БАЛТИЙСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ВОЕНМЕХ» имени Д. Ф. Устинова

Распределённые Информационные Системы.

Автор: Макаров Д.Ю.

Санкт-Петербург
2009 год

Холдинговая структура представляет собой организационную структуру, формирующуюся в результате присоединения предприятий различной отраслевой направленности, в которую включаются как предприятия, занимающиеся основным видом деятельности, предприятия, деятельность которых является вспомогательной по отношению к основному бизнесу (например, банк, страховая компания, служба безопасности), так и те, деятельность которых не затрагивает основную (например, профилакторий, поликлиника, сеть кинотеатров, казино и т.п.).

Основные задачи, решаемые в рамках проекта по ИТ-консалтингу на предприятии с холдинговой структурой:

- ликвидация информационных барьеров между подразделениями
- повышение управляемости
- повышение конкурентоспособности за счет использования новейших управленческих технологий

Рассмотрим эти моменты на примере программы “1С: Управление производственным предприятием 8”. “1С: Управление производственным предприятием 8” является комплексным прикладным решением, охватывающим основные контуры управления и учета на производственном предприятии. Решение позволяет организовать комплексную информационную систему, соответствующую корпоративным, российским и международным стандартам и обеспечивающую финансово-хозяйственную деятельность предприятия.

Общая концепция решения поясняется схемой:

*МСФО - международные стандарты финансовой отчетности.

Прикладное решение создает единое информационное пространство для отображения финансово-хозяйственной деятельности предприятия, охватывая основные бизнес-процессы. В то же время четко разграничивается доступ к хранимым сведениям, а также возможности тех или иных действий в зависимости от статуса работников.

На предприятиях холдинговой структуры общая информационная база может охватывать все организации, входящие в холдинг. Это существенно снижает трудоемкость ведения учета за счет повторного использования разными организациями общих массивов информации. При этом по всем организациям ведется сквозной управленческий и регламентированный (бухгалтерский и налоговый) учет, но регламентированная отчетность формируется раздельно по организациям.

Факт совершения хозяйственной операции регистрируется один раз и получает отражение в управленческом и регламентированном учете. Необходимость повторного ввода информации исключена. Средством регистрации хозяйственной операции является документ, причем для ускорения работы широко используются механизмы подстановки данных «по умолчанию», ввод новых документов на основании ранее введенных.

В прикладном решении принято следующее соотношение данных различных учетов:

- независимость данных управленческого, бухгалтерского и налогового учета;
- сопоставимость данных управленческого, бухгалтерского и налогового учета;
- совпадение суммовых и количественных оценок активов и обязательств по данным управленческого, бухгалтерского и налогового учета, при отсутствии объективных причин их расхождения.

Прикладное решение поставляется с комплектом интерфейсов, что обеспечивает каждому пользователю первоочередной доступ к нужным именно ему данным и механизмам прикладного решения.

Регламентированный (бухгалтерский и налоговый) учет по организациям ведется в национальной валюте, в том время как для управленческого учета по предприятию в целом может быть выбрана любая валюта. В разных организациях единой информационной базы могут использоваться разные системы налогообложения: в одних организациях — общая система налогообложения, в других — упрощенная; могут использоваться разные установки политики налогового и бухгалтерского учета. Кроме того, к отдельным видам деятельности организации может быть применена система налогообложения в виде единого налога на вмененный доход.

В дополнение к управленческому и регламентированному учетам можно вести учет по международным стандартам финансовой отчетности

(МСФО). С целью снижения трудоемкости учет по МСФО ведется не оперативно, с использованием трансляции (пересчета) данных других видов учета.

Базы данных и документов при этом могут быть собраны в одном месте (напр. на сервере основного предприятия) или распределены по серверам предприятий для более быстрого взаимодействия программы с данными для решения локальных операций. В обиходе СУБД, на основе которых создаются распределенные информационные системы, также характеризуют термином «Распределенные СУБД», и, соответственно, используют термин «Распределенные базы данных». При этом при создании распределённых информационных систем должны быть учтены основные требования к распределённым базам данных:

- прозрачность расположения данных для пользователя (иначе говоря, для пользователя распределенная база данных должна представляться и выглядеть точно так же, как и нераспределенная)
- изолированность пользователей друг от друга (пользователь должен «не чувствовать», «не видеть» работу других пользователей в тот момент, когда он изменяет, обновляет, удаляет данные)
- синхронизация и согласованность (непротиворечивость) состояния данных в любой момент времени
- локальная автономия (ни одна вычислительная установка для своего успешного функционирования не должна зависеть от любой другой установки)
- отсутствие центральной установки (следствие предыдущего пункта)
- независимость от местоположения (пользователю все равно где физически находятся данные, он работает так, как будто они находятся на его локальной установке)
- непрерывность функционирования (отсутствие плановых отключений системы в целом, например для подключения новой установки или обновления версии СУБД)
- независимость от фрагментации данных (как от горизонтальной фрагментации, когда различные группы записей одной таблицы размещены на различных установках или в различных локальных базах, так и от вертикальной фрагментации, когда различные поля-столбцы одной таблицы размещены на разных установках)
- независимость от реплицирования (дублирования) данных (когда какая-либо таблица базы данных, или ее часть физически может быть представлена несколькими копиями, расположенными на различных установках, причем «прозрачно» для пользователя), хотя в идеале такие случаи должны исключаться системой
- распределенная обработка запросов (оптимизация запросов должна носить распределенный характер — сначала глобальная оптимизация, а далее локальная оптимизация на каждой из задействованных установок)

- распределенное управление транзакциями (в распределенной системе отдельная транзакция может требовать выполнения действий на разных установках, транзакция считается завершенной, если она успешно завершена на всех вовлеченных установках)
- независимость от аппаратуры (желательно, чтобы система могла функционировать на установках, включающих компьютеры разных типов)
- независимость от типа операционной системы (система должна функционировать вне зависимости от возможного различия ОС на различных вычислительных установках)
- независимость от коммуникационной сети (возможность функционирования в разных коммуникационных средах)
- независимость от СУБД* (на разных установках могут функционировать СУБД различного типа, на практике ограничиваемые кругом СУБД, поддерживающих SQL)

* Данное свойство характеризуют также термином «интероперабельность».

При входе пользователя в распределенную систему ядро СУБД, идентифицируя пользователя, запускает запросы его ранее определенного и хранимого в базе данных представления и формирует ему «свое» видение базы данных, воспринимаемое пользователем как обычная (локальная) база данных. Так как представление базы данных виртуально, то «настоящие» данные физически находятся там, где они находились до формирования представления. При осуществлении пользователем манипуляций с данными ядро распределенной СУБД по системному каталогу базы данных само определяет, где находятся данные, вырабатывает стратегию действий, т. е. определяет, где, на каких установках целесообразнее производить операции, куда для этого и какие данные необходимо переместить из других установок или локальных баз данных, проверяет выполнение ограничений целостности данных. При этом большая часть таких операций прозрачна (т. е. невидима) для пользователя, и он воспринимает работу в распределенной базе данных, как в обычной локальной базе.

Несмотря на простоту и определенную изящность идеи «представлений», практическая реализация подобной технологии построения и функционирования распределенных систем встречает ряд серьезных проблем. Первая из них связана с размещением системного каталога базы данных, ибо при формировании для пользователя «представления» распределенной базы данных ядро СУБД в первую очередь должно «узнать», где и в каком виде в действительности находятся данные. Требование отсутствия центральной установки приводит к выводу о том, что системный каталог должен быть на любой локальной установке. Но тогда возникает проблема обновлений. Если какой-либо пользователь изменил данные или их структуру в системе, то эти изменения должны отразиться во

всех копиях системного каталога. Однако размножение обновлений системного каталога может встретить трудности в виде недоступности (занятости) системных каталогов на других установках в момент распространения обновлений. В результате может быть не обеспечена непрерывность согласованного состояния данных, а также возникнуть ряд других проблем.

Решение подобных проблем и практическая реализация распределенных информационных систем осуществляется через отступление от некоторых рассмотренных выше принципов создания и функционирования распределенных систем. В зависимости оттого, какой принцип приносится в «жертву» (отсутствие центральной установки, непрерывность функционирования, согласованность состояний данных и др.) выделились несколько самостоятельных направлений в технологиях распределенных систем — **технологии «клиент-сервер», технологии реплицирования, технологии объектного связывания**.

