

blogerator

рубрика эксклюзивного обзора



www.blogerator.ru

Актуальные обзоры событий, личностей
и технологий в современном IT

MySQL "на стероидах": развитие через клонирование

© 2011, Игорь Савчук / Bloggerator.ru

Приобретение корпорацией Oracle компании Sun поставило под вопрос существование и характер дальнейшего развития сразу множества известных свободных технологий. В этой обзорной статье мне бы хотелось кратко рассмотреть историю, современное состояние и динамику развития и перспективы такого известного и сверхзначимого для современного интернета проекта, как сервер баз данных MySQL. Здесь мы перечислим и подробно рассмотрим специфику всех популярных существующих ныне форков MySQL, которые не только активно развиваются в последнее время, но и во многом уже превзошли своего родителя - MySQL.

Остается лишь добавить, что описываемое ниже, касается положения дел сложившегося вокруг MySQL по состоянию на начало 2011 года.

Сайт-источник:

www.bloggerator.ru

Автор обзора:

Игорь Савчук

Обратная связь:

igor@eu.by

ICQ: 466040463

Jabber: remixa@jabber.ru

Дата обзора:

Январь 2011 г.

Дата сборки документа:

01.08.2011 @ 8:03

Выпуск номер:

№02 (все остальные выпуски можно найти на сайте проекта www.bloggerator.ru)

Адреса оригинальной публикации на сайте bloggerator.ru для ваших комментариев и пожеланий:

<http://bloggerator.ru/page/mysql-na-steroidah>

Этот электронный документ разрешается свободно распространять, цитировать или использовать любым другим способом, не нарушающим его целостность и не преследующий цели извлечения выгоды, без ограничений и каких-либо дополнительных уведомлений или разрешений автора. Издание и любая печать бумажной версии этого интервью в прессе и СМИ допустима только с письменного разрешения автора интервью.

Содержание выпуска:

	0
Часть I Введение	4
Часть II Краткая история MySQL	4
Часть III ExtSQL	7
Часть IV Drizzle	8
Часть V SkySQL	10
Часть VI MariaDB Server	11
Часть VII OurDelta	14
Часть VIII Percona	17
Часть IX NoSQL	19
Часть X Справочник движков	21
Часть XI Заключение	23
Часть XII Приложение	24
	0

§ 1 Введение



Приобретение корпорацией Oracle компании Sun поставило под вопрос существование и характер дальнейшего развития сразу множества известных свободных технологий. В этой обзорной статье мне бы хотелось кратко рассмотреть историю, современное состояние, динамику развития и перспективы такого известного и сверхзначимого для современного интернета проекта, как сервер баз данных MySQL. Последовавший массовый исход ведущих разработчиков MySQL из Oracle, включая и самого основателя проекта - Майкла Вайдиниуса, - невольно заставляет усомниться в будущем Oracle MySQL. Поэтому здесь мы перечислим и подробно рассмотрим специфику всех популярных существующих ныне форков MySQL, которые не только активно развиваются в последнее время, но и во многом уже превзошли своего родителя - MySQL.

Остается лишь добавить, что описываемое ниже, касается положения дел сложившегося вокруг MySQL по состоянию на начало 2011 года.

§ 2 Краткая история MySQL

Сервер реляционной [базы данных MySQL](#) (RDBMS MySQL), в короткое время успел стать сверхпопулярной базой данных, а также незаменимой частью современного Интернета, входя в священную связку из "большой четверки" открытых web-технологий LAMP (Linux-Apache-MySQL-PHP), которая и формирует технологически по большей части весь современный Web. Роль баз данных в этой связке, в наш насыщенный информацией век, все более и более приобретает исключительный характер, и поэтому архитектура современного сайта всегда подразумевает наличие быстрого и гибкого

хранилища информации, роль которого в современном интернете в большинстве случаев уготована именно MySQL.

Исторически, первая более-менее работоспособная публичная версия MySQL имела номер 3 и была выпущена в 2000 году. Эту версию можно лишь условно назвать базой данных, - это была скорее лишь заготовка для будущего сервера БД с самой минимальной поддержкой SQL. Затем в 2004 году, после очень продолжительного тестирования и обкатки, зарелизилась очень удачная и хорошо сбалансированная по функционалу и качеству кода версия 4, которая и получила самое широкое распространение в Web-проектах, создав ту самую популярность, которую проект MySQL пожинает и по сей день.

В ней впервые появились объединения, подзапросы, поддержка b- и r-деревьев, короче, тот минимум SQL, который реально позволял использовать MySQL как свободное и эффективное хранилище данных для реальных малых и средних проектов по всему миру. Ровно через год спешно была доработана и выпущена новая версия 5, которая представляет из себя уже вполне законченную и логически зрелую базу данных со всех точек зрения, содержащую в себе такие продвинутые возможности, как например, хранимые процедуры, представления, курсоры, триггеры, транзакции и многое другое. Версия 5.1, вышедшая в 2008 году - это дальнейшая шлифовка той же "пятерки", с добавлением планировщика событий, поддержки плагинов и расширений, хранения логов и статистик сервера в виде таблиц БД и многие другие декоративные изменения-удобства.

