Олег Бартунов

КЛЮЧЕВОЙ КОНТРИБЬЮТОР В POSTGRESQL

ПРОКАЧАТЬ ПАМЯТЬ СЛОНУ

Имя нашего собеседника хорошо известно любому, кто серьезно занимается базами данных. Именно благодаря Бартунову в PostgreSQL когда-то появилась поддержка локализации, и проект начал завоевывать популярность в России и других неанглоязычных странах. С тех пор Олег вот уже почти 20 лет принимает активное участие в разработке проекта и в развитии отрасли обработки и хранения данных в рунете.

БЕСЕДОВАЛ СТЕПАН ИЛЬИН

Наука и PostgreSQL

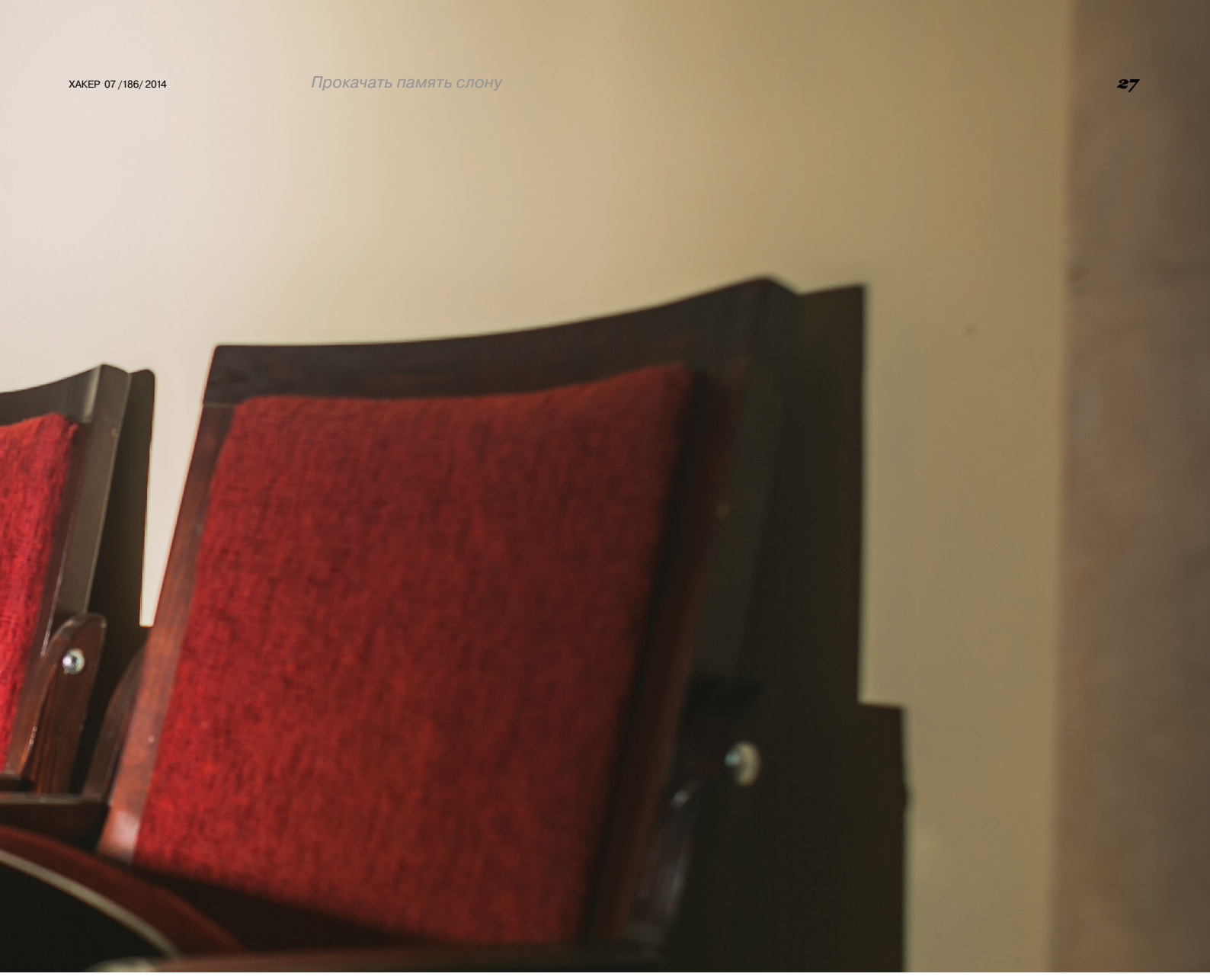
Как вышло, что вы совмещаете работу в институте, астрономию и PostgreSQL? Что основное?
Всю жизнь я хотел быть астрономом. Став им, был рад до смерти. Но мне все время приходилось работать с компьютерами, заниматься расчетами: мы считали взрывы сверхновых звезд. Работать начинали на «Мир-2», БЭСМ-4, БЭСМ-6, ЕС — в общем, застали все главные машины того времени.

