**Почему SQL "тормозит"?**

В последнее время в коференции fido.ru.1csoft стали появляться вопросы по клиент-серверным версиям (SQL) программ фирмы "1С". Аналогичные вопросы поступают к нам и по электронной почте и по телефону от клиентов нашей компании (Альтер Лого). Судя по содержанию этих вопросов (а также по содержанию ответов на них в фидошной конференции) могу с прискорбием констатировать, что люди, занимающиеся внедрением и эксплуатацией программных продуктов "1С" имеют весьма смутное представление о том, что такое клиент-серверная система, для чего она вообще нужна, как она работает и чего от нее можно и нужно ожидать, а чего нельзя.

Мимо такого непонимания я пройти спокойно не могу, поэтому беру на себя труд по сочинению этого текста, который, как я надеюсь, поможет многим из разработчиков и пользователей понять, нужен ли им SQL-продукт или нет.

Итак, первое, что необходимо понять и запомнить: SQL-системы не предназначены для ускорения выполнения выборок и печати отчетов. Если Вы ожидаете, что установив "1C-Торговлю для SQL" Вы ускорите работу своей системы, то Вы глубоко заблуждаетесь. Не будет она работать быстрее. Поэтому всякие разговоры о том, что "…SQL-Торговля - это тормоз…" абсолютно не имеют смысла. Несколько лет назад журнал PC Magazine проводил сравнительный анализ (в том числе и по быстродействию) систем управления базами данных, построенных на основе обычных файл серверов и с использованием клиент-серверных (SQL) систем. Естественно, условия испытаний по возможности были нивелированы, т.е. применялись одинаковые объемы баз данных, одинаковые их структуры, один и тот же компьютер в качестве сервера, одинаковое количество рабочих станций и т.д. Если мне не изменяет память, из клиент-серверных систем были испытаны Oracle, Interbase, Informix, Gupta и самый дешевый в то время Watcom SQL Server. Во всех случаях, средняя скорость выполнения запросов в SQL-системах была ниже, чем у файл-серверной системы (сейчас об этом эффекте можно прочесть в любой серьезной книге по SQL ). Испытатели не были удивлены полученным эффектом, поскольку были людьми грамотными и понимали причину такого поведения SQL-систем при заданных условиях эксперимента. Ведь задачей эксперимента было сравнение быстродействия двух видов систем и поэтому были выбраны условия, позволяющие произвести это сравнение. В частности для тестов использовались базы данных объемом 1.5-2Гб. Ведь если бы исследователи взялись проводить испытания, используя базы данных на порядок большего размера, то им просто не с чем было бы сравнивать SQL-варианты, поскольку обычная файл-серверная система на таких объемах информации просто заткнулась бы. Вот в этом то и состоит основное отличие и достоинство клиент-серверных систем: они будут работать со вполне приемлемой скоростью с базами данных такого объема, с которыми файл-серверная система работать просто не сможет ("не сможет" в том смысле, что ее функциональность, в том числе и быстродействие, станут неприемлемы для коммерческих приложений).

Посмотрите на обычный сетевой вариант 1С-Торговли. Она реализована на файл-серверном принципе, т.е. обработка данных ведется рабочей станцией, а сервер служит просто как дополнительное, доступное всем пользователям дисковое устройство. Это означает, что при выполнении задачи (например при сборке отчета) ВСЯ база данных (или значительная ее часть) прокачивается по сети на рабочую станцию и там обрабатывается процессором рабочей станции. Быстродействие такой системы зависит от быстродействия диска сервера, скорости передачи данных по сети, мощности процессора рабочей станции, объема ее ОЗУ и некоторых других факторов. Центральный процессор сервера играет второстепенную роль и должен просто обеспечивать передачу потока данных с сетевого канала на диск и обратно, по возможности не внося замедления в этот процесс. Главным в таком подходе является то, что практически вся база данных перегоняется по сети на рабочую станцию для последующей обработки. Если несколько станций одновременно выполняют сборку отчетов, то всем им закачивается база данных и естественно система "тормозит". Когда выполняются менее накладные операции, типа ввода нового документа, то объем перекачки данных значительно меньше, правда в реальности ввод документа как правило сопровождается поиском клиента в справочнике, вычислением задолженности клиента и т.п., что в свою очередь порождает перекачку большого количества информации с диска сервера на рабочую станцию. Не следует также забывать о необходимости синхронизации доступа рабочих станций к данным. Поскольку вся обработка ведется на уровне рабочих станций, а файл-сервер просто играет роль разделяемого дискового устройства, задачи синхронизации решаются в таких системах с помощью организации различных файлов блокировок (на диске файл-сервера) в которые каждая рабочая станция записывает информацию о данных, которые она модифицирует в данный момент, а при попытке считать данные проверяет не заняты ли эти данные другой рабочей станцией. Несмотря на ряд недостатков ("висячие" блокировки при аварийном выключении рабочих станций, "торможение" всей системы при модификации большого числа записей), способ вполне работоспособен.

