

УДК 004.415.28

АРХИТЕКТУРА ИНТЕРНЕТ-ПЛАТФОРМЫ ДЛЯ ПЕРСОНАЛИЗИРОВАННОЙ МЕДИЦИНЫ

С.П. Орлов¹, В.П. Золотов¹, А.В. Якимов²

¹ Самарский государственный технический университет
Россия, 443100, г. Самара, ул. Молодогвардейская, 244

² НКО «Фонд содействия модернизации здравоохранения»
Россия, 121099, г. Москва, Новый Арбат, 21

Рассматривается проблема обеспечения современными информационными технологиями системы здравоохранения населения. Показано, что данные генома человека должны быть в основе современной информационной системы, построенной на базе интернет-платформы персонализированной медицины. Сформулированы требования к инструментам интернет-платформы. Выделены основные модули, из которых строится функционал интернет-платформы. Предложена архитектура платформы, описаны принципы масштабирования проекта. На основе анализа трехзвенной клиент-серверной архитектуры выбран стек технологий, необходимых для реализации проекта. Решены вопросы горизонтального масштабирования и масштабирования системы во времени. Рассмотрены принципы построения интеллектуальной системы поддержки принятия решений в составе интернет-платформы.

Ключевые слова: информационные технологии, персонализированная медицина, архитектура интернет-платформы, сервисно-ориентированная архитектура, система поддержки принятия решений

Персонализированная медицина – это оказание комплексной медицинской помощи, осуществляемой в соответствии с характеристиками конкретного пациента. Этот термин учитывает не только используемые уже в течение долгого времени клинические признаки, такие как возраст, пол, телосложение, индекс массы тела, сопутствующие заболевания, биохимические показатели, семейный анамнез, позволяющий оценить риск наследственной заболеваемости, и проч., но и в первую очередь молекулярные характеристики организма пациента. Последние включают данные геномных и постгеномных исследований, в том числе протеомики и метаболомики, выполненные при помощи современных высокотехнологичных методов молекулярной биологии и медицины.

Основной целью персонализированной медицины является разделение популяции здоровых и больных людей на индивидуумов, отличающихся по вероятности возникновения заболевания и по способности реагировать на различные виды терапии. Такая классификация позволит рационально и целенаправленно применять терапевтические воздействия с максимальной пользой и минимумом неблагоприятных эффектов для отдельного человека.

Значительный прогресс в развитии персонализированной медицины про-

Сергей Павлович Орлов (д.т.н., проф.), заведующий кафедрой «Вычислительная техника».

Владимир Петрович Золотов (к.т.н., доц.), доцент кафедры «Вычислительная техника».

Андрей Владимирович Якимов, инженер-программист.

изошел благодаря декодированию генома человека, который, по сути, является генетической картой всего организма. Ученым удалось определить 20–25 тысяч человеческих генов и последовательности миллиардов конъюгаций хромосом внутри ДНК, что составляет элементарную структуру набора генов человека.

У любого пациента с течением времени возрастает объем персональных данных о личном здоровье. Как следствие, возникает необходимость их хранения, систематизации и обработки. По мере роста объема этих данных увеличивается и сложность их анализа врачом для постановки правильного диагноза. В то же время возрастает значимость и интенсивность использования информационных технологий в современной системе здравоохранения и персонализированной медицине в частности.

Обозначенные проблемы могут быть решены с помощью предлагаемой интернет-платформы, запуск которой позволяет повысить эффективность за счет оптимизации доступа пользователей и докторов к данным о здоровье пациента. В статье ставится задача разработки интернет-платформы персонализированной медицины как сервиса для централизованного хранения, систематизации, анализа и управления медицинской информацией с доступом к ней через Интернет. Стоит отметить, что функционал платформы должен содержать как возможности для персонального использования, так и функционал для врачебных заведений с целью организации их работы.

Требования и функционал платформы

Анализ проблемы позволил сформулировать основные требования к платформе в целом.

