**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ**

**Кафедра веб-технологий и компьютерного моделирования**

ХАРИТОНОВА

Вероника Ренальдовна

**Облачные сервисы. Разработка веб-приложения  
с использованием Google Cloud Storage**

Дипломная работа

|  |  |
| --- | --- |
|  | Научный руководитель: кандидат физ.-мат. наук,  доцент К.Г. Кузьмин |
| Допущена к защите  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2017 г.  Заведующий кафедрой  веб-технологий и компьютерного моделирования доктор физ.-мат. наук, профессор И.И. Иванов |  |

Минск, 2017

Оглавление

[Перечень условных обозначений и сокращений 3](#_Toc463772482)

[Реферат 4](#_Toc463772483)

[Введение 5](#_Toc463772484)

[Глава 1. Основные сведения об облачных технологиях 6](#_Toc463772485)

[1.1 Общая информация 6](#_Toc463772486)

[1.2 Модели обслуживания 6](#_Toc463772487)

[1.3 Основные свойства облачных технологий 8](#_Toc463772488)

[1.4 Основные поставщики облачных услуг 8](#_Toc463772489)

[Глава 2. Реализация проекта 10](#_Toc463772490)

[2.1 Основная идея и возможности системы 10](#_Toc463772491)

[2.2 Архитектура проекта 11](#_Toc463772492)

[2.3 Основные используемые фреймворки 12](#_Toc463772493)

[2.3.1 Spring 12](#_Toc463772494)

[2.3.2 Hibernate 12](#_Toc463772495)

[2.3.3 Apache Tiles 13](#_Toc463772496)

[Глава 3. Перспективы развития 14](#_Toc463772497)

[Заключение 15](#_Toc463772498)

[Список использованных источников 16](#_Toc463772499)

[Приложение 1. Схема базы данных 17](#_Toc463772500)

Перечень условных обозначений и сокращений

ПО – программное обеспечение

SaaS – Software as a Service, программное обеспечение как услуга

PaaS – Platform as a Service, платформа как услуга

IaaS – Infrastructure as a Service, инфраструктура как услуга

Реферат

Дипломная работа содержит: 36 страниц, 6 иллюстраций (рисунков), 1 приложение, 8 использованных литературных источников.

Ключевые слова: GOOGLE CLOUD STORAGE, ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ, SPRING, HIBERNATE.

Объектом исследования являются облачные технологии и их использование при разработке веб-приложения.

Целью дипломной работы является исследование облачных технологий.

В результате исследования получены следующие результаты:

• описаны существующие на данный момент виды облачных технологий,

• описаны способы использования облачных технологий,

• написано приложение, в котором используется облачное хранилище.

Областью применения являются Интернет-ресурсы.

Дипломная работа выполнена автором самостоятельно.

Введение

В настоящее время происходит быстрое развитие облачных технологий. Существует множество различных облачных сервисов, например виртуальные машины, хранилища, базы данных. Они предоставляются различными компаниями, такими, как Google, Microsoft Azure, Amazon. И для большинства из них разработаны готовые API, которые значительно облегчают процесс разработки.

Облачные сервисы дают большие преимущества при их использовании, среди которых можно выделить надежность, доступность, экономичность, масштабируемость.

Их легко подключать и тем самым они помогают быстро получить все необходимые для системы ресурсы. При этом не нужно закупать оборудование для функционирования системы, например компьютеры, жесткие диски и др., что существенно снижает расходы.

Облачные сервисы позволяют не быть привязанным к конкретному физическому устройству или аппаратной платформе и получать доступ к своим данным в любой точке мира и с любого устройства, имеющего доступ в интернет. При этом оплата осуществляется только за реально использованные ресурсы.

Таким образом, в связи с все большим развитием облачных технологий, я решила изучить один из их видов, а именно Cloud Storage, и использовать возможности данного сервиса в разрабатываемом веб-проекте.

