

Теория связи

Один для всех и все на одного

С развитием сетевых технологий вычислительные операции стали усложняться, понадобились новые методы решения распределенных задач. Одной из удачных реализаций подобного метода распределенной обработки является технология «клиент-сервер».

Структура модели

Слово «сервер» происходит от английского *serve* — «служить», «обслуживать». Таким образом, сервер — это программа, которая предоставляет определенный класс услуг. Соответственно, клиент — это другая программа, которая обращается за предоставлением услуги. Когда программа-сервер и программа-клиент работают на разных компьютерах, этими терминами обозначают также и сами компьютеры.

Общение клиента и сервера проходит несложно: программа-клиент, соединяясь с программой-сервером, использует в качестве базовых наиболее удобные из имеющихся протоколов, реализуя над

ними собственные протоколы общения. Транспортные протоколы наиболее эффективны, но неудобны для пересылки данных объемом свыше нескольких сотен байт, а представительные протоколы дают повышенные накладные расходы, зато наиболее комфортны для программиста и надежны в применении. Это можно сравнить, например, с ремонтом квартиры: владелец может сделать все как самостоятельно, что требует от него специальных знаний и навыков, так и поручить ремонт специализированной организации, что будет дороже, но надежнее (если, конечно, организация действительно профессиональная). »

» Протокол общения может быть стандартизован как FTP, HTTP, POP3 и другие открытые протоколы либо держаться в секрете, подобно закрытым протоколам ICQ. В первом случае имеется полная и точная документация на правила общения, что позволяет писать надежные программы, во втором это право резервирует для себя создатель протокола, а сторонние программисты могут лишь изучать его поведение на фактических примерах.

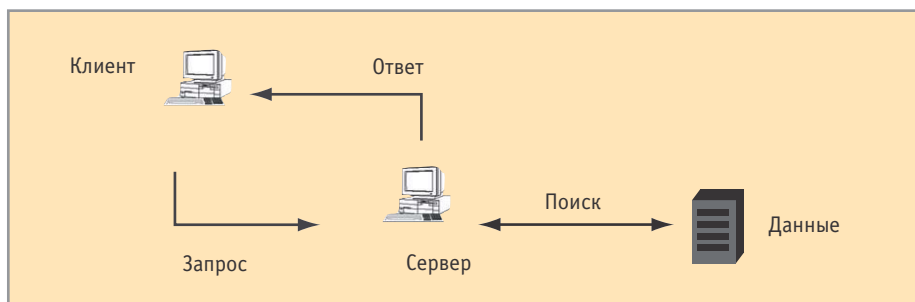
Взаимодействие клиента и сервера

Клиент может взаимодействовать с сервером посредством отдельных, не зависящих друг от друга обращений (транзакций), по методу «запрос-ответ» или же в рамках сеанса (сессии).

Независимые транзакции используются в тех случаях, когда для выполнения операции клиенту достаточно послать серверу один запрос, содержащий описание требуемой операции, и получить обратно один ответ, содержащий результат ее выполнения или причину отказа. После каждой такой транзакции сервер закрывает соединение с клиентом.

Сессия представляет собой виртуальное соединение, в течение которого между сервером и клиентом создается канал передачи, и они «помнят» друг о друге и выполняемых действиях. Сеансовое взаимодействие подобно заключению договора, оно налагает на клиента и сервер дополнительные обязательства, установленные для этого сеанса. Завершение сеанса приводит к разрыву отношений между сервером и клиентом и ликвидации их взаимных обязательств. Если одна сторона внезапно завершает сеанс, другая получает об этом уведомление и может выполнить соответствующую «чистку» (например, «откатить» последнюю операцию как незавершенную, вернув все в исходное состояние). Если связь разрывается по вине сети — уведомление получают обе стороны.

Сеансовое взаимодействие необходимо при двустороннем характере общения, когда требуются серия взаимных действий и обоюдная уверенность в успешном завершении процесса. Например, клиентская программа вкладчика банка связывается по его команде



▲ Клиент соединяется с сервером и просит найти информацию в базе данных; запрос обрабатывается, ответ идет клиенту через сервер

с сервером банка, чтобы выполнить перевод некой суммы на указанный счет. Сервер проверяет, достаточно ли денег на счете, верны ли реквизиты получателя, и просит подтверждения операции. Получив подтверждение, сервер выполняет перевод и сообщает о его результате, а клиент, соответственно, корректирует состояние счета в базе данных компьютера клиента. Если связь будет разорвана до завершения сеанса, возможна рассинхронизация баз данных сервера и клиента, которая может породить конфликт вкладчика с банком. Поэтому и сервер, и клиент в таких условиях ведут журналы всех сеансов и операций в них и при сбоях восстанавливают синхронизацию при следующем соединении.

