

# Кризис технологии СУБД и пути его преодоления

---

*Сергей Кузнецов*

**kuz@citforum.ru**

# План доклада

---

- **Кризис назрел?**
- **Что делать и с чего начать?**
- **Сомнения и размышления**
- **Благодарности**
- **Литература**

# Кризис назрел? (1)

---

- Львиная доля мирового рынка управления данными занята продуктами трех ведущих компаний-поставщиков технологии СУБД: Oracle, IBM и Microsoft
- Системы управления базами данных (СУБД), поставляемые этими компаниями, с каждым новым выпуском становятся все объемнее
- В них появляются все новые и новые возможности, и, похоже, что полный набор возможностей этих СУБД уже неизвестен не только их пользователям, но и рядовым системным разработчикам

# Кризис назрел? (2)

---

- При наращивании возможностей своих продуктов основные поставщики технологии СУБД руководствуются двумя главными соображениями.
- Во-первых, конечно, на них воздействуют новые требования рынка (существующие или предсказываемые аналитиками)
- Во-вторых, похоже, что больше всего поставщики опасаются переделок ядер своих СУБД
- Это очень сильно влияет как на выбор набора новых средств, так и на особенности их реализации

# Кризис назрел? (3)

---

- Тем не менее, благодаря своей массе, клиентской базе, авторитету на рынке и т.д., основные поставщики СУБД закрывают дорогу на массовый рынок новым продуктам, которые, возможно, объективно лучше соответствуют новым требованиям рынка
- Накапливаются все больше интересных исследовательских результатов и экспериментальных реализаций в области управления данными с очень малыми шансами на практическое применение

# Кризис назрел? (4)

---

- Все менее понятно, что такое язык SQL.
- В существующем стандарте SQL:2003 можно выделить модельный слой, действительно определяющий общие черты СУБД основных поставщиков
- Но в этом стандарте содержится множество спецификаций, выходящих за пределы этого модельного слоя (например, средства OLAP, XML и т.д.)
- Нет ни одного человека, который мог бы с уверенностью сказать, что он знает современный SQL полностью
- Нет и, видимо, никогда не будет ни одной системы, в который был бы полностью реализован стандарт языка SQL

# Кризис назрел? (5)

---

- Часто приходится слышать, что стандарт SQL – это некоторый ориентир
- Компании стремятся двигаться в нужном направлении, считая, что это направление указывается именно этим стандартом
- Но стандарт SQL сегодня настолько велик и многообразен, что в каком бы направлении не развивалась некоторая СУБД, почти всегда можно найти некоторую черту SQL, которую можно было бы считать путеводной звездой

# Кризис назрел? (6)

---

- В 1990-м г. в «Манифесте систем баз данных третьего поколения» говорилось, что SQL стал «межгалактическим» языком общения
- Похоже, что с того времени возможности использования SQL для общения резко уменьшились



# Что делать и с чего начать?

## (1)

---

- Ответ на этот вопрос сильно зависит от того, кому его задают
- Очевидная стратегия основных поставщиков СУБД состоит в том, чтобы не допустить революционной ситуации
- Они всячески пытаются сохранять и расширять свою клиентскую базу, добавляя в свои продукты все новые возможности, стараясь при этом не утратить надежность и производительность
- В некоторых случаях (как это было, например, в 1995 г. с DB2) они даже решаются на существенные изменения ядра своих СУБД, не отказываясь при этом от традиционной общей архитектуры системы

# Что делать и с чего начать?

## (2)

---

- Поставщики традиционных СУБД «второго эшелона» (в том числе, и тех систем, которые относятся к категории open source), фактически, пытаются найти свое место на рынке, предлагая облегченный набор средств с интересным для многих пользователей соотношением «цена/эффективность/надежность/доступность».
- Двигаясь по этому пути, ни одна из СУБД «второго эшелона» никогда не сможет стать серьезным конкурентом «большой тройки» СУБД в той части рынка, которая приносит наибольшую часть прибыли

