

## Сетевые и распределенные операционные системы

Сетевые и распределенные системы являются принципиально различными.

В сетевых операционных системах для того, чтобы задействовать ресурсы другого сетевого компьютера, пользователи должны знать о его наличии и уметь это сделать. Каждая машина в сети работает под управлением своей локальной операционной системы, отличающейся от операционной системы автономного компьютера наличием дополнительных сетевых средств (программной поддержкой для сетевых интерфейсных устройств и доступа к удаленным ресурсам), но эти дополнения существенно не меняют структуру операционной системы.

Распределенная система, напротив, внешне выглядит как обычная автономная система. Пользователь не знает и не должен знать, где его файлы хранятся, на локальной или удаленной машине, и где его программы выполняются. Он может вообще не знать, подключен ли его компьютер к сети. Внутреннее строение распределенной операционной системы имеет существенные отличия от автономных систем.

Все перечисленные выше цели объединения компьютеров в вычислительные сети не могут быть достигнуты без организации взаимодействия процессов на различных вычислительных системах. Будь то доступ к разделяемым ресурсам или общение пользователей через сеть – в основе всего этого лежит взаимодействие удаленных процессов, т. е. процессов, которые находятся под управлением физически разных операционных систем. Поэтому мы в своей работе сосредоточимся именно на вопросах кооперации таких процессов, в первую очередь выделив ее отличия от кооперации процессов в одной автономной вычислительной системе (кооперации локальных процессов), о которой мы говорили в лекциях 4, 5 и 6.

Изучая взаимодействие локальных процессов, мы разделили средства обмена информацией по объему передаваемых между ними данных и возможности влияния на поведение другого процесса на три категории: сигнальные, каналные и разделяемая память. На самом деле во всей этой систематизации присутствовала некоторая доля лукавства. Мы фактически классифицировали средства связи по виду интерфейса обращения к ним, в то время как реальной физической основой для всех средств связи в том или ином виде являлось разделение памяти. Семафоры представляют собой просто целочисленные переменные, лежащие в разделяемой памяти, к которым посредством системных вызовов, определяющих состав и содержание допустимых операций над ними, могут обращаться различные процессы. Очереди сообщений и рп'ы базируются на буферах ядра операционной системы, которые опять-таки с помощью системных вызовов доступны различным процессам. Иного способа реально передать информацию от процесса к процессу в автономной вычислительной системе просто не существует. Взаимодействие удаленных процессов принципиально отличается от ранее рассмотренных случаев. Общей памяти у различных компьютеров физически нет. Удаленные процессы могут обмениваться информацией, только передавая друг другу пакеты данных определенного формата (в виде последовательностей электрических или электромагнитных сигналов, включая световые) через некоторый физический канал связи или несколько таких каналов, соединяющих компьютеры. Поэтому в основе всех средств взаимодействия удаленных процессов лежит передача структурированных пакетов информации или сообщений.

При взаимодействии локальных процессов и процесс–отправитель информации, и процесс–получатель функционируют под управлением одной и той же операционной системы. Эта же операционная система поддерживает и функционирование промежуточных накопителей данных при использовании не прямой адресации. Для организации взаимодействия процессы пользуются одними и теми же системными вызовами, присущими данной операционной системе, с одинаковыми интерфейсами. Более того, в автономной операционной системе передача информации от одного процесса к другому, независимо от используемого способа адресации, как правило (за исключением микроядерных операционных систем), происходит напрямую – без

участия других процессов-посредников. Но даже и при наличии процессов-посредников все участники передачи информации находятся под управлением одной и той же операционной системы. При организации сети, конечно, можно обеспечить прямую связь между всеми вычислительными комплексами, соединив каждый из них со всеми остальными посредством прямых физических линий связи или подключив все комплексы к общей шине (по примеру шин данных и адреса в компьютере). Однако такая сетевая топология не всегда возможна по ряду физических и финансовых причин. Поэтому во многих случаях информация между удаленными процессами в сети передается не напрямую, а через ряд процессов-посредников, "обитающих" на вычислительных комплексах, не являющихся компьютерами отправителя и получателя и работающих под управлением собственных операционных систем. Однако и при отсутствии процессов-посредников удаленные процесс-отправитель и процесс-получатель функционируют под управлением различных операционных систем, часто имеющих принципиально разное строение.

Вопросы надежности средств связи и способы ее реализации, рассмотренные нами в лекции 4, носили для случая локальных процессов скорее теоретический характер. Мы выяснили, что физической основой "общения" процессов на автономной вычислительной машине является разделяемая память. Поэтому для локальных процессов надежность передачи информации определяется надежностью ее передачи по шине данных и хранения в памяти машины, а также корректностью работы операционной системы. Для хороших вычислительных комплексов и операционных систем мы могли забыть про возможную ненадежность средств связи. Для удаленных процессов вопросы, связанные с надежностью передачи данных, становятся куда более значимыми. Протяженные сетевые линии связи подвержены разнообразным физическим воздействиям, приводящим к искажению передаваемых по ним физических сигналов (помехи в эфире) или к полному отказу линий (мыши съели кабель). Даже при отсутствии внешних помех передаваемый сигнал затухает по мере удаления от точки отправления, приближаясь по интенсивности к внутренним шумам линий связи. Промежуточные вычислительные комплексы сети, участвующие в доставке информации, не застрахованы от повреждений или внезапной перезагрузки операционной системы. Поэтому вычислительные сети должны организовываться исходя из предпосылок ненадежности доставки физических пакетов информации.