1. Основные сведения об облачных технологиях
   1. Общая информация

Идея облачных вычислений появилась еще в 1960 году, когда Джон Маккарти высказал предположение, что когда-нибудь компьютерные вычисления будут производиться с помощью «общенародных утилит». Считается, что идеология облачных вычислений получила популярность с 2007 года благодаря быстрому развитию каналов связи и стремительно растущим потребностям пользователей. [7]

Под облачными вычислениями обычно понимается предоставление пользователю компьютерных ресурсов и мощностей в виде интернет-сервиса. Таким образом, вычислительные ресурсы предоставляются пользователю в «чистом» виде, и пользователь может не знать, какие компьютеры обрабатывают его запросы, под управлением какой операционной системы это происходит и т.д.

В настоящее время крупные вычислительные облака состоят из тысяч серверов, размещенных в центрах обработки данных (ЦОД). Они обеспечивают ресурсами десятки тысяч приложений, которые одновременно используют миллионы пользователей. [3]

Основное отличие «облачного» программного решения от обычного в том, что вся информация сохраняется не на пользовательском жестком диске, а на удаленном сервере. Аналогично с производимыми операциями: они нагружают не персональный компьютер или ноутбук, а мощности серверов компании, предоставляющей то или иное приложение. Пользователь же получает лишь результат, отправляемый на монитор через интернет.

Существует большое количество сервисов, среди которых можно выделить сервисы для создания и управления виртуальными машинами, различные хранилища, сервисы для развертывания приложения в облаке, для работы с Big Data и др.

Причины возрастающей популярности облачных технологий понятны: возможности их применения очень разнообразны и позволяют экономить как на обслуживании и персонале, так и на инфраструктуре.

* 1. Модели обслуживания

Инфраструктура как услуга (IaaS) предоставляется как возможность использования облачной инфраструктуры для самостоятельного управления ресурсами обработки, хранения, сетей и другими фундаментальными вычислительными ресурсами, например, потребитель может устанавливать и запускать произвольное программное обеспечение, которое может включать в себя операционные системы, платформенное и прикладное программное обеспечение. Потребитель может контролировать операционные системы, виртуальные системы хранения данных и установленные приложения, а также ограниченный контроль набора доступных сервисов (например, межсетевой экран, DNS). Контроль и управление основной физической и виртуальной инфраструктурой облака, в том числе сети, серверов, типов используемых операционных систем, систем хранения осуществляется облачным провайдером.[8]

Примерами услуг инфраструктуры служат IBM SmartCloud Enterprise, VMWare, Amazon EC2, Windows Azure, Google Cloud Storage, Parallels Cloud Server и многие другие.

Платформа как услуга (PaaS) — модель, когда потребителю предоставляется возможность использования облачной инфраструктуры для размещения базового программного обеспечения для последующего размещения на нём новых или существующих приложений (собственных, разработанных на заказ или приобретённых тиражируемых приложений). В состав таких платформ входят инструментальные средства создания, тестирования и выполнения прикладного программного обеспечения — системы управления базами данных, связующее программное обеспечение, среды исполнения языков программирования — предоставляемые облачным провайдером. Контроль и управление основной физической и виртуальной инфраструктурой облака, в том числе сети, серверов, операционных систем, хранения осуществляется облачным провайдером, за исключением разработанных или установленных приложений, а также, по возможности, параметров конфигурации среды (платформы).[3]

Примерами услуг платформы служат IBM SmartCloud Application Services, Amazon Web Services, Windows Azure, Boomi, Cast Iron, Google App Engine и другие.