Реальные распределенные информационные системы, как правило, построены на основе сочетания всех трех технологий, но в методическом плане их целесообразно рассмотреть отдельно. Дополнительно следует также отметить, что техника представлений оказалась чрезвычайно плодотворной также и в другой сфере СУБД—защите данных. Авторизованный характер запросов, формирующих представления, позволяет предоставить конкретному пользователю те данные и в том виде, которые необходимы ему для его непосредственных задач, исключив возможность доступа, просмотра и изменения других данных.

Технологии «клиент-сервер»

Системы на основе технологий «клиент-сервер» исторически выросли из первых централизованных многопользовательских автоматизированных информационных систем, интенсивно развивавшихся в 70-х годах (системы main frame), и получили, вероятно, наиболее широкое распространение в сфере информационного обеспечения крупных предприятий и корпораций.

В технологиях «клиент-сервер» отступают от одного из главных принципов создания и функционирования распределенных систем — отсутствия центральной установки. Поэтому можно выделить две основные идеи, лежащие в основе клиент-серверных технологий:

- общие для всех пользователей данные на одном или нескольких серверах
- много пользователей (клиентов) на различных вычислительных установках, совместно (параллельно и одновременно) обрабатывающих общие данные

Иначе говоря, системы, основанные на технологиях «клиент-сервер», распределены только в отношении пользователей, поэтому часто их не относят к «настоящим» распределенным системам, а считают отдельным классом многопользовательских систем.

Важное значение в технологиях «клиент-сервер» имеют понятия сервера и клиента. Под сервером в широком смысле понимается любая система, процесс, компьютер, владеющие каким-либо вычислительным ресурсом (памятью, временем, производительностью процессора и т. д.) Клиентом называется также любая система, процесс, компьютер, пользователь, запрашивающие у сервера какой-либо ресурс, пользующиеся каким-либо ресурсом или обслуживаемые сервером иным способом.

В своем развитии системы «клиент-сервер» прошли несколько этапов, в ходе которых сформировались различные модели систем «клиент-сервер». Их реализация и, следовательно, правильное понимание основаны на разделении структуры СУБД на три компонента:

1. компонент представления, реализующий функции ввода и отображения данных, называемый иногда еще просто как интерфейс пользователя (см. рис. 2.1);
2. прикладной компонент, включающий набор запросов, событий, правил, процедур и других вычислительных функций, реализующий предназначение автоматизированной информационной системы в конкретной предметной области;
3. компонент доступа к данным, реализующий функции хранения, извлечения, физического обновления и изменения данных (машина данных).

Исходя из особенностей реализации и распределения (расположения) в системе этих трех компонентов различают четыре модели технологий «Клиент-сервер»:

1. **модель файлового сервера (File Server — FS)** - является наиболее простой и характеризует собственно не столько способ образования фактографической информационной системы, сколько общий способ взаимодействия компьютеров в локальной сети. Один из компьютеров сети выделяется и определяется файловым сервером, т. е. общим хранилищем любых данных. В FS-модели все основные компоненты размещаются на клиентской установке. При обращении к данным ядро СУБД, в свою очередь, обращается с запросами на ввод-вывод данных за сервисом к файловой системе. С помощью функций операционной системы в оперативную память клиентской установки полностью или частично на время сеанса работы копируется файл базы данных. Таким образом, сервер в данном случае выполняет чисто пассивную функцию.
2. **модель удаленного доступа к данным (Remote Data Access—RDA)** - основана на учете специфики размещения и физического манипулирования данных во внешней памяти для реляционных СУБД. В RDA-модели компонент доступа к данным в СУБД полностью отделен от двух других компонентов (компонента представления и прикладного компонента) и размещается на сервере системы. Компонент доступа к данным реализуется в виде самостоятельной

программной части СУБД, называемой SQL-сервером, и инсталлируется на вычислительной установке сервера системы. Функции SQL-сервера ограничиваются низкоуровневыми операциями по организации, размещению, хранению и манипулированию данными в дисковой памяти сервера. Иначе говоря, SQL-сервер играет роль машины данных. В файле (файлах) базы данных, размещаемом на сервере системы, находится также и системный каталог базы данных, в который помещаются в том числе и сведения о зарегистрированных клиентах, их полномочиях и т. п. На клиентских установках инсталлируются отделенные программные части СУБД, реализующие интерфейсные и прикладные функции. Пользователь, входя в клиентскую часть системы, регистрируется через нее на сервере системы и начинает обработку данных. Прикладной компонент системы (библиотеки запросов, процедуры обработки данных) полностью размещается и выполняется на клиентской установке. При реализации своих функций прикладной компонент формирует необходимые SQL-инструкции, направляемые SQL-серверу. SQL-сервер, представляющий специальный программный компонент, ориентированный на интерпретацию SQL-инструкций и высокоскоростное выполнение низкоуровневых операций с данными, принимает и координирует SQL-инструкции от различных клиентов, выполняет их, проверяет и обеспечивает выполнение ограничений целостности данных и направляет клиентам результаты обработки SQL-инструкции, представляющие как известно наборы (таблицы) данных.