По мере прихода известности и начала активного развития проекта, у него стали появляться и серьезные проблемы, качество разработки только ухудшили многочисленные перепродажи и смены собственника компании-разработчика MySQL AB. Например даже последняя версия MySQL 5.1 содержит ряд серьезных ошибок, которые хорошо документированы и известны, приводящих к краху сервера или выдаче неверных результатов - их исправление откладывается постоянно на потом, якобы в силу их серьезной архитектурной природы. Также в новых версиях серьезно упала производительность по сравнению с версией 4.1, короче, я бы назвал это типичной проблемой роста известного проекта. На этом переходном этапе, когда смелые желания, вскруженные успехом головы, и порождаемые столь быстрым развитием сложные проблемы стали опережать реальные возможности разработчиков, где-то в районе 2008-2009 годов появилось много недовольных вышеописанным положением дел, после чего стали как грибы после дождя появляться альтернативные форки-реинкарнации хорошо известного нам MySQL.



Майкл Вайдиниус (Michael "Monty" Widenius) - основатель MySQL

Из множества приведу только один пример, зато самый яркий, - это увольнение самого автора и создателя MySQL Майкла Вайдиниуса (Michael "Monty" Widenius) из Sun, именно из-за причины чрезвычайно некачественного производственного цикла разработки MySQL установленного в стенах Sun Microsystems, когда ещё объективно не готовую программу (кстати, предназначенную для промышленной эксплуатации) административными мерами принуждают релизить к заранее жестко установленным датам, объясняя это какими-то туманными маркетинговыми соображениями. По мнению Майкла Вайдиниуса, разработка также должна вестись в едином открытом пространстве, без разделения исходных текстов на коммерческую и комьюнити ветки, как это сделала Sun. Так, например, в ветке MySQL 5.1 до сих пор не исправлены 20 хорошо известных критических ошибок, приводящих к краху процесса или к искажению данных.

Число критических ошибок можно смело увеличить еще на 35, если учитывать

нерешенные проблемы ветки 5.0, которые, скорее всего, присутствуют и в 5.1. На фоне этого нарастающего бардака терпение Майкла лопнуло и он решил уволиться после того, когда даже критическая ошибка, связанная со сбоем при изменении первичных ключей в режиме построчной репликации, не остановила Sun от выпуска финального "стабильного" релиза MySQL 5.1. Надо ли говорить, что с переходом MySQL в собственность Oracle, ситуация с разработкой только ещё больше ухудшилась. Все эти чрезвычайно деструктивные тенденции очень сильно подхлестнули творчество разработчиков этой популярной БД "на стороне", и если посмотреть ретроспективно, то 2008-2009 год - это резкий всплеск создания проектов-форков MySQL, и многие из них, забегая чуть вперед, оказались очень даже перспективными и успешными. Теперь давайте обзорно рассмотрим главные из них на начало 2011 года.

§ 3 ExtSQL

В развитии разных веток их разработчики пошли разными концептуальными путями. С одной стороны сделана попытка наращивания ещё большей функциональности и добавление множества новых фишек, - яркий например такого подхода, это проект компании Software Workshop [под названием ExtSQL](#).

Этот проект параллельно ведет две ветви разработки, базирующихся соответственно на кодовой ветке 4-ой и 5-ой версий оригинального MySQL. ExtSQL разрабатывался с сильным уклоном для специализированного использования в системах web-хостинга и призван решить проблемы, связанные с организацией учета потребления ресурсов. Администраторы ExtSQL получают возможность полного мониторинга активности пользователей, всех баз и соединений.

Например, запрос `"SHOW STATISTICS select, insert FROM user HISTORY"` позволит узнать число запросов "select" и "insert" совершенных пользователями за последний час. Естественно, для этого в бывший MySQL добавлены новые команды и расширен диалект SQL, при этом сохранена полная обратная совместимость с MySQL. Кроме поддержки своей версии MySQL, организация-разработчик Software Workshop входит в состав технического комитета INCITS H2, участвующего в развитии стандарта SQL, и активно пытается добиться расширения SQL в плане добавления возможностей для учета потребления серверных ресурсов. В целом, если говорить языком цифр, то независимые тестирования показывают, что плата за подобные надстройки над традиционным MySQL выражается в потере производительности сервера в среднем на 5-6% на обычных задачах, и на 12-16% при интенсивной работе с реально большими базами данных.

Итак, подводя итог и выделяя главный видовой признак этого клона: ExtSQL - это своего рода редакция "**MySQL Server Advanced Hosted Edition**", для особенно требовательных к статистике и мониторингу администраторов СУБД этого класса.

Дополнительные ссылки по этой теме:<http://www.extsql.com/intro.php><http://www.extsql.com/faq.php>

§ 4 Drizzle

Совершенно противоположной дорогой пошел другой, уже достаточно популярный и перспективный форк MySQL - [проект Drizzle](#). Этот проект основан бывшим директором MySQL по архитектуре Брайаном Эйкером, и представляет собой упрощенный и более быстрый вариант MySQL, в котором тщательно отобраны и удалены все ресурсоемкие и маловостребованные возможности MySQL 5. Часть из этих возможностей всё же возможно реализовать через подключаемые плагины. Эта СУБД позиционируется как высокоскоростная и высоконадежная БД, с поддержкой ACID-транзакций. В качестве хранилища используется InnoDB и PBXT. Что интересно, что весь сишный исходный код из MySQL был полностью переписан на языке C++. Управление проектом находится в руках независимого сообщества.

