РАЗРАБОТКА ОБЪЕКТНОЙ МОДЕЛИ РАСПРЕДЕЛЕННОГО ХРАНИЛИЩА ДАННЫХ

Лаздынь С.В., Петров А.В.
Кафедра автоматизированных систем управления
Донецкий национальный технический университет
E-mail: slazd@ukr.net, petrovalex@strannik.org.ua

Abstract

Lazdyn S. V., Petrov A. V. Development of the distributed Data Warehouse object model. *The main components of logical and physical architecture are defined in the structure of the distributed Data Warehouse. With the use of the object-oriented approach the models of main components of Data Warehouse are developed. On the basis of co-operation of models of components the general object model of the distributed Data Warehouse is built.*

Характеристика проблемы

В современных системах поддержки принятия решений для управления крупными предприятиями и корпорациями широко используются хранилища данных, имеющие распределенную архитектуру. Одной из проблем, которую приходится решать при создании таких систем — это определение структуры хранилища данных и схемы распределения данных по узлам компьютерной сети, при которых обеспечивается высокая производительность всей информационной системы.

Поэтому при проектировании и анализе распределенного хранилища данных (РХД) необходима разработка модели, достоверно отражающей хранилище данных на всех этапах его функционирования. Это особенно важно при практическом внедрении проектов распределенных хранилищ данных, так как проведение экспериментов на достаточно приближенной к реальности модели позволяет выявить слабые звенья, исправить возможные недостатки и повысить эффективность создаваемого хранилища данных.

Краткий анализ проведенных исследований по моделированию РХД

Под распределенным хранилищем данных будем понимать «интегрированную, предметно-ориентированную, нестационарную, долговременную базу данных, обеспечивающую поддержку принятия решений», при этом фрагменты базы территориально удалены друг от друга [1].

В настоящее время существуют следующие технологии моделирования хранилищ данных:

– реляционная ER-модель, предложенная Питером П. Ченом в 1976 году. С помощью ER-модели удалось формализовать методы моделирования данных для оперативных систем обработки транзакций. Основная идея ER-моделирования состоит в выделении в базе данных «сущностей» и «связей» между сущностями [2]. Эта технология хорошо зарекомендовала себя при создании реляционных, операционных баз данных;

– объектно-ориентированная модель, в которой для описания сущностей данных и связей между ними используется одно понятие — объект, включающее в себя информацию о сущности и о связях существующих внутри объекта, а также информацию о его связях с внешними объектами [4].

При моделировании распределенное хранилище данных рассматривается как единое логическое хранилище данных, основные концепции моделирования хранилищ данных, касающиеся структуры данных, агрегирования данных и метаданных, сохра-

няют свое значение. Важную роль начинают играть специфические свойства хранилища данных, распределенного по узлам компьютерной сети. Поэтому необходимо при моделировании в первую очередь учитывать местонахождение фрагментов данных распределенного хранилища данных, а также учитывать распределение и возможное дублирование данных распределенного хранилища. Указанные выше модели обеспечивают только описание структуры данных РСД [4]. Однако для адекватного моделирования необходима такая модель, которая описывала бы не только структуру данных в хранилище, но и содержала в себе описания всех компонентов распределенного хранилища, таких как: измерения и фрагменты данных, сервера и каналы связи, источники данных и рабочие станции. Кроме того, модель должна позволять проанализировать функционирование хранилища данных и выявить слабые места и недостатки.

Разработка объектных моделей типовых компонентов РХД

Проведенный анализ структуры РХД показал, что в распределенном хранилище данных можно выделить две большие группы типовых компонентов: компоненты логической архитектуры и компоненты физической архитектуры. К компонентам логической архитектуры относятся все компоненты хранилища данных связанные с данными, и не связанные с технической стороной функционирования хранилища. К компонентам физической архитектуры относятся те компоненты хранилища данных, которые отвечают за техническую сторону функционирования РХД [3, 4].

