

Пронин С.В., канд. техн. наук. (ХНАДУ)

ОРГАНИЗАЦИЯ СИСТЕМЫ МОНИТОРИНГА ГОРОДСКОГО ТРАНСПОРТА

Анотація: У статті розглядається підхід до побудови систем моніторингу транспорту з використанням сховищ даних.

Ключові слова: сховище даних, база даних, СУБД, моніторинг, сервер.

Аннотация: В статье рассматривается подход к построению систем мониторинга транспорта с использованием хранилищ данных.

Ключевые слова: хранилище данных, база данных, СУБД, мониторинг, сервер.

Abstract: This paper presents an approach to the construction of systems for monitoring transport using data warehouses.

Keywords: data warehouse, database, database, monitoring server.

Введение

Современные информационные технологии без лишних затрат на программное обеспечение позволяют реализовать мониторинг и оценку функционирования городского транспорта, что доказывает возможность внедрения таких систем на базе существующих компьютерных сетей в условиях большого города. Достаточным является соответствие имеющейся компьютерной сети и протокола HTTP. Со стороны респондента необходимо наличие обычного Web-браузера, на любом современном компьютере, независимо от установленной операционной системы. В качестве сервера базы данных могут использоваться коммерческие решения на базе Microsoft SQL Server 2005/2008 и старше, или свободная версия MySQL, поддерживаемая корпорацией Oracle.

Анализ публикаций

В основе современного подхода к построению аналитических систем лежит идея интегрированного хранилища данных, обеспечивающего единый доступ к информации, разбросанной по разнообразным источникам в том числе внешним. При этом существенно, что данные в хранилище имеют исторический характер, т. е. обеспечивается интеграция не только разнородных источников, но и архивных данных, возникающих в процессе функционирования той или иной оперативной системы [1-3].

Данные из оперативных систем и внешних источников подвергаются различным преобразованиям, согласованию и загружаются в централизованное хранилище, которое содержит всю информацию, необходимую для всевозможных процессов принятия решений. Для того чтобы существующие хранилища данных способствовали принятию управленческих решений, информация должна быть представлена аналитику в нужной форме, т.е. должны быть развиты инструменты доступа к данным хранилища и их обработки. Для информационного обеспечения отдельных функционально замкнутых задач используются так называемые витрины данных, в которые информация попадает либо из хранилища (зависимые витрины) либо непосредственно из источников данных, проходя предварительные согласования и преобразования. Витрины данных строятся на основе реляционных или, что более популярно, многомерных СУБД (Система управления базами данных). Дело в том, что для решения большинства задач анализа оказываются полезными принципы многомерной модели данных и соответствующие им многомерные базы данных [4-8]. На основе анализа моделей данных была выбрана темпоральная (временная) модель данных, которые хорошо учитывают изменения в данных в течении времени [9-11], что характерно для транспортной отрасли.

Архитектура СУБД транспортных организаций

При разработке хранилища данных, транспортной СУБД будем исходить из того, что её архитектура должна отвечать особенностям решаемой задачи. В данном случае этому наиболее полно соответствует архитектура «клиент-

сервер». Применительно к транспортным системам баз данных архитектура «клиент-сервер» интересна и актуальна тем, что обеспечивает простое и относительно дешевое решение проблемы коллективного доступа к базам данных в локальной сети.

Технология транспортной СУБД предлагается выстраивать на основе технологии «клиент-сервер» которая будет иметь две части – приложение-клиент (front-end) и сервер базы данных (back-end). Такой подход позволит обеспечить с одной стороны централизованное администрирование, безопасность, надежность, а с другой при применении технологии “файл-сервер” система будет обладать низкой стоимостью и возможностью распределенной обработки данных. Эти два типа качеств важны в связи с особенностями устройства транспортной отрасли. Кроме того, архитектура «клиент-сервер» значительно упрощает и ускоряет разработку приложений за счет того, что правила проверки целостности данных находятся на сервере.

При извлечении данных по результатам работы транспорта для их анализа необходимо организовать доступ к базе данных от прикладной программы анализа или пользователя производится путем обращения к клиентской части системы. В качестве основного интерфейса между клиентской и серверной частями используем язык баз данных SQL.

Серверы баз данных, интерфейс которых основан 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. В общем виде предлагаемая система мониторинга представлена на рис. 1.