При организации взаимодействия локальных процессов каждый процесс (в случае прямой адресации) и каждый промежуточный объект для накопления данных (в случае не прямой адресации) должны были иметь уникальные идентификаторы – адреса – в рамках одной операционной системы. При организации взаимодействия удаленных процессов участники этого взаимодействия должны иметь уникальные адреса уже в рамках всей сети.

Физическая линия связи, соединяющая несколько вычислительных комплексов, является разделяемым ресурсом для всех процессов комплексов, которые хотят ее использовать. Если два процесса попытаются одновременно передать пакеты информации по одной и той же линии, то в результате интерференции физических сигналов, представляющих эти пакеты, произойдет взаимное искажение передаваемых данных. Для того чтобы избежать возникновения такой ситуации (race condition!) и обеспечить эффективную совместную работу вычислительных систем, должны выполняться условия взаимоисключения, прогресса и ограниченного ожидания при использовании общей линии связи, но уже не на уровне отдельных процессов операционных систем, а на уровне различных вычислительных комплексов в целом.

Распределенные компьютерные системы и особенности их ОС

В распределенной системе (distributed system) вычисления распределены между несколькими физическими процессорами (компьютерами), объединенными между собой в сеть.

Слабо связанная система (loosely coupled system) – распределенная компьютерная система, в которой каждый процессор имеет свою локальную память, а различные процессоры

взаимодействуют между собой через линии связи – высокоскоростные шины, телефонные линии, беспроводную связь (Wi-Fi, EVDO, Wi-Max и др.).

Преимущества распределенных систем:

Разделение (совместное использование) ресурсов: в распределенной системе различные ресурсы могут храниться на разных компьютерах. Нет необходимости дублировать программы или данные, храня их копии на нескольких компьютерах.

Совместная загрузка (load sharing ): каждому компьютеру в распределенной системе может быть поручено определенное задание, которое он выполняет параллельно с выполнением другими компьютерами своих заданий.

Надежность: при отказе или сбое одного из компьютеров распределенной системы его задание может быть перераспределено другому компьютеру, чтобы сбой в минимальной степени повлиял или вовсе не повлиял на итоговый результат.

Связь: в распределенной системе все компьютеры связаны друг с другом, так что, например, при необходимости возможен удаленный вход с одного компьютера на другой с целью использования ресурсов более мощного компьютера.

В распределенной системе компьютеры связаны в сетевую инфраструктуру, которая может быть:

локальной сетью (local area network - LAN);

глобальной или региональной сетью (wide area network - WAN).

По своей организации распределенные системы могут быть клиент-серверными (client-server) или одноранговыми (peer-to-peer) системами. В клиент-серверной системе определенные компьютеры играют роль серверов, а остальные – роль клиентов, пользующихся их услугами. Подобная организация распределенных систем наиболее распространена, и мы рассмотрим ее подробнее. В одноранговой распределенной системе все компьютеры равноправны.

Структура клиент-серверной системы изображена на рис. 3.2.

Структура клиент-серверной системы

Рис. 3.2. Структура клиент-серверной системы

Виды серверов в клиент-серверных компьютерных системах

Клиент-серверная архитектура распределенных систем весьма широко распространена и поддерживается операционными системами. Поэтому очень важно знать, какие виды и функции серверов предлагают современные распределенные системы.

Файл-сервер (file server) – компьютер и программное обеспечение, предоставляющие доступ к подмножеству файловых систем, расположенных на дисках компьютера-сервера, другим компьютерам локальной сети (LAN). Пример – серверное программное обеспечение SAMBA (SMB – сокращение от Server Message Block) для ОС типа UNIX (Linux, FreeBSD, Solaris и т.д.), обеспечивающее доступ с Windows-компьютеров локальной сети к файловым системам UNIX-машин. Samba также реализована для платформы Macintosh / MacOS.

Сервер приложений (application server) – компьютер и программное обеспечение, предоставляющее вычислительные ресурсы (память и процессор) и необходимое окружение для удаленного запуска определенных классов (как правило, больших) приложений с других компьютеров локальной сети. Примеры серверов приложений - WebSphere (IBM), WebLogic (BEA) – наилучшие из известных серверов приложений, работающих в Java Enterprise Edition (JEE).

Сервер баз данных (database server) – компьютер и программное обеспечение, предоставляющее доступ другим компьютерам сети к базам данных, расположенным на компьютере-сервере. Пример: серверное программное обеспечение для доступа к базам данных Microsoft SQL Server.

Веб-сервер (Web server) – компьютер и программное обеспечение, предоставляющее доступ клиентам через WWW к Web-страницам, расположенным на компьютере-сервере. Пример: свободно распространяемый Web-сервер Apache.