Для построения моделей типовых компонентов РХД использован объектно-ориентированный подход. При этом, для каждого компонента был разработан соответствующий класс объектов, определены его основные свойства и методы.

Компоненты логической архитектуры:

Измерения — сущности хранилища данных, которые наглядно представляют информацию о фактах через их атрибуты [5, 6]. Характеризуются названием измерения, диапазоном значений измерения(в связи с занесением новых данных может изменяться в течении функционирования РХД), и списком атрибутов. Например измерение «Дата» может иметь диапазон значений с 1997-го года до 2006-го года, и список атрибутов «Год», «Месяц», «День».

Объектная модель измерения хранится в классе TDimension. Класс TDimension содержит свойства: ID_Dimension — уникальный идентификатор измерения, Name — наименование измерения, ValueCount — ширина диапазона значений измерения, целое число, Attr — массив атрибутов измерения. Методов данных класс не содержит.

Фрагмент хранилища данных — это часть таблицы фактов. Этот компонент характеризуется архитектурой фрагмента данных — таблица фактов, или гиперкуб, размером одной записи, количеством записей, расположением и размерностью — границами, ограничивающими подкуб данных, расположенный во фрагменте.

Класс TFragment содержит следующие свойства: ID_Fragment — уникальный идентификатор фрагмента, FieldSize — вектор размеров полей фрагмента, FieldNames — вектор названий полей фрагмента, RecordCount — количество записей, RecordSize — размер одной записи, Location — указатель но объект класса TServer — местоположение фрагмента данных, Type — тип фрагмента хранилища данных: таблица фактов или гиперкуб, DimList, UpDimValue, LowDimValue свойства, определяющие размерность фрагмента хранилища данных — верхние и нижние границы подкуба данных. Также класс TFragment содержит следующие методы: InsertRecord — моделирование вставки записи, GetRecord — моделирование выборки одной записи.

Источник данных — под источником данных понимают, как правило, оперативную базу данных, содержащую детализованные данные о текущем состоянии предприятия. Объект «источник данных» характеризуется местонахождением, количеством записей и размером одной записи.

Класс TSource содержит следующие свойства: ID_Source — уникальный идентификатор источника данных, Location — указатель но объект класса TServer — местоположение источника данных, RecordCount — количество записей в источнике, RecordSize — размер одной записи. Также класс TSource будет содержать один метод GetRecord — моделирование выборки записи данных.

Запросы на выборку — всякое хранилище данных характеризуется возможностью возникновения нерегламентированные запросов. Таким образом, теоретически пользователь может сформировать какой угодно запрос. Запрос на выборку характеризуется местом формирования, границами, очерчивающими отбираемый подкуб данных и релевантностью — средней частотой возникновения в день. Во время выполнения запросов на выборку возникает необходимость в транспортировке больших объемов данных между серверами сети. Поэтому во время передачи определяется оптимальный маршрут с учетом загруженности каналов связи и выполняется последовательная передача данных по всем каналам связи.

Класс TSelectQuery будет содержать следующие свойства: ID_SelQuery — уникальный идентификатор запроса, Source — указатель на рабочую станцию-источник запроса, Schedule — расписание выполнения запросов на протяжении дня, DimList, UpDimValue, LowDimValue размерность запроса, границы отбираемого подкуба данных, Name — наименование запроса, и Freq — релевантность. Класс TSelectQuery будет содержать метод Execute — моделирование выполнения запроса.

Запросы на вставку — данные извлекаются из разных источников, проходят процессы трансформации, унификации, фильтрации, очистки, агрегирования и после этого записываются во фрагмент данных. Запрос на вставку характеризуется фрагментом данных, в который выполняется вставка записи, релевантностью и источником данных, из которого осуществляется вставка к фрагменту. В случае если одни и те же данные нужно вставить в несколько фрагментов — формируются несколько запросов на вставку.

