patterson and Hennessy (1996), Воеводин В.В. и Воеводин Вл.В. (2002)):

- **SISD** (Single Instruction, Single Data) – системы, в которых существует одиночный поток команд и одиночный поток данных; к данному типу систем можно отнести обычные последовательные ЭВМ;
- **SIMD** (Single Instruction, Multiple Data) – системы с одиночным потоком команд и множественным потоком данных; подобный класс составляют многопроцессорные вычислительные системы, в которых в каждый момент времени может выполняться одна и та же команда для обработки нескольких информационных элементов; подобной архитектурой обладают, например, многопроцессорные системы с единым устройством управления; данный подход широко использовался в предшествующие годы (системы ILLIAC IV или CM-1 компании Thinking Machines), в последнее время его применение ограничено, в основном, созданием специализированных систем;
- **MISD** (Multiple Instruction, Single Data) – системы, в которых существует множественный поток команд и одиночный поток данных; относительно данного типа систем нет единого мнения – ряд специалистов говорят, что примеров конкретных ЭВМ, соответствующих данному типу вычислительных систем, не существует, и введение подобного класса предпринимается для полноты системы классификации; другие же относят к данному типу, например, *систолические вычислительные системы* (см. Kung (1982), Kumar et al. (1994)) или системы с конвейерной обработкой данных;
- **MIMD** (Multiple Instruction, Multiple Data) – системы с множественным потоком команд и множественным потоком данных; к подобному классу систем относится большинство параллельных многопроцессорных вычислительных систем.

Следует отметить, что хотя систематика Флинна широко используется при конкретизации типов компьютерных систем, такая классификация приводит к тому, что практически все виды параллельных систем (несмотря на их существенную разнородность) относятся к одной группе MIMD. Как результат, многими исследователями предпринимались неоднократные попытки детализации систематики Флинна. Так, например, для класса MIMD предложена практически общепризнанная структурная схема (см. Xu and Hwang (1998), Buyya (1999)), в которой дальнейшее разделение типов многопроцессорных систем основывается на используемых способах организации оперативной памяти в этих системах (см. рис. 1.4).

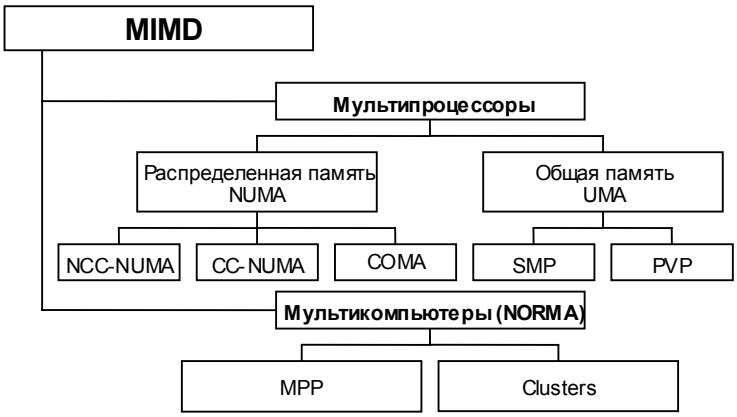
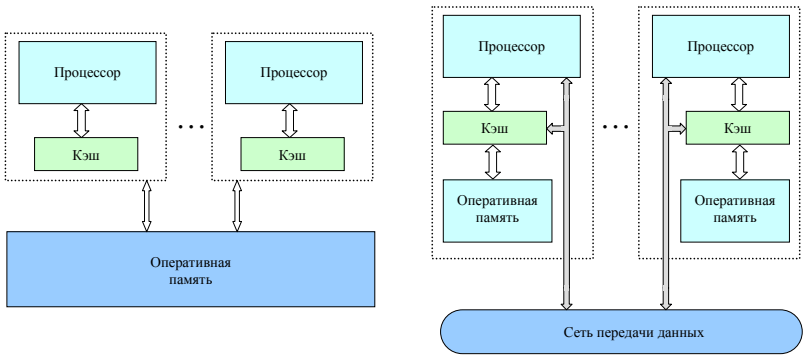


Рис. 1.4. Классификация многопроцессорных вычислительных систем

Данный поход позволяет различать два важных типа многопроцессорных систем – *multiprocessors* (*мультипроцессоры* или системы с общей разделяемой памятью) и *multicomputers* (*мультикомпьютеры* или системы с распределенной памятью).