Службы

Набор услуг, предоставляемых каким-либо сервером, называют службой (service). Службы имеют специальные названия и способы доступа к ним: функции операционной системы, стандартные номера портов в протоколах UDP/TCP, известные адреса интернет-узлов и т. п. Службы, разработанные для внутреннего применения, могут использовать любые доступные способы установки соединения и передачи информации.

Большинство служб использует в качестве базовых основные протоколы Интернета — UDP и TCP. Одиночные транзакции могут реализовываться как непосредственно на UDP, так и на протоколах более высокого уровня. Сеанс между сервером и клиентом обычно организуется с помощью протокола TCP, создающего между сервером и клиентом «виртуальный канал», по которому они пересылают друг другу произвольные потоки данных. В протоколе TCP уже реализованы средства установки,

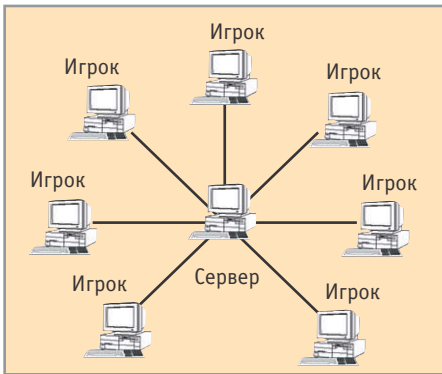
поддержания и завершения соединения. Сеанс может быть реализован и посредством протокола UDP, как это было сделано, например, в ранних версиях протокола ICQ, но в этом случае серверу и клиенту необходимо брать все специальные функции по поддержанию сеанса на себя.

В последнее время сетевые службы все чаще используют в качестве базового протокол HTTP. Первоначально он был предназначен для получения с серверов гипертекстовых страниц, однако впоследствии стал использоваться и в других приложениях. Базовым протоколом здесь служит TCP, а HTTP дополнительно вводит понятие страницы как объекта более высокого уровня по отношению к потоку данных. Соответственно, запрос клиента к серверу содержит идентификатор ресурса и ряд дополнительных данных, а сервер возвращает страницу на языке HTML/XML, содержащую информацию в структурированном виде.

Однако наряду с удобством применения и высокой распространенностью протокол HTTP обладает существенным недостатком: он предполагает текстовое представление всех данных, а для изображений и звука оно не подходит. Поэтому передача такой информации в рамках HTTP выполняется специальными методами с использованием потоковой модели базового TCP.

Программирование клиент-серверной модели

Программирование клиентов и серверов — более сложная задача, нежели создание локальных программ. Строго говоря, локальная программа тоже обращается за обслуживанием к различным службам операционной системы, но эти запросы чаще всего носят синхронный характер. Синхронность вы-»



▲ **Quake — первая компьютерная игра, использующая динамическое подключение и синхронизацию клиента с сервером**

» ражается в том, что управление передается из программы в ОС и возвращается только после завершения операции; обработка выполняется последовательно. Соответственно, службам ОС нет необходимости ждать, пока программа обратится с запросом, а программе — пока ОС ответит, потому что последовательный характер обработки определяется использованием единственного процессора.

Напротив, сервер и клиент — это всегда два независимых, асинхронных процесса, каждый из которых работает на собственном процессоре со своей скоростью и загрузкой. Даже в однопроцессорной системе процессы сервера и клиента следует считать истинно параллельными, поскольку ОС с вытесняющей многозадачностью может переключать процессор в произвольные моменты времени. Среду передачи данных (канал связи) между этими процессами уже нельзя считать абсолютно надежной.

При посылке запроса серверу программа-клиент не приостанавливается автоматически, как при последовательной обработке. Она может продолжать выполнять какие-то рутинные операции, ожидая, когда придет ответ от сервера, либо остановиться совсем, имитируя традиционную форму вызова подпрограммы. В любом случае есть вероятность, что ответ не придет никогда — сервер может зависнуть, а сетевые средства не всегда обнаруживают аварийные ситуации. Для такого случая должен быть предусмотрен тайм-аут — временной интервал, по истечении которого клиент отключается от сервера. Наряду с тайм-аутом желательно предус-

матривать в клиентской программе возможность прерывания затянувшейся операции самим пользователем.

