

4

КОМПЬЮТЕРНЫЕ СИСТЕМЫ
И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 658.512.011.56

ВЫБОР И ОБОСНОВАНИЕ КРИТЕРИЯ ЭФФЕКТИВНОСТИ
ПРИ ПРОЕКТИРОВАНИИ РАСПРЕДЕЛЕННЫХ БАЗ ДАННЫХ

В.Б. Новосельский, Т.А. Павловская

В статье рассматривается жизненный цикл проектирования распределенных баз данных (РБД), описываются этапы проектирования. На основании взаимозависимости этапов фрагментации данных, размещения данных и выбора стратегии исполнения запросов формулируется задача проектирования РБД. Производится выбор и обоснование критерия эффективности при проектировании РБД. Авторы рассматривают составляющие критерия и исследуют зависимость критерия от различных параметров.

Ключевые слова: распределенные базы данных, фрагментация, размещение, распределенное исполнение запроса.

Введение

Создание распределенных информационных систем является весьма актуальной задачей. Это связано с возрастающими потребностями в приложениях, доступ к которым осуществляется из различных географических местоположений. Увеличиваются требования к оперативности и достоверности информации. Управление информацией происходит с помощью баз данных (БД). Для достижения высокой производительности распределенных приложений, работающих с базами данных, необходимы эффективные методы проектирования распределенных баз данных (РБД).

В статье рассматривается жизненный цикл проектирования РБД, описываются этапы проектирования. На основании взаимозависимости этапов фрагментации данных, размещения данных и выбора стратегии исполнения запросов формулируется задача проектирования РБД. Производится выбор и обоснование критерия эффективности проекта РБД. Авторы рассматривают составляющие критерия и исследуют зависимость критерия от различных параметров.

Жизненный цикл проектирования РБД

Жизненный цикл проектирования РБД состоит из двух фаз – начального проектирования и репроектирования. Большинство исследователей под начальным проектированием понимают фрагментацию БД и размещение фрагментов по узлам вычислительной сети (ВС). С течением времени возможно ухудшение производительности приложений, работающих с РБД, вызванное изменениями в инфраструктуре распределенной среды (ИРС).

Под инфраструктурой понимаются физические и логические параметры функционирования системы, а именно – транзакции и их частоты, топология ВС, характеристики узлов ВС и т.д. При возникновении таких изменений требуется репроектирование РБД для сохранения производительности приложений. Фаза репроектирования, помимо этапов, входящих в начальное проектирование, дополнительно содержит методы материализации обновленного дизайна. Жизненный цикл процесса проектирования изображен на рис. 1.

Далее кратко описаны основные этапы проектирования.

