

# НАДМИН - АДМИНИСТРАТИВНО-РАСЧЕТНАЯ СИСТЕМА ДЛЯ РЕГИОНАЛЬНЫХ СЕТЕЙ

А.В. Карлаш, СМ. Абрамов, П.Г. Жбанов, А.С. Нестеров, Б.В. Ермилова, Ю.В. Шевчук

В течение предыдущих пяти лет администрирование городской сети г. Переславля-Залесского и расчеты с абонентами велись при помощи системы Nadmin. Система зарекомендовала себя надежной и удобной в эксплуатации. Однако за пять лет прошел прогресс в программных и аппаратных средствах, а так же было накоплено множество пожеланий к системе. Все это привело к необходимости разработки новой версии системы.

Известно, что в России региональные сети не могут рассчитывать на постоянное внешнее финансирование, поэтому они должны самостоятельно покрывать свои эксплуатационные расходы. Источником финансирования, на который, можно рассчитывать, являются средства абонентов сети. Чтобы их можно было эффективно использовать, необходимо решить две задачи:

1. выработать политику цен на услуги;
2. создать систему поддержки расчетов с абонентами (billing system).

В той или иной мере эти задачи были решены в старой системе Nadmin. Однако в ней отсутствовал ряд полезных функциональных возможностей. Например, сегодня очень простые запросы, которые абонент мог бы исполнить сам, выглядят как электронные письма с просьбой на адрес группы технической поддержки. Это отвлекает обслуживающий персонал от более важных дел и вынуждает его заниматься рутинными задачами, хотя с этой работой мог бы справиться сам абонент. Тем самым возник актуальный вопрос о создании новой административной системы, которая обладала бы более полной и удобной функциональностью для решения задач администрирования.

Новая версия системы реализуется, как для выполнения следующих функций:

- автоматизация деятельности администратора сети;
- учет трафика абонентов в реальном времени;
- предоставление абонентам возможности управления собственными услугами;

- поддержка новых услуг, таких как: IP-телефония, веб-хостинг;
- возможность создания тарифных планов;

так и для удовлетворения следующих экономических потребностей:

- необходимость поддержания конкурентной способности СТ "Ботик" на рынке высоких технологий;
- повышение привлекательности СТ "Ботик" среди абонентов;
- перенос рутинных операций с персонала СТ "Ботик" на абонентов;
- снижение расходов абонентов СТ "Ботик".

Также в новой версии системы Nadmin особое внимание было уделено дизайну и гибкости системы, и возможности дальнейшего расширения без кардинальной перестройки всей системы.

Для создания новой версии системы необходимо было решить следующие задачи:

- исследовать и реализовать более эффективную и удобную базу данных;
- исследовать и реализовать более эффективные статистическую, расчетную и административную подсистемы;
- исследовать и реализовать веб-интерфейсы (пользовательский и администраторский).

Архитектура системы

Система Nadmin состоит из следующих подсистем (рис1):