Сервер пассажирских перевозок представляет собой программный компонент вычислительной системы, который выполняет сервисные обслуживающие функции, по запросам клиентов предоставляя им доступ к ресурсам и услугам, связанным с транспортной отраслью, такие как проведение анкетных обследований, определение местоположения транспортных средств, информация о работе транспорта и другие. Также решается задача по управлению СУБД, выполнению запросов от SQL-операторов и осуществление управления базой данных. Здесь СУБД действует в традиционной роли, заключающейся в манипулировании данными получаемыми от мониторинга результатов работы транспорта. СУБД на сервере базы данных не создает форм, отчетов или меню, а выполняет исключительно функции ядра: прием SQL-запросов и обработку строк в таблицах.

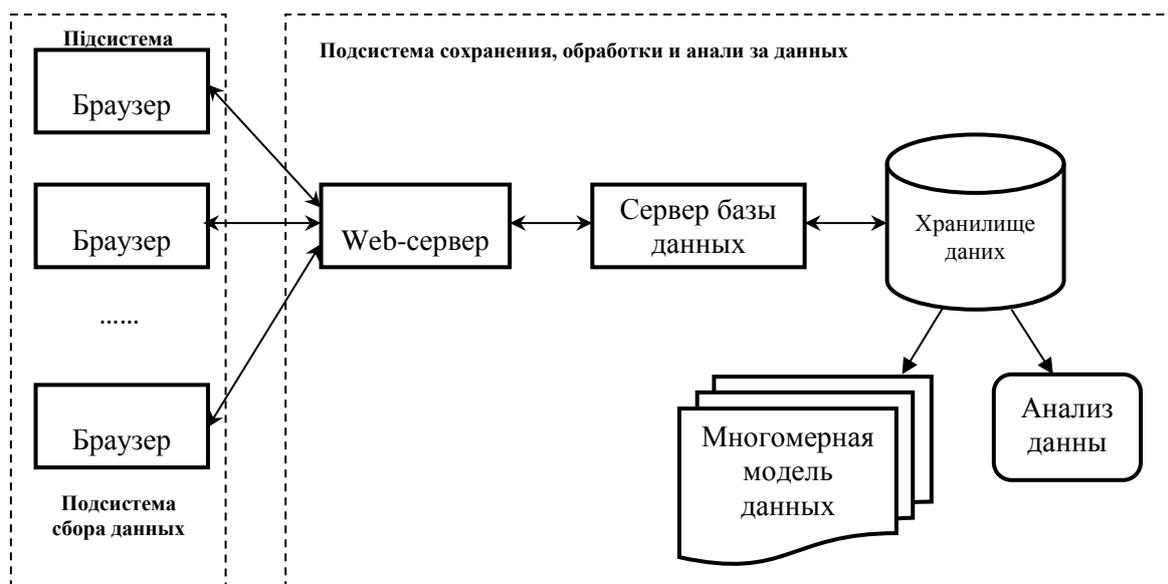


Рисунок 1 – Общая схема системы СУБД

В нашем случае Web-сервер (webserver) выполняет три основные функции. Во-первых, это HTTP-сервер: он обрабатывает протокол HTTP, принимая запросы и генерируя отклики в формате HTTP. Во-вторых, на web-сервере располагаются сценарный интерпретатор (scriptingengine), который позволяет писать код на VBScript и JavaScript и запускать его на web-сервере. Наконец, в приложениях баз данных третьей функцией web-сервера является создание, чтение, обновление и удаление экземпляров представлений. В случае,

обычно используемые, когда web-сервер работает под управлением операционной системы Windows роль HTTP-сервера почти всегда будет играть IIS, поскольку эта система является частью Windows. IIS предоставляет интерфейс под названием ISAPI (Internet Server Application Program Interface), интерфейс прикладных программ Интернет-сервера), с помощью которого другие программы могут перехватывать и обрабатывать HTTP-сообщения. Одной из таких программ является ASP (Active Server Processor, обработчик активных серверных страниц), Он обрабатывает все web-страницы с расширением .asp. Когда IIS получает такую страницу, она посылает ее ASP через интерфейс ISAPI. ASP обрабатывает страницу и генерирует отклик, который передается IIS через ISAPI и далее посылается клиенту. На рис. 2 показаны стандарты и интерфейсы прикладных программ, которые используются при передаче данных между сервером базы данных и web-сервером.

