УДК 004.428.2

СИСТЕМА МОНИТОРИНГА ВЕНДИНГОВЫХ АППАРАТОВ В РЕАЛЬНОМ ВРЕМЕНИ

А. Ю. Смолянинов, В. А. Крайванова
ФГБОУ ВПО «Алтайский государственный технический университет им. И.И. Ползунова»,
г. Барнаул

Статья посвящена описанию архитектуры программного комплекса для мониторинга состояния вендинговых аппаратов. Описан общий подход к построению архитектуры, основные модули и процессы взаимодействия между ними.

Ключевые слова: архитектура ПО, мониторинг технических систем, системы реального времени

Современные технологии позволяют оперировать большими объемами данных с высокой скоростью, открывая ранее недоступные возможности для предприятий торговли. Одна из них – сети торговых автоматов. Однако сейчас торговые автоматы не предоставляют достаточных средств удаленного мониторинга, что накладывает существенное ограничение на расширение таких сетей. Когда количество автоматов превышает сотню, для их поддержания требуется большое количество обслуживающего персонала: необходимо вовремя отслеживать поломки неисправности, рассчитывать показатели логистики для маршрутов экспедиторов и пр. Разработка системы удаленного мониторинга значительно снизит трудоемкость поддержки, а значит, увеличит прибыль. Целью работы является реализация системы удаленного мониторинга вендинговых автоматов, которая собирает информацию о состоянии автоматов и остатке товара и передает ее диспетчерам – пользователям системы. Кроме того, система рассчитывает прогноз расхода продукта и вероятности возможных неисправностей на основе доступной статистики.

Архитектура разработанной системы представлена на рисунке 1. На каждом торговом автомате имеется устройство выхода в сеть и настройки подключения. При включении автомат переходит в режим передачи данных о своем состоянии и следующих показателях: остаток товара, текущий режим взаимодействия (продажа, обслуживание, блокировка), состояние аппаратных узлов (для отслеживания поломок). При изменении любого параметра данные передаются на Сервер 1 по специально разработанному двоичному протоколу. В протоколе содержится около 50 команд, специфичных для оборудования автомата.

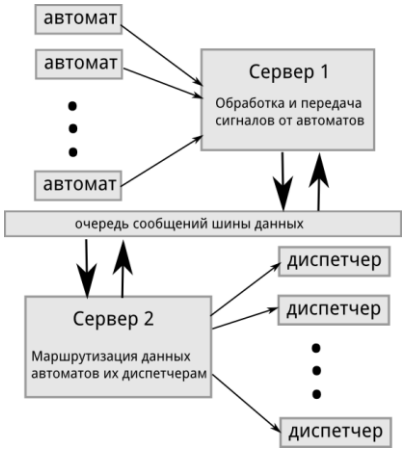

Рисунок 1 − Общая архитектура системы

Протокол гарантирует доставку сообщения на Сервер 1. Получив сообщение, Сервер 1 «на лету» перекодирует его бинарного вида в формат JSON и размещает на шине данных. Также сообщение записывается в базу данных, играющую роль лога событий, на основе которого составляются отчеты и осуществляется просмотр всех предыдущих событий.

Сервер 2 отвечает за отправку управляющих команд. Это служебные команды для управления аппаратом, такие

как: перезагрузка, переход в режим блокировки, очистка и оперативное отображение данных для диспетчеров.

Внутри оба сервера представляют собой кластеры серверов с балансировщиком нагрузки – специальным компонентом системы, осуществляющим маршрутизацию сообщений между аналогичными элементами системы с целью максимально оптимального распределения нагрузки между ними. Взаимодействие между Сервером 1 и Сервером 2 осуществляется с использованием технологии шины данных. Благодаря этому обеспечивается отказоустойчивость и абстрагирование от механизма соединения. Рассмотрим подробнее внутреннюю архитектуру Сервера 2. Для обеспечения работы в многопользовательском режиме Сервер 2 использует компонент авторизации на основе механизма RBAC (управление доступом на основе ролей). Механизм RBAC позволяет распределять привилегии между пользователями системы. Набор логически схожих привилегий группируется в роль, каждый пользователь может иметь несколько ролей. Таким образом, в результате комбинирования ролей, каждый пользователь имеет свой набор привилегий. Кроме индивидуального набора привилегий каждый пользователь имеет набор аппаратов, которые ему назначены. Также, каждый аппарат имеет множество пользователей, которые его отслеживают. При подключении клиента сервер получает список аппаратов, назначенных пользователю, и список его привилегий. Сообщения о событиях от вендинговых аппаратов отсылаются только тем пользователям, которые имеют права на их получение. Для ускорения процесса рассылки уведомлений, на Сервере 2 формируется набор «комнат» (рисунок 2).

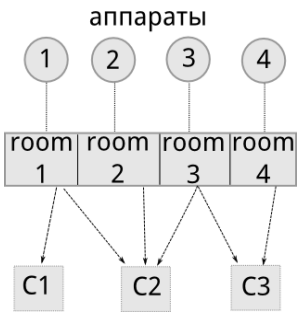

Рисунок 2 – логическая схема системы комнат.