# Что делать и с чего начать?

## (3)

---

- По-другому отвечает на этот вопрос Майкл Стоунбрейкер
- Мое понимание его позиции состоит в следующем
- Стоубрейкер в таком виде свою позицию никогда не излагал

# Что делать и с чего начать?

## (4)

- Архитектура современных SQL-ориентированных СУБД появилась более 30 лет тому назад, когда рынок систем управления данными был единым, не фрагментированным на специализированные секторы
- СУБД вынужденно делались «безразмерными», пригодными для использования в любой области приложений баз данных.
- Эта «безразмерность» присутствует сегодня в продуктах основных поставщиков (и в универсальных СУБД «второго эшелона»)
- Плюсами основных SQL-ориентированных СУБД является надежность и общая высокая производительность. Минусы – сложность, объемность и высокие накладные расходы, свойственные универсальности

# Что делать и с чего начать?

## (5)

---

- За прошедшие 30 с лишним лет рынок систем управления данными сильно фрагментировался
- Стали известными большие секторы рынка, для которых очень существенна высокая производительность приложений, которая не достигается или достигается с недопустимо большими затратами при использовании «безразмерных» СУБД
- Экономически целесообразной стала разработка специализированных систем, которые ориентируются на эффективную поддержку заранее известных сценариев использования

# Что делать и с чего начать?

## (6)

---

- За эти же тридцать лет в области управления данными была выполнена громадная исследовательская работа, результаты которой можно успешно применять для разработки специализированных систем

# Что делать и с чего начать?

## (7)

---

- В связи с быстро меняющимися требованиями рынка успешными могут быть только такие новые продукты, которые можно вывести на рынок достаточно быстро – через год или два после начала разработки
- Это еще один довод в пользу специализированных систем, нацеленных на решение узкого класса задач
- В таких систем требуются более простые языковые средства, упрощается оптимизация запросов и другие аспекты, являющиеся традиционным камнем преткновения «безразмерных» систем

# Что делать и с чего начать?

## (8)

---

- В последние десять лет Стоунбрейкер последовательно воплощает в жизнь эти идеи
- Пробным шаром явилась разработка специализированных средств управления потоковыми данными
- На основе исследований и разработок, выполненных в ряде университетов США, была создана компания и промышленная система StreamBase, которая была хорошо принята финансовыми компаниями с Уолл-Стрит
- На этом этапе Стоунбрейкер практически не конкурировал с «безразмерными» СУБД, для которых рыночный сектор потоковых данных, по-видимому, был слишком узок



# Что делать и с чего начать?

## (9)

- Следующая попытка Стоунбрейкера состояла в создании нового SQL-ориентированного средства поддержки хранилищ данных с хранением данных по столбцам
- Созданная компания и промышленная система Vertica основывается на предыдущих университетских исследованиях и разработках, которые, в свою очередь, опираются на многолетние работы других исследователей
- Стоунбрейкер уже начинает потенциально конкурировать как с «безразмерными» СУБД, так и с СУБД, изначально ориентированными на поддержку хранилищ данных
- Приводятся результаты тестовых испытаний, показывающие, что в некоторых сценариях использования приложение, основанное на использовании Vertica, демонстрирует производительность, на два порядка более высокую, чем при использовании «безразмерной» коммерческой СУБД

# Что делать и с чего начать?

## (10)

---

- Наконец, теперь Стоунбрейкер полностью выходит на тропу войны с «безразмерными» СУБД, покушаясь на их основной, традиционный сектор рынка – OLTP
- В исключительно интересном, пока еще университетском проекте H-Store приводятся результаты испытаний этой системы на эталонном тестовом наборе TPC-C, демонстрирующие превосходство над «безразмерной» коммерческой СУБД почти на два порядка

# Что делать и с чего начать?

## (11)

---

- Кроме того, в статье «Пригоден ли один размер для всех? Часть 2: результаты тестовых испытаний» приводится краткая характеристика и показатели производительности экспериментальной системы ASAP, ориентированной на поддержку математических баз данных
- Результаты тоже впечатляют, хотя опубликованных данных относительно общей организации и интерфейсов системы явно недостаточно, чтобы можно было хорошо понять принципы ее организации

