адсутнічае пагаршэнне прадукцыйнасці.Яго нястачай з'яўляецца праблема апраксімацыі палігонаў і неабходнасць стварэння індэкса для ўсёй Зямлі.

Для параўнання або пабудовы індэксаў для радковых і двайковых дадзеных можна выкарыстоўваць хэш-індэксы. Яны прадугледжваюць захоўванне не саміх значэнняў, а іх хэшаў, дзякуючы чаму змяншаецца памер (а, адпаведна, і павялічваецца хуткасць апрацоўкі) індэксаў з вялікіх палёў. Такі тып індэксаў карысны пры аперацыях сартавання і параўнання дадзеных, аднак, выкарыстоўваючы яго, нельга вызначыць, колькі прыкладна радкоў ёсць паміж двума значэннямі, і для яго выкарыстання патрабуюцца толькі цэлыя значэнні ключоў.

Там, дзе ёсцьвялікія мноства з нізкай магутнасцю і добрай кластарызацыяй па іх значэннях, выкарыстоўваюцца бітавыя індэксы (bitmap). Бітывыя індэксы выкарыстоўваюцць бітавыя масівы івынікі выконвання пабітавых лагічных аперацый на гэтых масівах. Гэты індэкс эфектыўны пры вялікіх табліцах і калі слупкі часта ўдзельнічаюць у аперацыях ўстаўкі / абнаўлення / выдалення. Аднак, пры аднаўленні мноства бітаў спатрэбіцца час, каб выканаць DML-аперацыю і аднавіць індэксы.

Для манатонна нарастальных значэнняў (напрыклад аўтаінкраментны ідэнтыфікатар) выкарыстоўваюць індэкс зваротнага ключа. Індэкс зваротнага ключа, па параўнанні са стандартным індэксам, звяртае кожны байт слупка, захоўваючы пры гэтым парадак слупкоў. Калі слупок індэксуецца ў зваротным рэжыме, значэнні слупка будуць захоўвацца ў індэксе ў розных блоках па меры таго, як зыходнае значэнне адрозніваецца. Такая кампаноўка можа дапамагчы пазбегнуць пагаршэння прадукцыйнасці ў індэксах, дзе змены індэкса сканцэнтраваны на невялікім наборы блокаў. Індэксы зваротнага ключа змяншаюць «гарачыя пункты» ў індэксах, асабліва індэксы першасных ключоў, шляхам змены байтаў блокаў ліста. І, такім чынам, знішчаюць канкурэнцыю за ліставыя блокі па ўсіх асобніках.

Пры выбары тыпу індэкса трэба ўлічваць структуру дадзеных, якія індэксуюцца, а таксама характар працы з імі. Кожны тып па-свойму унікальны, моцны і карысны. І няма ўніверсальнага алгарытму і структуры дадзеных, якія былі б карысны ўсім. Важна памятаць, што выбар тыпу індэксавання можа як дапамагчы вырашыць праблему, гэтак і нашкодзіць, дадаўшы дадатковы расход памяці і замаруджванне часу апрацоўкі запытаў.

Спіс выкарыстаных крыніц:
1. Рассел, Д. Индекс (базы данных) / Д. Рассел, Р. Кон. – М. : Оникс, 2013. – 520 с.
2. Цэнтр ведаў IBM [Электронны рэсурс]. – 2018. – Рэжым доступу: https://www.ibm.com/support/knowledgecenter.

# СИЛЬНЫЕ СТОРОНЫ ФРЕЙМВОРКА ANGULAR

*Белорусский государственный университет информатики и радиоэлектроники*
*г. Минск, Республика Беларусь*

*Харитонов Н.В., Дроздов А.С.*

*Стержанов М.В. – к.т.н., доцент*

В современном мире, где разработка программного обеспечения развивается быстрыми темпами, наибольшую популярность приобретают веб-приложения. Основными плюсами использования веб-приложений являются простота, удобство, доступность и скорость. Для реализации таких приложений существует множество различных фреймворков разработки, среди которых одним из наиболее популярных в последние три года является Angular.

Angular – Javascript-фреймворк, созданный на основе TypeScript, который позволяет разрабатывать сложные SPA-приложения (Single Page Application). Написание приложения состоит из создания шаблонов с помощью языка разметки HTML, написания классов-компонентов для работы с данными шаблонами, а также выделения общей логики приложения в сервисы. Все вместе это можно объединять в отдельные модули. Angular 2 написан в соответствии с шаблоном проектирования «модуль», что позволяет изолировать части логики приложения от глобального контекста, и содержит собственную систему модулей (NgModules) с тем, чтобы приложение имело понятную структуру и позволяло переиспользовать необходимые части без циклических зависимостей. Разработанный и поддерживаемый компанией Google, он описывается как JavaScript MVW-фреймворк. Angular (он же – Angular 2+", он же – Angular 2 или ng2) является переписанным преемником AngularJS (он же – Angular.js или AngularJS 1.x). Несмотря на то, что AngularJS (ранняя версия) был выпущен в октябре 2010 года, его создатели до сих пор устраняют недоработки данного фреймворка. Новый же Angular (без окончания «JS») был выпущен в свет в сентябре 2016 года в качестве версии №2 своего предка. На данный момент самой последней версией является 5-я. Фреймворк Angular используется такими компаниями, как Google, Wix, weather.com, healthcare.gov и Forbes.