Астрономия — это наука о данных. Мы каталогизируем факты. Все объекты наших наблюдений имеют координаты, названия и другие параметры, и этих объектов миллиарды. Какое-то время мы работали с наблюдениями по-старому, на магнитных лентах, — но в какой-то момент я понял, что это ужасно и нужно использовать базы данных. В то время БД только-только получали развитие, но на Западе они уже появились. Это был 1993 год. Я полез на FTP, скачал Ingres и стал с ним играться. Первые мои базы были на нем.

Тогда же я начал читать статьи, в том числе Стоунбрейкера (один из ключевых специалистов в мире БД, создатель Ingres и Postgres. — Прим. ред.), и вдохновился его идеями. В 1995 году студенты Стоунбрейкера написали в комьюнити письмо, мол, мы закончили работу и хотим выдать Postgres наружу, и я, естественно, присоединился к движению. С той поры нас осталось всего несколько человек. По-моему, только я и Брюс Момжан. То есть из самых первых, кто некогда все это начинал.

С чего вы начинали с PostgreSQL?
В 1995–1996 годы меня пригласили сделать архив одной газеты.Так как из БД я знал PostgreSQL, я поставил туда именно ее. И обнаружилось, что PostgreSQL не понимает кириллицу. Изначально она была 7-битная, 8-битного текста не понимала. Пришлось две-три недели во всем разбираться, как это обычно и делается в open source. Помогало то, что у меня уже был опыт локализации Perl — я серьезно разбирался с ним, хорошо знал Ларри Уолла еще в Америке. И мы занимались его локализацией, чтобы тот понимал locale. Так что у меня уже было представление, что нужно делать, чтобы организовать поддержку и в PostgreSQL.

А также было понимание того, как работает PostgreSQL.
На самом деле нет. Тогда особенного понимания еще



не было. Его до сих пор нет. Потому что PostgreSQL неисчерпаем. Вряд ли найдется человек, который полностью знает все его части.

Словом, тогда, работая над архивом, я сделал патч. В те времена все было очень демократично: послал патч, его сразу закоммитили. Так PostgreSQL получил локализацию и стал широко использоваться в Европе. До этого ее знали только Канада и Штаты.

Но тогда PostgreSQL все еще был для вас хобби?
Да, до какой-то поры Postgres был хобби. Потом наступил период, когда мы занялись Rambler. Это уже отдельная история: Rambler тоже стоит на Postgres из-за нас, в частности из-за меня.

В то же время мы сделали одну из первых CMS — систему под названием Discovery. Нам нужен был своеобразный конструктор, удобная площадка, которая позволила бы нам заниматься популяризацией науки и быстро создавать сайты для любых проектов. Это сейчас подобное реализуется без проблем: качаешь любую CMS и ставишь. Тогда этого не было, приходилось все делать самим.