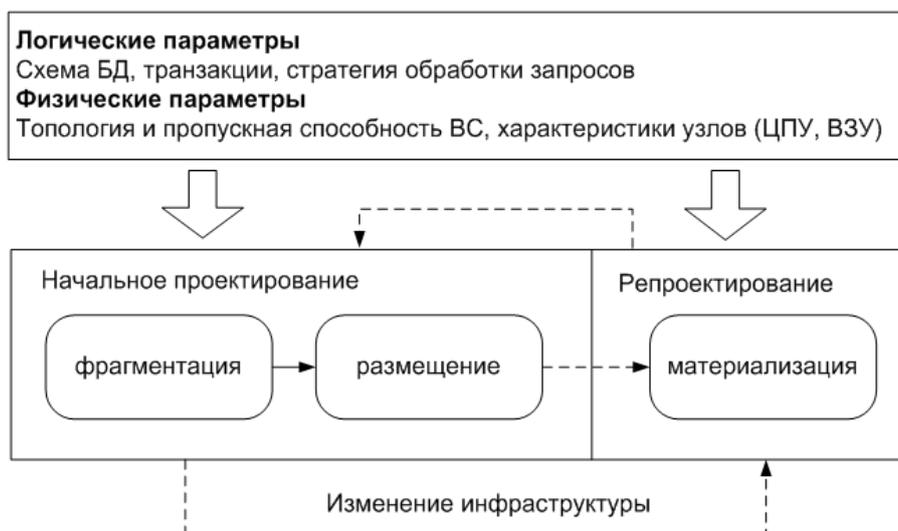


Рис. 1. Жизненный цикл проектирования РБД

Фрагментация данных

Основной целью фрагментации является сужение пространства поиска при исполнении запроса. Фрагментация данных допускает разбиение отношения на два или более сегмента или фрагмента. Каждый фрагмент может храниться на любом узле ВС. Принято выделять две базовых стратегии фрагментации данных – горизонтальную и вертикальную фрагментации. Вертикальная фрагментация – это разделение атрибутов на группы, горизонтальная фрагментация – разделение отношения на подмножества таким образом, что каждое подмножество содержит полный набор атрибутов. Примеры горизонтальной и вертикальной фрагментации показаны на рис. 2, 3 соответственно.



Рис. 2. Горизонтальная фрагментация

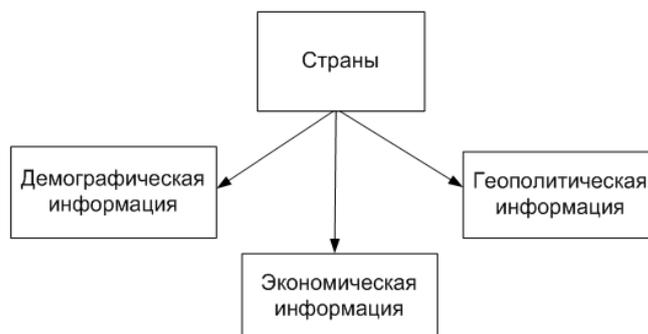


Рис. 3. Вертикальная фрагментация

Горизонтальная фрагментация разделяется на первичную (primary) и вторичную (derived) фрагментацию. Целью первичной горизонтальной фрагментации является оптимизация операций над множествами, позволяющая, во-первых, сузить пространство поиска и, во-вторых, обеспечить возможность параллельного выполнения операций. Целью же вторичной горизонтальной фрагментации является повышение скорости навигационных операций, достигаемое за счет объединения множеств объектов различных типов.

Размещение данных

Размещение данных представляет собой процесс принятия решения о месте хранения данных с целью минимизации целевой функции при выполнении запросов. Выделяют следующие типы стратегии размещения данных:

- централизованное размещение данных – вся база данных хранится на одном узле;
- секционированное размещение данных – каждый фрагмент БД хранится на определенном узле;
- реплицированное размещение данных – одна или более копий фрагментов БД хранятся на нескольких узлах.

Репликация данных связана с хранением копий данных на нескольких узлах сети. Поскольку копии фрагментов повышают уровень доступности данных и уменьшают время отклика, репликация уменьшает общие затраты на коммуникации при выполнении запросов.

Критерий эффективности РБД

Проектирование схем фрагментации и размещения отношений основывается на информации о способах и методах использования РБД. Методы использования зависят от стратегии исполнения запросов, которая, в свою очередь, должна учитывать схемы фрагментации и размещения. Таким образом, проектирование фрагментации, размещения и стратегии исполнения запросов должно производиться одновременно. Следовательно, задачу проектирования РБД следует формулировать так: для данной логической схемы БД, множества запросов и конфигурации ВС описать схему фрагментации, схему размещения фрагментов и стратегии исполнения каждого запроса таким образом, чтобы оптимизировать целевую функцию.