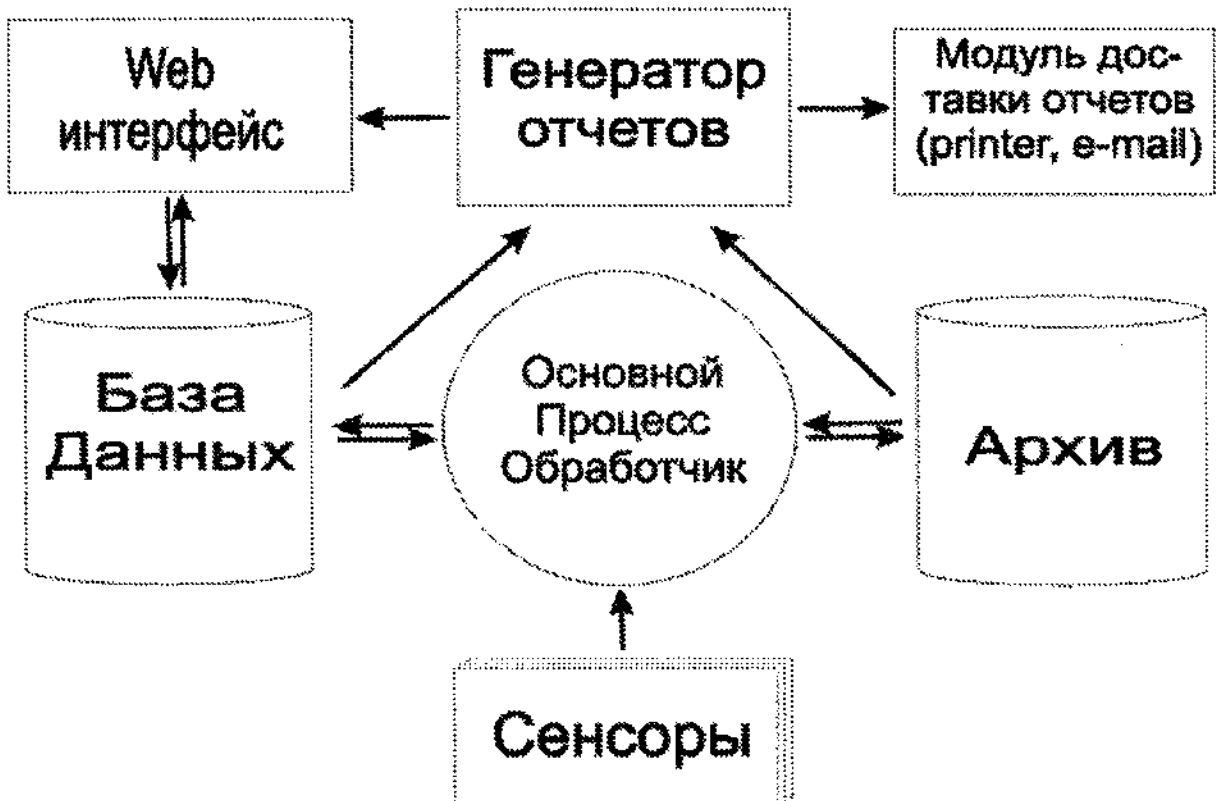


Рис.1

- База данных (БД). БД используется для хранения следующих объектов: абоненты, подключения и т. д. В системе Nadmin реализована собственная БД с использованием пакета RCS.
- Архив. Архив используется для долгосрочного хранения информации. В нем поддержана возможность сжатия данных, что позволяет хранить большие объёмы информации.
- Сенсоры. Сенсоры обрабатывают статистику, отбирают информацию, необходимую для расчета, формируют порции статистики и передают для расчета основному процессу-обработчику.
- Основной процесс-обработчик (ОПО). ОПО является центральным процессом системы, который выполняет сбор ключевой статистики от сенсоров, расчет стоимости оказанной услуги, модификацию лицевого счетов абонентов и архивацию полученной статистики в архив.

Генератор отчетов. Генератор отчетов по запросу, используя информацию из базы данных и из архива, строит отчеты. Сгенерированные отчеты бывают трёх видов: web-страница, печатная форма, отчет для отправки по e-mail.

- Web-интерфейс. Web-интерфейс используется для настройки системы, изменения информации и получения части отчетов.

#### Сенсоры

Сенсор является обработчиком сырой статистики. Естественным источником статистики по использованию ресурсов пользователями через различные серверы системы являются лог-файлы этих серверов. Каждый сенсор умеет разбираться в конкретном формате сырой статистики. В ходе одного такта работы сенсора анализируется одна (очередная) строка из лог-файла. В ходе анализа сенсор определяет потребителя ресурса и объем потребленной информации. Эти данные в виде порции статистики (рис.2) передаются для ОПО. За счет заголовка порции статистики, где содержится ключевая информация, необходимая для расчета, отпадает необходимость в дальнейшем разборе детальной статистики. Детальная статистика предназначена только для показа пользователю. Сенсор не обращается к БД и к другим конфигурационным файлам системы, он максимально автономен. Это сделано для того, чтобы UNIX-процесс, исполняющий код сенсора, требовал как можно меньше ресурсов (память, процессорное время). Также эта независимость позволяет сенсору работать на удаленном компьютере.