Класс TInsertQuery будет содержать следующие свойства: ID_InsQuery —уникальный идентификатор запроса, Name — наименования запроса, Source — указатель на источник данных, Target — указатель на объект класса TFragment фрагмент данных, в который осуществляется вставка, Schedule — расписание выполнения запроса на протяжении моделирования, Freq — релевантность запроса. Класс TinsertQuery будет содержать один метод — Execute — моделирование процесса выполнения запроса.

Компоненты физической архитектуры:

Сервер — аппаратно-программная платформа, под управлением которой функционируют один, или несколько фрагментов данных. Главный параметр сервера — скорость обработки транзакций. Этот параметр определяется аппаратной платформой сервера, операционной системой, и прикладным программным обеспечением. Также важным параметром является вместимость жесткого диска. Также сервер характеризуется текущей загруженностью и состоянием.

Класс TServer будет содержать следующие свойства: ID_Server — уникальный идентификатор, Name — наименование сервера, HDDSize — размер жесткого диска, HDDFree — размер свободного места на жестком диске, TPS — параметр "транзакций-в-секунду", Free — время освобождения сервера. Класс TServer будет выполнять следующие функции: DiskRead — моделирования процесса чтения данных с жесткого диск, DiskWrite — моделирование процесса записи данных на жесткий диск, Down — моделирование отказа сервера, ChannelWrite — запись данных в один из присоединенных каналов связи, ChannelRead — чтение данных из канала связи.

Канал связи — это физическая среда передачи данных. Характеризуется помехоустойчивостью — вероятность возникновения помехи, отказоустойчивостью, скоростью передачи данных, и текущей загруженностью канала.

Класс TChannel будет иметь следующие свойства: ID_Channel — уникальный идентификатор, Capacity — пропускная способность канала, Otkaz — отказоустойчивость канала, Pomeh — помехоустойчивость, Free — время освобождения канала, LocationFirst — указатель на первый сервер к которому подключен канал связи, LocationSecond — указатель на второй сервер, к которому подключен канал связи. Класс TChannel будет содержать следующие методы: TransmitData — моделирование процесса передачи данных между двумя серверами, Down — моделирование отказа канала связи.

Сетевое устройство — маршрутизатор, мост или коммутатор соединяет между собою рабочие станции и сервер, или соединяет несколько серверов. Характеризуется перечнем серверов, к которому он подключен, скоростью передачи данных, размером одного пакета, и размером заголовка пакета.

Класс TNetDevice будет содержать следующие свойства: ID_NetDevice — уникальный идентификатор сетевого устройства, Server — перечень присоединенных к сетевому устройству серверов, Speed — скорость передачи данных, PackageSize — размер пакета данных, PackageHead — размер заголовка пакета данных. Также класс TNetDevice будет содержать один метод TransmitData, что будет выполнять функцию моделирования процесса передачи данных между сервером, и рабочей станцией.

Рабочая станция. Характеризуется названием и сетевым устройством, которое объединяет рабочую станцию с сервером.

Класс TWorkStation будет иметь следующие атрибуты: ID_WorkStation уникальный идентификатор, TNetDevice — указатель на сетевое устройство, к которого подключена рабочая станция, Name — наименование рабочей станции.

Класс TWorkStation также будет содержать один метод — CreateQuerySchedule — создания очереди запросов, которые должны формироваться на данной рабочей станции.

Описанные выше объектные модели типовых компонентов РХД программно реализованы в виде библиотеки классов на языке C++ .

Построение общей объектной модели РХД.

Общая объектная модель распределенного хранилища данных построена как система взаимодействующих объектных моделей типовых компонентов. Схема взаимодействия объектов в модели РХД показана на рис. 1.

Для удобства моделирования и организации взаимодействия между объектами введен еще один объект, который содержит методы, работающие со всем хранилищем данных. Поскольку одним из главных процессов влияющих на быстродействие распределенного хранилища данных является передача данных по каналам связи, основной задачей этого объекта будет маршрутизация и управление передачей данных между серверами.

