

Серверы баз данных

Термин "сервер баз данных" обычно используют для обозначения всей СУБД, основанной на архитектуре "клиент-сервер", включая и серверную, и клиентскую части. Такие системы предназначены для хранения и обеспечения доступа к базам данных.

Хотя обычно одна база данных целиком хранится в одном узле сети и поддерживается одним сервером, серверы баз данных представляют собой простое и дешевое приближение к распределенным базам данных, поскольку общая база данных доступна для всех пользователей локальной сети.

Принципы взаимодействия между клиентскими и серверными частями

Доступ к базе данных от прикладной программы или пользователя производится путем обращения к клиентской части системы. В качестве основного интерфейса между клиентской и серверной частями выступает язык баз данных SQL.

Это язык по сути дела представляет собой текущий стандарт интерфейса СУБД в открытых системах. Собирательное название SQL-сервер относится ко всем серверам баз данных, основанных на SQL. Соблюдая предосторожности при программировании, некоторые из которых были рассмотрены на предыдущих лекциях, можно создавать прикладные информационные системы, мобильные в классе SQL-серверов.

Серверы баз данных, интерфейс которых основан исключительно на языке SQL, обладают своими преимуществами и своими недостатками. Очевидное преимущество - стандартность интерфейса. В пределе, хотя пока это не совсем так, клиентские части любой SQL-ориентированной СУБД могли бы работать с любым SQL-сервером вне зависимости от того, кто его произвел.

Недостаток тоже довольно очевиден. При таком высоком уровне интерфейса между клиентской и серверной частями системы на стороне клиента работает слишком мало программ СУБД. Это нормально, если на стороне клиента используется маломощная рабочая станция. Но если клиентский компьютер обладает достаточной мощностью, то часто возникает желание возложить на него больше функций управления базами данных, разгрузив сервер, который является узким местом всей системы.

Одним из перспективных направлений СУБД является гибкое конфигурирование системы, при котором распределение функций между клиентской и пользовательской частями СУБД определяется при установке системы.

Преимущества протоколов удаленного вызова процедур

Упомянувшиеся выше протоколы удаленного вызова процедур особенно важны в системах управления базами данных, основанных на архитектуре "клиент-сервер".

Во-первых, использование механизма удаленных процедур позволяет действительно перераспределять функции между клиентской и серверной частями системы, поскольку в тексте программы удаленный вызов процедуры ничем не отличается от удаленного вызова, и следовательно, теоретически любой компонент системы может располагаться и на стороне сервера, и на стороне клиента.

Во-вторых, механизм удаленного вызова скрывает различия между взаимодействующими компьютерами. Физически неоднородная локальная сеть компьютеров приводится к логически однородной сети взаимодействующих программных компонентов. В результате пользователи не обязаны серьезно заботиться о разовой закупке совместимых серверов и рабочих станций.

Типичное разделение функций между клиентами и серверами

В типичном на сегодняшний день случае на стороне клиента СУБД работает только такое программное обеспечение, которое не имеет непосредственного доступа к базам данных, а обращается для этого к серверу с использованием языка SQL.

В некоторых случаях хотелось бы включить в состав клиентской части системы некоторые функции для работы с "локальным кэшем" базы данных, т.е. с той ее частью, которая интенсивно используется клиентской прикладной программой. В современной технологии это можно сделать только путем формального создания на стороне клиента локальной копии сервера базы данных и рассмотрения всей системы как набора взаимодействующих серверов.

С другой стороны, иногда хотелось бы перенести большую часть прикладной системы на сторону сервера, если разница в мощности клиентских рабочих станций и сервера чересчур велика. В общем-то при использовании RPC это сделать нетрудно. Но требуется, чтобы базовое программное обеспечение сервера действительно позволяло это. В частности, при использовании ОС UNIX проблемы практически не возникают.

Требования к аппаратным возможностям и базовому программному обеспечению клиентов и серверов

Из предыдущих рассуждений видно, что требования к аппаратуре и программному обеспечению клиентских и серверных компьютеров различаются в зависимости от вида использования системы.

Если разделение между клиентом и сервером достаточно жесткое (как в большинстве современных СУБД), то пользователям, работающим на рабочих станциях или персональных компьютерах, абсолютно все равно, какая аппаратура и операционная система работают на сервере, лишь бы он справлялся с возникающим потоком запросов.

Но если могут возникнуть потребности перераспределения функций между клиентом и сервером, то уже совсем не все равно, какие операционные системы используются.

Информация, данные, информационные объекты

В информатике есть принципиально три разные сущности:

1. Информация – это факты, сведения, команды и т.д.. Когда говорим об информации, имеем ввиду ответы на вопросы: о чем это? что означает, в чем смысл? чему соответствует в реальном мире? зачем это? и т.д. Плохая весть – это информация. Обратим внимание, что "плохая весть" не подразумевает обсуждения способа записи, структуры, какого-то синтаксиса, носителя и т.д.
2. Данные -- это представление информации с использованием какого-то формализма. Например, данные хорошей вести, представленная в RDF, или даже просто по-русски в KOI-8. Когда говорим о данных, подразумевается их структура и возможность разобраться (человеку или компьютеру). Данные абстрактны, т.е. не существуют в материальном мире. Действительно, одни и те же данные (например, текстовая строка "ABC" в UNICODE или небольшая база данных со сложной схемой) могут быть представлены на самых разных носителях -- оставаясь теми же самыми данными. Модели, описания -- это всё данные, ибо налицо использование формализмов (схемы данных, нотации и т.д.).
3. Информационные объекты Газеты, журналы, документы (бумажные и даже электронные -- файлы), базы данных -- это информационные объекты, т.е. физические объекты, содержащие данные. Эти данные можно прочесть. Одни и те же данные могут содержаться на разных информационных объектах (например, быть в газете и базе данных). Нужно понимать, что данные не живут без носителя - а носитель с данными всегда является физическим объектом: файл, например, это обычно маленькие магнитные частички, специальным образом ориентированные, расположенные во множестве разных мест. Ну, или какие-то части микросхем памяти, имеющие разные физические характеристики. Документы, хранилища данных -- это как раз информационные объекты.

Пример: начальник сказал "у нас есть база данных, учтите это в вашей работе" -- что имелось ввиду?

- а) есть информация -- можно позвонить по телефону, и оператор ответит на вопросы, обращаясь к каким-то своим записям то ли в компьютере, то ли в записной книжке
- б) есть данные: известна схема базы данных, и можно подумать о том, как запрашивать информацию из этой базы данных и маршрутизировать ответы в правильной кодировке нужным людям, используя нужные для них каналы передачи (кому твиттер, а кому письмом)
- в) есть носитель с базой данных (купленный на рынке DVD), при этом неведомо, в каком эта база данных формате, и как из нее вытащить хоть что-нибудь.
- г) и так далее -- число вариантов огромно, и чтобы их подробно обсуждать, как раз и нужно договориться о картине мира (онтологии) обработки данных, хотя бы между профессионалами -- архитекторами корпоративных информационных систем.