В отличие от SQLite, Drizzle не претендует на роль встраиваемого решения и реализован в виде сервера. Архитектура Drizzle построена на основе идеи микро-ядра, исповедует максимальное упрощение структуры БД и вынос логики на сторону приложений. В частности, такой дизайн СУБД позволяет организовать обработку просто ОГРОМНОГО числа параллельных запросов, при выполнении которых в полной мере задействуются мощности современных многоядерных CPU, как результат - Drizzle'овские пиковые показатели интенсивности обмена запросами-ответами с web-приложением завесят любой стандартный сервер MySQL 5.1, который просто захлебнётся под такой нагрузкой (тысячи или десятки тысяч сетевых соединений в случае "обычного" MySQL почти гарантированно вызывают проблемы с памятью сервера БД - см. открытые ошибки [bug#26590](#), [bug#33948](#), [bug#49169](#), так что не думайте, что при такой нагрузке достаточно просто обновить железо сервера).



Брайан Эйкер, основатель Drizzle, бывший директор MySQL по архитектуре, ведущий разработчик Memcached, автор оригинального балансировщика на Slashdot.org (сайт входит в TOP20 самых посещаемых ресурсов мира)

Кроме этого, в Drizzle дополнительно реализованы встроенные средства для разнесения данных по ключевому полю (шардинг) на кластер из нескольких машин, для создания эффективной балансировки нагрузки для по-настоящему сверхнагруженных проектов. По сравнению с MySQL в Drizzle удалена поддержка хранимых процедур (вместо CREATE FUNCTION следует использовать связываемые объекты), триггеров, кэша запросов (query cache), представлений (view), операции GRANT и ALTER, ограничений ACL, команды SHOW, предварительно подготовленных запросов (prepared statement) и др. Прекращена поддержка малоиспользуемых типов данных из MySQL. На вопросы оппонентов, как же можно использовать БД, например, без view'ов, разработчики отвечают в том ключе, что "все те, кому действительно серьезно нужны view'ы - уже давно сидят на PostgreSQL или Oracle, это же касается и всех других продвинутых фичей".

Да, действительно, для запуска очень многих, например, блоговых движков написанных в связке с MySQL уже под Drizzle - понадобится модификация и некоторый тюнинг кода этих движков, впрочем, как успокаивают разработчики, изменения эти невелики и возможностей Drizzle на самом деле более чем достаточно для полноценного функционирования большинства популярных CMS, тем более, что сообщество уже кастомизировало некоторые известные php-движки под Drizzle, что позволяет показывать их рекордную производительность на том же железе, на котором раньше крутился старый-добрый MySQL. В качестве приятного побочного эффекта, на Drizzle перестали проходить многие популярные разновидности техник по sql-injections для MySQL, так что безопасность стала невольным бонусом этого проекта, попутно демонстрируя нам всем глубокие философские выводы о том, что *минимализм* есть почти тождество слова *безопасность*.

До прихода в MySQL AB, *Брайан Эйкер*, создатель Drizzle, долгое время работал разработчиком в Slashdot.org – сверхпопулярного и культового веб-сайта, где он реализовал уникальный балансировщик нагрузки. Стало уже нарицательным [понятие слэшдот-эффект](#), но, по правде говоря, от такого наплыва посетителей Slashdot и сам регулярно падал, и наиболее узким местом всегда становились базы данных. И хотя в MySQL не всегда получалось реализовать найденные в то время эффективные решения, в свой Drizzle, по словам Брайана, он попытался вложить все, что ему известно о специфике работы баз данных в режиме сверхвысоких нагрузок. Поэтому эта СУБД изначально позиционируется как высокоскоростная и высоконадежная облачная БД, с поддержкой ACID-транзакций. В качестве хранилища данных используется стандартные для MySQL InnoDB или PBXT, поэтому миграция должна пройти без проблем.

Дополнительные ссылки по этой теме:

<http://docs.drizzle.org/>

<http://wiki.drizzle.org/>

§ 5 SkySQL

Рассмотрев, так сказать крайности, макси- и мини-варианты оригинального MySQL, давайте посмотрим какие есть т.с. более традиционно-классические, продолжения этого известного проекта. На фоне тотального исхода из Oracle разработчиков MySQL (из Oracle добровольно "катапультировалось" уже более 70% оригинальной команды MySQL AB), они, разбредаясь по миру, пытаются как-то заново пристроиться уже под новую историческую эпоху для MySQL.

Так появилась [компания SkySQL](#), основанная уволившимися из Oracle разработчиками MySQL, частично финансируемая несколькими бывшими инвесторами MySQL AB и

возглавляемая *Ульфом Санбергом*, бывшим вице-президентом по предоставлению глобальных сервисов MySQL. Новая компания пока не намерена развивать свой форк MySQL, а займется предоставлением коммерческой технической поддержки, консалтинговых услуг, обучением и продажей готовых решений на базе MySQL. Кроме того, SkySQL занялась выпуском и поддержкой альтернативной промышленной сборки MySQL - SkySQL Enterprise. Продукт распространяется по подписке и снабжен полноценной технической поддержкой (помощь в решении проблем, консультации по настройке, устранение последствий сбоев) и консалтинговым сервисом (планирование внедрения и проведение оптимизации), т.е. фактически полностью перезапустила бизнес своего прототипа, ныне доставшегося Oracle, шведской компании MySQL AB. Так вот, здесь очень интересно, что входит в состав этой самой SkySQL Enterprise? А входит туда, кроме сопутствующих утилит, "прозрачная замена" MySQL - MariaDB Server.