Этот объект будет иметь название TDW, содержать списки всех объектов хранилища данных. Класс TDW также будет содержать следующие методы: Initialize — инициализация модели РХД, Marshrout — поиск оптимального маршрута между двумя серверами с учетом помехоустойчивости и отказоустойчивости и занятости каналов, OpenModelDay — начало моделирования, CloseModelDay — завершение моделирования, TransmitData — передача данных между двумя серверами сети, Clock — отсчет тактов модельного времени.

В начале моделирования формируется структура РХД. Из базы данных считывается информация об измерениях, фрагментах данных, серверах, каналах связи и об остальных компонентах модели. В памяти создаются все необходимые объекты и происходит инициализация очередей запросов. Для каждого запроса с учетом его релевантности создается очередь вызовов в течении всего времени моделирования.

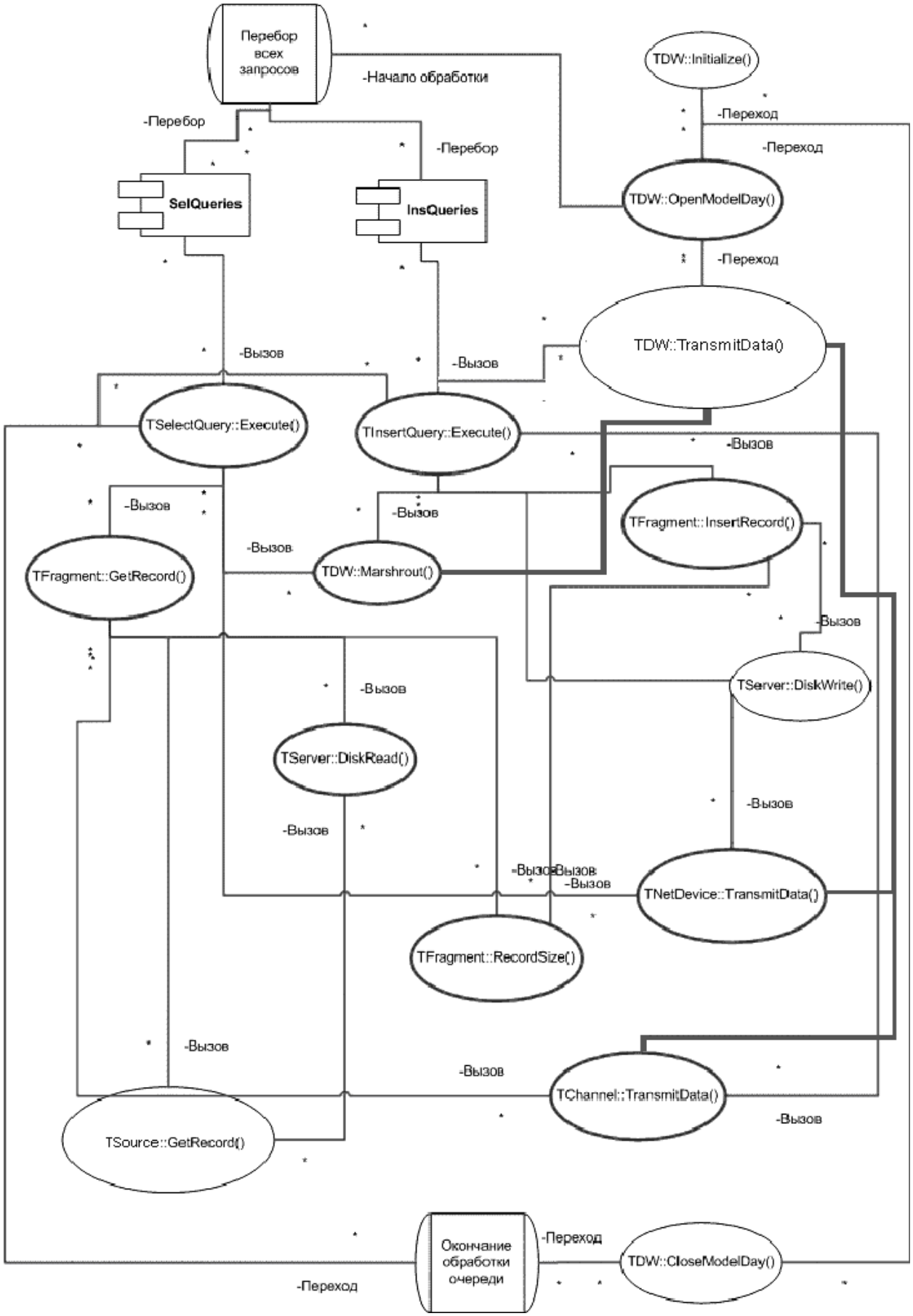

Рисунок 1 — Схема взаимодействия объектов в модели РХД

В процессе моделирования из очереди запросов выбирается запрос с наименьшим временем возникновения и начинается его обработка. Моделируются процессы считывания и записи данных в фрагменты. Время, в течение которого серверы реагируют на запрос, записывается в выходную базу данных. Когда наступает необходимость в передаче данных между серверами, происходит обращение к объекту TDW.

Объект TDW управляет передачей данных, производит поиск наилучшего маршрута между серверами с учетом помехоустойчивости, отказоустойчивости и текущей загруженностью каналов связи и сетевых устройств. При передаче данных по каналам связи передаваемые данные разбиваются на пакеты и передаются по кратчайшему маршруту в пункт назначения.

При передаче данных по каналам связи в выходную базу данных записывается информация о времени, в течение которого канал был занят. Если в ходе передачи канал вышел из строя — объект TDW инициирует поиск обходного маршрута для передачи данных. Если такой маршрут имеется, то TDW организовывает повторную передачу всех неполученных в пункте назначения данных. Если такого пути не существует, то запрос возвращается в очередь и из очереди берется следующий запрос. Как только последний пакет достигает пункта назначения, запрос считается выполненным. В базу данных записывается общее время выполнения запроса. Когда выполнятся все запросы из очереди запросов — происходит завершение моделирования.