Здесь, в ГАИШ, она работает до сих пор. Сайт astronet.ru — самый крутой сайт об астрономии в России — до сих пор стоит на том старом Rambler, на том старом движке. То есть мы сделали движок для astronet.ru, а потом в Rambler вставляли его в продакшен.

Rambler здорово помог нам с PostgreSQL. У меня появился напарник — Федя Сигаев, который работает и здесь, и в Mail.Ru. Вместе с ним мы стали заниматься PostgreSQL уже серьезно.

Как сейчас распределяете время между работой в институте и PostgreSQL?
Да никак не распределяю. За прошедшие годы институт понял, что PostgreSQL нужна астрономии. Все наши ресурсы и сервисы, все БД завязаны на PostgreSQL. У нас есть отдельная, приоритетная тема (всего таких тем в институте тринадцать), она называется «информационные проблемы астрономии». Я ей руковожу. Наши пространственные индексы используются на многих обсерваториях мира. МГУ поддерживает PostgreSQL. Наука без IT сейчас невозможна.

Сам ГАИШ уникален еще и тем, что в свое время здесь собирались многие большие деятели интернета.

ЗАНИМАЕТСЯ POSTGRESQL С 1996 ГОДА

Начало работы с PostrgeSQL

Какова сейчас ваша роль в PostgreSQL?
После Rambler мы с Фёдором Сигаевым продолжили работать над кодом. Было очень интересно, на каждом шагу возникали челленджи, нужно было разбираться, что-то придумывать.

PostgreSQL очень хорошо спроектировали изначально. В эту базу заложены идеи, которые нам очень нравились. Скажем, идея расширяемости. Стоунбрейкер как-то сказал, что расширяемость баз данных — это необходимая вещь. Ситуации в жизни бывают разные, и, поставив однажды какую-то БД, потом, как правило, трудно с нее слезть. То есть ты зависишь от производителя. PostgreSQL — такая база… если тебе чего-то в ней не хватает, можно просто дописать это самостоятельно. Меня это очень вдохновило, и именно этим я занялся. С тех пор мы с Федей отвечаем за расширяемость PostgreSQL.

За счет чего реализована эта расширяемость?
В PostgreSQL существуют интерфейсы. Они позволяют стороннему человеку (специалисту в области данных, не разработчику) описать свои данные, предложить какие-то функции. В ответ вы автоматически получаете различные плюшки от PostgreSQL, например, индексы. Это не просто обычный тип данных. Этот тип данных будет искаться с помощью индексов. Вы получаете конкурентный доступ к данным, восстановление после сбоев.

PostgreSQL позволяет разработчику заниматься только спецификой своих данных. А с тем, как сделать навигацию по дереву, как его построить, БД разберется сама. В PostgreSQL все это было заложено, но таким образом, что изначально оно не работало. Была директория, но никто ей не пользовался. Мы раскопали директорию под названием gist и сделали так, что из академической разработки GiST (Generalized Search Tree, обобщенное поисковое дерево) стал рабочим.

Чем вы занимались потом?
GiST нам нужен был для эффективной работы с массивами. До нас массивами в постгресе особенно не пользовались — это было совсем нереляционно, да и особой поддержки не было, но мы понимали, что времена меняются и люди нуждаются в массивах. Чтобы нормально использовать массивы, нужна индексная поддержка. Эту поддержку мы сделали с помощью инфраструктуры расширяемости GiST. Тогда мы во всем разобрались и сделали.

Первым нашим модулем стал intarray, он до сих пор очень популярен. Use case простой: у вас есть категории, товар из нескольких заданных категорий. Простая, тривиальная задача, но при реляционном подходе она решается медленно. Соответственно, мы сделали работу с массивами в PostgreSQL обычной задачей. Сейчас люди работают и не задумываются о том, что раньше это было невозможно.