Дополнительные ссылки по этой теме:

<http://www.skysql.com/>

<http://www.databasejournal.com/features/mysql/article.php/3908646/SkySQL-Takes-Aim-at-Delivering-MySQL.htm>

§ 6 MariaDB Server

Давайте зададимся вопросом - что же такое [MariaDB Server](#)?

Для этого немного откатимся в краткий исторический обзор, чтобы понять, откуда и зачем взялся этот очередной, новый форк MySQL, роль которого на данный момент чрезвычайно важна, и поэтому стоит на нем остановиться по-подробнее. Майкл "Монти", создатель и бессменный лидер проекта СУБД MySQL, покинув в конце 2008 года компанию Sun Microsystems и после прекращения непосредственного участия в разработке MySQL, задумал создать максимально совместимый с MySQL, но идейно новый сервер, базирующийся на принципиально новом движке хранилища данных. Используемое новое хранилище данных Maria - это устойчивый к сбоям клон MyISAM. Многие специалисты отмечают, что Maria Engine это, как минимум, один из лучших движков для выборки данных, из всех что сейчас существуют, тогда как в классическом MySQL дефолтный MyISAM - её самое слабое место.



Майкл Вайдиниус (Michael "Monty" Widenius) - основатель MySQL и MariaDB

В новой компании вместе с Монти над ним работает около 20 сотрудников - бывших разработчиков из MySQL AB, ушедших из Sun вслед за своим идейным лидером. Создатель MySQL заявляет, что MariaDB Server - это максимально точная и совместимая интерфейсная копия SQL используемого в оригинальном MySQL 5, тогда как внутри - он уже во многом превосходит свой протип, давший ему жизнь и стартовую кодовую базу. Более того, Монти заявил, что MariaDB отныне - это единственная официальная версия MySQL, тогда как Oracle MySQL - "is now dead". Самое страшное в этом молодом проекте это то, что многие недовольны женским названием новой БД: ну так одну из дочерей автора MySQL зовут My, а вторую - Maria. Отсюда и названия баз данных. Есть правда ещё один сын, и зовут его, на всякий случай, Макс. И MaxDB уже, кстати, создана, но здесь о ней не будет сказано ни слова.

Таким образом, суммируя, часть разработчиков после ухода из Sun и Oracle осели в компании Monty Program, где день и ночь пилят новый-старый MySQL но под уже новым названием - [MariaDB Server](#), тогда как вторая часть программистов и сервисного персонала из бывшего MySQL AB - основали компанию SkySQL по сопровождению и поддержке MariaDB Server. И кстати, по-моему мнению, MariaDB Server - это самая качественная и сбалансированная замена для MySQL, после того как Oracle его похоронит (а по моему субъективному мнению, ждать этого осталось уже не долго, и первые звоночки как бы даже уже [тревожно позванивают](#)).

За основу взята кодовая база 6-той версии MySQL, после чего в неё были внесены существенные изменения в целях ускорения производительности, повышения стабильности работы (например, реализована новая методика тестирования сборок), реализованы продвинутая статистика работы сервера и расширенная версия механизма

pool-of-threads, в сравнении с тем, что мы можем увидеть в MySQL 6.0, а также многое другое. Важно ещё раз подчеркнуть: сверхзадача проекта, при всех этих достаточно масштабных улучшениях (процент отличия кода от стартовой базы MySQL уже достиг примерно 25%, по словам Монти), - сохранить полностью совместимый SQL и API с оригинальным MySQL, вплоть до сохранения аналогичной структуры каталогов сервера и названия/ключи всех сервисных утилит. Итак, суммируя, перечислим основные свойства этого популярного клона MySQL:

- MariaDB - основанное на MyISAM высоконадежное хранилище, отличающиеся повышенной устойчивостью и сохранению целостности данных после краха, при полной совместимости с MyISAM. Благодаря ведению лога операций, в случае краха производится откат результатов выполнения текущей операции. Также поддерживается возможность восстановления состояния из любой точки в логе операций (включая поддержку CREATE/DROP/RENAME/TRUNCATE).
- В качестве замены InnoDB используется движок XtraDB, основанный на коде InnoDB-plugin и полностью совместимый с ним, но отличающийся заметно более высокой производительностью, благодаря интеграции патчей от компаний Google и Percona. В частности, в XtraDB улучшен механизм работы с памятью, улучшена работа подсистемы ввода/вывода InnoDB, добавлена поддержка нескольких потоков чтения и записи, поддержка управления пропускной способностью, реализация упреждающей выборкой данных (read-ahead), адаптивная установка контрольных точек (adaptive checkpointing), расширены возможности по масштабированию для больших проектов, система организации блокировок адаптирована для работы на системах с большим числом CPU, добавлены дополнительные возможности для накопления и анализа статистики.
- В состав включено хранилище PBXT (PrimeBase XT), разработанное с нуля и поддерживающее мультиверсионный метод организации хранения данных MVCC (multi-version concurrency control), позволяющий избавиться от блокировок при выполнении операций чтения. PBXT поддерживает ACID-совместимые транзакции, быстрый откат транзакций и восстановление после некорректного завершения работы сервера. Имеются средства для обеспечения ссылочной целостности данных, поддержка определения внешних ключей (foreign key), каскадных обновлений и удалений данных. Поддерживается возможность прямого потокового ввода и вывода бинарных данных (BLOB) в БД.
- В качестве замены разработанного в Sun Microsystems и уже не поддерживаемого хранилища Federated задействован проект FederatedX. FederatedX позволяет организовать обращение к удаленным таблицам как к локальным. Имеется поддержка транзакций, одновременной установки нескольких соединений к удаленной СУБД, использования операций "LIMIT".
- Реализована возможность создания пула потоков, позволяющего использовать заданное число mysql-нитей для обработки всех запросов, вместо механизма "одна нить на одно соединение". В итоге можно значительно увеличить производительность БД, обрабатывающих большое число соединений или имеющих большое число длительно выполняющихся запросов;
- Ликвидация таблиц - новый вид оптимизации операций по объединению таблиц