1.3.1. Мультипроцессоры

Для дальнейшей систематики **мультипроцессоров** учитывается способ построения общей памяти. Возможный подход – использование единой (централизованной) *общей памяти* (*shared memory*) – см. рис. 1.5. Такой подход обеспечивает *однородный доступ к памяти* (*uniform memory access or UMA*) и служит основой для построения *векторных параллельных процессоров* (*parallel vector processor or PVP*) и симметричных *мультипроцессоров* (*symmetric multiprocessor or SMP*). Среди примеров первой группы суперкомпьютер Cray T90, ко второй группе относятся IBM eServer, Sun StarFire, HP Superdome, SGI Origin и др.



а) б)

Рис. 1.5. Архитектура многопроцессорных систем с общей (разделяемой) памятью:
системы с (а) однородным и (б) неоднородным доступом к памяти

Одной из основных проблем, которые возникают при организации параллельных вычислений на такого типа системах, является доступ с разных процессоров к общим данным и обеспечение, в этой связи, *однозначности (когерентности) содержимого разных кэшей* (*cache coherence problem*). Дело в том, что при наличии общих данных копии значений одних и тех же переменных могут оказаться в кэшах разных процессоров. Если в такой ситуации (при наличии копий общих данных) один из процессоров выполнит изменение значения разделяемой переменной, то значения копий в кэшах других процессорах окажутся не соответствующими действительности и их использование приведет к некорректности вычислений. Обеспечение однозначности кэшей обычно реализуется на аппаратном уровне – для этого после изменения значения общей переменной все копии этой переменной в кэшах отмечаются как недействительные и последующий доступ к переменной потребует обязательного обращения к основной памяти. Следует отметить, что необходимость обеспечения когерентности приводит к некоторому снижению скорости вычислений и затрудняет создание систем с достаточно большим количеством процессоров.

Наличие общих данных при выполнении параллельных вычислений приводит к необходимости *синхронизации взаимодействия* одновременно выполняемых потоков команд. Так, например, если изменение общих данных требует для своего выполнения некоторой последовательности действий, то необходимо обеспечить *взаимоисключение* (*mutual exclusion*) с тем, чтобы эти изменения в любой момент времени мог выполнять только один командный поток. Задачи взаимоисключения и синхронизации относятся к числу классических проблем, и их рассмотрение при разработке параллельных программ является одним из основных вопросов параллельного программирования.

Общий доступ к данным может быть обеспечен и при физически распределенной памяти (при этом, естественно, длительность доступа уже не будет одинаковой для всех элементов памяти) – см. рис. 1.5. Такой подход именуется как *неоднородный доступ к памяти* (*non-uniform memory access or NUMA*). Среди систем с таким типом памяти выделяют:

– Системы, в которых для представления данных используется только локальная кэш-память имеющихся процессоров (*cache-only memory architecture or COMA*); примерами таких систем являются, например, KSR-1 и DDM;

– Системы, в которых обеспечивается когерентность локальных кэшей разных процессоров (*cache-coherent NUMA or CC-NUMA*); среди систем данного типа SGI Origin 2000, Sun HPC 10000, IBM/Sequent NUMA-Q 2000;

– Системы, в которых обеспечивается общий доступ к локальной памяти разных процессоров без поддержки на аппаратном уровне когерентности кэша (*non-cache coherent NUMA or NCC-NUMA*); к данному типу относится, например, система Cray T3E.