1. Взаимодействие с сервером платформы должно осуществляться по принципу «клиент – сервер» с использованием RESTfulAPI. Это позволит без проблем масштабировать возможность доступа к данным с различных устройств – к примеру, доступа с помощью обычного браузера и с помощью мобильного приложения со смартфона.
2. Доступ к функционалу платформы должен осуществляться через Интернет только по защищенному протоколу (HTTPS). Главное преимущество защищенного соединения (HTTPS) в том, что данные между сервером и клиентом нельзя прочесть.
3. Платформа должна являться многопользовательским приложением, распределенной системой, работающей более чем на одном сервере. По умолчанию предполагается, что в данный момент платформой могут пользоваться несколько пользователей. Операции над данными могут производиться несколькими пользователями одновременно.
4. Пользователем платформы по умолчанию является пациент, но дополнительно пользователь может принадлежать к группам «Эксперт» или «Доктор». В свою очередь, группы «Эксперт» и «Доктор» могут быть объединены в отдельную группу «Медицинское учреждение».
5. Доступный пользователю функционал платформы должен определяться группами, к которым принадлежит данный пользователь.
6. Архитектура приложения сервера должна быть выполнена по модульному принципу. Это позволит добавлять новые функции по мере необходимости без кардинального изменения ядра платформы.
7. Платформа должна поддерживать мультиязычный интерфейс.
8. Обязательна поддержка Open API – системы для сторонних разработчи-

ков, которая предоставляет возможность легко с согласия пользователей получать доступ к их информации для создания на ее базе новых приложений.

На основании этих требований предлагается разработать следующий функционал интернет-платформы персонализированной медицины.

1. Доступ к функционалу платформ возможен только зарегистрированным и авторизованным в системе пользователям. Авторизация осуществляется проверкой правильности ввода логина и пароля посетителя.

2. Авторизованный пользователь-пациент получает доступ к основному элементу системы – электронной медицинской карте (ЭМК), хранилищу личных медицинских данных. В основе создания ЭМК – международные стандарты организации, хранения и обмена медицинскими данными HL7 и OpenEHR. Пациенты – пользователи системы смогут хранить все сведения о своем здоровье в одном месте и иметь к ним доступ в любое удобное время через аккаунт на сайте. ЭМК интегрируется с любыми медицинскими информационными системами. Данные попадают в ЭМК из этих систем, от пациентов, добавляющих свои сканированные медицинские документы или загружающих по Bluetooth данные весов, тонометров, глюкометров и других домашних приборов, и от врачей, которые вносят свои заключения по результатам телемедицинских консультаций. Пациент самостоятельно управляет собственной ЭМК и по своему желанию предоставляет к ней доступ врачам или медицинским учреждениям, с которыми он взаимодействует.

3. Обеспечивается использование телемедицины и удаленных консультаций с любым доктором в системе независимо от текущего геоположения.

4. Консультация должна включать в себя несколько типов:

- видеочат;
- текстовый чат;
- вопрос-ответ с возможностью массового обсуждения.

5. Предусмотрен автоматический анализ параметров организма и мутаций в геноме, которые добавлены пользователем-пациентом.

6. Предусмотрена автоматическая расшифровка анализов пациента, которые добавлены в систему.

7. Структурирована информация справочного характера.

Анализ сформулированных требований и функционала интернет-платформы показывает, что проект будет высоконагруженным по данным. Высоконагруженными ресурсами в первую очередь являются многопользовательские приложения, многие из которых представляют собой распределенные информационные системы, работающие на нескольких серверах. Такая конфигурация необходима для обеспечения обработки больших объемов данных, возникающих при пиковых нагрузках, а также их репликации.

Стек технологий для разработки и запуска платформы

Интернет-платформа персонализированной медицины реализует общепринятую трехзвенную архитектуру.

