

**УДК 004.22**

**К.А. Лаврентьев,**

*ассистент кафедры информационных систем и технологий  
Хабаровской государственной академии экономики и права,*

*аспирант ВЦ ДВО РАН*

**Е.А. Титова,**

*студентка 3-го курса*

*факультета управления специальности «Прикладная информатика в экономике»  
Хабаровской государственной академии экономики и права*

### ПРОБЛЕМЫ ПРОЕКТИРОВАНИЯ АРХИТЕКТУРЫ РАСПРЕДЕЛЕННЫХ БАЗ ДАННЫХ

*The existing strategies of storing data in distributed databases are analyzed in this article. A list of criteria that must be met for distributed database is defined and arisen design problems of a distributed database when using each strategy are described.*

**Keywords:** *distributed database, replication, fragmentation, centralization, design.*

Развитие информационных технологий обуславливает новые требования и возможности организации баз данных. В то время когда почти каждое электронное устройство имеет выход в глобальную сеть Интернет и использует свою локальную базу данных, необходимо использовать какие-то отличные от клиент-серверных технологии для построения баз данных. Такими технологиями являются распределённые базы данных (РБД). Целью данной статьи является описание распределённых баз данных и анализ неотъемлемой части РБД – стратегий хранения данных, выявление их достоинств и недостатков.

Распределённая база данных – это территориально распределённая совокупность локальных баз данных, объединён-

ных согласованными принципами организации, комплектования и эксплуатации, а также каналами связи, и доступная для совместного использования. Такая база данных состоит из узлов, где каждый узел является полноценной СУБД, и узлы связаны между собой таким образом, что пользователю не заметна «распределённость» БД. Логически целая БД делится на узлы, каждый из которых хранится на одном компьютере, а все компьютеры соединены линиями связи. Каждый из этих узлов работает под управлением своей СУБД. При проектировании РБД следует придерживаться так называемого фундаментального принципа (или правило 0): для пользователя распределённая система должна выглядеть так же, как нераспределённая система. Другими словами,

пользователь должен работать с распределённой базой данных так, как будто все данные находятся на его компьютере, а не удалены от него географически.

Из фундаментального принципа проектирования распределённых баз данных вытекают критерии распределённости, которые вывел К. Дейт в своей работе «Введение в системы баз данных»[1]. Среди всех критериев, авторы выделили следующие:

1. Локальная автономность.

Локальные данные принадлежат локальным узлам и управляются администраторами локальных БД. Локальные процессы в РБД остаются локальными. Все процессы на локальном узле контролируются только этим узлом.

2. Отсутствие опоры на центральный узел.

В системе не должно быть узла, без которого система не может функционировать, то есть не должно быть центральных служб.

3. Независимость от местоположения.

Пользователь должен получать доступ к любым данным в системе независимо от того, являются эти данные локальными или удалёнными.

4. Независимость от фрагментации.

Доступ к данным не должен зависеть от наличия или отсутствия фрагментации и от типа фрагментации.

5. Независимость от типа оборудования.

СУРБД должна функционировать на оборудовании с различными вычислительными платформами. СУРБД должна быть способной функционировать в сетях с различной архитектурой и типами носителя.

6. Независимость от типа СУБД.

СУРБД должна быть способной функционировать поверх различных локальных СУБД, возможно, с различными моделями данных (требование гетерогенности). С точки зрения пользователя, любая информация система, в том числе и распределённая база данных, представляет собой «чёрный ящик». Пользователь вводит какие-то данные в интерфейсе системы или выбирает какую-то функцию и хочет в результате получить ответ. Чем быстрее пользователь получит ответ, тем лучше, по его мнению, работает система. Соответственно главной характеристикой РБД является малая величина времени отклика. Чтобы сделать величину времени отклика минимальной, необходимо при проектировании РБД выбрать стратегию хранения данных, соответствующую предметной области.

Существует три типа стратегий хранения данных: централизация, фрагментация, репликация. Каждая из них имеет свои преимущества и недостатки.

### Централизация

Централизованная БД имеет традиционную клиент-серверную архитектуру данных.

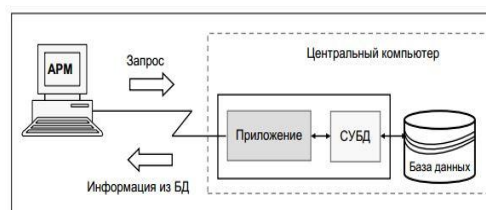


Рисунок 1 – Централизованная БД

При подобной архитектуре все необходимые данные и СУБД размещены на центральном компьютере (сервере).