Целевой функцией этапа фрагментации данных является уменьшение времени исполнения запроса. В работах, посвященных размещению данных, в качестве критерия эффективности выбирается минимизация общего количества передаваемых по сети данных [1] или минимизация времени ответа на запрос [2]. Целью репликации является повышение надежности и коэффициента готовности транзакций [3, 4]. Коэффициент готовности транзакций отражает вероятность того, что транзакция успешно завершится за ожидаемое время. При выборе стратегии исполнения запроса критерием эффективности выступает время исполнения запроса и коэффициент готовности транзакций [5]

Производительность системы определяется отношением объема работы ко времени, за которое она была совершена. Следовательно, критерием эффективности РБД должно быть время ответа на запрос. Однако время ответа и коэффициент готовности транзакций взаимосвязаны: с уменьшением времени ответа на запрос повышается загрузка ресурсов ВС, что приводит к уменьшению коэффициента готовности транзакций. Для учета этой зависимости формализуем критерий эффективности следующим образом:

$$\xi = \sum_{i=1}^k \sum_{j=0}^n (c_t \cdot Q_{ij} + c_a \cdot W_{ij}) \cdot f_{ij} \rightarrow \min,$$

где k – количество узлов ВС; n – количество запросов; f_{ij} – частота возникновения j -го запроса в i -ом узле; Q_{ij} – временной коэффициент j -ого запроса, порожденного в i -м узле; W_{ij} – коэффициент использования ресурсов при обработке j -го запроса, порожденного в i -м узле; c_t и c_a – коэффициенты важности времени ответа и готовности транзакции, которые лежат в пределах $[0,1]$. Значения коэффициентов определяются на основании требований к РБД, например, для РБД реального времени $c_t=1$, $c_a=0$.

Временной коэффициент j -го запроса определяется как $Q_{ij}=T_{ij}^d/T_{ij}^l$, т.е. равен отношению времени ответа на запрос, исполненный в РБД, к расчетному времени ответа на запрос, исполненный локально в узле i , при условии наличия в узле всех необходимых фрагментов. Коэффициент использования ресурсов равен сумме коэффициентов использования центрального процессора и внешнего запоминающего устройства узлов, вовлеченных в операцию, и коэффициента использования сетевых ресурсов.

Такой критерий обеспечивает оптимальность решения, но не обеспечивает допустимость, так как минимизируется общее время ответа системы в целом. Поэтому необходимо ввести ограничение на максимально допустимое время ответа на запрос в каждом узле, $Q_j < \overline{Q}_j$, где $j = 0 \dots n$, \overline{Q}_j – максимально допустимое время ответа на запрос j .

Время ответа на запрос

Время ответа в РБД имеет две составляющие: время локальной обработки и время ответа сети. Время локальной обработки складывается из задержек в центральном процессоре и внешнем запоминающем устройстве, конкурентного доступа к сетевой среде и времени выборки и обработки данных.

Время ответа сети формируется из времени передачи данных, задержек в очередях и промежуточных узлах сети и латентности, т.е. задержки на передачу сигнала [2]. Время передачи данных зависит от пропускной способности сети. Задержки в очередях и промежуточных узлах вызваны обработкой, которая происходит в узлах, находящихся между передающим и принимающим узлами. Латентность напрямую зависит от расстояния между узлом-приемником и передатчиком. Как отмечено в [2], эта составляющая особенно важна в высокоскоростных сетях передачи данных (СПД).

В РБД существует возможность параллельного доступа к ресурсам, поэтому для оценки времени ответа необходимо рассмотреть процесс исполнения запроса в распределенной среде.

Распределенное исполнение запроса

Наиболее часто план исполнения запроса представляется в виде дерева операторов, в котором листья представляют отношения, которые участвуют в запросе, а промежуточные вершины – операторы. Параллельная обработка при таком описании разделяется на нескольких видов:

- межоператорная (inter-operation) – параллельное исполнение операторов, лежащих на разных ветвях дерева;
- внутриоператорная (intra-operation) – параллельное исполнение на различных фрагментах одного отношения суб-операторов, полученных в результате декомпозиции оператора;
- конвейерная (pipeline) – узел дерева «потребитель» может начать выполнение до того, как «производитель» завершит свою работу.

Для оценки взаимовлияния стратегии исполнения запроса и схем фрагментации и размещения оценим время исполнения одного оператора o при локальном и распределенном исполнении. Процесс распределенного исполнения оператора на трех узлах показан на рис. 4.