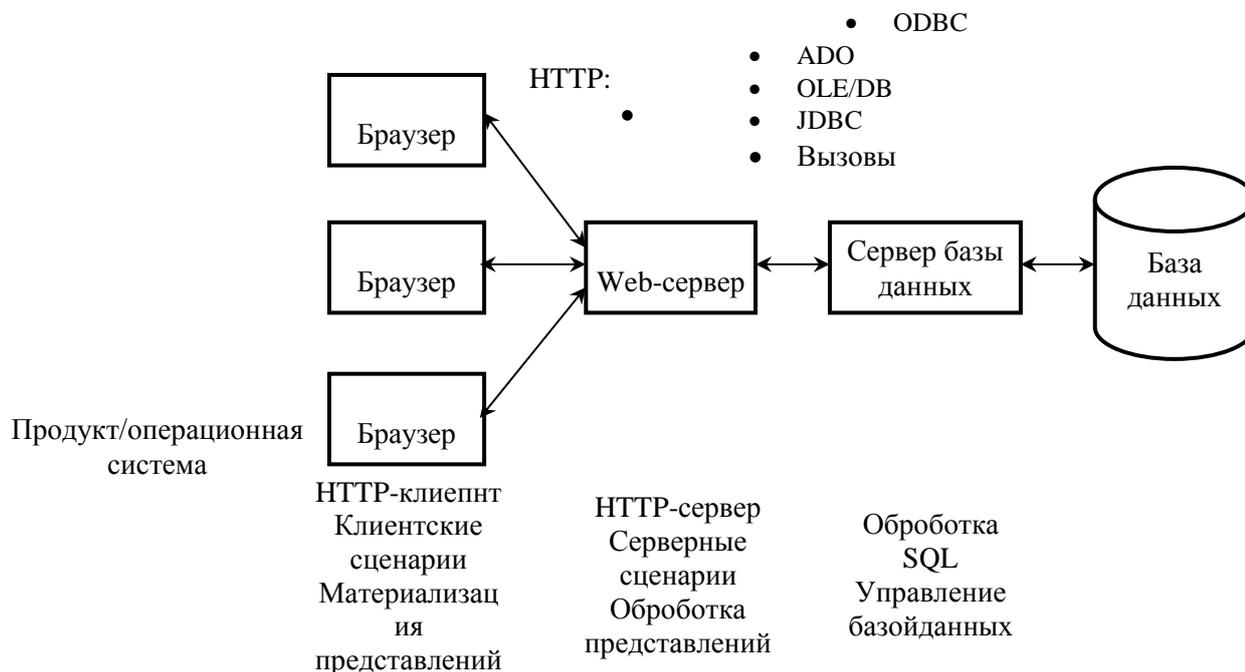


Рисунок 2 – Функции уровней (стандарты и интерфейсы)

Модель хранилища данных

Так как работа городского транспорта отличается динамикой и изменением состояний в различные моменты времени для задачи анализа

информации содержащейся в базах данных оптимально использовать темпоральные хранилища данных которые в отличие от традиционных моделей, обеспечивающих хранение лишь мгновенного снимка объектов предметной области, позволяют хранить информацию об эволюции объектов, т.е. для любого объекта, который был создан в момент времени $t(I)$ и закончил свое существование в момент времени $t(I+n)$, в БД будут сохранены все его состояния на данном временном интервале.

Тогда под темпоральностью объекта следует понимать явную или неявную связь объекта с определенными датами или промежутками времени, а темпоральные данные – это данные, которые могут изменяться с течением времени.

В соответствии с устоявшимися понятиями темпоральная модель данных (ТМД) – это модель данных, ориентированная на хранение темпоральных данных, все аспекты которой также должны быть темпоральными. В отличие от традиционной модели данных которая обычно состоит из трех компонент, таких как структура данных, операции и ограничения целостности, темпоральная модель также включает эти компоненты но с учетом изменений данных во времени и в общем имеет следующий вид:

$$MT=(DS(t), OP(t), C(t)), \quad (1)$$

где $DS(t)$ - структура данных в момент времени;

$OP(t)$ – операции над данными в момент времени;

$C(t)$ – целостность данных в момент времени.