Потом мы сделали первый вариант полнотекстового поиска. Потом сделали поиск с ошибками. Сейчас мы имеем уже три инфраструктуры расширяемости — GiST, GIN и SP-GiST, с помощью которых можно разрабатывать разнообразные индексы.

Как PostgreSQL развивается сейчас?
Очень активно. К примеру, сейчас мы сделали поддержку JSON, этого нет ни в MySLQ, ни в Oracle, ни в MS SQL. Осенью выйдет релиз 9.4 с нормальной поддержкой JSONB.

Можете рассказать об этом проекте подробнее?
Проект родился еще в 2003 году. Прошло уже больше десяти лет. Тогда Министерство образования заказало нам создать каталог субъектов министерства. Нам дали схемы таблиц — школы, университеты, колледжи и так далее. Их набралось штук пятьдесят. Каждый из них сам по себе вполне реляционный, но, если тебе хочется поискать в общем, придется делать пятьдесят запросов, что очень неудобно. С другой стороны, если объединить все в одну таблицу, получается таблица размером, кажется, 500 колонок. Потому что там очень много индивидуальных, специфических полей. Есть, конечно, общие поля, но очень много и индивидуальных. Представляете, сколько это? Грубо говоря, на экране не помещалось. Вот мы с Федей Сигаевым сидели и думали, что с этим делать.

Нам подумалось, что в Perl есть такая штука, как hash. То есть набор пар ключ — значение. Мы решили выделить общие поля как отдельные таблицы, а все остальные 500 полей поместить в отдельную строку, в виде ключ — значение. Пусть они там живут, на них никто не смотрит. Так как мы занимаемся расширяемостью, мы создали новый тип данных, называется hstore. Этот тип данных мы положили не в виде строки, а в виде бинарного хранения. В базе он хранился бинарно. Сделали индекс, все, как положено. И стали пользоваться. Тогда мы не знали, что это key-value БД. Более того, в те годы не было JSON, он появился только через несколько лет.

Hstore сейчас широко используется?
Сейчас hstore является самым часто используемым модулем. Люди бросились его использовать для всяких админок, добавлять новые поля. Традиционные базы данных страдают тем, что изменить их схему очень тяжело, а здесь просто добавляется новый ключ. Клепать какие-нибудь интерфейсы — милое дело.

Несколько лет назад, когда уже появился JSON и MongoDB, мы подумали, что нужно сделать нашему hstore (который просто ключ — значение) поддержку вложенности. Чтобы внутри hstore могли быть еще hstore, массивы и так далее. К этому времени hstore уже был очень популярен и раскручен, мы получили поддержку на эту работу от компании Engine Yard. Мы взялись делать, сделали, в Канаде показали, как это работает. А потом понемногу склонились к идее, что нам незачем делать новый hstore, лучше сделать JSON. Фактически там нет никакой разницы. Бинарное хранение было одинаковым и там и там, разнились только внешние интерфейсы. Мы их переписали и сделали из hstore JSONB (пришлось ввести новый тип данных, чтобы не было проблем с совместимостью с существующим типом данных json, который есть просто текстовый тип данных и появился еще в версии 9.2).

Уже в ходе работы я сравнил нас с MongoDB и вдруг увидел, что мы даже быстрее их. Проект стал стремительно развиваться, нам снова дали поддержку, но мы пошли даже дальше. То, что я описал, закоммитили в 9.4, то есть осенью уже будет JSONB с индексами. Но мы также сделали язык запросов (jsonb query language) и назвали его JSquery.

Заодно мы обнаружили кучу проблем. Как всегда и бывает, когда начинаешь чем-то заниматься, — возникают челленджи. Мы решили с ними побороться. Я подумал, что нужно сделать новый метод доступа. Придумали название VODKA. Потому что один метод доступа мы уже назвали GIN — обобщенный обратный индекс. А тут я сказал: хочу, чтобы теперь была «водка». Чтобы от России в PostgreSQL было такое название. Как расшифровывать, мы потом поймем :). Всем понравилось. Мы начали делать новый метод доступа, который решал бы те проблемы, что мы обнаружили. За прошедшее время мы плотно поработали и уже показали первую версию VODKA. Доклад с его презентацией назывался «Create index … using VODKA». Сейчас работаем над ней дальше.

