

М.А. СМЯТКИН,
Р.А. СИМАКОВ

**Предпосылки создания
распределенной массивно-
реляционной СУБД**

УДК 004.65

Муромский институт
(филиал) ФГБОУ ВПО
«Владимирский
государственный
университет имени
Александра
Григорьевича и Николая
Григорьевича
Столетовых», г. Муром

Целью данной статьи является обоснование актуальности проектирования новой распределенной СУБД. Опираясь на приведенные проблемы современных СУБД, замедляющие прогресс в ряде областей науки, выдвигаются требования. На базе этих требований будет вестись дальнейшая научная работа по проектированию и разработке СУБД.

The purpose of the article is a motivation of designing of new distributed DBMS. There are two general problems of modern distributed DBMSs that are slowing progress in some sciences. Basing on these problems, fundamental requirements on DBMS are proposed. All further science work on designing DMBS will be based on these requirements.

Введение. В настоящее время сформировались как минимум две проблемы в области проектирования СУБД, потенциально способные замедлить научный прогресс:

Проблема 1. Возможность отставания поддерживаемых современными СУБД объемов данных от объемов, получаемых с производимых научных сенсоров. Данная проблема уже подлежала обсуждению на ряде конференций, например [1], и одной из основных целей ряда современных СУБД [2, 3] является её решение.

Проблема 2. Отсутствие гибкой площадки для проведения исследований в области распределенных СУБД. В последнее время в области проектирования СУБД многие исследователи сходятся во

мнении, что от традиционного подхода к решению любых задач в одной универсальной СУБД, необходимо перейти к противоположному подходу, предполагающему наличие множества СУБД, оптимизированных под конкретные задачи [4]. Отсюда большинство существующих решений являются строго специализированными и какие-либо общие научные исследования на их архитектурах затруднительны даже в случае открытых исходных кодов. Поэтому многие ученые выбирают за основу распределенных систем традиционные реляционные СУБД, такие как PostgreSQL и др. [3, 4].

Основной идеей проектирования СУБД является попытка решения обозначенных выше проблем.

Требования. С учетом удовлетворения описанных выше проблем СУБД можно определить как расширяемую распределенную СУБД для хранения больших объемов данных. Ниже представлены фундаментальные требования к СУБД, построенные с учетом поставленных задач:

1. Характер хранимых данных – множества и многомерные массивы кортежей. В соответствии с *проблемой 1* наиболее приоритетным направлением является сборка, анализ и извлечение знаний из больших объемов научных данных. Для этих задач наиболее естественным является выбор многомерной модели хранения, обоснованной проектировщиками SciDB [1, 5]. Однако ряд факторов ставят необходимость поддержки в СУБД реляционной модели:

1.1. Несмотря на то, что в теории любой набор данных, представляемый в реляционной модели, можно также представить и в многомерном виде, в ряде случаев возможны потери в производительности или в требуемом дисковом пространстве.

1.2. Существуют области применения, которые естественным образом соответствуют реляционной модели: различного рода журналы, метаданные базы данных и др.

1.3. Два обозначенных выше ограничения могли бы решаться применением двух отдельных СУБД для поддержки многомерной и реляционной моделей. Однако в данном случае возникает дополнительная нагрузка на обработку запросов, появляются проблемы с фиксацией изменений на обеих СУБД (т.е. возникают блокировки, которые в соответствии с идеями СУБД недопустимы). Кроме того,

для обеспечения достаточной расширяемости и эффективности реляционная СУБД должна либо соответствовать поставленным требованиям (за исключением модели хранения), что значительно ограничивает выбор, либо использоваться на каждом узле как хранилище соответствующего набора данных, что создает ряд дополнительных накладных расходов.

2. Прозрачное распределение данных и нагрузки по узлам. Данный пункт нацелен на удовлетворение *первой* из обозначенных проблем. У клиента не должна возникать необходимость (соответственно, не должна предоставляться и возможность) знать о физическом расположении ресурсов и влиять на него, например, с целью изменения производительности. Это накладывает на систему дополнительные сложности: обеспечение автоматической балансировки нагрузки и независимость выполнения пользовательских функций (операторов) от распределения данных. Определенной группе пользователей, например, администраторам могут предоставляться возможности мониторинга физического состояния системы.

3. Обеспечение линейного роста узлов, предоставляющих, как минимум, прежний уровень производительности при соответствующем увеличении объема хранимых и обрабатываемых данных. Стоит отметить, что соблюдение линейного роста относится к случаям, когда линейный рост возможен алгоритмически. Например, время выполнения перемножения матриц с ростом их размерности на одноузловой системе будет возрастать не линейно, подобным же образом оно должно увеличиваться и в распределенной системе (при пренебрежении распределенной обработкой).