Традиционно, веб-приложение на Angular 2 состоит из набора компонентов (виджетов), которые образуют древовидную структуру. В ней, в свою очередь, родительский компонент имеет ссылку на все вложенные в него дочерние. Обмен данными реализован следующим способом. При получении данных родитель отправляет их часть дочерним компонентам. Они, в свою очередь, либо передают их часть дальше по дереву компонентов, либо отображают эти данные с использованием различных элементов интерфейса. Также, в Angular 2 дочерние компоненты имеют возможность оповестить родительский о различных пользовательских событиях: клик мыши или нажатие клавиши. Также возможно внедрение так называемых сервисов (services) - специальных объектов, предоставляющих компоненту возможность получить данные в

любой момент, вне зависимости от его расположения в дереве компонентов. Стоит упомянуть, что в Angular 2 реализован прием, позволяющий любому объекту веб-приложения получить доступ к другому, предварительно зарегистрированному объекту (provider) при необходимости.

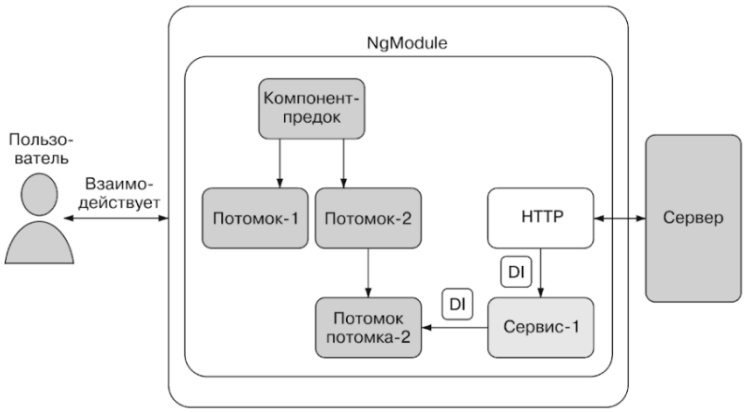


Рис. 1. Пример архитектуры Angular-приложений

Так же хотелось ты отметить, что Angular 2 разработан с нуля для мобильных приложений и оптимизирован с точки зрения эффективности памяти и меньшего количества циклов центрального процессора. Здесь объектами первого класса являются touch-события и жесты, работающие на всех устройствах. Все тесты производительности, проводимые командой Angular, находятся в открытом доступе вGithub. Так что абсолютно любой может на них взглянуть и даже провести их.

Таким образом, фреймворк Angular.js является полноценным инструментом, полноценно реализующим компонентный подход, позволяя разработчикам создавать отдельные компоненты, а из компонентов создавать полноценные приложения. Созданные компоненты можно использоваться в других приложениях, так как они независимы.

Список использованных источников:

1. Angular 2 [Электронный ресурс]. – Режим доступа: https://angular.io/.
2. Fain Y., Moiseev A. Angular 2 Development with TypeScript. – Manning Publications, 2016

# ПРИМЕНЕНИЕ СКРЫТЫХ МАРКОВСКИХ МОДЕЛЕЙ ДЛЯ СОПОСТАВЛЕНИЯ КАРТ

*Белорусский государственный университет информатики и радиоэлектроники*
*г. Минск, Республика Беларусь*

*Иванин Н.С.*

*Волорова Н.А. – к.т.н., доцент*

В последнее время было разработано множество подходов, позволяющих улучшить навигацию и управление транспортным средством на основе данных, собираемых его различными датчиками. Одним из таких датчиков является датчик Глобальной Системы Позиционирования(GPS). Зачастую требуется определить по какой дороге движется транспортное средство на основе данных датчика GPS – эта проблема известна как проблема сопоставления карт. В настоящей работе предлагается подход на основе использования Скрытых Марковских Моделей(СММ) для решения задачи сопоставления карт. Этот подход позволяет бороться с такими проблемами, как шумы в GPS данных и разреженность данных.

Сопоставление карт – это процесс определения дороги, по которой совершало движение транспортное средство на основе данных, собранных специальными датчиками. Как правило, датчики собирают данные GPS, поскольку система GPS доступна повсеместно. Например, сопоставление карт используется навигационными системами. В последние время сопоставление карт также используется для измерения скорости движения по дороге, а также построения статистических моделей пробок. В дальнейшем такие модели могут быть использованы для нахождения оптимального пути движения транспортного средства в обход пробок [1].

В работе [2] приводятся основные элементы структуры дорожной сети:

Узел. Дорога, являющаяся линейным объектом, при цифровом представлении хранится как последовательность точек(узлов). Узлами являются также точки пересечения дорог, начала и концы дорог, по которым транспортные средства могут двигаться.

Сегменты дорог. При существовании прямого пути между двумя соседними узлами такой путь будет называться сегментом дороги. Начальный и конечный узел такого пути будут называться начальными и конечными узлами сегмента.