**Date1: 20030606160105**

**Date2: 20030606160110**

**Id: 193.232.174.2**

**Count-in: 19356**

**Class-in: HIGH**

**Count-out: 1346**

**Class-out: NORMAL**

**144:120 81.117.239.98.3333 x.8081 tcp**

**144:120 81.117.239.98.3233 x.3128 tcp**

**144:120 81.117.239.98.3162 x.80 tcp**

**144:120 81.117.239.98.3268 x.8080 tcp**

**144:0 81.117.239.98.2982 x.1080 tcp**

Рис.2

Сенсоры реализованы как обычные процедурные программы на языке Perl. Каждый сенсор запускается как программа-демон. Сенсор "помнит", до какого места в лог-файле он уже обработал статистику. Для этого сенсор запоминает последнюю успешно обработанную строку из лог-файла и другие данные, необходимые для восстановления состояния сенсора после сбоя.

Серверы с лог-файлами используют понятие ротации лог-файлов. Сенсор отслеживают данное событие как смену индексного описателя файла (mode). В случае изменения inode происходит переоткрытие базового лог-файла.

В случае сбоя сенсор может простаивать в течение длительного времени. За это время могут произойти ротации лог-файлов (одна или несколько). Поэтому при запуске сенсор первоначально определяет все накопленные лог-файлы и ищет в них последнюю успешно обработанную строку.

#### База данных

БД реализована как дерево каталогов с текстовыми файлами под управлением системы RCS (рис.3). За счет использования RCS автоматически реализуются два важных оригинальных свойства БД:

- ретроспективность БД;
- хранение информации о дате и авторе каждого изменения объектов в БД.

Каждому корневому каталогу соответствует сущность БД (объекты Org, Ftp и др.). Экземпляры сущностей представлены в виде подкаталогов с файлами под RCS. Именем подкаталога является идентификатор - Ш. ГО является сквозным для всех каталогов-сущностей.

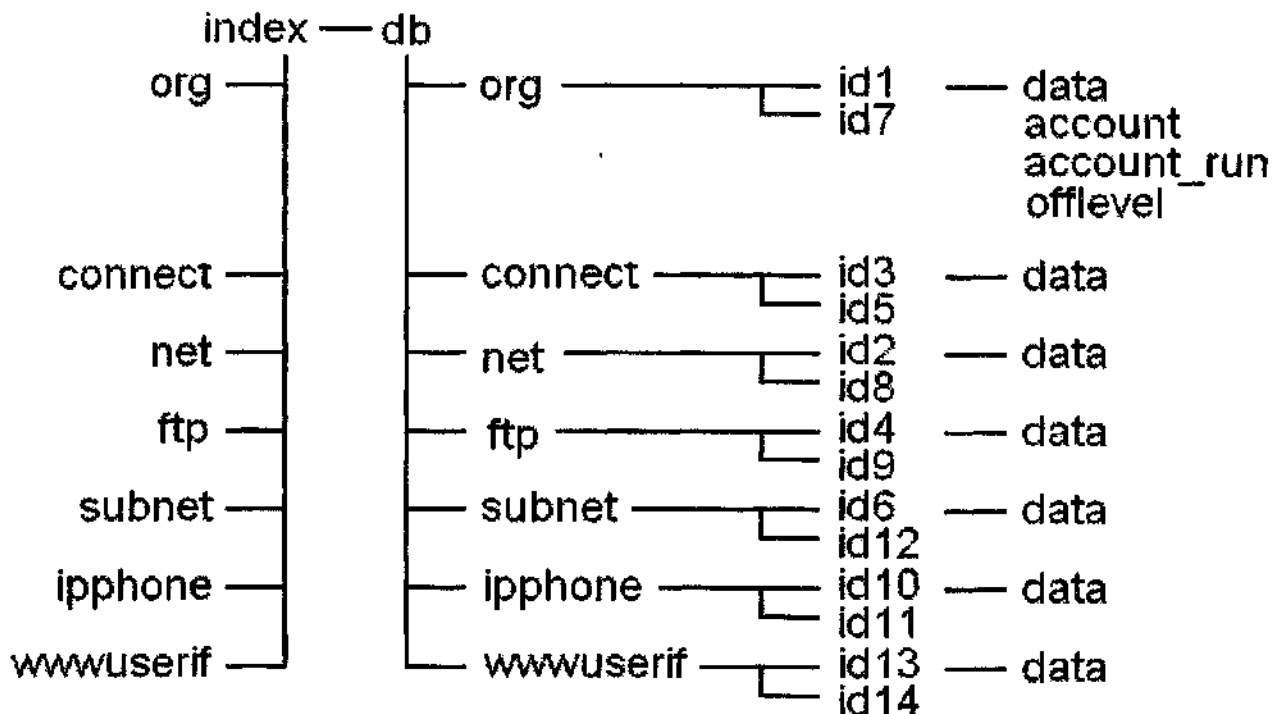


Рис.3

БД также включает в себя также набор индексных файлов. Индексные файлы содержат в себе основную информацию из БД на текущий момент времени, необходимую для быстрой работы системы.