Программное обеспечение как услуга (SaaS) — модель, в которой потребителю предоставляется возможность использования прикладного программного обеспечения провайдера, работающего в облачной инфраструктуре и доступного из различных клиентских устройств или посредством тонкого клиента, например, из браузера (например, веб-почта) или интерфейс программы. Контроль и управление основной физической и виртуальной инфраструктурой облака, в том числе сети, серверов, операционных систем, хранения, или даже индивидуальных возможностей приложения (за исключением ограниченного набора пользовательских настроек конфигурации приложения) осуществляется облачным провайдером.

|  |  |
| --- | --- |
|  |  |

Примерами SaaS являются Gmail, Google Docs, Netflix, Photoshop.com, Acrobat.com, Intuit QuickBooks Online, IBM LotusLive, Unyte, Salesforce.com, Sugar CRM и WebEx. Значительная часть растущего рынка мобильных приложений также является реализацией SaaS. [9]

* 1. Основные свойства облачных технологий

Национальный Институт стандартов и технологий NIST (National Institute of Standards and Technology, USA) в своем документе “The NIST Definition of Cloud Computing” определяет следующие характеристики облаков:

* возможность в высокой степени автоматизированного самообслуживания системы со стороны провайдера;
* наличие системы Broad Network Access;
* сосредоточенность ресурсов на отдельных площадках для их эффективного распределения;
* быстрая масштабируемость (ресурсы могут неограниченно выделяться и высвобождаться с большой скоростью в зависимости от потребностей);
* управляемый сервис (система управления облаком автоматически контролирует и оптимизирует выделение ресурсов).[1]
  1. Основные поставщики облачных услуг

На сегодняшний день существует большое множество поставщиков облачных платформ, хранилищ и ПО.

В настоящий момент основными поставщиками облачной инфраструктуры считаются Amazon, Google и Microsoft. У каждой из компаний имеется целая линейка предоставляемых услуг, среди которых для сравнения выбраны наиболее популярные.

Google App Engine – сервис хостинга сайтов и web-приложений на серверах Google. Бесплатно предоставляется до 1 Гб дискового пространства, 10 Гб входящего трафика в день, 10 Гб исходящего трафика в день, 200 миллионов гигациклов CPU в день и 2 000 операций отправления электронной почты в день. Приложения, разворачиваемые на базе App Engine, должны быть написаны на Python, Java либо Go. Предлагается набор API для сервисов хранилища datastore API (BigTable) аккаунтов Google, набор API для загрузки данных по URL, электронной почты и т.д. [9]

Платформа Google конкурирует с аналогичными сервисами от Amazon, которые предоставляют возможность размещать файлы и веб-приложения, используя свою инфраструктуру. В отличие от многих обычных размещений приложений на виртуальных машинах, таких как Amazon EC2, платформа App Engine тесно интегрирована с приложениями.

Сравнение представлено на рис.

Amazon Elastic Compute Cloud (Amazon EC2) – веб-сервис, предоставляющий вычислительные мощности в облаке. Он дает пользователям полный контроль над вычислительными ресурсами, а также доступную среду для работы. Amazon EC2 позволяет пользователям создать Amazon Machine Image (AMI), который будет содержать их приложения, библиотеки, данные и связанные с ними конфигурационные параметры, или использовать заранее настроенные шаблоны образов для работы Amazon S3.

Windows Azure Compute – компонент, реализующий вычисления на платформе Windows Azure, предоставляет среду выполнения на основе ролевой модели.[3]

Google Cloud Storage – сервис хостинга файлов, основанный на IaaS. Все файлы, которые записываются или перезаписываются на серверы, автоматически шифруются по алгоритму AES-128.

Amazon Simple Storage Service (Amazon S3) – веб-служба, предлагаемая Amazon Web Services, предоставляющая возможность для хранения и получения любого объема данных, в любое время из любой точки сети, так называемый файловый хостинг. В марте 2012 года компания Nasuni провела опыт, в течение которого поочередно передавала массивный объем данных (12 Тб) из одного облачного сервиса в другой. В эксперименте участвовали наиболее рейтинговые облака: Amazon S3, Windows Azure и Rackspace. К удивлению исследователей, скорость передачи данных сильно отличалась в зависимости от того, какое облако принимало данные. Самый лучший показатель скорости записи данных оказался у Amazon S3, передача данных из двух других сервисов занимала всего 4–5 часов, в то время как передача данных в Rackspace заняла чуть меньше недели, а в Windows Azure – 40 часов.