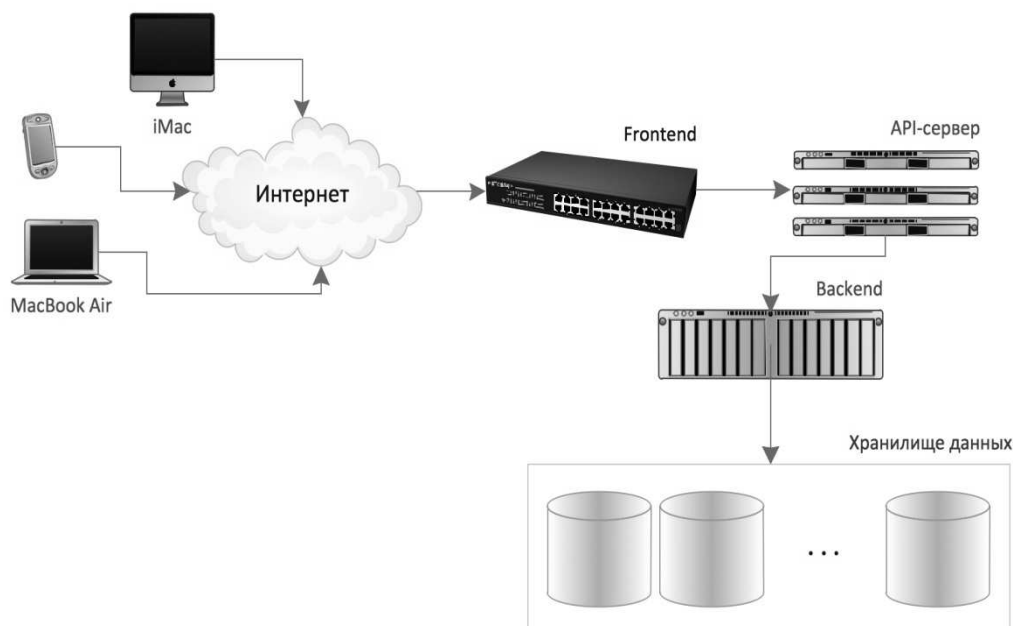
1. Frontend – пул физических серверов, которые отдают статику; передает все, что требует вычислений, дальше на сервер Backend. Обычно на Frontend также возлагается функция балансировки нагрузки.

2. Backend – сервер, который выполняет расчеты и готовит web-страницу или json-данные. На Backend данные не хранятся, он не имеет «памяти» и каждый запрос обрабатывает с нуля. Это позволяет легко добавлять новые серверы в

пул обработчиков.

3. Хранилище – обычно это реляционная база данных с возможностью персистентного хранения данных. Фактически это место, где данные пользователя физически хранятся.

Архитектура интернет-платформы для персонализированной медицины приведена на рисунке.



Архитектура интернет-платформы

Опишем стек технологий, которые будут использоваться в разработке данной платформы.

1. В качестве основного языка программирования для написания ядра платформы на сервере Backend используется скриптовый язык PHP. Текущая версия языка поддерживает ООП, пространство имен и содержит большое количество сторонних библиотек. Для ускорения интерпретации скриптов необходимо использовать eAccelerator – программу, кэширующую байт-код PHP.

2. В качестве движка сервера баз данных для постоянного хранения используется реляционная база данных MySQL. Выбор продиктован хорошими характеристиками быстродействия данного продукта, а также поддержкой шардинга и репликации по умолчанию [1].

3. В качестве движка хранения множеств и списков используется Redis – сетевое журналируемое хранилище данных типа «ключ – значение». Хранилище Redis содержит базу данных в оперативной памяти и снабжено механизмами снимков и журналирования для обеспечения постоянного хранения. Это обеспечивает очень высокую скорость доступа к данным, при этом гарантируется их персистентность.

3. Для кэширования данных используется Memcached – программное обеспечение, реализующее сервис кэширования данных в оперативной памяти на основе хеш-таблицы.

4. На стороне web-клиента (браузер) применяется основной язык программирования – JavaScript. Современная мощь этого языка позволяет сделать из сай-

та по-настоящему интерактивную систему и минимизировать трафик между клиентом и сервером за счет технологии AJAX.

5. Функционал платформы предполагает организацию обычного и видеочата. Для их реализации необходима программная прослойка с неблокирующим вводом/выводом. Для этого идеально подходит Node.js – программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код) и превращающая JavaScript из узкоспециализированного языка в язык общего назначения [2].

6. В качестве web-сервера для Frontend логичнее всего использовать Nginx – HTTP-сервер и обратный прокси-сервер. На данный момент это один из лучших web-серверов для использования в высоконагруженном проекте.