Perl-интерфейс базы данных системы Nadmin представляет собой слой ПО, обеспечивающий доступ к объектам БД.

#### Лицевой счет

Каждый абонент системы Nadmin имеет свой лицевой счет. На лицевой счет заносятся платежи и снимаются деньги за различные услуги. Лицевой счет представлен RCS-файлом ACCOUNT. Строки файла ACCOUNT бывают двух видов - строки расходов и строки платежей. Иногда абонент может иметь несколько лицевых счетов. Регистрация расходов абонента ведется в реальном времени.

Одной из особенностей новой версии системы является введение понятия уровня отключения. Уровень отключения - это минимальное значение суммы на лицевом счете, при котором происходит отключение абонента. Уровень отключения формируется совместно системой и абонентом. По умолчанию он равен нулю.

Абонент может сделать уровень своего отключения положительным (но не больше суммы на его лицевом счете). Это необходимо, например, если абонент ставит перед собой цель тратить в некоторый период фиксированную сумму со своего лицевого счета.

Уровень отключения меньше нуля допустим не для всех абонентов. Если расходы абонента в течение нескольких расчетных периодов не опускались до уровня отключения, то получается, что абонент кредитовал СТ "Ботик". Поэтому СТ "Ботик" может кредитовать данного абонента, разрешив ему делать свой уровень отключения меньше уровня по умолчанию. При достижении суммы на лицевом счете уровня отключения абонент автоматически отключается от сети. Если абонент был отключен и после платежа новая сумма на лицевом счете превышает уровень отключения, абонент автоматически подключается к сети.

#### Основной процесс-обработчик (ОПО)

ОПО является ядром расчетной системы (рис.4). В его обязанности входит получение порций статистики от сенсоров и запуск соответствующей расчетной (биллинговой) функции, которая производит расчет стоимости потребления некоторого ресурса. После того, как подсчитана стоимость и проведены все бухгалтерские транзакции, порция статистики архивируется с помощью специального класса Archive

В состав ОПО входит справочник - вспомогательная часть ОПО. В его обязанности входит:

- построение ретроспективного кэша на заданную глубину истории объектов;
- определение услуги по идентификатору и дате;
- обновления кэша в случае изменения БД.

Биллинговые функции являются специфичным механизмом определения стоимости оказанной услуги. По этому они вынесены в конфигурационный файл. С одной стороны, язык описания должен быть достаточно простым, с другой, автор биллинговых функций не должен быть ограничен специализированным метаязыком. Исходя из того, что авторами биллинговых функций чаще всего являются системные программисты, было решено использовать язык Perl для их написания.

#### Веб-интерфейс

Для реализации пользовательского интерфейса был разработан инструментарий для создания веб-страниц на основе шаблонов. Шаблоном веб-страницы является обычный html-документ с псевдокомментариями, которые заменяются на необходимые данные из системы Nadmin.