(JOIN), позволяющий исключить обращения к некоторым таблицам, на которые ссылается запрос пользователя;

- Добавлено множество патчей, созданных компанией Percona и сообществом независимых разработчиков. Например, добавлен патч для выявления узких мест при работе с базой. Кроме отслеживания запросов выполняющихся более заданного времени, поддерживается занесение в лог информации о запросах приводящих к перебору большого числа строк, сложным сортировкам, задействованию временного файла на диске или удовлетворяющих параметрам заданного фильтра (qc_miss, full_scan, full_join, tmp_table, tmp_table_on_disk, filesort, filesort_on_disk). В отличии от стандартного механизма ведения лога медленных запросов, msl оперирует не секундами, с округлением в большую сторону, а отрезками 0.005 сек.

Например, на моем хостинге MariaDB прекрасно работает в связке с PHP5, обслуживая обычные WordPress'ы и прочие популярные движки, для работы с БД настроен обычный PhpMyAdmin, сами пользователи хостинга о существовании такой БД даже и не подозревают, принимая её за всамделишный MySQL. Хотя ради справедливости стоит заметить, что сами разработчики говорят, что расхождение с кодовой базой MySQL уже сейчас достигло примерно 20 процентов, и будет конечно нарастать и дальше. Также мною проверялась работа MariaDB в связке с [MySQL Proxy](#) и специализированным файрволлом для MySQL - [GreenSQL](#): обе софтины работали с MariaDB прямо как с родной MySQL, так что думаю, что как минимум одна достойная и очень перспективная альтернатива для MySQL уже состоялась.

Хочется добавить, что MariaDB Server, также как и Drizzle - очень активно развиваются и эволюционируют, и кроме этого, каждый из этих двух проектов очень хорошо смотрится для своих, несколько разных целевых ниш. Кстати говоря, буквально на днях, у MariaDB вышел первый официально стабильный релиз за всю её молодую историю, подробности можно [почитать тут](#).

Дополнительные ссылки по этой теме:

<http://mariadb.org/>

<http://kb.askmonty.org/en/mariadb>

§ 7 OurDelta

Следующий проект, который мы кратко рассмотрим - это дистрибутивы от [проекта OurDelta](#). На самом деле это популярные ре-сборки двух upstream-проектов: для MySQL 5 поддерживаются два параллельных дистрибутива, это OurDelta и OurDelta-Sail. Если в первый включаются билды собранные с применением к стандартным сырцам MySQL 5 только хорошо протестированной коллекции сторонних "неканонических" патчей, то во вторую, экспериментальную отчасти ветвь, включается всё самое прогрессивное и

максимально свежее что имеется на момент релиза, но, как понятно, часто оно толком ещё не обкатано в реальных условиях как следует, со всеми вытекающими отсюда последствиями.

Кроме этого, OurDelta точно также в параллельной ветке расширяет функциональность и MariaDB 5, которую считает прямым конкурентом и приемником MySQL, но здесь на данный момент есть только одна ветвь сборок - стабильная для MariaDB. Что же это за такие патчи, которыми регулярно потчует нас проект OurDelta?

Итак, OurDelta – это попытка отобрать самые интересные патчи/доработки от независимых разработчиков, после чего собрать их все в одном продукте. Сбором сторонних патчей, их интеграцией и синхронизацией – и занимается проект OurDelta. Чтобы дать возможность выбора, OurDelta ведет две параллельные ветви своих популярных ребилдов: на базе исходников классического MySQL 5.0 и вышерассмотренного MariaDB 5.0. Чтобы хотя бы немного дать представление, что же включается в билды от OurDelta, остановимся на одном из её крупных доноров – проекте Facebook.

Чтобы справиться с нагрузкой, которую создают 750 миллионов пользователей социальной сети, Facebook оперирует четырьмя тысячами экземпляров MySQL (при этом очень активно используется шардинг), всего задействовано девять тысяч инсталляций memcached. Конечно, стандартный MySQL не предназначен для подобных нагрузок, поэтому уже более 5 лет, под руководством известнейшего разработчика Марка Каллагана, идет интенсивная доработка MySQL под собственные экстремальные потребности Facebook (за историей этого проекта можно наблюдать на странице MySQL@Facebook). Неудивительно, что patchset от Facebook – один из самых интересных в составе OurDelta.