Можно подробнее о VODKA, что она даст?
Это связано с индексацией вложенных структур. Идея в том, чтобы сделать конструктор индексов. Сейчас существуют B-tree, Hash, GiST, SP-GiST и GIN. Пять индексных методов доступа. Мы хотим использовать их вместе с помощью VODKA.

Это откроет нам дорогу для индексации многих интересных вещей. Например, полнотекстовый поиск можно совместить с пространственным поиском. Скажем, «найти все рестораны, близкие ко мне» — это пространственный поиск. А можно найти только те рестораны, в меню которых находится «водка». Это комбинация. Пространственный поиск осуществляется с помощью GiST, скажем дерева R-tree, а полнотекстовый поиск с помощью обратного индекса.

АСТРОНОМ, ОКОНЧИЛ АСТРОНОМИЧЕСКОЕ ОТДЕЛЕНИЕ ФИЗФАКА МГУ, ОСНОВНОЕ МЕСТО РАБОТЫ — ГАИШ МГУ

ЗАНИМАЕТСЯ АУДИТОМ И КОНСАЛТИНГОМ, ЧИТАЕТ ЛЕКЦИИ, ЧАСТО ВЫСТУПАЕТ НА КРУПНЫХ МЕЖДУНАРОДНЫХ КОНФЕРЕНЦИЯХ

Совместить, и все будет работать в одном индексе. Это очень интересно, такого нет нигде и ни у кого. Здесь мы впереди всех, делаем доклады, и Mongo уже смотрят на нас косо.

Что ждет PostgreSQL в ближайшем будущем?
Ближайшее, о чем сейчас все думают, — это Pluggable Storage. Известно, что PostgreSQL — основа нескольких десятков коммерческих БД: Greenplum, AsterData… В общем, почти все коммерческие БД берут PostgreSQL и вытаскивают из нее storage manager. Проблема такая — storage manager в постгресе захардкожен. Сейчас обсуждается идея сделать Pluggable Storage API, чтобы можно было подключать различные хранилища. Например, для аналитики нужно вертикальное хранилище. Идея в том, чтобы охватить все нужные кейсы. У нас сейчас нас row-oriented storage, и оно хорошо для целостности данных. Здесь запись либо записалась, либо нет. Но это создает оверхеды, особенно для аналитики — когда вам нужна только одна колоночка, а все равно приходится читать всю строку. Сейчас это активно обсуждается

Кроме того, мы движемся в сторону автоматического шардинга. Крупная европейская компания 2ndquadrant получила грант от Европейского сообщества, в рамках FP7, на работы по расширяемости, по масштабированию PostgreSQL. Основы логической репликации уже закоммичены. Работа начата. У нас есть встроенная потоковая репликация — асинхронная, синхронная, каскадная, и сейчас делается логическая. Логическая репликация — это очень… сильный и важный шаг.

Недавно анонсировали также проект Postgres-XL. У нас ранее существовал проект Postgres-XC, это масштабируемый мультимастер. Он остался и по сей день исследовательским проектом, в который добавляют все новые и новые фичи, его поддерживают японцы. Теперь анонсировали коммерческую компанию Postgres-XL (не знаю, почему не XXL, наверное, оставили на будущее). Postgres-XL ориентирован именно на масштабируемый Postgres-мультимастер, который можно купить с поддержкой. Так что работа в этом направлении тоже идет, мы как раз недавно обсуждали, как будем осуществлять взаимодействие.