Скорость работы такой системы прямо связана с объемом обрабатываемой базы данных. Они начинают ощутимо "тормозить", когда база данных достигает объема свыше 200-300Мб и при приближении к 1Гб практически просто перестает работать. Цифры конечно приблизительные и зависят от используемого программного обеспечения и формата базы данных. Например, при использовании формата таблиц базы данных Paradox торможение наступает значительно позже, чем при использовании формата DBase. Когда "торможение" ощутимо дает о себе знать, пользователи системы идут на различные хитрости: закрывают старую базу и открывают новую каждый квартал, пытаются удалить старые данные и т.п. Однако любой бухгалтер скажет Вам, что данные нужны ему не за квартал, а минимум за год и предпочтительно в динамике, а не в виде отдельных кусков. Да и долги клиентов иногда тянутся годами. Временным решением проблемы в такой ситуации может быть увеличение пропускной способности сети за счет установки 100-мегабитной сети вместо 10-мегабитной и интеллектуальных маршрутизаторов вместо тупых хабов. Однако, маршрутизаторы чрезвычайно дороги, а 100Мбит сеть даст повышение пропускной способности 2.5-3 раза (а не в 10 раз, как можно было бы ожидать!!!). Да и зачем увеличивать пропускную способность сети, если жесткий диск сервера уже работает на пределе своей производительности? Через пол-года Ваша база данных вырастет еще на 300-500Мб и система опять заткнется, пустив по ветру все вложенные в модернизацию денежки. Не следует забывать и еще об одной значительной детали. Это архивирование. Попробуйте заархивировать базу данных объемом 1-1.5Гб - за время потребное для архивации вы можете успеть пообедать, посмотреть кино и поругаться с начальником. А ведь это нужно делать каждый день. И при этом во время архивирования ни один из пользователей с базой работать не может.

Теперь рассмотрим SQL-систему ( т.е. клиент-серверную систему). Если кто-то говорит Вам, что у него сильно тормозит SQL - спросите его, какой у него сервер. Если он ответит, что-то вроде: "…Pentium 266, 64Мб ОЗУ и UDMA IDE", то можете смело сказать ему, что он … не слишком квалифицированный специалист. Серверы для SQL-систем должны иметь значительно большие ресурсы. PentiumII 300Mhz со 128Мб ОЗУ и Ultra Wide SCSI дисками - это пожалуй тот минимум, на котором может НОРМАЛЬНО функционировать программное обеспечение MS SQL Server с 5-8 подключенными клиентами. Заметьте при этом, что сеть с пропускной способностью 100Мбит совсем не обязательна. Все дело в том, что при работе с SQL-сервером рабочая станция не качает базу данных к себе по сети. Она просто передает по сети достаточно компактный запрос на сервер, который выполняет запрос, например производит выборку, и передает результат запроса обратно на рабочую станцию. Таким образом, трафик по сети значительно снижается, поскольку перекачки базы не происходит, а по сети передаются только запросы и результаты их выполнения. Конечно, если при разработке клиентской части программного обеспечения будет допущена ошибка и будет запрограммирован запрос, результатом выполнения которого является вся база данных или значительная ее часть, то вся эта информация будет качаться на рабочую станцию, выдавшую такой запрос. Но это уже по большей части лежит на совести разработчиков прикладных задач, т.е. применительно к программам "1С" - на совести тех, кто занимается настройками (и на совести разработчиков из "1С" - в части запрещения оптимизации таких запросов).