Использование распределенной общей памяти (*distributed shared memory* or *DSM*) упрощает проблемы создания мультипроцессоров (известны примеры систем с несколькими тысячами процессоров), однако, возникающие при этом проблемы эффективного использования распределенной памяти (время доступа к локальной и удаленной памяти может различаться на несколько порядков) приводят к существенному повышению сложности параллельного программирования.

1.3.2. Мультикомпьютеры

Мультикомпьютеры (многопроцессорные системы с распределенной памятью) уже не обеспечивают общий доступ ко всей имеющейся в системах памяти (*no-remote memory access or NORMA*) – см. рис. 1.6. При всей схожести подобной архитектуры с системами с распределенной общей памятью (рис. 1.5 б), мультикомпьютеры имеют принципиальное отличие – каждый процессор системы может использовать только свою локальную память, в то время как для доступа к данным, располагаемых на других процессорах, необходимо явно выполнить *операции передачи сообщений* (*message passing operations*). Данный подход используется при построении двух важных типов многопроцессорных вычислительных систем (см. рис. 1.4) - *массивно-параллельных систем* (*massively parallel processor or MPP*) и *кластеров* (*clusters*). Среди представителей первого типа систем - IBM RS/6000 SP2, Intel PARAGON, ASCI Red, транспьютерные системы Parsytec и др.; примерами кластеров являются, например, системы AC3 Velocity и NCSA NT Supercluster.

Следует отметить чрезвычайно быстрое развитие многопроцессорных вычислительных систем **кластерного типа** – общая характеристика данного подхода приведена, например, в обзоре, подготовленном под редакцией Barker (2000). Под ***кластером*** обычно понимается (см., например, Xu and Hwang (1998), Pfister (1998)) множество отдельных компьютеров, объединенных в сеть, для которых при помощи специальных аппаратно-программных средств обеспечивается возможность унифицированного управления (*single system image*), надежного функционирования (*availability*) и эффективного использования (*performance*). Кластеры могут быть образованы на базе уже существующих у потребителей отдельных компьютеров, либо же сконструированы из типовых компьютерных элементов, что обычно не требует значительных финансовых затрат. Применение кластеров может также в некоторой степени снизить проблемы, связанные с разработкой параллельных алгоритмов и программ, поскольку повышение вычислительной мощности отдельных процессоров позволяет строить кластеры из сравнительно небольшого количества (несколько десятков) отдельных компьютеров (*lowly parallel processing*). Это приводит к тому, что для параллельного выполнения в алгоритмах решения вычислительных задач достаточно выделять только крупные независимые части расчетов (*coarse granularity*), что, в свою очередь, снижает сложность построения параллельных методов вычислений и уменьшает потоки передаваемых данных между компьютерами кластера. Вместе с этим следует отметить, что организация взаимодействия вычислительных узлов кластера при помощи передачи сообщений обычно приводит к значительным временным задержкам, что накладывает дополнительные ограничения на тип разрабатываемых параллельных алгоритмов и программ.

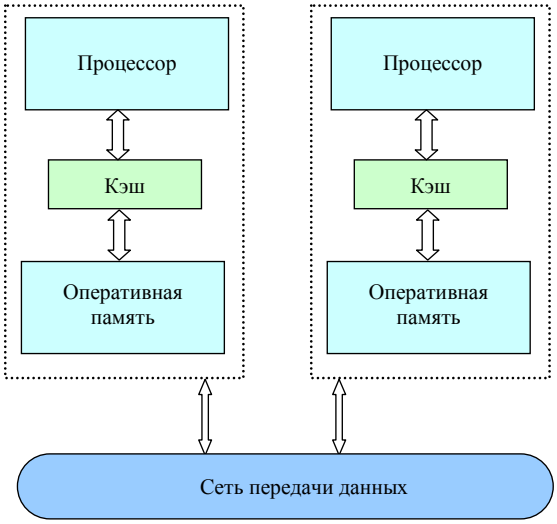


Рис. 1.6. Архитектура многопроцессорных систем с распределенной памятью

Отдельные исследователи обращают особое внимание на отличие понятия кластера от *сети компьютеров* (*network of workstations* or *NOW*). Для построения локальной компьютерной сети, как правило, применяют более простые сети передачи данных (порядка 100 Мбит/сек). Компьютеры сети обычно более рассредоточены и могут быть использованы пользователями для выполнения каких-либо дополнительных работ.