Принцип построения архитектуры платформы – разделение приложения на логические модули, когда приложение делится на составные части (web, БД, кеш) и эти части работают на разных серверах (или группах серверов). С точки зрения производительности и дальнейшего масштабирования это оптимальная схема построения платформы [3].

Однако такое деление не решает задачу надежности и отказоустойчивости. Каждый узел не резервирован, и в случае выхода из строя любого из них перестает работать весь проект в целом. Эту проблему возможно решить с помощью кластера приложений. Обеспечить представление приложения в виде кластера можно с помощью следующих технологий, которые позволяют масштабировать базу данных, кеш, web и представить каждую из составляющих в виде группы взаимозаменяемых серверов:

- вертикальный шардинг (вынесение модулей на отдельные серверы MySQL);
- репликация MySQL и балансирование нагрузки между серверами;
- распределенный кэш данных (memcached);
- непрерывность сессий между веб-серверами (хранение сессий в базе данных);
- кластеризация веб-сервера: синхронизация файлов; балансирование нагрузки между серверами.

С проектной точки зрения создание такого приложения в виде веб-кластера нужно выполнять в два шага:

- разделение веб-сервера и базы данных на два логических уровня;
- построение собственно кластера приложения.

Основная идея, заложенная в организации разработки модулей проекта, – сервис-ориентированная архитектура (SOA). Сервис – это отдельная часть приложения, выполняющая конкретную задачу и имеющая API, через который предоставляется доступ к своим функциям [4].

Сервисы в разрабатываемой интернет-платформе:

- сервис регистрации и авторизации в системе пациентов;
- хранение статистики, к примеру сканов анализов;
- чат и видеоконференции;
- хранилище данных;
- очередь выполнения автоматического анализа введенных пациентом данных.

Сервис-ориентированная архитектура предполагает, что все приложение разделяется на независимые сервисы [5]. Страница веб-приложения, которую видит конечный пользователь-пациент, – это результат работы многих сервисов.

Использование такой стратегии обеспечивает ряд положительных качеств. Во-первых, сервисы проще разрабатывать: есть конкретная задача; имеется API; проще наладить тестирование; разработка не будет зависеть от других частей

приложения. Во-вторых, программисту не надо знать все приложение, достаточно работать со своей частью. В-третьих, сервис, а значит и приложение в целом, проще масштабировать.

Масштабирование

Вопрос масштабирования очень важен для проектов с ожидаемой высокой нагрузкой по трафику и объему обрабатываемых и хранимых данных. Сразу можно обозначить, что вертикальное масштабирование совершенно не подходит для проекта. Предлагаются два типа масштабирования, которые будут активно применяться в процессе работы интернет-платформы персонализированной медицины.

1. Горизонтальное масштабирование – разбиение системы на более мелкие структурные компоненты и разнесение их по отдельным физическим машинам (или их группам) и/или увеличение количества серверов, параллельно выполняющих одну и ту же функцию. Масштабируемость в этом контексте означает возможность добавлять к системе новые узлы, серверы, процессоры для увеличения общей производительности. Этот способ масштабирования требует аккуратного проектирования на начальном этапе, чтобы сразу предусмотреть в программной части работу с распределенным пулом серверов.

2. Масштабирование во времени – очень важная парадигма проектирования высоконагруженных проектов в интернете. Смысл в том, что не все действия в интернет-приложении должны происходить немедленно. Допустим, врач добавил ответ-комментарий на вопрос пациента и пользователю должно прийти уведомление по e-mail. Нет необходимости сразу посылать email-письмо. Гораздо правильнее поставить задание "послать email" в очередь на специальном сервере и обработать по возможности. Такой подход призван разгрузить Backend и сделать отклик системы фактически реактивным, т. е. с минимальной задержкой, невидимой для пользователя.

Принятие решений

Важнейшим моментом для эффективного использования интернет-платформы в персонализированной медицине является процедура принятия решений врачом или медицинским работником (в том числе и управленческим персоналом) по обслуживаемому пациенту. Очевидно, что эта процедура многоступенчатая и проводится в условиях неопределенности данных. Следовательно, необходимо использовать интеллектуальную систему поддержки принятия решений [6, 7], которая должна генерировать альтернативные решения по диагностике, методам медикаментозного и терапевтического лечения и т. д. Предлагается построить систему, которая, опираясь на хранилище данных, будет основываться на продукционных моделях вывода с использованием баз знаний [8, 9]. Технологии этого типа строятся на принципах многомерного представления и анализа данных (OLAP).