После завершения моделирования на основании данных о загрузке каналов, серверов и временах выполнения запросов, занесенных в выходную базу данных, рассчитываются итоговые коэффициенты загрузки каналов и среднее время выполнения запросов на выборку и вставку.

Заключение

Разработана новая объектная модель распределенного хранилища данных, учитывающая его специфические свойства. Для этого для основных компонентов физической и логической архитектуры распределенного хранилища данных были разработаны объектные модели в виде классов на языке программирования C++. Также было установлено взаимодействие между классами, входящими в модель РХД и их методами, что обеспечивает отражение основных процессов, связанных с выполнением запросов в распределенном хранилище данных. Модель позволяет проводить исследование эффективности распределенного хранилища данных на стадии разработки и анализировать время выполнения запросов, загруженность серверов и каналов.

В дальнейшем возможно усовершенствование модели путем учета мене значимых факторов, таких как репозитарий метаданных, фрагментация таблиц измерений и др.

Литература

1. Bill Inmon, Chuck Kelley, The Twelve Rules of Data Warehouse for Client/Server World, Data Management Review, 4(5) 1994, с. 6–17.
2. W. H. Inmon. Building The Data Warehouse (Second Edition). — NY, NY: John Wiley. — 1993.
3. Спирли Э. Корпоративные хранилища данных. Планирование, разработка, реализация. Том. 1.: Пер. с англ. — М.: Издательский дом «Вильямс», 2001. — 400 с.
4. Роб П., Коронел К. Системы баз данных: проектирование, реализация и управление. — 5–е изд., перераб. и доп.: Пер. с англ. — СПб.: БХВ-Петербург, 2004. — 1040 с.
5. Сураджит Чаудхури, Умешвар Дайал, Венкатеш Ганти. Технология баз данных в системах поддержки принятия решений // Открытые системы. — №1. — 2002.
6. Риккарди Г. Системы баз данных. Принципы и практика использования в Internet и среде Java.: Пер. с англ. — СПб.: БХВ-Петербург, 2005. — 582 с.