

УДК 681.5

**ВЛИЯНИЕ РЕЛЯЦИОННОЙ МОДЕЛИ ДАННЫХ НА  
КАЧЕСТВЕННЫЕ ЭКСПЛУАТАЦИОННЫЕ  
ХАРАКТЕРИСТИКИ ВЫСОКОНАГРУЖЕННЫХ  
ИНТЕРНЕТ-ПРОЕКТОВ**

*к. т. н., доцент Доманецкая И. Н.\**

*Матейко Я. В.\*\**

*к. т. н., доцент Федусенко Е. В.\**

*к. т. н., доцент Хроленко В. Н.\**

*аспирант Федусенко А. А.\**

*\*ГВУЗ «Киевский национальный университет строительства и архитектуры», г. Киев*

*\*\* Научно-образовательный центр «Инновационные решения», г. Киев*

**Постановка проблемы.** Стремительно растущие объемы информации в условиях многопользовательского режима работы современных интернет-проектов порождают новые сложные задачи по организации ее хранения и обработки. Как правило, осознание проблемы, связанной с большими нагрузками на сервис, приходит эволюционным путем. То есть тот функционал, который еще вчера работал хорошо, сегодня «вдруг» начинает работать с неудовлетворительной скоростью и отказоустойчивостью. Анализ первопричин и механизмов недееспособности традиционной модели масштабирования системы хранения и обработки данных в условиях больших объемов данных составляет предмет исследований данной работы.

**Связь с научными и практическими задачами и анализ последних исследований и публикаций.** Реляционные системы управления данными(СУБД) на сегодняшний день являются одним из самых распространенных средств представления и манипулирования информацией в компьютерных системах. Однако реляционная модель не обеспечивает обработку неструктурированной информации, и, более того, практически не имеет возможности хранения мультимедийных данных. К тому же персонализация услуг в рамках транзакционных баз данных и рост объемов информации приводят к снижению качественных эксплуатационных характеристик обработки данных средствами реляционных СУБД.

Вопросам оптимизация эффективности работы баз данных посвящено множество академических исследований, печатаются работы компаний, направленные на решение этой задачи в рамках отдельных СУБД. Кроме того, по данной тематике проведено большое количество конференций, таких как HighLoad ++ (2007, 2008, 2009, 2010, 2011 и 2012, 2013)[2,4]. С появлением большого количества высоконагруженных Интернет-проектов, которые оперируют большими объемами данных и имеют большое количество посетителей, для которых важнейшей задачей является обеспечение постоянной высокой пропускной способности хранилища при неограниченном увеличении объемов данных начал развиваться направление NoSQL. На сегодняшний день создано большое количество NoSQL решений, написано много работ и проведено специализированных конференций (NoSQL matters Conference (2013), NoSQL NOW (2013)) [5].

**Формулировка целей статьи.** Цель данной публикации состоит в анализе наиболее распространенных шаблонов проектирования интерфейсов интернет-приложений и определения соответствующих им типовых запросов к реляционной базе; изложении методики проведения и результатов тестирования времени выполнения типовых запросов к базе данных в зависимости от объема данных; определении потенциальных факторов уменьшения производительности высоконагруженных систем.

**Изложение основного материала исследований** Современные WEB системы выставляют все более возрастающие требования к системным ресурсам как: процессорное время, оперативная память, дисковое пространство, загруженность дисковой подсистемы количеством операций в секунду. Для них типичной становится ситуация, когда рост аудитории опережает наращивание мощностей системы и наступает момент, когда уже достигнута верхняя граница возможностей архитектуры (и добавления нового компьютера ничего не решает, или решает на очень короткое время) [3].

Архитектура интернет-приложений, как правило, состоит из трех уровней доступа. Первый уровень: HTTP сервер. Выполняет роль маршрутизатора запросов к серверам бизнес-логики. Второй уровень: серверы бизнес-логики, могут быть выполнены на любом языке программирования. Выполняют основные операции с базой данных, обрабатывают все запросы, формируют ответы на запросы. Третий уровень: серверы баз данных. Выполняет авторизацию запросов к источникам данных, их обработку и выдачу результатов на запросы. Очень часто вся инфраструктура построена на базе реляционной

модели работы с данными. Таким образом, все бизнес-процессы зависят от особенностей реализации реляционной базы данных. Традиционный путь масштабирования - увеличение системных ресурсов системы, перестает быть эффективным в случае приближения характеристик системы к эксплуатационным лимитам.

Интерактивность – одно из основополагающих качеств высоконагруженной системы интернет-проектов. При проектировании интерфейсов интернет-приложений, дизайнеры интерфейсов обычно используют определенные правила, шаблоны и методики проектирования. В работе [1] определены шаблоны проектирования интерфейсов WEB-приложений типа интернет-магазинов. К шаблонам проектирования интерфейсов можно отнести следующие:

- динамично дополняемая страница результатов;
- страница выдачи результатов с распределением (пагинацией) данных на сегменты;
- страница выдачи результатов с распределением (пагинацией) данных на страницы и фильтрацией по критериям (каталоги, фильтры).