Марк Каллаган, руководитель группы разработчиков “High Availability MySQL” в Facebook

Проект также аккумулирует все интересные патчи от следующих других upstream-проектов: Percona, Google, Open Query, Proven Scaling, Sphinx Search и многих других. Наверное, благодаря этому, OurDelta можно назвать одной из самых “фичастых” сборок MySQL из всех ныне существующих. Традиционно, OurDelta особенно благоволит дистрибутиву CentOS, для которого делается специальная оптимизированная сборка.

Таким образом, проект OurDelta - это своего рода **"MySQL/MariaDB на стероидах"**, делающий продвинутые и накачанные новой функциональностью сборки MySQL/MariaDB на любой цвет и вкус. Хотелось лишь отметить, что в OurDelta очень охотно используют патчи также и от такого самобытного проекта-форка, как Percona, на котором давайте сейчас немного остановимся поподробнее.

Дополнительные ссылки по этой теме:

<http://ourdelta.org/patches>

<http://ourdelta.org/centos>

§ 8 Percona

[Компания Percona](#) основана в 2008 году двумя отечественными разработчиками, уже бывшими членами MySQL dev.team, Петром Зайцевым и Вадимом Ткаченко. Их БД-сервер основывается на кодовой базе MySQL 5.1, которая дополнена собственными многочисленными, и что хочется отдельно подчеркнуть, весьма качественными патчами, направленными на добавление новой функциональности, повышения стабильности работы и удобства администрирования. Но главная фишка проекта - интеграция собственного движка XtraDB, основанного на коде InnoDB-plugin и полностью совместимого с ним, но отличающегося существенно более высокой производительностью. По умолчанию XtraDB не изменяет дефолтный формат хранения данных в InnoDB. Вы можете прозрачно переключаться между XtraDB и InnoDB по нескольку раз в день.

Кроме того, Percona поставляет вместе с этим движком очень мощную бэкаповую систему [XtraBackup](#), которая позволяет решать и автоматизировать даже самые экзотические задачи из области сохранности вверенных на хранение серверу БД данных.



Петр Зайцев, основатель Percona, бывший менеджер по поддержке из "High Performance Group" в MySQL AB, со-автор известной книги издательства O'Reilly - "High Performance MySQL"

Так вот, в экспертных кулуарах сейчас (на конец 2010 года) как бы считается, что основная острая конкуренция разгорается именно между движками [Maria Engine](#) (движок совсем недавно [был переименован](#) в Agia) и как раз [Percona XtraDB](#) - никаких более продвинутых на данный момент storage engines просто нет в природе, так что старичок MySQL будет потихоньку по-любому сдавать свои позиции на фоне его более активных и озабоченных развитием конкурентов, если конечно Oracle вдруг не опомнится и не станет вкладывать силы/ресурсы в его дальнейшую глубокую разработку.

Поскольку по словам самих разработчиков, базовая формула проекта - **Percona Server = MySQL + XtraDB + XtraBackup**, - давайте хотя бы кратко ещё пройдемся по составным частям этого равенства, отчасти уже упомянутым выше.

XtraDB – это новый движок данных, который был специально разработан для Percona,

как замена стандартного InnoDB. Сам XtraDB основан на InnoDB plugin. Важно, что XtraDB на 100% совместим с InnoDB, поэтому вы можете использовать его как полную и прозрачную замену для InnoDB. XtraDB разрабатывался для улучшения масштабируемости на современном железе, а также включает в себя множество других возможностей и патчей, специально оптимизированных для высоко нагруженных систем. Установить этот новый движок очень просто. Первый, ручной вариант, это пересобрать MySQL по аналогичной схеме с InnoDB Plugin – т.е. просто заменить библиотеки на XtraDB. Второй вариант более предпочтителен и проще – довериться сборке MySQL от компании Percona, которая уже включает в себя подключенный XtraDB. Думаю, факт включения движка XtraDB в конкурирующую с Percona дистрибуцию MariaDB – это лишнее подтверждение качества этого решения.

Очень кратко остановимся и на второй значимой компоненте Percona - **Percona XtraBackup**. Говоря кратко – это мощнейшее средство для организации сколь угодно сложных схем бэкапа БД. Не создавая дополнительной нагрузки на сервер, XtraBackup позволяет делать полные, инкрементальные, частичные резервные копии баз данных, работающих под управлением СУБД MySQL (и его клонов). В качестве эпитета достаточно сказать, что авторитетная конференция O'Reilly MySQL Conference & Expo наградила это решение титулом “*Application of the Year 2011*”.

Интересной особенностью дистрибутива MySQL от Percona является также и то, что упомянутый выше плагин HandlerSocket уже интегрирован в последние сборки Percona, поэтому, при желании, вы можете пользоваться режимом NoSQL-запросов к базе данных на движке XtraDB прямо “из коробки”, что даст весьма существенное повышение производительности сервера БД.

