

Делаем свою Google Maps

Google Maps знают все, пользуются все и любят все.

Но. :) По всем пунктам есть определённые возражения.

Google Maps знают все: знают *все* но не *всё*. Система постоянно развивается, появляются новые возможности. Так же меняются правила пользования сервисами Google Maps, которые знают уже далеко не все.

Google Maps пользуются все. Все, при соблюдении [условий использования](#), которые вкратце гласят: использовать бесплатно можно только в публично доступных сервисах. То есть использование под паролем - пиратство.

Любят все. Не все. Сервисы Google настолько хороши, что захватывают всё большую часть рынка, и не все последствия этого очевидны.

Об этом можно долго рассуждать, но сейчас нас волнует конкретная задача: создать сервис, аналогичный (условно) Google Maps. Мотивация решения не использовать Google Maps:

- система не должна быть под контролем сторонней организации, которая может в любой момент прикрыть сервис совсем или положить свои рекламные баннеры на нашу карту;
- система должна использовать наши карты;
- система должна работать в областях, не покрываемых Terms of Service - в управлении или мониторинге транспорта, в закрытых паролем системах, и так далее.

Что такое вообще Google Maps: это три основные подсистемы:

1. карты как картинки - у нас таких нет и никогда не будет;
2. доставка карт по сети - это огромная CDN, обеспечивающая скорость до которой нам не подняться;
3. Google Maps API - мегамонстр, повторить и отладить который нереально.

Приходится признать, что воспроизвести полностью сервис невозможно. Но так как задача поставлена, посмотрим что можно сделать и в каких объёмах.

1.Карты.

В современной GIS системе карта строится из маленьких кусочков, обычно 256x256 или 512x512 пикселей, с фиксированной шкалой масштабов. Наша цель - получить для любой области покрытие этими кусочками карты.

В типичном GIS приложении карта состоит из двух принципиально разных частей: статической редко меняющейся части - растровой подложки (например карты местности), и векторных слоёв - данных, которые наносятся на карту как результат работы какого-то приложения, например маршруты, транспортные средства, точечные объекты (магазины, заправки и т.д.).

Проблемы, с которыми мы сталкиваемся при создании растровой подложки - это отсутствие обработанных спутниковых снимков. Максимум на что можно рассчитывать - это свободно распространяемые фрагменты местности в необработанном виде, без наложенных названий городов, в невысоком разрешении, но это хоть что-то и на практике использовать хоть и трудно, но можно. Есть и другой путь получить подложку - и в некоторых случаях не такой и плохой.

Существует класс программ-генераторов карт. Функция такой программы - получить координаты области и построить для неё карту. Источником данных для них могут выступать векторные карты - это ESRI DCW, [VMAP Level0](#) и Level1, сеточные данные глубин и высот, базы данных населённых пунктов, растровые спутниковые и другие карты - как проприетарные, так и свободнодоступные. Как правило, конечным продуктом является файл в формате GIF, PNG, JPG. Конкретные форматы определяется возможностями браузеров их отображать - например, GIF и PNG имеют прозрачные или полупрозрачные участки, или размером файлов - JPG даёт самые маленькие размеры для неконтрастных изображений.

У всех программ узкое место - проектирование карт, но проявляется это по разному.

Известные мне построители карт (только более или менее бесплатные), все как-то поддерживают форматы OpenGis:

[UMN Mapserver](#) - самая известная и, наверно, самая древняя. Бесплатно. Написан на C (или C++ - не помню) и работает как CGI или FastCGI скрипт, то есть достаточно быстро. Древняя, корявая, но в некоторые отношения всё ещё вне конкуренции. Для генерации шаблона карт есть язык MapScript (интерфейс к MapServer), довольно понятный и имеющий реализации для нескольких языков - например perl, php, что позволяет делать карты на лету. Вроде есть какие-то построители карт - проекту-то уже больше лет чем интернету, но хороших я не встречал. Отлаживать карты трудно - часты падения сервера без сообщений. Зато если работает - то работает.