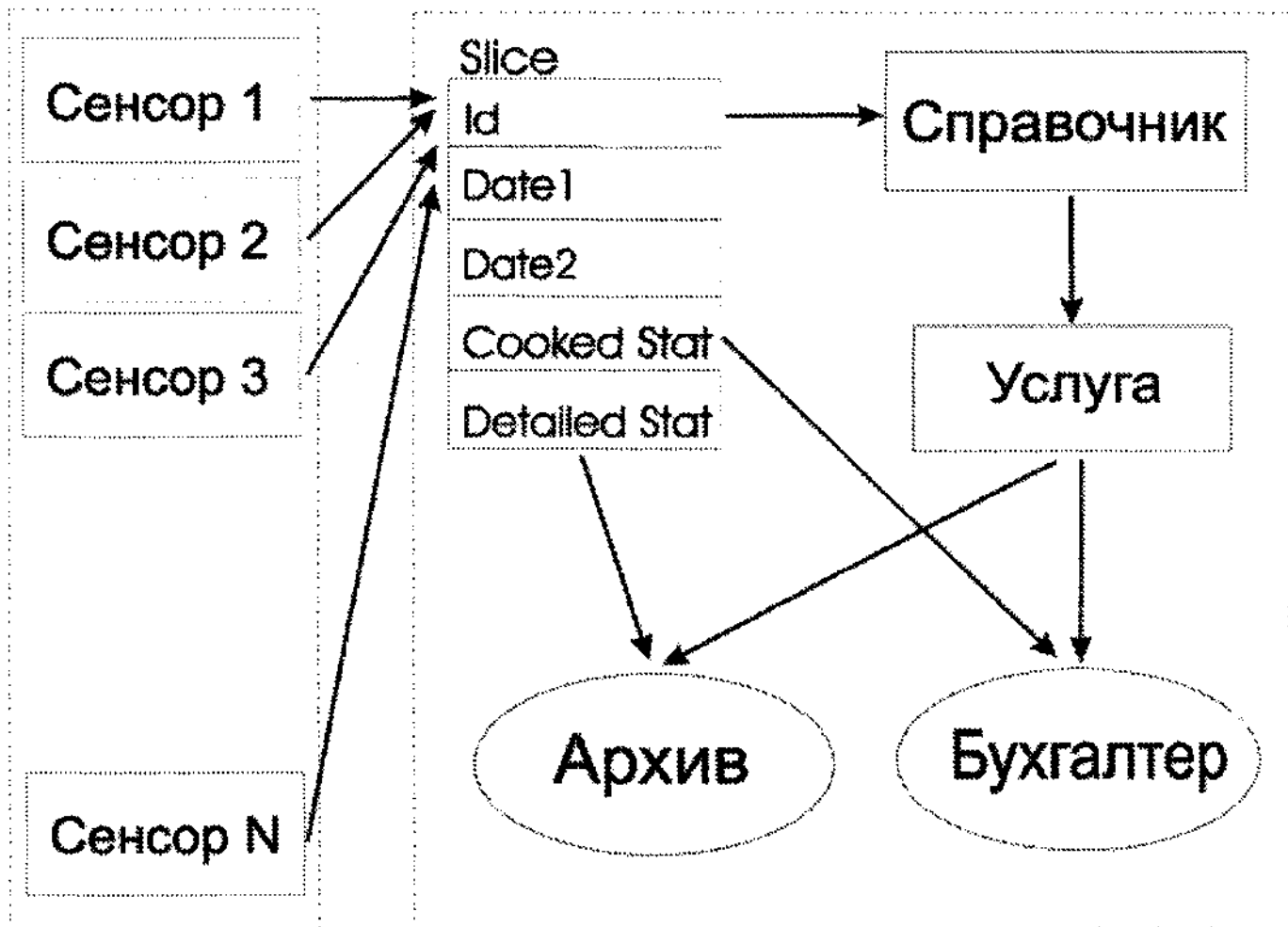


Рис. 4

Существует два вида интерфейсов: администраторский и пользовательский. Администраторский интерфейс предназначен для обслуживающего персонала СТ "Ботик". Пользовательский интерфейс предназначен для абонентов СТ "Ботик". На пользовательском интерфейсе может быть два вида пользователей:

- обычный пользователь;
- администратор.

#### Архив

Для аргументированного решения вопросов абонентов относительно происхождения трафика необходимо хранение статистической информации за весь расчетный период с достаточной степенью детализации. При возникновении претензий администратор должен иметь возможность предъявить абоненту данные, свидетельствующие о потреблении ресурса, со временем и указанием типа сервиса. Для этого вся статистика после анализа и расчета переносится в архив.

#### Состояние проекта

На сегодняшний день реализованы все основные компоненты системы. Успешно реализовано одно из основных свойств новой системы - обсчет статистики в реальном времени. Система запущена в экспериментальную экс-

плуатацию. По имеющимся результатам уже можно сделать вывод, что использованные технические решения полностью оправдывают себя.

Результаты текущей работы докладывались на VIII студенческой научной конференции (НОУ ИПС Университет города Переславля, 2004 г). Работа опубликована в сборнике трудов, посвященном 20-летию Института программных систем Российской академии наук "Программные системы: теория и приложения".