В заключении, отмечу, что Percona – очень интересный и сбалансированный дистрибутив MySQL, очень жестко конкурирующий по своей популярности с MariaDB, тем более приятно, что подавляющее число сотрудников этой компании – наши соотечественники. В дополнение, интересное видео с выступлением по-русски и подробным обзором проекта Percona можно [посмотреть здесь](#) с Евгением Степченко.

Дополнительные ссылки по этой теме:

<http://www.percona.com/docs/wiki/percona-xtradb:info:faq>

<http://www.percona.com/software/percona-xtrabackup/>

<http://www.mysqlperformanceblog.com/>

§ 9 NoSQL

В заключении хотелось бы сказать, что подбор наиболее подходящей к конкретному техническому заданию модификации MySQL вовсе не ограничивается выбором

оптимального дистрибутива и тюнинга его настроек, но также и принципиальным режимом работы самого сервера. В качестве примера приведу недавнюю успешную реализацию подхода NoSQL на базе MySQL-сервера. Яшинори Матсунобу (Yoshinori Matsunobu), автор этого метода и нового плагина HandlerSocket, ещё один бывший участник MySQL dev.team, в своей подробной [технической статье](#) "Использование MySQL в режиме NoSQL - История о том, как достичь обработки 750,000 запросов в секунду" рассказывает об этой новой методике, а также делится тестами и реальным кодом клиентов, который можно уже прямо сейчас использовать в своих высоконагруженных проектах.



Яшинори Матсунобу, бывший разработчик из MySQL AB, в настоящий момент работает в DeNA - японской игровой социальной сети (по его словам, крайне интенсивно использующей БД построенные на MySQL)

Ну что тут сказать - решение очень красивое, при этом позволяет гибко переключаться между обычным SQL-синтаксисом из обычных клиентов, так и собственным HandlerSocket-протоколом для переключения в режим сверхвысокой производительности через реализованный в виде плагина NoSQL-режим работы MySQL. И всё это чудо мгновенного преобразования в суперпроизводительную БД - на самом обычном железе! Работая в этом режиме, автор разработки, спроектированной для крупнейшей японской игровой социальной сети, где он работает, фактически уперся в пропускную способность имеющегося у него 1Gb канала, так что расти ещё есть куда ;-)

В качестве дополнения к этой теме, также хочется упомянуть и недавние экспериментальные сборки самой Oracle, которая делает попытки по интеграции **memcached** через специализированный плагин в оригинальный MySQL. Готовые

тестовые сборки “mysql-5.6-labs-innodb-memcached” можно скачать на официальном сайте *MySQL Labs*.

Дополнительные ссылки по этой теме:

<http://l-o-n-g.livejournal.com/153756.html>

<http://tokarchuk.ru/2010/12/handlersocket-protocol-and-php-handlersocket-extension/>

<http://habrahabr.ru/blogs/nosql/123452/>

http://blogs.oracle.com/mysql/entry/nosql_to_mysql_with_memcached

<http://blogs.innodb.com/wp/2011/04/nosql-to-innodb-with-memcached>

§ 10 Справочник движков

Если немного упрощать, то база данных — это обертка вокруг движка хранения данных. Она занимается приемом запросов и управлением ими, кэшированием и прочими обслуживающими функциями, обеспечивая работу с низкоуровневым API движка. Последний, в свою очередь, собственно и хранит данные (на диске или в памяти), работает с операционной системой и обеспечивает выдачу нужных выборок по запросу от сервера. Если раньше СУБД (связка "сервер + движок") была монолитная, то теперь в большинстве современных систем используется структура с плагинами.

Движок при такой организации является просто модулем, а сам сервер не зависит от системы хранения данных. В последних редакциях классического MySQL также используется плагиновая архитектура. Поэтому классический встроенный движок InnoDB (правда, обычно устаревшей версии) можно легко заменить на модуль другого проекта, который часто будет лучше. В альтернативных мускулу разработках, в том числе MariaDB или Drizzle, все движки изначально выполнены как плагины.

Попробую здесь кратко пробежаться по современным движкам хранения данных в MySQL-совместимых СУБД:

- **InnoDB** — основной движок для мускула, который с версии 5.5 наконец-то сделали дефолтным. Поддерживает транзакции, репликацию, построчную блокировку. Достаточно устойчив к сбоям.
- **MyISAM** — очень проблемный движок, плохо переносящий крах сервера. Не поддерживает транзакции, но зато может похвастаться полнотекстовыми индексами и быстротой работы. Долгое время был стандартным для всех версий MySQL, а потому до сих пор является самым популярным.
- **Aria** — замена для MyISAM с поддержкой транзакций и улучшенной работой с памятью. Движок гарантирует целостность данных и при этом не уступает в скорости MyISAM.
- **CVS** — специализированный движок на случай, когда требуется хранить и обрабатывать большие массивы строковых данных, разделяемых запятой.