В качестве примера рассмотрим централизованную БД на базе недавно созданной турфирмы, имеющей один офис. Предположим, что пользователь вводит запрос, требующий последовательного просмотра базы данных. СУБД получает запрос, просматривает БД, находит нужную запись, и возвращает её пользователю. При увеличении количества одновременных запросов к серверу пользователь начнет это «чувствовать» за счёт увеличения времени между вводом данных и выводом. Отсюда вытекает главный недостаток централизованных БД – увеличение времени отклика при большой нагрузке. Самым ярким примером централизованных РБД можно назвать сеть Интернет. Любой сайт имеет базу данных, которая хранится на сервере, и пользователи обращаются к этой единой базе данных по сети. Среди достоинств стратегии централизованного хранения данных можно выделить:

1. Простота поддержки, расширения и управления базой данных. За счёт того, что данные хранятся в одном месте и структуры хранения данных редко изменяются, администратор БД будет тратить немного усилий для поддержки и обслуживания базы данных.

2. Простота поддержки целостности базы данных. Так как данные хранятся в одном месте, то ситуации нарушения целостности из-за неправильного ввода или неверных транзакций очень маловероятны.

3. Малая стоимость функционирования системы. Для поддержания работы системы не нужны большие финансовые затраты за счёт того, что в системе есть только один сервер.

Кроме достоинств, у стратегии централизованного хранения данных есть и недостатки:

1. Увеличение времени отклика при возрастании нагрузки. За счёт того, что данные хранятся на одном сервере, при увеличении нагрузки на систему будет возрастать время отклика из-за последовательной обработки запросов пользователей.

2. Надёжность и безопасность хранения данных. При хранении данных на одном сервере администратору БД необходимо уделять повышенное внимание безопасности сервера и главное – надёжности хранения. Ведь если сервер выйдет из строя, то работа всей системы остановится.

Как было сказано выше, главная характеристика РБД – это минимальное время отклика на действия пользователя. При проектировании базы данных с использованием стратегии централизации возникает проблема минимизации времени отклика и надёжности хранения. Однако в случае использования этой стратегии можно получить простую в обслуживании базу данных.

Каждая ИС рано или поздно начинает расширяться, а значит, и увеличивается количество пользователей, которые могут быть распределены географически. В таком случае данные будут нужны в разных точках мира, а значит, появится потребность в децентрализованном подходе.

#### **Децентрализованный подход**

Децентрализованный подход, по своей сути, отражает организационную структуру многих компаний, логически или физически состоящих из отдельных подразделений, отделов, проектных групп и т.п., которые физически распределены по

разным точкам, причём каждая отдельная производственная единица имеет дело с собственным набором данных. Разработка распределённых баз данных, отражающих организационные структуры предприятий, позволяет сделать общедоступными данные, поддерживаемые каждым из существующих подразделений, обеспечив при этом их хранение именно в тех местах, где они чаще всего используются. Подобный подход расширяет возможности совместного использования информации, одновременно повышая эффективность доступа к ней.

Фрагментация – это разбиение базы данных на фрагменты и размещение их по разным узлам сети. Фрагментация является основным способом организации РБД. Она позволяет хранить данные на том узле, где они наиболее часто используются.

Например, пусть имеется компания с несколькими филиалами в разных городах страны. Основное подразделение и филиалы будут представлены фрагментами РБД, каждый фрагмент представляет собой самостоятельную базу данных. В каждом фрагменте будут храниться данные нужные только подразделению-владельцу этого фрагмента.

Фрагментация может быть вертикальная и горизонтальная.

E1
E2
E3

Рисунок 2 – Горизонтальная фрагментация

Горизонтальная фрагментация означает разбиение таблицы по строкам. Иными словами, суть горизонтальной фрагментации состоит в том, что делаются горизонтальные «срезы» в соответствии со значением какого-либо столбца таблицы. К примеру, в компании есть таблица с клиентами, в каждом узле РБД в этой таблице будут храниться данные, клиентах, которые обслуживаются в подразделении-владельце узла.

E1	E2	E3
----	----	----

Рисунок 3 – Вертикальная фрагментация

При вертикальной фрагментации разбиение таблицы осуществляется не по строкам, а по столбцам. В этом случае некоторая часть информации хранится в одном месте, а другая часть (относящаяся к той же таблице) – в другом. Допустим, данные о сотрудниках, такие как паспортные данные, прежнее место работы, будут храниться в центральном офисе (в отделе кадров), а номер пропуска (к примеру) непосредственно в подразделении, где работает сотрудник. Среди достоинств фрагментации можно выделить:

1. Соответствие структуры базы данных структуре предприятия. Это свойство позволяет проектировать более логичную структуру хранения данных.

2. Минимизация времени отклика. За счёт того, что данные узла хранятся в ло-

кальной базе данных, время отклика системы будет минимальным.