В нашей области работы тоже хватает. Есть прототип полнотекстового поиска, который мы уже пару раз показывали на конференциях. Мы с Сашей Коротковым сделали прототип, который работает быстрее, чем Sphinx.

Андрей Аксёнов (автор Sphinx. — Прим. редакции) расстроится :).
Да нет, не расстроится. Это разные вещи. Мы знаем, почему у нас все быстрее, и, если Аксёнов спросит, мы ему расскажем. Потому что, в принципе, мы не должны быть быстрее. Потому что Sphinx — это standalone поисковик, который волен делать, что угодно, у него нет кислотной нагрузки (ACID). У нас же традиционный оверхед всего, но мы быстрее за счет лучшего индекса. Опять же у нас очень хороший GIN, который имеет внутри множество навороченных хитростей. Часть этого прототипа уже закоммичена, но, чтобы закоммитить остальное, нужно серьезно и немало трудиться.

Можете рассказать, как работает GIN? Как работает B-tree, я знаю, а вот GIN…
B-tree — хорошая, универсальная структура, жупел всех БД. Но она плохо работает с дубликатами. Дело в том, что в JSON может быть много одинаковых ключей. Но B-tree не приспособлено к дубликатам, оно приспособлено для индексирования уникальных значений (primary key). В жизни же встречается множество дубликатов, во всем том мусоре, что мы индексируем JSON. В результате B-tree получается большое и не очень эффективное.

Обратный индекс (GIN) состоит из двух частей: есть ключи (то, что мы индексируем) и по нему строится B-tree. Это называется entry tree. Еще есть posting tree, это идентификаторы страниц, на которых встречаются эти ключи. Получается некая матрица. Ключ и массив. Эта структура гораздо компактнее для дубликатов, так как все дубликаты уходят в этот массив. Но у нас лежит не массив, все лежит в виде дерева, в виде B-tree. На выходе получается хитрая структура, которая очень эффективно работает с дубликатами. В этом большое отличие B-tree от GIN.

Комьюнити

Когда вы занялись PostgreSQL всерьез, профессионально, за этим стояла какая-то компания?
Нет. Мы так и не создали ни одной компании. Так и работаем сами по себе. Нас поддерживают частные компании, в том числе и в России. Как я уже говорил, первым был Rambler. Я считаю, что он дал нам жизнь. Rambler был первым большим порталом, который сказал: «Мы Oracle ставить не будем, лучше поставим PostgreSQL».

То есть уже много лет PostgreSQL существует на донейшн от разных крупных компаний?
Да. К примеру, у нас есть небольшой контракт с 1С. Они поддерживают PostgreSQL. Из крупных иностранных компаний это EnterpriseDB, Engine Yard, Heroku. Сейчас еще Salesforce. Это крупные, серьезные игроки, например Engine Yard и Heroku — это очень большие американские хостеры.

Много лет нас поддерживали французы — компания JFG Networks (привет, Жиль!). Это благодаря ей мы сделали GIN. Еще много лет мы получали гранты в РФФИ. За что большое им всем спасибо.

Интересно. Если расширить это на open source в целом, получается, что у хороших проектов есть шанс на жизнь при поддержке крупных компаний, грантов и так далее.
Да, но для этого нужно сделать карьеру. Нужно, чтобы тебя знали. На эту тему можно рассуждать долго. Ведь сейчас крупный open source — это фактически большая корпорация. PostgreSQL, Apache — большие компании, карьера в которых делается… «по заслугам», так сказать. Сколько ты сделал, столько и значит твой голос.

Да, у нас в open source демократия. Но у нас тоже нельзя сделать изменение и сразу сабмитить его в код. Все сначала выкладывается в ревью, проходит обсуждения и споры. Это сложный процесс. Поэтому, когда компании спрашивают, сколько нам понадобится денег, я отвечаю, что на саму разработку нужно столько-то и нужно еще вот столько, чтобы добиться того, чтобы код прошел дальше. У нас очень высокие требования к качеству кода, документации, ко всему. Это очень сложно.