Взаимодействие сервера и клиента в рамках распределенной сети можно сравнить с общением самостоятельных людей/организаций с различными службами, тоже самостоятельными. Даже если служба обязана принять запрос и дать ответ, взаимодействие может нарушаться по многим причинам, от некачественной связи до недобросовестности исполнителей. Поэтому программа, обращаясь по инициативе пользователя к какой-либо сетевой службе, должна предусматривать возможность любого развития событий и стараться решать большинство проблем самостоятельно, лишь в самых крайних случаях сообщая пользователю о неудаче.

Примеры реализации различных служб в Интернете

Регистрация и обновление программ

В последнее время все чаще внедряется технология автоматической регистрации и обновления программ, в которой разработчик создает сервер регистрации/обновления, а программа сама обращается к нему для получения уникального регистрационного кода, обновления баз данных, версии самой программы и т. п. Эта технология реализована в Windows 98/ME/2k/XP, программах ReGet, Antiviral Toolkit Pro и других. Кроме того, автоматически соединяясь с сервером регистрации, программа проходит проверку на легальность, и если оказывается, что она взломана, сервер отдает приказ клиенту перестать работать или изменить процесс (например, в пиратском ReGet после проверки программы некоторые скачанные файлы будут нулевого размера).

ICQ (интернет-пейджер)

Служба ICQ предназначена для обмена между пользователями (клиентами) короткими текстовыми сообщениями, изображениями и произвольными файлами. В отличие от обычной почтовой связи, ICQ предполагает сеансовое соединение клиента с сервером, в течение которого пользователь находится на связи. Первоначально система ICQ со-

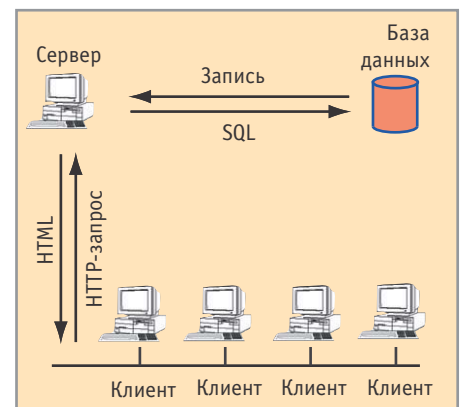
стояла только из центрального сервера icq.com (точнее — из набора, или кластера, однотипных серверов, который представляется клиенту как единый сервер). В современной системе клиентские программы могут также выступать серверами друг для друга — как при обычной пересылке сообщений и файлов, так и при организации общедоступных папок наподобие традиционных файл-серверов.

ICQ использует закрытые, не документированные в общедоступном виде протоколы, сохраняя таким образом свои права на все разработки. Однако большая часть протоколов ICQ изучена энтузиастами, в результате чего возникли такие альтернативные клиентские программы, как Miranda и Trillian.

Napster

(музыкальный p2p-сервис)

Это другой вид сообщества, ориентированный на обмен музыкой. Центральный сервер napster.com выступает лишь в роли главного сервера (координатора), обеспечивая клиентов информацией друг о друге. Любой пользователь при помощи клиентской программы может сообщить главному серверу о наличии у него какой-либо музыки в форматах MP3, WMA и тому подобных, сервер сохраняет эту информацию в своей базе данных, а клиентская программа становится при этом частным, дополнительным сервером, ожидающим запросов от других клиентов. Когда пользователь, работая со своей клиентской программой, формирует запрос на поиск музыки, его программа отправляет запрос главному серверу, по-»



▲ Система «клиент-сервер», построенная на протоколе HTTP с участием базы данных MySQL

лучая в ответ список частных серверов, предоставляющих запрошенные произведения. После этого программа связывается с одним из частных серверов, в роли которого выступает такая же клиентская программа, и скачивает файлы.

Подобным образом технология Napster эффективно обошла систему защиты авторских и имущественных прав на музыкальные произведения, ведь на главном сервере нет ни одного файла, распространение которого было бы запрещено законом, обмен происходит исключительно на личном уровне, а главный сервер выступает лишь в роли «справочного бюро». Тем не менее деятельность Napster была справедливо расценена как противоправная, и судебным решением служба была закрыта.

