

Н. С. Дудаков, Н. Е. Пирогов

РАЗРАБОТКА ГИБРИДНОЙ МОДЕЛИ СИСТЕМЫ БЕЗОПАСНОГО УПРАВЛЕНИЯ ХРАНЕНИЕМ ДАННЫХ

В настоящее время все большую важность приобретают вопросы эффективного и безопасного управления хранением, доступом и использованием информации. Внедряются системы автоматизированного управления для оперативной, гибкой и эффективной работы с информацией. Объемы информации и сложность производимых операций возрастают, и организация хранения, доступа и обработки данных становится одним из наиболее важных моментов взаимодействия в информационной среде [1].

В современной распределенной системе данных доступ к информации одновременно получает большое число пользователей, наделенных различными правами. Система защиты хранилища данных должна решать две принципиальные задачи: обеспечивать разделение прав пользователей и осуществлять безопасную передачу данных по локальной сети.

Используемая в большинстве информационных систем архитектура «клиент-сервер», предполагающая централизованное хранение данных, зачастую имеет существенные недостатки, такие как большая нагрузка на сервер и высокие требования к каналам передачи данных между клиентом и сервером. Подход, реализуемый в локальных базах данных, — хранение копии базы данных на каждом рабочем месте — решает указанные проблемы, однако появляется необходимость решения задачи синхронизации информации в клиентских и серверных копиях базы данных. Также при работе с локальной конфигурацией появляется ограничение на размеры баз данных. Возможной альтернативой может быть использование корпоративных высокопроизводительных баз данных или увеличение мощности серверного оборудования, но такие подходы являются дорогостоящими, и для некоторых проектов неприемлем закрытый код корпоративных СУБД.

В работе предлагается разделять хранимые данные и использовать существующие, взаимодействующие в той или иной степени программные решения. Актуальным и эффективным является разделение данных по темпам обновления и доступа на сравнительно более медленную статическую и быструю динамическую части. Статические данные характеризуются низкой частотой обновления и доступа, к ним можно отнести классификационные данные, различную справочную информацию. В то же время статическая информация имеет достаточно большой объем, и целесообразно хранить ее на серверах баз данных. Динамические данные, в свою очередь, характеризуются высоким темпом обновления, при этом динамическая часть имеет сравнительно меньший объем данных.

В качестве основы для статической части данных предлагается использовать объектно-реляционную СУБД PostgreSQL [2]. PostgreSQL является свободной альтернативой коммерческим СУБД, базируется на языке SQL и поддерживает стандарт SQL 2003. Основными достоинствами данной СУБД являются поддержка баз данных практически неограниченного размера и мощные и надежные механизмы транзакций и репликаций.

Для динамической части предлагается локальная конфигурация системы, согласно которой копии базы данных хранятся одновременно на серверах и на рабочих местах. Запросы на чтение осуществляются из локальных баз данных, а все изменения согласовываются с сервером и тиражируются по подключенным рабочим местам. В этом случае серверное программное обеспечение обрабатывает только запросы на модификацию данных, но более трудоемкой является задача поддержания целостности данных и синхронизации данных на рабочих местах. В соответствии с поставленными требованиями, а также по результатам серии исследований и



тестов была выбрана база данных SQLite — открытая встраиваемая библиотека, поддерживающая стандарт SQL 92. Интерфейс SQLite реализован в одной библиотеке, а сами данные возможно хранить в единственном файле. SQLite является производительной базой данных, благодаря высокоупорядоченной внутренней архитектуре [3]. При создании прототипа системы, основанной на локальной базе данных SQLite, была разработана схема взаимодействия баз данных в локальной сети, реализована клиент-серверная архитектура системы.

Далее системы управления данными, отвечающие за статические и динамические части, были объединены в единую гибридную систему управления хранением данных. Был разработан объединенный интерфейс гибридной системы, позволяющий разработчикам получать централизованный доступ как к статической СУБД, так и к клиент-серверной архитектуре хранилища динамических данных. Передача информации осуществляется по защищенным каналам связи, кроме того, обмен информацией между модулями системы позволяет применять любые дополнительные алгоритмы кодирования данных для обеспечения большей безопасности соединения. Кроссплатформенность созданной системы достигается за счет использования библиотеки Qt при проектировании модулей [4].

Таким образом, разработана гибридная модель системы безопасного управления хранением данных. Эффективность работы системы достигается за счет разделения хранимых данных и использования наиболее подходящих доработанных существующих программных решений. В качестве «фундамента» системы предложен вариант использования баз данных PostgreSQL и SQLite для статической и динамической частей данных соответственно. Система сохраняет всю функциональность и производительность исходных баз данных и является эффективным решением для хранения данных с различными темпами обновления с поддержанием политики безопасности.

СПИСОК ЛИТЕРАТУРЫ:

1. Дейт К. Дж. Введение в системы баз данных. М., 2001.
2. Документация по базе данных PostgreSQL. URL: <http://www.postgresql.org>.
3. Документация по базе данных SQLite. URL: www.sqlite.org.
4. Шлеер М. Qt4: Профессиональное программирование на C++. СПб., 2007.

А. А. Дураковский, С. В. Дворянкин

МОДЕЛИРОВАНИЕ УПРАВЛЯЮЩЕЙ ИНФОРМАЦИИ ДЛЯ СИСТЕМЫ ЛОГИСТИЧЕСКОЙ ПОДДЕРЖКИ АППАРАТНО-ПРОГРАММНЫХ СРЕДСТВ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ РАСПРЕДЕЛЕННЫХ СИСТЕМ ОБРАБОТКИ ДАННЫХ

Главным критерием безопасности эксплуатации является отсутствие недопустимого риска. Поскольку объективно риск причинения вреда зависит не только от характера и качества технических средств и качества процессов [1, 2], но также от поведения человека, требования по безопасности эксплуатации не могут быть абсолютными, а должны устанавливать допустимую степень риска. Следовательно, понятие «риск» неразрывно связано с понятием безопасности