Windows Azure Storage – компонент хранилища, предоставляющий масштабируемое хранилище.[3]

Использование услуг показано на диаграмме (рис).

Все эти платформы (Windows Azure, Google, Amazon) имеют API, построенное на REST, HTTP и XML, что позволяет разработчикам использовать облачные сервисы с любой операционной системой, устройствами и платформами.

1. Реализация проекта
   1. Основная идея и возможности системы

Основная идея проекта - создание событий, которые уже прошли или еще будут в будущем. События могут быть совершенно разного типа, например концерты, выставки, просто встреча друзей и др.

Вы, как пользователь, добавляете основную информацию о событии: название, тип, описание, даты и теперь другие пользователи смогут его просматривать.

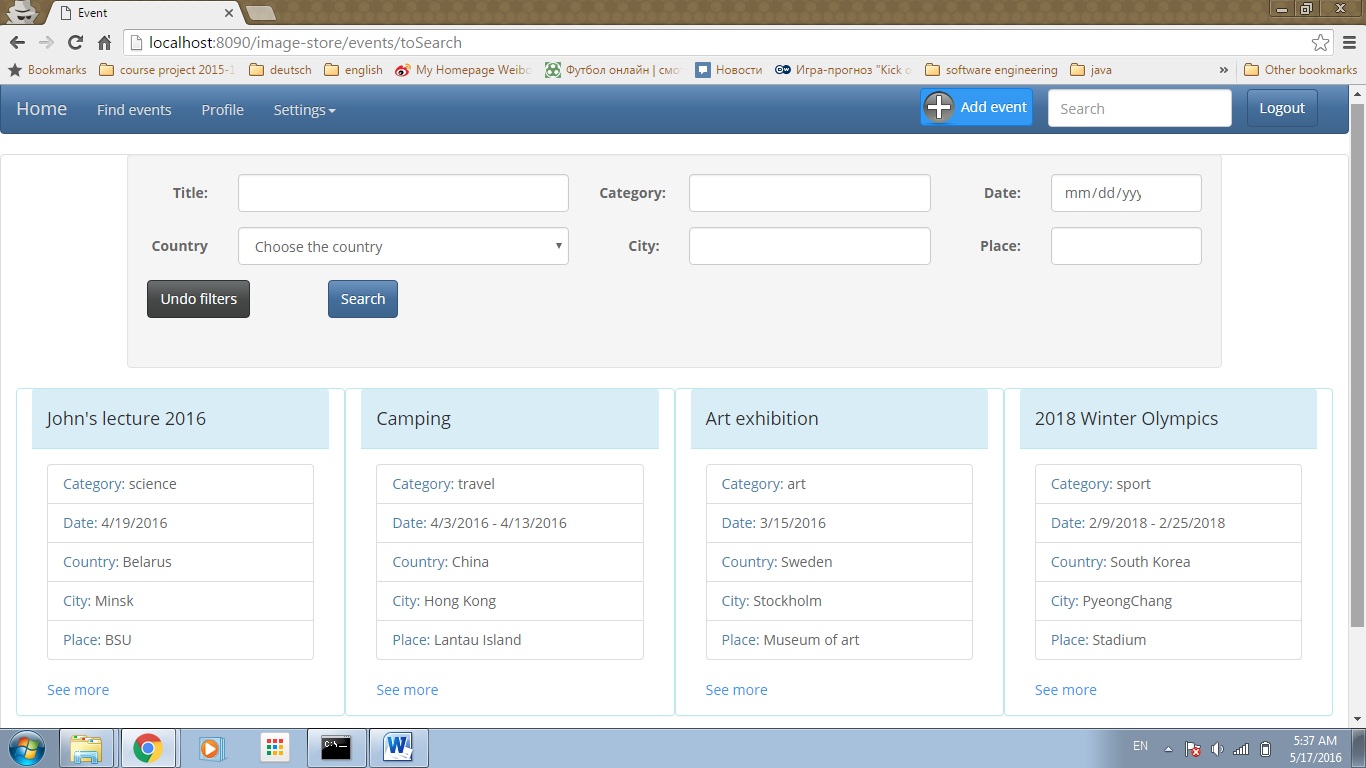
Основная цель создания события - делиться своими фотографии, которые были на нем сделаны. Пользователь, который участвовал в этом событии и у него есть фотографии, может запросто ими поделится и при этом просматривать чужие. Таким образом, данный проект позволяет объединить в одном месте большое количество фотографий события и просматривать их в любое время.