Заключение

Описанный подход к построению архитектуры интернет-платформы персонализированной медицины базируется в первую очередь на достижениях геномной медицины, современных методах диагностики с использованием информационных технологий. Развитие предлагаемой системы позволит повысить уровень жизни населения за счет нового качества работы органов здравоохранения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Шварц, Б. MySQL. Оптимизация производительности. 2-е изд. Пер. с англ. / Б. Шварц, П. Зайцев, В. Ткаченко, Дж. Заводны, А. Ленц, Д. Бэллинг. – СПб.: Символ-Плюс, 2010. – 832 с.
2. Кантелон М. K19 Node.js в действии / М. Кантелон, М. Харетер, Т. Головайчук. – СПб.: Питер, 2015. – 448 с.
3. Скляр Д. РНР. Рецепты программирования. 3-е изд. / Д. Скляр, А. Тахтенберг. – СПб.: Питер, 2015. – 784 с.
4. Архитектура высоконагруженных систем // InsightIT [2013–2013]. Дата обновления: 08.11.2013. URL: <http://www.insight-it.ru/highload/> (дата обращения: 02.10.2015).
5. Масштабирование нагрузки web-приложений // habrahabr. [2011–2011]. Дата обновления: 11.02.2011. URL: <http://habrahabr.ru/post/113992/> (дата обращения: 12.10.2015).
6. Нечаев Д.А. Модели анализа и принятия решений при управлении региональными программами / Д.А. Нечаев, С.П. Орлов // Системы управления и информационные технологии. – 2013. – № 2(52). – С. 35-38.
7. Орлов С.П. Система искусственного интеллекта при управлении целевыми программами развития регионов / С.П. Орлов // Вестник Волжского университета им. Татищева. – 2015. – № 2(24). – С. 63-71.
8. Орлов С.П. Разработка экспертной системы в составе интеллектуальной системы поддержки принятия решений в области водоподготовки и водоочистки природных вод / С.П. Орлов, А.В. Чуваков, Д.А. Нечаев // Современные наукоемкие технологии. – 2010. – № 5. – С. 44-52.
9. Батищев В.И. Комплексный подход к формированию алгоритмов логического вывода для анализа состояния сложных систем / В.И. Батищев, Н.Г. Губанов, С.П. Орлов // Мат-лы VII междунар. науч.-практ. конф. «Татищевские чтения: актуальные проблемы науки и практики: Информационные технологии в организации производства». – Тольятти: Волжский ун-т им. В.Н. Татищева, 2010. – С. 98-101.

Статья поступила в редакцию 12 октября 2015 г.

THE ARCHITECTURE OF THE WEB-BASED PLATFORM FOR PERSONALIZED MEDICINE

S.P. Orlov¹, V.P. Zolotov¹, A.V. Yakimov²

¹ Samara State Technical University
244, Molodogvardeyskaya st., Samara, 443100, Russian Federation

² NPO "Fund of assistance to modernize health care"
21, Novyi Arbat st., Moscow, 121099, Russian Federation

The paper deals with the problem of modern information technologies in the public health system. It is shown that the human genome should be placed to the heart of the modern information system designed on the basis of the web-based platform of personalized medicine. The requirements to the web-based platform tools are formulated. The basic modules which comprise the functional of the web-based platform are outlined. The paper offers the architecture of the web-based platform and describes the principles of the project scaling. On the basis of the three-tier client-server architecture analysis, a stack of techniques required for the project is selected. The issues of horizontal scaling and system-time scaling are discussed. The principles of designing an intelligent decision support system as part of the web-based platform are proposed.

Keywords: *information technologies, personalized medicine, architecture of internet-platform, service-oriented architecture, decision support system.*

Sergey P. Orlov (Dr. Sci. (Techn.)), Professor.

Vladimir P. Zolotov (Ph.D. (Techn.)), Associate Professor.

Andrey V. Yakimov, Engineer.