Каждая комната логически соответствует конкретному аппарату. В процессе подключения каждый пользователь «размещается» в соответствующих его аппаратам комнатах и, таким образом, получает уведомления только от назначенных ему аппаратов.

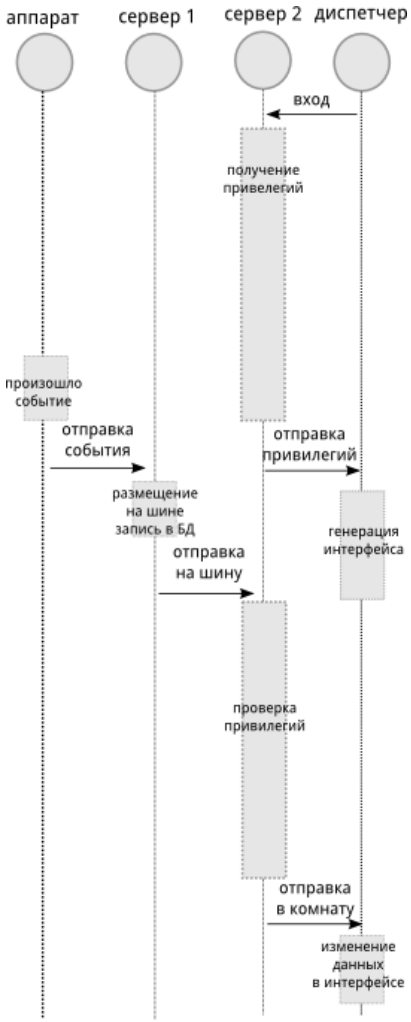

Рисунок 3 – Цепочка событий.

Для выполнения требования работы в реальном времени соединение между клиентом и сервером открыто постоянно, то есть клиент получает данные без предварительного запроса. Права пользователей не статичны, поэтому при каждой передаче данных сервер обязан проверить привилегии клиента.

Также набор привилегий используется клиентом для генерации интерфейса диспетчера (Рисунок 3). Получив список своих привилегий, клиентское приложение на

их основании генерирует пользовательский интерфейс.

Интерфейс диспетчера представляет собой одностраничное веб-приложение. После авторизации и получения/изменения набора привелегий для формирования структуры приложения используется компонент генерации пользовательского интерфейса. Этот компонент содержит в себе шаблоны управляющих элементов для каждой привилегии. Получая список привилегий, соответствующие шаблоны размещаются на экране пользователя в определенных местах. При получении оповещения об изменении набора привилегий происходит перегенерация интерфейса.

Для хранения структуры пользовательских ролей удобно пользоваться базой данных с реляционной моделью доступа. Реляционная модель позволяет гибко управлять ролевыми компонентами, прозрачно демонстрируя связи между пользователями и их привилегиями.

Однако поддержка ссылочной целостности создает большие накладные расходы на СУБД, что существенно сказывается на скорости получения данных. Для формирования списка привилегий, которыми обладает пользователь, необходимо выполнить запрос к базе данных, собирающий все роли пользователя и их привилегии. Такой же запрос необходимо выполнять так же и при их проверке.

Модель, построенная в терминах реляционной базы данных, дает возможность легко оперировать абстракциями ролей и привилегий, т. к. связи пользователей и их ролей очень логично описываются с помощью таблично-ссылочной системы. Однако цена использования таких средств – это скорость доступа к данным.

Таким образом, проверка прав доступа в данной цепочке событий (Рисунок 3) является узким местом. Разработанная модель позволяет эффективно решать поставленную задачу, однако обращение к БД занимает большую часть времени всего процесса и требует оптимизации, которую планируется провести на следующем этапе.

СПИСОК ЛИТЕРАТУРЫ

1. Rohit R. Socket.IO Real-time Web Application Development [Текст]. / R. Rohit – PACKT Publishing, 2013 – 140 с.

2. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования [Текст]. / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес – СПб: Питер, 2001. – 381 с.

3. PostgreSQL: Documentation [Электронный ресурс]. // postgresql.org – 2014. – Режим доступа: http://www.postgresql.org/docs, свободный.

4. Управление доступом на основе ролей [Электронный ресурс]. // wikipedia.org – 2014. – Режим доступа: ru.wikipedia.org/wiki/Управление_доступом_на_основе_ролей, свободный.

5. Ксензов М. Рефакторинг архитектуры программного обеспечения [Электронный ресурс]. // М. Ксензов, citforum.ru – 2004. – Режим доступа: http://citforum.ru/SE/project/refactor/, свободный.

6. Масштабируемая Веб-архитектура и распределенные системы [Электронный ресурс]. // rus-linux.net – 2014. – Режим доступа: rus-linux.net/MyLDP/BOOKS/Architecture-Open-Source-Applications/Vol-2/Scalable-Web-Architecture-01.html , свободный.

7. Особенности архитектур реального времени [Электронный ресурс]. // www.rae.ru – 2014. – Режим доступа: http://www.rae.ru/snt/section=content&op=show_article&article_id=2572, свободный.

Смолянинов Анатолий Юрьевич – студент: (923) 650 8313, e-mail: zarkone@ya.ru; Крайванова Варвара Андреевна – к. ф.-м.н.: +7 913 230 34 19, e-mail: krayvanova@yandex.ru.