4. Обеспечение расширяемости функционала. Данное правило необходимо для удовлетворения *первой проблемы* и заключается в предоставлении конечному пользователю возможности введения собственных функций, операторов, типов данных и методов доступа. В роли конечного пользователя выступает исследователь или разработчик СУБД, а клиент, применяющий систему в своих целях. Расширяемость системы на уровне исходных кодов также входит в список требований, способствуя решению *второй проблемы*, и косвенно реализуется при соблюдении требования 6.

5. Исключение из системы центральных элементов. Каждому узлу системы во время работы может назначаться любая роль: координатор выполнения запроса, реплика данных, исполнитель запроса и т.д. Кроме того, роли не являются взаимоисключающими. Например, узел, являющийся репликой, тем не менее, может быть координатором запроса, исполнителем или и тем, и другим одновременно. Так же, некоторые роли, которые могут стать узкими местами для пропускной способности или возможности отказа исключены из системы, а их задачи распределяются между всеми узлами, даже ценой дублирования информации, как, например, в случае с хранением метаданных.

6. Чёткое разделение системы на набор функционально целостных подсистем, поддерживающих интерфейсы, нацеленные на возможность замены той или иной подсистемы и разбитые на связные и минимально сцепленные модули, которые также могут подлежать замене. На данном этапе планируются следующие подсистемы:

- 6.1. Подсистема хранения.
- 6.2. Сетевая подсистема.
- 6.3. Подсистема обработки запросов.
- 6.4. Оптимизатор доступа к данным.
- 6.5. Подсистема репликации.
- 6.7. Подсистема балансировки нагрузки.

Заключение. В рамках статьи приводятся лишь фундаментальные требования, которые нельзя игнорировать при построении системы, основной задачей которой является решение обозначенных проблем. Коротко, поставленные требования можно записать как:

1. Поддержка многомерной и реляционной моделей данных.
2. Прозрачность системы.
3. Линейная масштабируемость.
4. Расширяемость функционала со стороны клиента.
5. Отсутствие центральных элементов.
6. Повышение ортогональности системы с целью увеличения внутренней расширяемости.

Первые пять требований носят проектный характер и будут максимально учитываться в дальнейшей научной работе. Требование 6 необходимо для решения *второй проблемы*, но имеет значе-

ние преимущественно на этапе разработки и, как правило, в научной работе упоминаться не будет.

Помимо этих требований, существуют более общие требования, которые будут соблюдаться в случаях, когда они не противоречат основным требованиям. К ним относятся требования к OLAP-системам, выдвинутые Коддом [6] и требования Дейта к распределенным СУБД [7].

Литература

1. *J. Becla, K-T Lim*. Report from the first workshop on extremely large databases. – Data Science Journal, Volume 7, 23 Feb. 2008. – Stanford Linear Accelerator Center, CA, USA, 2008, 13 p.

2. *M. Stonebreaker, P. Velikov, R. Simakov, A. Smirnov* [и др.]. Overview of SciDB: large scale array storage, processing and analysis. //Proceedings of the ACM SIGMOD International Conference on Management of Data, 06.06.2010-11.06.2010, Indianapolis, IN. P. 963-968.

3. *P. Bauman, A. Dehmel, P. Furtado* [и др.]. The multidimensional database system RasDaMan. // Proceedings of the ACM SIGMOD International Conference on Management of Data, New York, NY, USA, 1998. P. 575 – 577.

4. *С. Д. Кузнецов*. Транзакционные параллельные СУБД: новая волна. – Труды института системного программирования, т. 20, М., ИСП РАН, 2011, стр. 189 – 251.

5. *M. Stonebreaker, R. Simakov, P. Velikov* [и др.]. A demonstration of SciDB: a Science-Oriented DBMS. // VLDB'09: Proceedings of the 2009 VLDB Endowment. 2009.

6. *Codd E.F., Codd S.B., Salley C.T.* Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate. Codd & Date, 1993.

7. *К. Дж. Дейт*. Введение в системы баз данных, 8-у издание. Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 1328 стр.: ил. – Парал. тит. англ.

М. А. СМЯТКИН – ТЕЛ.: 8 (910) 171 82 76,

E-MAIL: SMYATKINMAXIM@GMAIL.COM

Р.А. СИМАКОВ

E-MAIL: ROMAN.SIMAKOV@GMAIL.COM