Главная цель создания проекта - изучение на его примере использования облачного хранилища.

Проект подразумевает огромное количество фотографий, при этом заранее неизвестен их объем. Поэтому использование локальных ресурсов не является целесообразным и эффективным.

Проект и база данных размещены на облачной виртуальной машине. Я использовала сервис Google Cloud Compute Engine. Использование облачной виртуальной машины позволяет подключаться к ней из любой точки мира и не быть привязанным к работе конкретного компьютера.

Пример страницы поиска изображен на Рис.

* 1.  Архитектура проекта

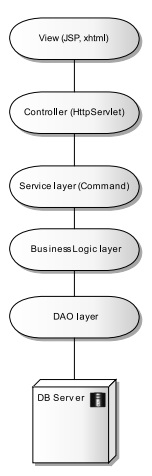
Система спроектирована на основе шаблона MVC. Model-view-controller — схема использования нескольких шаблонов проектирования, с помощью которых модель приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные.

Рисунок 1

Среди слоев модели можно выделить dao слой для работы с базой данных и сервисный слой, в котором происходит обращение к dao-классам.

Таким образом, архитектура проекта может быть представлена схемой, изображенной на рис.1

Схема базы данных представлена в приложении 1.

Серверная часть написана на языке Java с использованием таких фреймворков, как Spring Core, Spring MVC, Spring Security, Hibernate. Данные фреймворки в значительной степени облегчают процесс создания веб-приложения и общения с базой данных.

Для генерации страниц из частей использовался Apache tiles, для оформления - CSS, разметки - HTML, для динамики веб-страниц и отправки запросов - JS, JQuery, Ajax.

Реализована интеграция с Google Cloud Storage для хранения фотографий пользователей.

* 1. Основные используемые фреймворки
     1. Spring

Spring Framework — универсальный фреймворк с открытым исходным кодом для Java-платформы.

Spring Framework обеспечивает решения многих задач, с которыми сталкиваются Java-разработчики. [5]

Spring Framework может быть рассмотрен как коллекция меньших фреймворков. Большинство этих фреймворков может работать независимо друг от друга, однако они обеспечивают большую функциональность при совместном их использовании. Эти фреймворки делятся на структурные элементы типовых комплексных приложений: Inversion of Control-контейнер, фреймворк аспектно-ориентированного программирования, фреймворк доступа к данным, фреймворк управления транзакциями, фреймворк MVC, фреймворк аутентификации и авторизации (Spring Security), фреймворк работы с сообщениями и др.[5]

Центральной частью Spring Framework является контейнер Inversion of Control, который предоставляет средства конфигурирования и управления объектами Java с помощью отражения. Контейнер отвечает за управление жизненным циклом объекта: создание объектов, вызов методов инициализации и конфигурирование объектов путем связывания их между собой.