Отсюда следует, что структура данных должна быть адаптирована таким образом, чтобы имелась возможность хранить темпоральные данные. Дополнительно для каждого ограничения целостности в нетемпоральных моделях данных темпоральная модель должна поддерживать темпоральный аналог нетемпорального ограничения. Семантика темпоральных ограничений целостности также должна быть переопределена.

Так как данные, представленные в базе данных (БД), являются неким отражением текущего состояния действительности для отображаемого состояния

системы, каждая запись можно представить в виде факта, являющийся истинным в определенный момент или интервал времени. При переходе к темпоральной БД для каждого такого факта можно указать тот промежуток времени, в который этот факт являлся истинным в моделируемом мире, представленном в БД. Подобное представление времени, когда с данными связывается промежуток времени их актуальности с точки зрения моделируемого мира, называется модельным, или действительным временем.

Другим типом линии времени, который рассматривается исследователями темпоральных БД, является транзакционное время. В любой СУБД каждой записи БД можно сопоставить тот промежуток времени, когда данная запись была представлена в БД, то есть промежуток времени между моментами добавления записи и ее удаления из БД

Еще одной проблемой является тип представления времени в модели данных. Здесь различают дискретный и интервальный способы представления времени. При простоте дискретного представления данный тип имеет один недостаток, а именно усложнение запросов и операций из-за связи темпоральных объектов с одним атрибутом времени. Поэтому наиболее распространенные и простые в реализации являются модели основанные на интервальном способе представления времени.

Наиболее распространенной и простой в реализации является модель представления темпоральных данных, предложенная Р. Снодграсом [11], которая имеет следующий вид:

$$R=(A_1, \dots, A_n, T_s, T_e, V_s, V_e), \quad (2)$$

где R - битемпоральное отношение;

A_1, \dots, A_n - набор атрибутов;

T – битемпоральный атрибут, определенный на множестве битемпоральных элементов;

T_s, T_e, V_s, V_e - дополнительные атрибуты времени, содержащие дату начала и окончания транзакционного и модельного времени.

Вывод

В статье был рассмотрен и предложен подход к созданию систем мониторинга автомобильного транспорта на основе хранилищ данных.

Литература

1. Р. Пэддок Visual FoxPro 6.0 Разработка корпоративных приложений / Р. Пэддок, Д. Петерсен, Р. Тэлмейдж – М.: «ДМК», 1999. – 87 с.
2. Диго С. М. Проектирование и использование баз данных / С.М. Диго - Учебник. М.: Финансы и статистика, 1995. – 38 с.
3. Інтелектуальна технологія управління громадським пасажирським транспортом великих міст та регіонів України: звіт з наукової роботи № 09-53-07 / ХНАДУ - № держреєстрації 0107U001008 – Харків, 2008.– 257 с.
4. Вендров А. М. Проектирование программного обеспечения экономических информационных систем / А.М. Вендров – М.: «Финансы и статистика», 2000. – 130 с.
5. Билл Инмон. Производительность систем хранилищ данных. Performance In The Data Warehouse Environment / Билл Инмон. – 2000. – №4. – С. 41 – 48.
6. Вон Ким. Три основных недостатка современных хранилищ данных. Performance In The Data Warehouse Environment / Вон Ким. – 2000. – №4. – С. 120 – 127.
7. А. А. Барсегян, Методы и модели анализа данных / А. А. Барсегян, М. С. Куприянов – С.-Петербург: БХВ Петербург, 2004. – 134 с.
8. А. Бергер, MS SQL Server 2005 Analysis Services и многомерный анализ данных / А. Бергер, И. Горбач – С.-Петербург: БХВ Петербург, 2007. – 235 с.
9. Спандерашвили, Д.В. Особенности построения системы сбора статистики телекоммуникационной компании./ Д.В. Спандерашвили. // Информатика: проблемы, методология, технологии. Материалы пятой региональной научно-методической конференции. – Воронеж: Воронежский государственный университет, 2005. – Ч 2. – С.136-141. – ISBN 5-9273-0681-0.
10. Долгий А.И. Программный комплекс автоматического анализа георадиолокационных данных «GEORAILWAY+»/ Долгий А.И., Хатламаджиян А.Е., Окост М.В. // Тезисы докладов четвертой международной научно-практической конференции «Инженерная и рудная геофизика – 2008». – Геленджик, 2008.
11. Snodgrass R. Developing Time-Oriented Database Applications in SQL/ R Snodgrass – Morgan Kaufmann Publishers, 1999.