# Что делать и с чего начать?

## (12)

---

- Все эти работы представляются очень интересными и перспективными
- Однако имеется ряд сомнений относительно того, что они, как это предсказывает Стоунбрейкер, приведут к новой революции в области баз данных

# Сомнения и размышления

## (1)

---

- Хотят ли новой революции пользователи и разработчики приложений?
- В Манифесте систем баз данных третьего поколения, одним из основных авторов которого был Стоунбрейкер, звучала мысль о том, что 20-му веку не нужна еще одна революция
- Тогда это говорилось в связи с попытками передела мира баз данных сообществом объектно-ориентированных баз данных
- Как видно, этот передел не состоялся, и объектно-ориентированные базы данных заняли периферийную позицию на рынке управления данными

# Сомнения и размышления

## (2)

---

- Теперь мы живем уже в 21-ом веке, в котором, к счастью, революций пока еще не было
- Но почему Стоунбрейкер считает, что народ с нетерпением их ждет?
- К настоящему времени накоплено так много технологических возможностей, как аппаратных, так и программных, что большинство пользователей и разработчиков предпочтет потратить больше средств и/или пожертвовать некоторой долей производительности, чем перейти к использованию радикально других программных средств
- Наверное, это будет и дальше тормозить прогресс, но зато сохранит общественное спокойствие

# Сомнения и размышления

## (3)

---

- Удачный опыт внедрения принципиально новых средств управления потоковыми данными не опровергает эти соображения
- В этом случае речь идет об области приложений, которую принципиально не удовлетворяло существовавшее положение дел
- Финансовые организации, анализирующие потоки данных, которые поступают с бирж, использовали (и, в основном, продолжают использовать) малоэффективные уникальные программные средства, не обеспечивающие должный темп анализа
- Для них переход к использованию производственных систем — это не революция, а нормальный эволюционный процесс

# Сомнения и размышления

## (4)

---

- Теперь поговорим о технической стороне проблемы
- Начнем с СУБД Vertica
- В описываемых в статье «Пригоден ли один размер для всех? Часть 2: результаты тестовых испытаний» результатах испытаний Vertica побеждает традиционную СУБД на агрегатных запросах, в которых используется малая доля столбцов очень «широкой» таблицы



# Сомнения и размышления

## (5)

---

- Заметим, что речь идет о приложениях категории OLAP, в которых принципиально участвуют непредвиденные («ad hoc») запросы аналитиков
- В частности, в этих приложениях могут запрашиваться не только агрегатные, но и «атомарные» данные
- Но при этом хорошо известно, что в СУБД с хранением данных по столбцам операция извлечения строк таблицы является очень дорогостоящей
- Так что нужно еще разобраться, до каких пределов Vertica будет побеждать СУБД с хранением данных по строкам
- Здесь все не так очевидно

# Сомнения и размышления

## (6)

---

- Что касается преимуществ H-Store в приложениях OLTP, то, конечно, очень впечатляют примеры классов транзакций, для которых свойства ACID обеспечиваются автоматически, без накладных расходов на управление транзакциями
- Очень радуют возможности достижения высокого уровня доступности за счет репликации данных
- Однако рассмотрим более внимательно, за счет чего на основе H-Store удалось добиться таких высоких показателей на тестовом наборе TPC-C

# Сомнения и размышления

## (7)

---

- Приведение классов транзакций ТРС-С к форме, в которой для сериализации не требуются средства управления транзакциями
- Неизвестно, сколько времени это заняло у авторов статьи, но задача эта явно не тривиальна, поскольку для ее решения потребовалось специальным образом разделять и реплицировать базу данных
- Конечно, авторы говорят, что в будущем у них должны появиться специальные инструменты, автоматизирующие этот процесс, но подходы к созданию таких инструментов очень туманны
- Для транзакций, не обладающих специальными свойствами, преимущества H-Store не оценивались