В завершение обсуждаемой темы можно отметить, что существуют и другие способы классификации вычислительных систем (достаточно полный обзор подходов представлен в Hockney and Jesshope (1988), Patterson and Hennessy (1996), Воеводин В.В. и Воеводин Вл.В. (2002), см. также материалы сайта http://www.parallel.ru/computers/taxonomy/). При рассмотрении данной темы параллельных вычислений рекомендуется обратить внимание на способ структурной нотации для описания архитектуры ЭВМ, позволяющий с высокой степенью точности описать многие характерные особенности компьютерных систем.

1.4. Характеристика типовых схем коммуникации в многопроцессорных вычислительных системах

При организации параллельных вычислений в мультикомпьютерах для организации взаимодействия, синхронизации и взаимоисключения параллельно выполняемых процессов используется передача данных между процессорами вычислительной среды. Временные задержки при передаче данных по линиям связи могут оказаться существенными (по сравнению с быстродействием процессоров) и, как результат, *коммуникационная трудоемкость* алгоритма оказывает существенное влияние на выбор параллельных способов решения задач.

1.4.1. Примеры топологий сети передачи данных

Структура линий коммутации между процессорами вычислительной системы (*топология сети передачи данных*) определяется, как правило, с учетом возможностей эффективной технической реализации. Немаловажную роль при выборе структуры сети играет и анализ интенсивности информационных потоков при параллельном решении наиболее распространенных вычислительных задач. К числу типовых топологий обычно относят следующие схемы коммуникации процессоров (см. рис. 1.7):

- **Полный граф** (*completely-connected graph* or *clique*)– система, в которой между любой парой процессоров существует прямая линия связи; как результат, данная топология обеспечивает минимальные затраты при передаче данных, однако является сложно реализуемой при большом количестве процессоров;