Теперь посмотрим, что же происходит с сервером в SQL-системе. Ведь север сам выполняет полученный запрос, поэтому, легко представить себе, что если с SQL-системой работают 10 пользователей, то для сервера это практически то же самое, как если бы на нем были одновременно запущены 10 экземпляров программы, с которой работают пользователи (например 10 локальных копий 1С-Торговли). Попробуйте запустить локально на каком-нибудь Pentium 200 или 266 десять экземпляров 1C-Торговли и выполнить одновременно 10 отчетов об остатках на складе. Сделав это и оценив результат Вы поймете, что всякие разговоры о "торможении" SQL-системы без обсуждения параметров сервера не имеют смысла. SQL-системы крайне прожорливы на ресурсы сервера. По-хорошему для таких систем вообще следует использовать компьютеры с RISC процессорами в многопроцессорном варианте (что-нибудь из продукции SUN Microsystems). Однако, для наших бедных клиентов с их ограниченными финансовыми возможностями SQL-системы все равно имеют существенное преимущество: даже при использовании сервера стоимостью 4-5 тыс. долларов они будут работать со вполне приемлемой скоростью с базами данных такого объема, на котором обычные файл-серверные системы просто заткнутся. Причем, этот объем может достигать величин в десятки ГИГАБАЙТ. Ведь доступ к данным очень быстр - они расположены здесь же, на диске сервера и не требуют передачи по сети для последующей обработки. Все запросы поступают к серверу, так что кроме обычного файлового кэширования есть огромные возможности по оптимизации выполнения запросов, их распараллеливанию. Эти возможности уже заложены в программное обеспечение MS SQL Server, а от разработчиков из "1С" требуется так написать исполняющую часть программы, чтобы использовать эти возможности в полной мере. Возможно также применение различных хитростей, например серверов-репликаторов (для разделения групп пользователей на тех, кто пользуется только отчетами, т.е работает в режиме "только чтение", и тех, кто активно модифицирует данные) или разноски данных по разным дисковым устройствам. При перегрузке дисковой системы она легко модернизируется, например с помощью RAID-массива (не забывайте однако, что SQL-система - это вообще иная ценовая категория как по цене матобеспечения, так и по цене железа). В результате при правильном проектировании системы (если не обнаружатся любители выдавать крутые запросы, результатом которых будет перекачка всей базы данных по сети), 100Мбит-ная сеть может не понадобиться. Вот что в первую очередь дает SQL-система. Вообще, SQL-система предоставляет значительные возможности для оптимизации аппаратной части и настройки программной части. И говорить о "торможении" SQL-системы имеет смысл только тогда, когда эти возможности исчерпаны.

До сих пор мы говорили о принципиальных различиях файл-серверных систем от клиент-серверных. Теперь немного о дополнительных преимуществах клиент-сервера.

Надежность. Клиент-серверные системы имеют встроенный механизм работы с транзакциями, в том числе и их отката. В файл-серверных версиях программ "1С" также имеется механизм работы с транзакциями, однако способ реализации их принципиально отличается. В файл-серверных версиях механизм транзакций представляет собой ни что иное, как просто блокировку всей базы данных до завершения выполнения критических по времени операций одной из рабочих станций. Откат возможен только при сохранении работоспособности рабочей станции, инициировавшей транзакцию. В клиент-серверной системе этот механизм (который реализуется программным обеспечением SQL-сервера - в нашем случае MS SQL Server 6.5) значительно более сложен. Он позволяет получить "слепок" базы данных на момент начала транзакции без блокировки базы данных. И слепков таких может быть много: для каждой рабочей станции - свой. В случае "зависания" рабочей станции, открывшей транзакцию, транзакция может быть просто откачена (т.е. база данных будет восстановлена в том виде, в каком она была до начала инициации транзакции). Откат осуществляется либо по запросу рабочей станции (при сохранении ее работоспособности), либо при перезагрузке рабочей станции, либо администратором SQL-системы. Таким образом, выход из строя рабочей станции не столь опасен для целостности базы данных. Кроме того, SQL-система ведет так называемый журнал транзакций. По сути база данных хранится в виде ее начального содержимого и ее модификаций записанных в журнал транзакций. Такой способ хранения позволяет производить архивирование базы данных во время работы всей системы: просто состояние базы данных фиксируется на момент начала архивирования, отсекаются незавершенные транзакции, а основная база и часть журнала транзакций, содержащая завершенные транзакции записываются в архив. Процесс архивирования легко поддается автоматизации, т.е. присутствие оператора необязательно - SQL-сервер имеет встроенные средства для этого.