# Сомнения и размышления

## (8)

---

- Отсутствие сетевых взаимодействий между приложением и СУБД за счет использования для разработки приложения механизма хранимых процедур
- Более того, хранимая процедура в среде H-Store выполняется в том же адресном пространстве, что и СУБД
- Другими словами, фактически мы имеем дело с СУБД, встроенной в приложение
- И здесь возникают сразу два сомнения

# Сомнения и размышления

## (9)

---

- Во-первых, какой механизм разработки приложений, в конце концов, будет предложен пользователям?
- Пока в среде H-Store нет никакого механизма разработки приложений (используется программирование на языке C++ на том же уровне, что и программирование самой СУБД)
- Как отмечают авторы в конце статьи, они планируют использовать средства Ruby-on-Rails
- Но будут ли счастливы разработчики приложений, если их всех, вне зависимости от пристрастий, заставят использовать Ruby-on-Rails?

# Сомнения и размышления

## (10)

---

- Во-вторых, всех ли устроит технология встроенной СУБД?
- Марго Зельцер в своей статье про BerkeleyDB писала, что в среде ее системы разработчики приложений обладают не меньшей квалификацией, чем разработчики СУБД, и поэтому, мол, нечего защищать код СУБД от кода приложений
- Но разработчиков приложений OLTP гораздо больше, чем системных программистов вообще, а не только программистов СУБД
- Поиск ошибок в приложении станет гораздо более тяжелым делом, если любая такая ошибка сможет приводить к непредсказуемому поведению СУБД
- Будь я разработчиком приложений или заказчиком таких приложений, я бы, все-таки, выбрал защищенный режим

# Сомнения и размышления (11)

---

- Я затронул только часть своих сомнений
- По всей видимости, однозначного пути к революции в области технологии баз данных все-таки нет
- Нужны компромиссы

# Благодарности

---

- Хочу выразить благодарность Майклу Стоунбрейкеру и его коллегам за их неутомимость, за попытки расшевелить исследователей и разработчиков, за усилия по внедрению в практическое использование ранее полученных и новых исследовательских результатов



# Литература

---

- A Conversation with Michael Stonebraker and Margo Seltzer, ACM Queue, Volume 5, Number 4, May/June 2007. Перевод на русский язык <http://citcity.ru/16453/>
- Margo Seltzer. Beyond Relational Databases: There is more to data access than SQL, ACM Queue, Vol. 3, No. 3 - April 2005. Перевод на русский язык <http://www.citforum.ru/database/articles/seltzer/>
- Michael Stonebraker, Uğur Çetintemel. «One Size Fits All»: An Idea Whose Time Has Come and Gone. Proceedings of ICDE 2005. Перевод на русский язык [http://www.citforum.ru/database/articles/one\\_size\\_fits\\_all/](http://www.citforum.ru/database/articles/one_size_fits_all/)
- Michael Stonebraker, Chuck Bear, Uğur Çetintemel, Mitch Cherniack, Tingjian Ge, Nabil Hachem, Stavros Harizopoulos, John Lifter, Jennie Rogers, and Stan Zdonik. One Size Fits All? – Part 2: Benchmarking Results. Proceedings of the 3rd Biennial Conference on Innovative Data Systems Research (CIDR). Перевод на русский язык [http://www.citforum.ru/database/articles/one\\_size\\_fits\\_all\\_2/](http://www.citforum.ru/database/articles/one_size_fits_all_2/)
- Michael Stonebraker, Samuel Madden, Daniel J. Abadi, Stavros Harizopoulos, Nabil Hachem, Pat Helland. The End of an Architectural Era (It's Time for a Complete Rewrite). Proceedings of VLDB, 2007, Vienna, Austria. Пересказ на русском языке [http://www.citforum.ru/database/articles/end\\_of\\_arch\\_era/](http://www.citforum.ru/database/articles/end_of_arch_era/)
- Сергей Кузнецов. Универсальность и специализация: время разбивать камни?. [http://www.citforum.ru/database/articles/time\\_to\\_break\\_stones/](http://www.citforum.ru/database/articles/time_to_break_stones/)