Объекты, создаваемые контейнером, также называются управляемыми объектами (beans). Обычно конфигурирование контейнера осуществляется путем загрузки XML-файлов, содержащих определение bean’ов и предоставляющих информацию, необходимую для создания bean’ов. [1]

* + 1. Hibernate

Hibernate является ORM-решением для языка Java (Рисунок 2). Данная технология не только заботится о связи Java классов с таблицами базы данных (и типов данных Java в типы данных SQL), но также предоставляет средства для автоматического построения запросов и извлечения данных и может значительно уменьшить время разработки, которое обычно тратится на ручное написание SQL и JDBC кода. Hibernate генерирует SQL вызовы и освобождает разработчика от ручной обработки результирующего набора данных и конвертации объектов, сохраняя приложение портируемым во все SQL базы данных.[4]

На мой взгляд, самое главное преимущество Hibernate в том, что он делает код независимым от типа сервера баз данных. При этом он позволяет программисту не думать об устройстве и особенностях использования какого-то конкретного типа базы данных. Можно использовать один и тот же код, как для Oracle, так и я для MySql.



Рисунок 2

* + 1. Apache Tiles

Apache Tiles - это фреймворк для построения и использования шаблонов страниц. Первоначально разрабатывался, чтобы упростить разработку интерфейсов пользователей в веб-приложениях, но его использование не ограничено средой JavaEE.

Tiles позволяет разработчикам определять фрагменты страниц, которые могут быть собраны в отдельную страницу во время исполнения. Эти фрагменты могут использоваться как включения в основную страницу, чтобы сократить количество повторов общих элементов страниц, или встраивания с другими элементами, чтобы разработать серию шаблонов, которые можно многократно использовать. Данные шаблоны можно использовать повсеместно в приложении.[6]

Конфигурация шаблонов и страниц осуществляется в xml файлах.

* 1. Перспективы развития

В будущем планируется дальнейшее развитие проекта, а именно:

* Error tracking. Интеграция с одним из инструментов для логирования и сбора информации об ошибках в системе. Представителями таких инструментов являются: Raygun, Takipi, Airbrake, StackHunter, Sentry и др.
* Cloud Database. Экспорт существующей базы данных в облако и хранение там данных. Это обеспечит защиту данных и улучшит их доступность.
* Publish on cloud. Опубликование проекта в облаке и предоставление его конечным пользователям для использования.
* Добавление нового функционала. Планируется добавление подтверждения электронной почты, возможности оценивать фотографии, приглашать присоединиться к событию, создание закрытых событий и др.

Планируемое увеличение пользователей за 5 лет показано на графике (рис)

Заключение

В ходе работы над курсовым проектом мной были изучены Java-фреймворки для работы с веб-проектом и базами данных такие, как Spring, Hibernate, Apache Tiles. Эти фреймворки использовались для реализации веб-проекта, основной функциональностью которого является хранение и предоставление пользователям фотографий с различных событий.

