

Computerworld Россия №22, 2011

NewSQL берет все лучшее от мира SQL и NoSQL

Легенда мира СУБД Майкл Стоунбрейкер критикует отказ от использования SQL в распределенных системах новой архитектуры

Джаоб Джексон

Служба новостей IDG, Нью-Йорк



МАЙКЛ СТОУНБРЕЙКЕР:

«Те, кто отдает предпочтение набирающим популярность NoSQL-системам перед реляционными СУБД, возможно, выплескивают из ванны ребенка вместе с водой»

Фото: CC0 1.0 Dcoetzee

Пользователи, которые отдают предпочтение набирающим популярность NoSQL-системам перед реляционными СУБД, возможно, выплескивают из ванны ребенка вместе с водой, заявил на прошедшей в Сан-Хосе конференции NoSQL Now Майкл Стоунбрейкер, один из крупнейших авторитетов в области СУБД. Стоунбрейкер — главный архитектор Ingres и Postgres, активный участник разработки многих других систем. Он также является сооснователем компании Vertica, создателя одноименной столбцовой СУБД, которая в феврале нынешнего года была приобретена Hewlett-Packard. Сейчас Стоунбрейкер — директор по технологиям новой компании VoltDB, предлагающей программную систему распределенной обработки данных.

По убеждению патриарха СУБД, широко известный язык запросов SQL можно было бы после небольшого усовершенствования использовать с новыми горизонтально масштабируемыми NoSQL-системами, благодаря чему они могли бы обрести полноценную гибкость.

Этот новый подход Стоунбрейкер называет NewSQL. Компания VoltDB и сама предлагает СУБД, основанную на принципах NewSQL, однако аргументация Стоунбрейкера в пользу новой архитектуры звучит более убедительно, чем типичные рекламные восхваления.

Реляционные СУБД — действительно «вымирающий вид», как об этом заявляют сторонники концепции NoSQL, полагает Стоунбрейкер. Однако виноваты в этом поставщики СУБД, а не SQL как таковой. Реляционные СУБД, уверен он, медлительны вовсе не из-за того, что поддерживают SQL.

Большинство коммерческих реляционных систем управления базами данных, доступных на рынке, появились 30 лет назад или больше, отметил Стоунбрейкер. Они не были рассчитаны на применение в условиях нынешних транзакционных сред, обрабатывающих гигантские объемы данных. Десятилетиями реляционные СУБД обростали новыми возможностями сомнительной ценности, в результате превратившись в неимоверных размеров программные «пузыри».

«Традиционные реляционные СУБД дальше не масштабируются, — заявил Стоунбрейкер. — Если вам не нужна производительность, это не имеет значения, но если вам все же требуется высокое быстродействие, то традиционные SQL-системы не ваш выбор».

Медлительность баз данных можно отнести на счет нескольких факторов, отметил он. Реляционные системы обслуживают буферный пул, ведут журналы операций для нужд восстановления, а также управляют блокировками полей данных, предотвращающими их перезапись конкурирующими операциями. Согласно результатам испытаний, проведенных в VoltDB, все эти задачи отнимают до 96% системных ресурсов.

Становящиеся популярными NoSQL-системы, такие как MongoDB и Cassandra, многие рассматривают как альтернативу, лишенную ограничений, которые присущи обычным реляционным СУБД.

На одном из заседаний NoSQL Now консультант Дэн Маккрири разъяснил недостатки традиционных баз данных, побудившие разработчиков создать системы NoSQL. По мнению Маккрири, реляционные базы данных не обладают необходимой гибкостью. Их архитектура, разработанная еще в эпоху перфокарт, реализует фиксированный подход к моделированию данных. Если организации нужно добавить новый столбец к таблице, приходится изменять схему базы, что может вызвать определенные трудности. При этом сама схема не всегда точно отражает исходную модель данных, отметил эксперт: «Существует множество видов данных, которые нельзя разместить в таблице. Табличный формат накладывает слишком много ограничений».

Еще один недостаток традиционных SQL-систем состоит в том, что они плохо масштабируются за пределами одиночного сервера, полагает Маккрири. Когда объем данных превышает возможности одного сервера, их приходится делить между несколькими системами, а с этим могут быть определенные сложности. Кроме того, при исполнении СУБД на группе серверов могут возникнуть трудности с выполнением некоторых операций,

например внешних соединений, при которых консолидируются данные из нескольких таблиц.

Хотя NoSQL-системы обладают большей гибкостью и масштабируемостью, у них тоже есть свои ограничения, указал Стоунбрейкер. Ввиду отсутствия поддержки SQL такие системы лишены способности выполнять структурированные запросы с математической точностью. Созданный на основе алгебры отношений и реляционного исчисления, SQL дает гарантию того, что хорошо структурированный запрос, даже очень сложный, захватит из базы все необходимые данные.

Есть и другие проблемы. Например, NoSQL-системы не обеспечивают соответствия операций требованиям ACID (atomicity, consistency, isolation, durability — «атомарность, непротиворечивость, изолированность, долговечность») — стандарта, который гарантирует точность выполнения оперативных транзакций средствами СУБД, даже если работа системы прерывалась.

Средства обеспечения соответствия ACID можно реализовать на уровне приложения, однако написание соответствующего кода, по словам Стоунбрейкера, это «хуже смерти». Наконец, у каждой NoSQL-системы есть свой собственный язык запросов, в связи с чем затруднена стандартизация интерфейсов приложений.

NewSQL же, по убеждению Стоунбрейкера, обеспечивает гарантии качества выполнения транзакций, свойственные SQL-системам, и при этом обладает масштабируемостью на уровне NoSQL-систем.

Подход NewSQL основан на ряде инновационных архитектурных решений, указал Стоунбрейкер. В NewSQL не используется ресурсоемкий буферный пул, поскольку база данных целиком находится в основной памяти. Новый метод также устраняет потребность в краткосрочных блокировках данных, поскольку система исполняется на сервере строго в виде одиночного потока (хотя иные типы блокировки все же будут создавать определенную нагрузку). А «дорогостоящие» операции восстановления исключаются за счет применения дополнительных серверов для тиражирования и переключения нагрузки при отказе.

По утверждению Стоунбрейкера, система VoltDB, основанная на методах NewSQL, может выполнять транзакции в 45 раз быстрее, чем типичная реляционная СУБД. Кроме того, возможности горизонтального масштабирования позволяют распределять VoltDB между 39 серверами, а при исполнении на 300 процессорных ядрах система выполняет до 1,6 млн транзакций в секунду. Ей также требуется гораздо меньше серверов, чем типичной инсталляции Hadoop, — VoltDB выполняет тот же объем работы на 20 узлах, для которого Hadoop потребуются тысяча узлов.

Несмотря на то что аудитория слушателей состояла в основном из пользователей и разработчиков NoSQL, многие согласились с доводами Стоунбрейкера.

«Стоунбрейкер верно идентифицировал основной недостаток SQL-систем, — считает Маккрири. — Процессоры быстрее уже не станут, но число ядер продолжит расти. Поэтому проблему горизонтального масштабирования СУБД между множеством процессоров необходимо решать».

Маккрири также сходит в мнении со Стоунбрейкером о том, что пользователям NoSQL не помешал бы единый язык запросов, отсутствие которого замедляет внедрение NoSQL в целом. Однако, по предположению Маккрири, в качестве стандарта для новых СУБД можно было бы взять не SQL, а другие языки, например XQuery — язык запросов для XML-документов.

Постоянный URL статьи: <http://www.osp.ru/cw/2011/22/13010433/>

© «Открытые системы», 1992-2011. Все права защищены.