Прокси-сервер – компьютер и программное обеспечение, являющиеся частью локальной сети и поддерживающие эффективное обращение компьютеров локальной сети к Интернету, фильтрацию трафика, защиту от внешних атак. Прокси-сервер обычно встроен в операционную систему.

Сервер электронной почты – компьютер и программное обеспечение, выполняющие отправку, получение и "раскладку" электронной почты для компьютеров некоторой локальной сети. Могут обеспечивать также шифрование почты (email encryption) – шифрование электронных писем перед отправкой адресатам из определенного сетевого домена (как правило, заказчику) и их дешифровку после получения от заказчика.

Серверный бэк-энд (Server back-end) – группа (пул) связанных в локальную сеть серверных компьютеров, используемых вместо одного сервера, в целях большей надежности и предоставления большего объема ресурсов. Другой термин, близкий к этому, - центр обработки данных (data center). Эти понятия особенно актуальны в связи со все более широким распространением облачных вычислений, являющихся, с этой точки зрения, наиболее современной реализацией клиент-серверной схемы взаимодействия.

Кластерные вычислительные системы и их ОС

Компьютерные кластеры весьма популярны для научных вычислений. Компьютеры в кластере, как правило, связаны между собой через быструю локальную сеть. Кластеризация позволяет двум или более системам использовать общую память. Кластеризация обеспечивает высокую надежность. Различают компьютерные кластеры двух видов:

асимметричная кластеризация (asymmetric clustering) – организация компьютерного кластера, при которой один компьютер выполняет приложение, а остальные простаивают;

симметричная кластеризация (symmetric clustering) - организация компьютерного кластера, при которой все машины кластера исполняют одновременно различные части одного большого приложения.

Различают также:

кластеры с высокоскоростным доступом (high-availability clusters) – компьютерные кластеры, обеспечивающие оптимальный доступ к ресурсам, предоставляемым компьютерами кластера, например, к базам данных;

кластеры с балансировкой загрузки (load-balancing clusters) – компьютерные кластеры, которые имеют несколько входных компьютеров, балансирующих запросы (front-ends), распределяющих задания между компьютерами серверного back-end'a (серверной фермы).

Кластеры часто используются в университетах (например, установлены на нескольких факультетах СПбГУ) и в исследовательских центрах (например, CERN, Швейцария). Операционные системы для кластеров: Windows 2003 for clusters; Windows 2008 High-Performance Computing.

#### Системы и ОС реального времени

Системы реального времени часто используются как управляющие устройства для специальных приложений, - например, для научных экспериментов; в медицинских системах, связанных с изображениями; системах управления в промышленности; системах отображения (display); системах управления космическими полетами, АЭС и др. Для таких систем характерно наличие и выполнение четко определенных временных ограничений (время реакции – response time; время наработки на отказ и др.).

Различаются системы реального времени видов hard real-time и soft real-time.

Hard real-time – системы – системы реального времени, в которых при нарушении временных ограничений может возникнуть критическая ошибка (отказ) управляемого ею объекта. Примеры: система управления двигателем автомобиля; система управления кардиостимулятором. В таких системах вторичная память ограничена или отсутствует; данные хранятся в оперативной памяти (RAM) или постоянном запоминающем устройстве (ПЗУ, ROM). При использовании таких систем возможны конфликты с системами разделения времени, не имеющие места для ОС общего назначения. Выражаясь более простым языком, при работе подобных систем не допускаются прерывания; все необходимые данные для основного цикла работы системы должны предварительно быть загружены в память; процесс, выполняющий код такой системы, не должен подвергаться откатке на диск. ОС для таких систем обычно упрощены, вместо виртуальной памяти выделяется физическая, все другие виды виртуализации ресурсов исключены. Популярной практикой разработки ОС реального времени является практика разработки таких ОС на основе открытых исходных кодов ОС общего назначения путем "отсечения всего лишнего". Однако при этом следует соблюдать осторожность. Автору приходилось консультировать разработчиков системы реального времени для "Эльбруса", которые использовали для своей системы низкоуровневую процедуру выделения физической памяти, но не учли ее возможных конфликтов с общей системой виртуальной памяти ОС Эльбрус; в результате выделяемая память иногда "портилась" ... в результате изменения связующей информации в списке областей свободной памяти, который использовался механизмом виртуальной памяти "Эльбруса".

Soft real-time – системы – системы реального времени, в которых нарушение временных ограничений не приводит к отказу управляемого ею объекта. Обычно это системы управления несколькими взаимосвязанными системами с постоянно изменяющейся ситуацией. Пример - система планирования рейсов на коммерческих авиалиниях. В случае какой-либо задержки в работе такой системы, в худшем случае, пассажирам некоторых рейсов придется немного подождать в аэропорту, но никаких фатальных последствий не будет. Подобные системы имеют ограниченную полезность для промышленных систем управления и в роботике. Они также полезны в современных приложениях (например, для мультимедиа и виртуальной реальности), требующих развитых возможностей ОС.