3. **модель сервера базы данных (DataBase Server — DBS)** - её сердцевиной является механизм хранимых процедур. В отличие от RDA-модели, события, правила и процедуры, описанные средствами языка SQL, хранятся вместе с данными на сервере системы и на нем же выполняются. Иначе говоря, прикладной компонент полностью размещается и выполняется на сервере системы. На клиентских установках в DBS-модели размещается только интерфейсный компонент (компонент представления) ИС, что существенно снижает требования к вычислительной установке клиента. Пользователь через интерфейс системы на клиентской установке направляет на сервер базы данных только лишь вызовы необходимых процедур, запросов и других функций по обработке данных. Все затратные операции по доступу и обработке данных выполняются на сервере и клиенту направляются лишь результаты обработки, а не наборы данных, как в RDA-модели. Этим обеспечивается существенное снижение трафика сети в DBS-модели по сравнению с RDA-моделью.
4. **модель сервера приложении (Application Server — AS)** - суть AS-модели заключается в переносе прикладного компонента ИС на специализированный в отношении повышенных ресурсов по быстродействию дополнительный сервер системы. Как и в DBS-

модели, на клиентских установках располагается только интерфейсная часть системы, т.е. компонент представления. Однако вызовы функций обработки данных направляются на сервер приложений, где эти функции совместно выполняются для всех пользователей системы. За выполнением низкоуровневых операций по доступу и изменению данных сервер приложений, как в RDA-модели, обращается к SQL-серверу, направляя ему вызовы SQL-процедур, и получая, соответственно, от него наборы данных.

Технология объектного связывания данных решает задачу обеспечения доступа из одной локальной базы, открытой одним пользователем, к данным в другой локальной базе (в другом файле), возможно находящейся на другой вычислительной установке, открытой и эксплуатируемой другим пользователем. На практике другой локальной базой может быть и база данных сервера централизованной многопользовательской корпоративной системы.

Технология реплицирования

Во многих случаях узким местом распределенных систем, построенных на основе технологий «клиент-сервер» или объектного связывания данных, является недостаточно высокая производительность из-за необходимости передачи по сети большого количества данных. Определенную альтернативу построения быстродействующих распределенных систем предоставляют технологии реплицирования данных.

Репликой называют особую копию базы данных для размещения на другом компьютере сети с целью автономной работы пользователей с одинаковыми (согласованными) данными общего пользования.

Основная идея реплицирования заключается в том, что пользователи работают автономно с одинаковыми (общими) данными, растиражированными по локальным базам данных, обеспечивая с учетом отсутствия необходимости передачи и обмена данными по сети максимальную для своих вычислительных установок производительность. Программное обеспечение СУБД для реализации такого подхода соответственно дополняется функциями тиражирования(реплицирования) баз данных, включая тиражирование как самих данных и их структуры, так и системного каталога с информацией о размещении реплик, иначе говоря, с информацией о конфигурировании построенной таким образом распределенной системы.

При этом, однако, возникают две проблемы обеспечения одного из основополагающих принципов построения и функционирования распределенных систем, а именно — непрерывности согласованного состояния данных:

- обеспечение согласованного состояния во всех репликах количества и значений общих данных;

- обеспечение согласованного состояния во всех репликах структуры данных.

Обеспечение согласованного состояния общих данных, в свою очередь, основывается на реализации одного из двух принципов:

1. принципа непрерывного размножения обновлений (любое обновление данных в любой реплике должно быть немедленно размножено);
2. принципа отложенных обновлений (обновления реплик могут быть отложены до специальной команды или ситуации).

Источники:
http://www.profiz.ru/se/3_2006/1074/
http://www.1c.cf1.ru/doc/product/pred8/enterprise_management
http://auto-is.ru/auto_t5r1part1.html - http://auto-is.ru/auto_t5r9part2.html