Защита. В общем-то защищенность данных в значительной мере зависит от прикладной задачи (1С-Торговли или 1С-Расчета), однако по сравнению с обычным вариантом, в котором любой начинающий хакер спокойно может срубить пароль, в клиент-серверном варианте защита данных опирается на средства, предоставляемые SQL-сервером, что гораздо более недежно.

Гибкость применения. Системы на основе SQL-сервера позволяют построить сложные сетевые конфигурации со многими десятками и даже сотнями пользователей. При этом разработчику предоставляются широкие возможности по оптимизации системы, ее разделению по группам сложности и способам доступа. Серверы-репликаторы, например, дают прекрасный механизм для организации системы учета в крупной организации с разветвленной системой удаленных офисов, складов и т.п. При этом, работа на такой системе может вестись в реальном режиме времени, без перерывов для переноса и синхронизации данных - достаточно лишь организовать постоянные каналы связи 32-128 Кбит, что вполне осуществимо на наших телефонных линиях и не слишком дорого (конечно, в масштабе крупной компании).

Ну и наконец пора поговорить о недостатках. Недостатков у SQL-систем много - и крупных и мелких - тех же самых, которые присущи и файл-серверным системам. Однако есть два и весьма существенных.

Первый Вы наглядно увидите в прайсе фирмы "1С" - это цена. Цена не только программного обеспечения, но и цена железа на котором оно может функционировать и цена обслуживания. Ну, что же делать - SQL это продукт высоких технологий. А продукт высоких технологий всегда стоит дорого и должен эксплуатироваться грамотным персоналом, поэтому и системный администратор, способный грамотно работать с SQL-системой обойдется дороже, чем аналогичный специалист для обычной файл-серверной системы.

Второй - это существенные недоработки клиентской части программ комплекса 1C:Предприятие для SQL. В тех версиях, которые пока выпущены в продажу, далеко не полностью используются преимущества SQL сервера. Разработчик или поленился или не успел реализовать в полной мере возможности, предоставляемые средствами SQL сервера, поэтому во многих случаях там, где можно ожидать выполнения несложного SQL-запроса программа просто (как это делалось и в файл-серверной версии) перекачивает на локальный компьютер всю базу данных, вернее ее часть, и затем продолжает работу с локальной копией. Это с "внутренней" точки зрения, а внешне, для пользователя это выливается в те самые "тормоза" о которых пишут в конференции. Остается только надеяться, что со временем фирма "1С" доработает свои SQL-продукты таким образом, что функциональность SQL-сервера будет использоваться значительно более полно. В оправдание фирмы "1С" могу заметить, что программный продукт такой сложности всегда развивается постепенно - ведь если пытаться выпустить на рынок полностью "вылизанный" продукт высочайшего качества, то этого можно просто не дождаться. Все софтверные компании, выпускающие продукты высокой степени сложности идут именно по пути постепенного улучшения первоначально выпущенного продукта. Тот, кто пытается нарушить эту традицию обычно терпит крах на этом пути. Достаточно вспомнить историю с выпуском Windows 95 фирмой Microsoft: выпуск продукта все оттягивался и оттягивался, несмотря на многократные анонсы и переносы сроков, и в конце концов на рынок был выброшен недоработанный продукт. А уже потом были выпущены OSR1 и OSR2, однако продукт так и стался "сырым".

Теперь пора подвести некоторые итого. Не ждите, что SQL-системы будут работать быстро. Но они смогут работать там, где файл-серверный вариант не сможет, или же будет создавать существенные неудобства (удаленный доступ). Однако и обойдется это Вам не дешево.

Вот, пожалуй, и все. Надеюсь, что этот краткий обзор позволит Вам хоть немного сориентироваться в вопросе, что такое SQL и зачем он нужен.