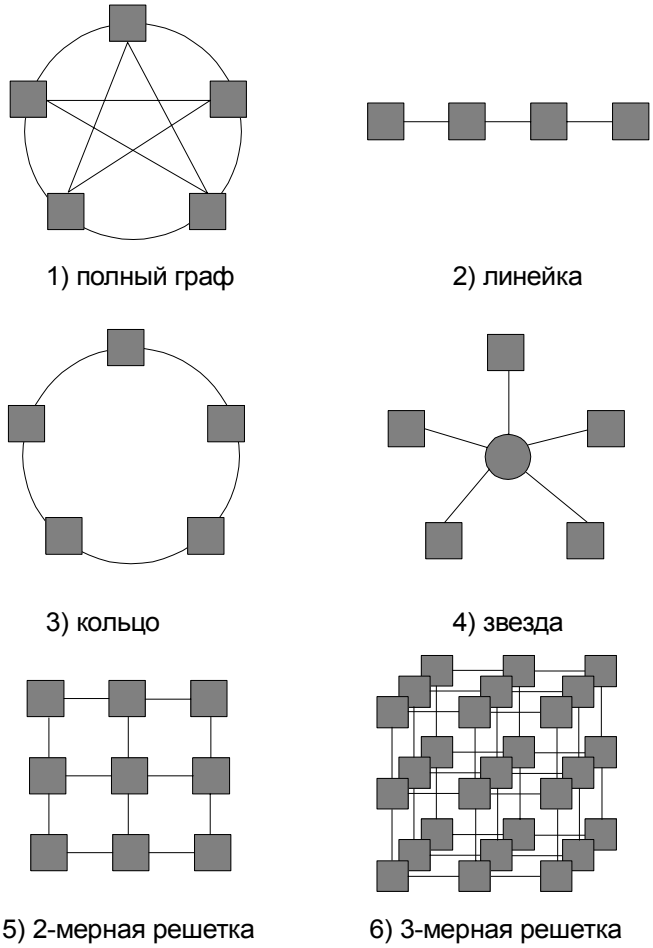


Рис. 1.7. Примеры топологий многопроцессорных вычислительных систем

- **Линейка** (*linear array* or *farm*) – система, в которой все процессоры перенумерованы по порядку и каждый процессор, кроме первого и последнего, имеет линии связи только с двумя соседними (с предыдущим и последующим) процессорами; такая схема является, с одной стороны, просто реализуемой, а с другой стороны, соответствует структуре передачи данных при решении многих вычислительных задач (например, при организации конвейерных вычислений);
- **Кольцо** (*ring*) – данная топология получается из линейки процессоров соединением первого и последнего процессоров линейки;
- **Звезда** (*star*) – система, в которой все процессоры имеют линии связи с некоторым управляющим процессором; данная топология является эффективной, например, при организации централизованных схем параллельных вычислений;
- **Решетка** (mesh) – система, в которой граф линий связи образует прямоугольную сетку (обычно двух - или трехмерную); подобная топология может быть достаточно просто реализована и, кроме того, может быть эффективно использована при параллельном выполнении многих численных алгоритмов (например, при реализации методов анализа математических моделей, описываемых дифференциальными уравнениями в частных производных);
- **Гиперкуб** (*hypercube*) – данная топология представляет частный случай структуры решетки, когда по каждой размерности сетки имеется только два процессора (т.е. гиперкуб содержит 2^N процессоров при размерности N); данный вариант организации сети передачи данных достаточно широко распространен в практике и характеризуется следующим рядом отличительных признаков:
  - два процессора имеют соединение, если двоичные представления их номеров имеют только одну различающуюся позицию;
  - в N - мерном гиперкубе каждый процессор связан ровно с N соседями;
  - N -мерный гиперкуб может быть разделен на два $(N-1)$ -мерных гиперкуба (всего возможно N различных таких разбиений);

– кратчайший путь между двумя любыми процессорами имеет длину, совпадающую с количеством различающихся битовых значений в номерах процессоров (данная величина известна как *расстояние Хэмминга*).

Дополнительная информация по топологиям многопроцессорных вычислительных систем может быть получена, например, в Hockney and Jesshope (1988), Patterson and Hennessy (1996), Culler and Singh (1998), Xu and Hwang (1998), Buyya (1999), Корнеев (1999), Воеводин В.В. и Воеводин Вл.В. (2002). При рассмотрении вопроса следует учесть, что схема линий передачи данных может реализовываться на аппаратном уровне, а может быть обеспечена на основе имеющейся физической топологии при помощи соответствующего программного обеспечения. Введение *виртуальных* (программно-реализуемых) топологий способствует мобильности разрабатываемых параллельных программ и снижает затраты на программирование.

1.4.2. Топология сети вычислительных кластеров

Для построения кластерной системы во многих случаях используют *коммутатор* (*switch*), через который процессоры кластера соединяются между собой. В этом случае топология сети кластера представляет собой полный граф (рис. 1.7), в соответствии с которым передача данных может быть организована между любыми двумя процессорами сети. При этом, однако, одновременность выполнения нескольких коммуникационных операций является ограниченной – *в любой момент времени каждый процессор может принимать участие только в одной операции приема-передачи данных*. Как результат, параллельно могут выполняться только те коммуникационные операции, в которых взаимодействующие пары процессоров не пересекаются между собой.

1.4.3. Характеристики топологии сети

В качестве основных характеристик топологии сети передачи данных наиболее широко используется следующий ряд показателей:

- *Диаметр* – показатель, определяемый как максимальное расстояние между двумя процессорами сети (под расстоянием обычно понимается величина кратчайшего пути между процессорами); данная величина может характеризовать максимально-необходимое время для передачи данных между процессорами, поскольку время передачи обычно прямо пропорционально длине пути;
- *Связность* (*connectivity*) – показатель, характеризующий наличие разных маршрутов передачи данных между процессорами сети; конкретный вид данного показателя может быть определен, например, как минимальное количество дуг, которое надо удалить для разделения сети передачи данных на две несвязные области;
- *Ширина бинарного деления* (*bisection width*) – показатель, определяемый как минимальное количество дуг, которое надо удалить для разделения сети передачи данных на две несвязные области одинакового размера;
- *Стоимость* – показатель, который может быть определен, например, как общее количество линий передачи данных в многопроцессорной вычислительной системе.