Однако есть и недостатки:

1. Сложность управления, интеграции и контроля целостности базы данных. За счет того, что данные распределены по нескольким узлам возможно дублирование данных на разных узлах. Так, разные узлы могут использовать разные СУБД, повышается сложность управления базой данных и уменьшается степень контроля целостности базы данных.

2. Сложность выполнения запросов, затрагивающих более одного узла. Иногда при работе с распределённой базой данных возникает задача выполнить запрос, в котором будут участвовать несколько узлов. В этом случае возникает сложность за счёт того, что узлы могут работать под управлением различных СУБД, и узлы распределены географически. Для решения этой проблемы предпринимаются очевидные действия: выбрать один из узлов главным для осуществления запроса. Однако в этом случае нарушаются критерии распределённости, которые были описаны выше.

При проектировании РБД с применением стратегии фрагментирования данных можно получить базу данных, которая будет логически соответствовать структуре организации и обеспечивать должное время отклика при работе внутри фрагмента. Однако возникает несколько важных проблем с оптимизацией за-

просов с несколькими узлами и соответственно увеличением времени отклика.

### Репликация данных

Репликация – это поддержание двух и более идентичных копий (реплик) данных на разных узлах РБД.



Рисунок 4 – Реплицированная БД

К основным достоинствам механизма репликации можно отнести повышение доступности и надёжности данных. Основным недостатком репликации является сложность поддержания идентичности реплик: если в одну копию данных вносятся изменения, то эти изменения также должны быть внесены в другие копии. Тут важно правильно выбрать подход к организации репликации.

#### 1. Репликация с основной копией:

– классический подход заключается в наличии одной основной копии, в которую можно вносить изменения; остальные копии создаются с определением *read only*;

– асимметричная репликация: основная копия фрагментирована и распределена по разным узлам РБД, и другие узлы могут являться подписчиками отдельных фрагментов (*read only*);

– рабочий поток. При использовании этого подхода право обновления не принадлежит постоянно одной копии, а пере-

ходит от одной копии к другой в соответствии с потоком операций. В каждый момент времени обновляться может только одна копия. Например, при последовательной обработке заказов на поставку товаров сначала в отделе приёма заказ принимается, затем на складе определяют, может ли он быть выполнен и сколько это будет стоить, затем эти данные передаются в финансовый отдел, а после оплаты – в отдел доставки. При репликации данных из узла  $S_i$  на узел  $S_{i+1}$  вместе с новыми данными передаётся право на обновление реплики.

2. Симметричная репликация (без основной копии). Все копии реплицируемого набора могут обновляться одновременно и независимо друг от друга, но все изменения одной копии должны попасть во все остальные копии.

При проектировании РБД важно правильно выбрать стратегию хранения данных, ведь от неё зависит время обработки данных как главный критерий оценки архитектуры РБД. При выборе нужно учитывать достоинства и недостатки каждой из них и подобрать подходящий вариант для автоматизируемой предметной области.

Рассмотрев три основные стратегии хранения данных в РБД, можно сказать, что централизация подходит для баз данных, которые не будут сильно масштабироваться. Для географически распределённых и масштабируемых баз данных необходимо применять фрагментацию либо репликацию. Однако у обеих этих стратегий есть существенные недостатки (увеличение времени отклика при меж-узловых запросах и сильное дублирование данных). Для того чтобы избежать

негативных последствий этих недостатков и при этом в полной мере задействовать достоинства фрагментации и репликации, предлагается использовать так называемую «смешанную» стратегию хранения данных. Такая стратегия позволяет хранить в узлах только нужные данные и при этом выборочно реплицировать таблицы.

#### Список использованных источников

1. Дейт К. Дж. Введение в системы баз данных. М. : Вильямс, 2006. 1316 с.
2. Заботина Н. Н. Проектирование информационных систем : учеб. пособие. М. : Инфра-М, 2013. 331 с.
3. Лаврентьев К. А., Смагин С. И., Пономарчук Ю. В. Проблема интеграции новых баз данных в распределённую информационную систему предприятия // Высокие технологии, исследования, финансы, промышленность : сб. статей Восемнадцатой международной научно-практической конференции «Фундаментальные и прикладные исследования, разработка и применение высоких технологий в промышленности и экономике». 4–5 декабря 2014 г. / под науч. ред. А. П. Кудинова, И. А. Кудинова. СПб. : Изд-во Политехн. ун-та, 2014. 200 с.
4. Основные понятия распределённых баз данных. URL: [gema44.ru/resurs/study/dbmat/ddb.doc](http://gema44.ru/resurs/study/dbmat/ddb.doc) (дата обращения: 08.02.2015).