[GeoServer](#) - написан на Java. Бесплатно. Сильно полагается на XML при конфигурировании карт, что с одной стороны неплохо - перспективно, а с другой стороны приходится этот XML делать вручную, что меня не порадовало вообще. Теоретически описания карт можно делать из любой программы, но как - я не представляю. Официальный способ (пока) - это работа через web интерфейс. Если надо просто вывести данные из базы, или из .SHP файлов - получается очень быстро и просто, подключение к PostGIS делается из меню, есть источники данных и так далее - всё по современному. Портят картину две неприятности: карту с сотней слоёв в разных масштабах вручную сделать практически невозможно. Какой-то студент работает над визуальным редактором карт - вот-вот будет, некоторые используют uDev - но это не решение. В общем, пока всё глухо. Отлаживать карты трудно - часто нажимаем кнопку и ничего не происходит и не сообщений ни ругани ни изменений. Иногда по неизвестной причине начитает грузить процессор, и хотя из-за многопоточности как-то всё равно работает, но именно как-то.

[Demis Web Map Server](#) - самая неизвестная, проприетарная, под Windows, в общем полная кака, если бы не его достоинства, а именно он единственный способен произвести красивую карту - так как имеет вменяемый визуальный редактор. Падений заметных пока не замечал. Дополнительно к визуальному редактору, который с серверами баз данных общаться не умеет, есть возможность добавлять данные скриптами. Это я знаю только теоретически.

[MapGuide Open Source](#) - самое странное создание. Бесплатный (но в варианте [MapGuide Enterprise](#) - платный). Теоретически должен быть наследником MapGuide той же фирмы -

Autodesk, но никакого отношения к MapGuide 6 не имеет. Монстр во всех смыслах - есть и редактор карт, и компонент для отображения карт, и сервер - но всё жутко тяжёлое и падающее. Теоретически он должен был бы быть идеальным сервером, но из-за кривизны и недоброй памяти, оставленной его тёзкой, доверия к нему нет. Хотя то что он поддерживается Autodesk, должно прибавить ему привлекательности.

[TileCache](#) - не совсем картографический сервер. Бесплатный. В принципе, любая из вышеперечисленных программ способна сделать подложку, то, что мы видим на карте Google при первом заходе на [maps.google.com](#). Есть несколько незаметных проблем - генерация карты происходит долго (на кусок 256x256 может уходить несколько секунд для навороченных карт), карты сами весят много, исходники карт тоже не маленькие (гигабайты) и не всегда их можно просто так распространять. Хороший выход - использование кэширующих серверов, которые один раз получают картинки с картами, сохраняют их на диске, и отдают потом по запросу. Теоретически можно любой squid приспособить, но практически имеет смысл использовать TileCache - написанный на Питоне сервер, ориентированный на кэширование маленьких кусочков карт - из которых, как правило, на клиентской стороне и собирается карта. При работе с Google Maps видно, что карта состоит из квадратиков, вот именно эти квадратики мы должны сгенерировать мапсервером, отдать их TileCache серверу, а уж он отдаст их клиентскому браузеру. Кстати, так как обычно используется стандартный протокол WMS - протокол общения клиента и картогенератора, то между клиентом и сервером легко вставить посредника, клиент и не замечает, что карту получает от прокси. По стандарту WMS надо указать список слоёв, средствами TileCache этот список можно указать, а приложению позволить использовать простое символическое имя.

Важная часть кэширования - использование фиксированных стандартных масштабов. Это позволяет хранить конечный набор фрагментов.

2. Доставка карт по сети

Предположим, мы научились генерировать картинки, поставили TileCache сервер - он отдаёт картинки достаточно быстро. Но один сдвиг карты или изменение масштаба приводит к перекачке сотни (или сотен) килобайт - трафик от сервера получается очень большой, и начинает играть роль ширина серверного канала. С другой стороны, канал от клиента до конкретного сервера может быть узкий даже при бесконечно широком серверном канале. Google решает проблему установкой кучи серверов по всему миру, на которые копируются карты и до которых имеются быстрые каналы из почти любого места. Мы должны поступить так же. Поможет в этом тот же TileCache - теоретически, можно нагенерировать карты, покрывающие весь земной шар, и эти карты скопировать на вторичные сервера - сделав свою маленькую сеть доставки контента, как у гугля. В принципе, такой прокси вместе с кэшем карт может быть записан на DVD и запускаться на машине без винчестера - надёжность высочайшая, и даже может запускаться на пользовательском компьютере, без серьёзного переписывания софта. Вариантов масса.