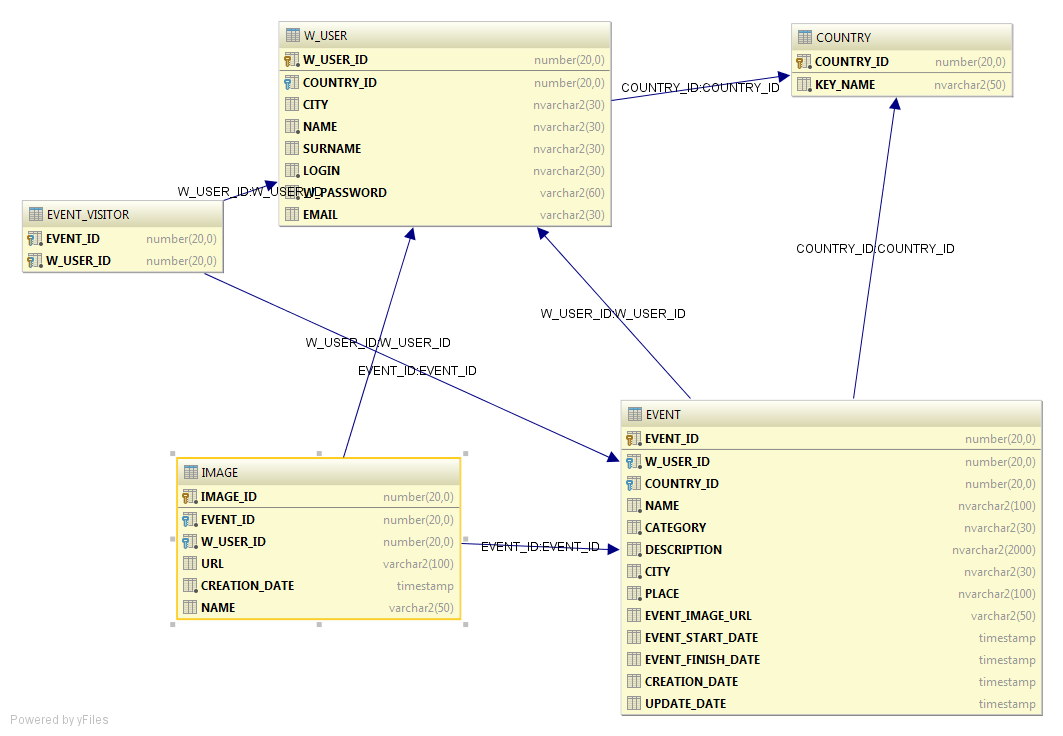
Для хранения фотографий использовались облачные технологии, а именно Google Cloud Storage. Использование облачного хранилища позволяет значительно уменьшить объем данных, хранящихся на локальном сервере. Также это обеспечивает надежность и доступность данных, упрощает расширение пространства при необходимости.

Мною была изучена общая информация об облачных технологиях, а также преимущества и недостатки их использования. Среди основных недостатков можно выделить следующие: зависимость от поставщика, отсутствие полного контроля над ресурсами. Среди преимуществ - надежность, доступность, экономичность, масштабируемость, простота подключения и использования, мобильность, арендность и др.

В настоящее время облачные технологии постоянно развиваются. Качество и количество предоставляемых поставщиками услуг постоянно растет. Количество встраиваемых компьютеров увеличивается благодаря снижению цен на процессоры и повсеместному распространению Интернета. Растут также и объемы передаваемых данных с последующей их обработкой (часто в режиме реального времени).

Поэтому можно с уверенностью говорить, что в ближайшие годы роль облачных вычислений будет еще больше увеличиваться.

Список использованных источников

1. *Блинов,* И.Н. Методы программирования: учеб.-метод. Пособие /И.Н. Блинов, В.С. Романчик - Минск: Издательство Четыре четверти, 2013, - 370-372 с.
2. **Google Cloud Storage Documentation**. [Электронный ресурс]. - Режим доступа: *https://cloud.google.com/storage/docs/overview.*- Дата доступа: 22.03.2016
3. **Бизнес в облаках. Чем полезны облачные технологии для предпринимателя.** [Электронный ресурс]. - Режим доступа: *https://kontur.ru/articles/225. -* Дата доступа: 13.04.2016
4. **Основы Hibernate.** [Электронный ресурс]. - Режим доступа: *https://habrahabr.ru/post/29691. -* Дата доступа: 13.04.2016
5. **Spring Framework.** [Электронный ресурс]. - Режим доступа: *https://ru.wikipedia.org/wiki/Spring\_Framework.* -Дата доступа: 21.05.2016
6. **Apache Tiles.** [Электронный ресурс]. - Режим доступа: *https://tiles.apache.org/framework/index.html. -* Дата доступа: 03.04.2016
7. *Martin, Richard J.* Guide to cloud computing./Martin, Richard J. and Hoover, Nicholas - Минск: Издательство Четыре четверти, 2013, - 370-372 с.
8. *Крупин,* И.И. Cloud Computing: высокая облачность./И.И. Крупин, Nicholas – М.: Издательство Мир, 2014, - 205с.
9. Облачные технологии. / К.П. Воленюк [и др.] - 2-е изд. - Минск: ТетраСистемс, 2004. - 495 с.
   * + 1.  Схема базы данных