- **Federated/FederatedX** — этот движок специализируется на прозрачном разнесении данных по нескольким серверам (физическим) на уровне таблицы.
- **PBXT** — призванный заменить InnoDB новый движок, в котором реализованы полная поддержка транзакций, многоверсионность, автоматическая обработка дедлоков. Движок оптимизирован для большого количества одновременных транзакций.
- **Blackhole** — служебный движок, представляющий собой, по сути, /dev/null для СУБД и фактически не производящий никаких записей на диск. Используется для репликации.
- **Archive** — движок, который максимально быстро работает на запись. Используется в тех случаях, когда необходимо хостить большие массивы данных. Для эффективности хранения используется сжатие, что приводит к медлительности во время выборки. Движок хорошо подходит для долговременного хранения логов и другой служебной информации.
- **XtraDB** — расширенная и исправленная в некоторых проблемных местах InnoDB от компании Percona.
- **MERGE** — схожий с Federated движок для разнесения данных в одной таблице на несколько разных.
- **MEMORY** — движок, использующийся для хранения данных не на диске, а в памяти. Информация из базы доступна только во время работы сервера, но это дает колоссальный прирост в производительности.
- **BlitzDB** — еще одна замена для MyISAM с хорошей производительностью за счет встроенного построчного кэширования и автоматического восстановления после сбоя. Движок не поддерживает транзакции.
- **NDB** — движок для кластера, обладающий, впрочем, кучей проблем и удручающе плохой производительностью.
- **Falcon** — легендарный движок от компании MySQL AB, разрабатываемый еще со времен Sun, когда было принято решение заменить оракуловский InnoDB.
- **SphinxSE** — очень интересный полнотекстовый движок от создателя поискового сервера Sphinx. Лучший вариант для полнотекстового поиска и индексации по правилам русского языка. Легко оперирует терабайтами данных, обеспечивая при этом все возможности современной БД.

Так как я, что называется, не ровно дышу в сторону **SphinxSE**, воспользуюсь своим положением редактора и очень кратко опишу его ниже.

Так как автор этого движка (Andrew Aksyonoff) проживает в России, то в Sphinx имеется отличная поддержка русского языка, включая возможность подключения русскоязычного стеммера. Что касается поискового механизма Sphinx (выше речь шла о патче для MySQL входящем в комплект Sphinx), то кроме MySQL имеется поддержка PostgreSQL. При измерении производительности Sphinx выполнил запрос в 15 раз быстрее чем Mongoose, в среднем потратив 0.1 сек при индексации 2-4 Гб текста. Разработчики заявляют, что Sphinx неплохо себя чувствует при объемах индексируемых данных до 100 Гб или 100 миллионов документов.

Основные особенности Sphinx:

- Высокая скорость индексации (до 10 Мб/сек);
- Высокая скорость поисковой выборки (0.1 сек. для запроса в базе с 2-4 Гб проиндексированного текста);
- Высокая масштабируемость (один CPU может использоваться для обслуживания 100 Гб текста и 100 млн. документов);
- Возможность организации распределенного на несколько узлов поискового механизма;
- Нативная поддержка MySQL, поддерживаются как стандартные движки MyISAM и InnoDB, так и собственное хранилище SphinxSE
- Поддержка поиска по заданным фразам;
- Возможность вероятностного поиска с использованием ранжирования с элементами нечеткой логики;
- Поддержка стеммеров для русского и английского языков;
- Поддержка индексации любого числа полей в документах, вес для которых может изменяться на лету;
- Поддержка групп документов;
- Возможность определения запрещенных слов
- Поддержка различных моделей поиска (совпадение всех ключей, совпадение фразы целиком, совпадение отдельных слов);
- XML интерфейс для интеграции со сторонними проектами;
- Доступно API для PHP, Python, Java, Perl и Ruby.

Дополнительные ссылки по этой теме:

<http://habrahabr.ru/blogs/mysql/64851/>

<http://dev.mysql.com/doc/refman/6.0/en/storage-engines.html>

<http://www.softwareprojects.com/resources/programming/t-mysql-storage-engines-1470.html>

<http://sphinxsearch.com/info/faq/>

<http://www.supportsages.com/blog/2010/08/mysql-storage-engines-an-overview-their-limitations-and-an-attempt-for-comparison/>

<http://forge.mysql.com/projects/storage-engines-list>

§ 11 Заключение

В заключении, уж коль мы закончили на критерии производительности - [здесь можно](#) почитать про реальное сравнительное тестирование некоторых наших героев, ну а выводы делайте сами.

- Таким образом, подводя итог и резюмируя, на данный момент можно констатировать, что в MySQL доступны три рабочих интерфейса: **Memcached**, **HandlerSocket** и **традиционный SQL**.

Завершая свою обзорную статью о жизненном цикле MySQL, хочется пожелать вам вдумчивого выбора, приемлемых нагрузок на ваших серверах БД, и конечно же всем нам - дальнейшего развития замечательных форков, такое дикое разнообразие которых объединяет только одно - общий знатный родитель - замечательная СУБД MySQL!

§ 12 Приложение

Этот раздел появится чуть-чуть позже. Пожалуйста, для этого заново скачайте это руководство с сайта bloggerator.ru в сентябре 2011г. (vdfretyuu3)

Выпуск №02 от 01.08.2011

*все остальные выпуски и интервью можно найти
на сайте проекта (www.blogerator.ru)*

*Этот электронный документ разрешается свободно
распространять, цитировать или использовать любым другим
способом, не нарушающим его целостность и не преследующий
цели извлечения выгоды, без ограничений и каких-либо
дополнительных уведомлений или разрешений автора. Издание и
любая печать бумажной версии этого интервью в прессе и СМИ
допустима только с письменного разрешения автора интервью.*



www.blogerator.ru

Актуальные обзоры событий, личностей
и технологий в современном IT