Динамично дополняемая страница результатов обычно используется для отображения результирующего набора данных, который неудобно просматривать полностью из соображений удобства пользователя этот набор данных дополняется по мере того, как пользователь, просматривает его. Например: стена сообщений социальных сетей, некоторые поисковые системы. Страница выдачи результатов с распределением (пагинацией) данных на сегменты обычно используется в поисковых и других системах, где необходимо разделить результирующий набор данных на сегменты для более удобной навигации. Например: поисковые системы, порталы, каталоги товаров. Страница выдачи результатов с распределением (пагинацией) данных на сегменты и фильтрацией по критериям (каталоги, фильтры), дополняет обычную страницу с пагинацией фильтрами. Фильтры позволяют выделить некоторое множество данных из результирующего набора этих данных, по определенному критерию.

На примере работы действующего интернет-магазина по продаже автозапчастей Drozd.kiev.ua рассмотрим с учетом каких шаблонов проектирования интерфейсов построены функциональные блоки типового интернет-магазина. Так, блок новостей и поиск могут быть реализованы как динамично дополняемая страница. Доступ к каталогу может быть реализован в свою очередь, с учетом особенностей страницы выдачи результатов с распределением (пагинацией) данных на сегменты и фильтрацией по критериям.

С точки зрения архитектора программного обеспечения реализация функциональности интернет-магазина на уровне работы с данными будет сведена к решению следующих задач:

- выборка из набора данных первых N-элементов
- выборка из набора данных последних N-элементов
- поиск в наборе данных
- фильтрация набора данных
- поиск элемента по его идентификатору

Для определения временных затрат при типичных сценариях работы интернет-приложений, было проведено тестирование зависимости затрат времени на выполнение типовых выборок на больших объемах данных.

Конфигурация стенда: Processor: Intel (R) Core (TM) 2 Quad CPU Q8300 @ 2.50GHz (4 CPUs) ~ 3.0GHz; Memory: 8192MB RAM; Fixed hard disk: SAMSUNG SP2004C 7200 rpm 200gb. Программное обеспечение: Microsoft Windows7 Enterprise Version 6.1.7601 Service Pack 1; СУБД mysql-5.5.15-win32. Общий объем тестовых данных составляет полных 1,000,000 и 10,000,000 строк указанного выше формата.

Тестировались следующие запросы: выборка из набора данных порции из 10 записей с заданным смещением; выборка из набора данных порции из 10 записей с заданным смещением при условии использования индексов по целочисленному и символьному полю с обеспечением сортировки записей в прямом и обратном порядке. Анализируя результаты тестов можно сделать следующие выводы: 1) скорость выборок на массиве данных деградирует прямо пропорционально объему этих данных (общая тенденция); 2) без использования индексов, скорость выборок лежит в одном диапазоне при упорядочении результирующего набора; 3) при использовании индексов и упорядочения система страдает от чрезмерной нагрузки, прямо пропорционально объему. Деграция в скорости напрямую влияет на такие функциональные блоки, где необходимо получать последовательный доступ к данным. Это могут быть новости, товары, заказы и т.д.

Таким образом, опыт показывает, что в случае высоконагруженных интернет-проектов реляционная модель не обеспечивает приемлемых значений временных параметров и суть проблемы лежит в плоскости реализации внутренних механизмов выполнения запросов в реляционных СУБД.

**Выводы.** В данной работе была проанализированы особенности влияния реляционной модели данных на временные характеристики выполнения запросов пользователей. Как показали проведенные тесты увеличение времени выполнения запросов с ростом объемов информации заложено в самой природе реляционной модели. Таким образом, актуальной задачей является разработка такой архитектуры, которая в будущем с минимальными затратами времени и ресурсов, сможет масштабироваться горизонтально, при этом катализатором таких изменений будет выступать потребность в увеличении параллельно выполненных операций в секунду.

### ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

1. Доманецька І.М., Матейко Я. В, Федусенко О.В., Хроленко В.М., Федусенко А.О. *Дослідження впливу моделі даних на ефективність роботи високонавантажених систем // Управління розвитком складних систем.* – 2014. – Вип. 17. – С. 81-89.
2. Клеменков П.А. *Большие данные: современные подходы к хранению и обработке // Труды Института системного программирования*, т. 23, М., ИСП РАН, 2012, стр. 143-158.
3. Martin L. Abbott, Michael T. Fisher. *The art of scalability.* – Addison-Wesley, 2009.
4. Mark A. Beyer, Douglas Laney. *The Importance of 'Big Data': A Definition*, Mark Beyer, Douglas Laney, G00235055
5. Rick Cattell *Scalable SQL and NoSQL Data Stores / Cattell Rick / SIGMOD Record*, December 2010 (Vol. 39, No. 4)