Для сравнения в таблице 1.1 приводятся значения перечисленных показателей для различных топологий сети передачи данных.

Таблица 1.1. Характеристики топологий сети передачи данных
(p – количество процессоров)

Топология	**Диаметр**	**Ширина бисекции**	**Связность**	**Стоимость**
Полный граф	1	$p^2/4$	p-1	p(p-1)/2
Звезда	2	1	1	p-1
Полное двоичное дерево	2log((p+1)/2)	1	1	p-1
Линейка	p-1	1	1	p-1
Кольцо	$\lfloor p/2 \rfloor$	2	2	p

Решетка N=2	$2(\sqrt{p}-1)$	$\sqrt{p}$	2	2(p-$\sqrt{p}$)
Решетка-тор N=2	$2\lfloor\sqrt{p}/2\rfloor$	$2\sqrt{p}$	4	2p
Гиперкуб	log p	p/2	log p	(p log p)/2

1.5. Характеристика системных платформ для построения кластеров

(находится в разработке)

1.6. Краткий обзор раздела

В разделе приводится общая характеристика способов организации параллельных вычислений и дается различие между многозадачным, параллельным и распределенным режимами выполнения программ. Для демонстрации возможных подходов рассматривается ряд примеров параллельных вычислительных систем. На основе данного рассмотрения отмечается существенное разнообразие вариантов построения параллельных систем.

Многообразие компьютерных вычислительных систем приводит к необходимости их классификации. В разделе дается описание одного из наиболее известных способов – *систематики Флинна*, в основу которой положено понятие потоков команд и данных. Данная классификация систем является достаточно простой и понятной, однако, в рамках такого подхода почти все многопроцессорные вычислительные системы попадают в одну группу – *класс MIMD*. С целью дальнейшего разделения возможных типов систем в разделе приводится также широко используемая структуризация класса многопроцессорных вычислительных систем, что позволяет выделить две важных группы систем с общей разделяемой и распределенной памятью – *мультипроцессоры* и *мультикомпьютеры*. Наиболее известные примеры систем первой группы - *векторные параллельные процессоры (parallel vector processor or PVP)* и *симметричные мультипроцессоры (symmetric multiprocessor or SMP)*. К мультикомпьютерам относятся *массивно-параллельные системы (massively parallel processor or MPP) и кластеры (clusters)*.

Далее в разделе обращается внимание на характеристику сетей передачи данных в многопроцессорных вычислительных системах. Приводятся примеры топологий сетей, отмечаются особенности организации сетей передачи данных в кластерах и обсуждаются параметры топологий, существенно влияющие на коммуникационную сложность методов параллельных вычислений.

В завершение раздела дается общая характеристика системных платформ для построения кластеров.

1.7. Обзор литературы

Дополнительная информация об архитектуре параллельных вычислительных систем может быть получена, например**,** Hockney and Jesshope **(**1988**),** Patterson and Hennessy (1996), Culler, Singh and Gupta (1998), Корнеев (1999), Воеводин В.В. и Воеводин Вл.В. (2002), Таненбаум (2002)**;** полезная информация содержится также в Xu and Hwang (1998), Buyya (1999).

В качестве обзора возможных топологий сетей передачи данных в многопроцессорных системах и технологий для их реализации может быть рекомендована, например, работа Dally and Towles, B.P. (2003).

Подробное рассмотрение вопросов, связанных с построением и использованием кластерных вычислительных систем, проводится в Xu and Hwang (1998), Buyya (1999). Практические рекомендации по построению кластеров для разных систем платформ могут быть найдены в Sterling (2001, 2002).