CDDB/FreeDB (Compact Disk DataBase)

Эта служба тоже предназначена для любителей музыки, однако в отличие от Napster она полностью легальна. Серверы CDDB/FreeDB работают с каталогами компакт-дисков, как CD-DA, так и CD-ROM. Клиентская программа, используя длительности дорожек (песен) диска, формирует 32-разрядный идентификатор диска, отправляя его в составе запроса на сервер, и получает в ответ либо описание диска из базы, либо сообщение об отсутствии диска с таким идентификатором. Описание диска включает сведения об авторах/исполнителях, названии альбома, годе его издания, жанре, а также полный список названий дорожек диска. Пользователь может также внести в ба-

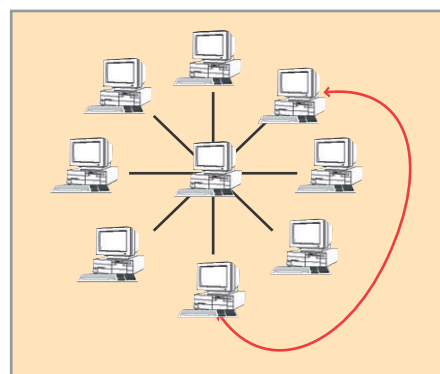
зу данных информацию об отсутствующих там дисках.

Службу CDDB представляет компания GraceNote (www.cddb.com, www.cddb.org, www.gracenote.com). Ее обслуживание вначале было бесплатным, сейчас требуется предварительная регистрация для получения платной лицензии на определенный срок, после истечения которого служба отказывает в доступе. Служба FreeDB (www.freedb.org) поддерживается группой энтузиастов и является бесплатной. FreeDB обеспечивает все основные функции CDDB, однако ее информационное наполнение пока несколько хуже.

WebMoney Transfer

Если Napster, организовав пользователям обмен музыкальными файлами, обходит ряд законов об авторских/имущественных правах, то служба WebMoney Transfer (www.webmoney.ru) выводит из-под государственного контроля ряд финансовых операций. С точки зрения закона любой страны эта служба выглядит безупречно: она всего лишь организует пользователям учет и обмен так называемых титульных знаков. Каждый пользователь может создавать кошельки, содержащие титульные знаки определенного типа: WMR — эквивалент российского рубля, WMZ — доллара США. Пользователи могут переводить друг другу знаки из своих кошельков и выполнять другие типично банковские операции с единственной разницей — сама служба WM к денежным средствам не имеет ровно никакого отношения.

Единственная функция службы WM — это учет знаков и обеспечение их движе-



▲ Один сервер может обслужить до сотен тысяч клиентов одновременно — хватило бы процессорных мощностей

ния. Для этого ее сервер имеет базу данных с информацией о пользователях и их кошельках, а пользователи соединяются с сервером посредством клиентской программы WM Keeper. Подобно программам «клиент-банк» WM Keeper общается с сервером по защищенному шифрованием RSA с длиной ключа 1024 бита каналу связи, что гарантирует полную конфиденциальность. В протоколах общения клиента и сервера предусмотрен также «откат» операции, прерванной из-за обрыва связи.

При работе с WM реальные деньги находятся исключительно на руках у конечных пользователей, служба лишь связывает пользователей друг с другом. Для покупки и обналичивания титульных знаков, а также для обмена «валют» существуют обменные пункты — независимые организации или частные лица, принимающие дензнаки у клиентов, но являющиеся, с точки зрения службы WM, ее рядовыми пользователями.

■ ■ ■ Евгений Музыченко

СТОИК

МЫ ЗНАЕМ КАК...

ИСПОЛЬЗОВАТЬ КОМПЬЮТЕР ДЛЯ
ОЦИФРОВКИ, ОБРАБОТКИ И
ВЫВОДА ВИДЕО ВСЕХ ФОРМАТОВ

- ПЛАТЫ ДЛЯ ОЦИФРОВКИ ВИДЕО
- АППАРАТНЫЕ MPEG-2/4 КОДЕРЫ
- УСТРОЙСТВА ЗАПИСИ DVD
- DV ↔ ВИДЕО КОНВЕРТОРЫ
- SVGA ↔ TV ПРЕОБРАЗОВАТЕЛИ
- ДОКУМЕНТ-КАМЕРЫ
- КОМПЬЮТЕРНЫЕ СИСТЕМЫ
ВИДЕОНАБЛЮДЕНИЯ/ОХРАНЫ

● КОМПЛЕКСЫ
ДЛЯ
МОНТАЖА
ВИДЕО
НА РС



PINNACLE SYSTEMS

НОВОЕ ПОКОЛЕНИЕ

EDITION

ДОСТАВКА ПО РОССИИ
ПРОГРАММА ДЛЯ ДИЛЕРОВ

www.stoik.ru ☎ Москва: 3669006 3664508 9628243 ☎ Н.Новгород: 438937