Как сейчас разделяются роли внутри сообщества, которое разрабатывает PostgreSQL?
Разделение, конечно, есть. Есть те, кто занимается разработкой, как мы, есть те, кто занимается PR. То есть разработка происходит в стиле free search: все время нужно что-то придумывать, исследовать, делать прототипы. Есть еще люди, без которых точно нельзя обойтись и которых мы не очень любим, они жуткие зануды, они придираются ко всему… это ревьюверы. Люди, которые отвечают за то, чтобы все было хорошо. Они прогоняют тесты, memory-checker'ы. Без них никуда.

Сами по себе мы никогда не сделали бы тот код, который сейчас есть в PostgreSQL. Для этого нужны другие скилы. Одно дело — исследовать и сделать прототип, показав, что все хорошо и отлично. И совсем другое дело — заставить все это работать на всех ОС, чтобы было 30% комментариев, чтобы названия переменных и функций были специальные. Для этого у нас есть хранители. Они говорят: «Так функцию назвать нельзя». Отвечаешь им, ладно, хорошо, тогда придумай, как хочешь.

Не пробовали подсчитать, сколько человек сейчас занимается PostgreSQL?
Основных разработчиков нашего уровня человек шестнадцать. Но мы находимся на высоком уровне, вверху. Нас часто приглашают на конференции, мы все знаем. А людей, которые просто сабмитят патчи, очень много. Просто нужно понимать разницу: одно дело — засабмитить патч и что-то поправить. Другое дело — самим задизайнить проект с нуля и сделать большой его кусок.

Как проходит разделение? Я видел, у вас есть major contributor?
Да, есть. Еще есть steering committee, которому не лень заниматься всей этой текучкой. Кто-то отвечает за серверы, кто-то за веб-сайты, кто-то за PR, кто-то проводит конференции. Кто-то же должен объявить о том, что, скажем, вчера открыли девелопмент-ветку 9.5. Для таких задач и существует данный комитет.

Как проходят ваши обсуждения?
Очень просто — mailing list. Плюс мы каждый год собираемся в Канаде, перед большой конференцией у нас есть один день, приезжают основные разработчики и обсуждают основные нужные фичи. Но в основном старый, классический mailing list. У нас есть hackers mailing list, commiters mailing list, users, general и так далее.

Что представляет собой компания PostgreSQL Global development group?
Это не компания, это то, как мы называемся. Формально это никак не зарегистрировано.

Вся сила PostgreSQL в том, что нас нельзя купить. Хотя, конечно, идея создать компанию возникала.

Почему? Почему не создать компанию, которая профессионально предлагала бы консалтинг?
Компании, занимающиеся консалтингом, существуют и так. Разработчикам компания, в общем-то, не нужна. Крупные компании, в свою очередь, сами нанимают к себе разработчиков.

К примеру, несколько человек работает в EnterpriseDB. В SalesForce взяли на работу Тома Лейна. VMware держат у себя несколько человек. Mail.Ru держит у себя Федю Сигаева.

У компаний, поддерживающих PostgreSQL, наверняка есть какие-то «хотелки». Вас просят разработать какие-то конкретные вещи, говорят, что в таком-то направлении чего-то не хватает?
Чем хорош PostgreSQL — мы ориентируемся не на что-то абстрактное, а на совершенно конкретные «хотелки» и use case. Если компания заявляет, что это нормальный, распространенный use case, и при этом дает нам денег, — это прекрасно. Но все это проходит обсуждение. Просто так «платите деньги, мы все сделаем» — нельзя.

В СВОБОДНОЕ ВРЕМЯ ЗАНИМАЕТСЯ ГОРНЫМ ТУРИЗМОМ, ИГРОЙ В ВОЛЕЙБОЛ, БЕГОМ, ПУТЕШЕСТВУЕТ