1.8. Контрольные вопросы

1. В чем заключаются основные способы достижения параллелизма?
2. В чем могут состоять различия параллельных вычислительных систем?
3. Что положено в основу классификация Флинна?
4. В чем состоит принцип разделения многопроцессорных систем на мультипроцессоры и мультикомпьютеры?
5. Какие классы систем известны для мультипроцессоров?

6. В чем состоят положительные и отрицательные стороны симметричных мультипроцессоров?

7. Какие классы систем известны для мультикомпьютеров?

8. В чем состоят положительные и отрицательные стороны кластерных систем?

9. Какие топологии сетей передачи данных наиболее широко используются при построении многопроцессорных систем?

10. В чем состоят особенности сетей передачи данных для кластеров?

11. Каковы основные характеристики сетей передачи данных?

12. Какие системные платформы могут быть использованы для построения кластеров?

1.9. Задачи и упражнения

1. Приведите дополнительные примеры параллельных вычислительных систем.

2. Выполните рассмотрение дополнительных способов классификации компьютерных систем.

3. Рассмотрите способы обеспечения когерентности кэшей в системах с общей разделяемой памятью.

4. Подготовьте обзор программных библиотек, обеспечивающих выполнение операций передачи данных для систем с распределенной памятью.

5. Рассмотрите топологию сети передачи данных в виде двоичного дерева.

6. Выделите эффективно реализуемые классы задач для каждого типа топологий сети передачи данных.

Литература

Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2002.

Корнеев В.В. Параллельные вычислительные системы. – М.: Нолидж, 1999.

Таненбаум Э. (2002) . Архитектура компьютера. – СПб.: Питер.

Barker, M. (Ed.) (2000). Cluster Computing Whitepaper at http://www.dcs.port.ac.uk/~mab/tfcc/WhitePaper/.

Buyya, R. (Ed.) (1999). High Performance Cluster Computing. Volume1: Architectures and Systems. Volume 2: Programming and Applications. - Prentice Hall PTR, Prentice-Hall Inc.

Culler, D., Singh, J.P., Gupta, A. (1998) Parallel Computer Architecture: A Hardware/Software Approach. - Morgan Kaufmann.

Dally, W.J., Towles, B.P. (2003). Principles and Practices of Interconnection Networks. - Morgan Kaufmann.

Flynn, M.J. (1966) Very high-speed computing systems. Proceedings of the IEEE 54(12): P. 1901-1909.

Hockney, R. W., Jesshope, C.R. (1988). Parallel Computers 2. Architecture, Programming and Algorithms. - Adam Hilger, Bristol and Philadelphia. (русский перевод 1 издания: Хокни Р., Джессхоуп К. Параллельные ЭВМ. Архитектура, программирование и алгоритмы. - М.: Радио и связь, 1986)

Kumar V., Grama A., Gupta A., Karypis G. (1994). Introduction to Parallel Computing. - The Benjamin/Cummings Publishing Company, Inc. (2nd edn., 2003)

Kung, H.T. (1982). Why Systolic Architecture? Computer 15 № 1. P. 37-46.

Patterson, D.A., Hennessy J.L. (1996). Computer Architecture: A Quantitative Approach. 2d ed. - San Francisco: Morgan Kaufmann.

Pfister, G. P. (1995). In Search of Clusters. - Prentice Hall PTR, Upper Saddle River, NJ (2nd edn., 1998).

Sterling, T. (ed.) (2001). Beowulf Cluster Computing with Windows. - Cambridge, MA: The MIT Press.

Sterling, T. (ed.) (2002). Beowulf Cluster Computing with Linux. - Cambridge, MA: The MIT Press.

Tanenbaum, A. (2001). Modern Operating System. 2nd edn. – Prentice Hall (русский перевод Таненбаум Э. Современные операционные системы. – СПб.: Питер, 2002)

Xu, Z., Hwang, K. (1998). Scalable Parallel Computing Technology, Architecture, Programming. – Boston: McGraw-Hill.