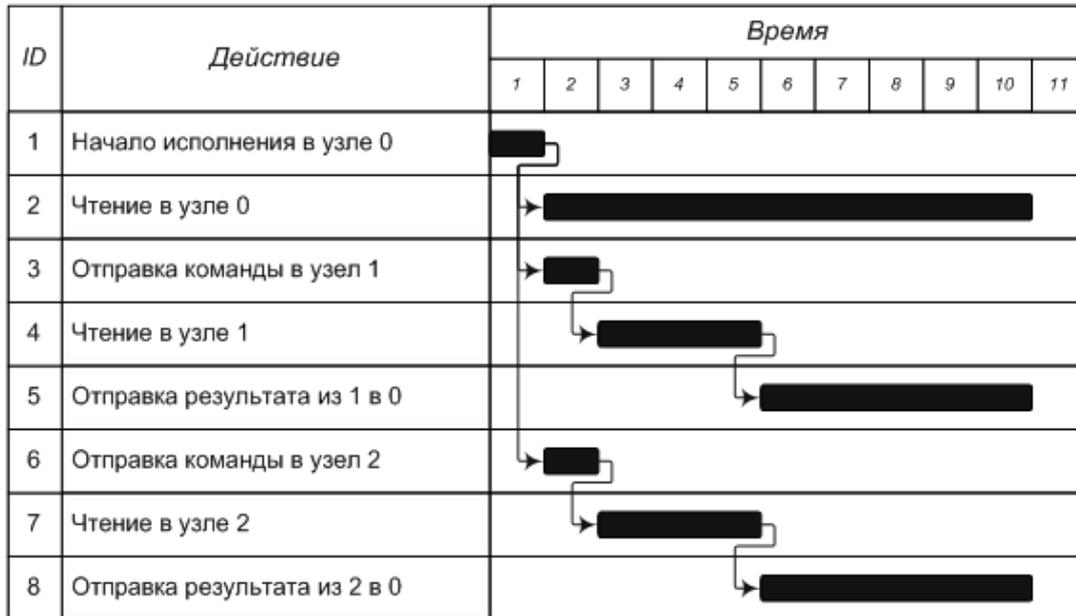


Рис. 4. Диаграмма Ганта распределенного исполнения оператора

Для простоты положим, что все узлы равноудалены от локального, их скорость чтения одинакова и равна скорости чтения в локальном узле, на всех узлах достаточно ресурсов для параллельного доступа к сети.

Положим Q – общее количество читаемых данных, S – скорость чтения единицы данных в k -м узле. Если все данные помещены в узел k , время выборки $T_{loc} = \frac{Q}{S}$. Разместим часть записей f ($f=[0..1]$) из выборки Q в n различных узлах, причем, так как все узлы равнозначны, в каждом узле будет размещена равная доля. На основании диаграммы процесса исполнения запроса запишем следующее равенство:

$$(1 - nf) \frac{Q}{S} = P + f \frac{Q}{S} + f \frac{Q}{R},$$

где R – скорость передачи единицы данных по сети; P – время передачи запроса от i -го узла до узла k . Обозначим разницу скоростей локального и сетевого чтения $R = \Delta S$ и выразим общее количество данных, считываемых с удаленных узлов. Оно прямо пропорционально сокращению времени ответа: $T' = n \left(1 - \frac{PS}{Q} \right) / \left(n + 1 + \frac{1}{\Delta} \right)$. Как видно

из уравнения, ускорение будет наблюдаться только в том случае когда $\frac{PS}{Q} < 1$. Так

как основным компонентом времени передачи небольшого сообщения, особенно в высокоскоростных сетях (почти 90% в сети T1 при расстоянии 900 км, согласно [2]), является задержка на распространение сигнала, то она может быть оценена из расстояния между узлами. Например, при $P = 5$ мс, $S = 50$ Мб/с данное неравенство будет выполняться при $S \geq 250$ Кб.

График зависимости сокращения времени исполнения запроса от соотношения скоростей локального и сетевого чтения при чтении 1Мб данных ($\frac{PS}{Q} = 0,25$) приведен на рис. 5.

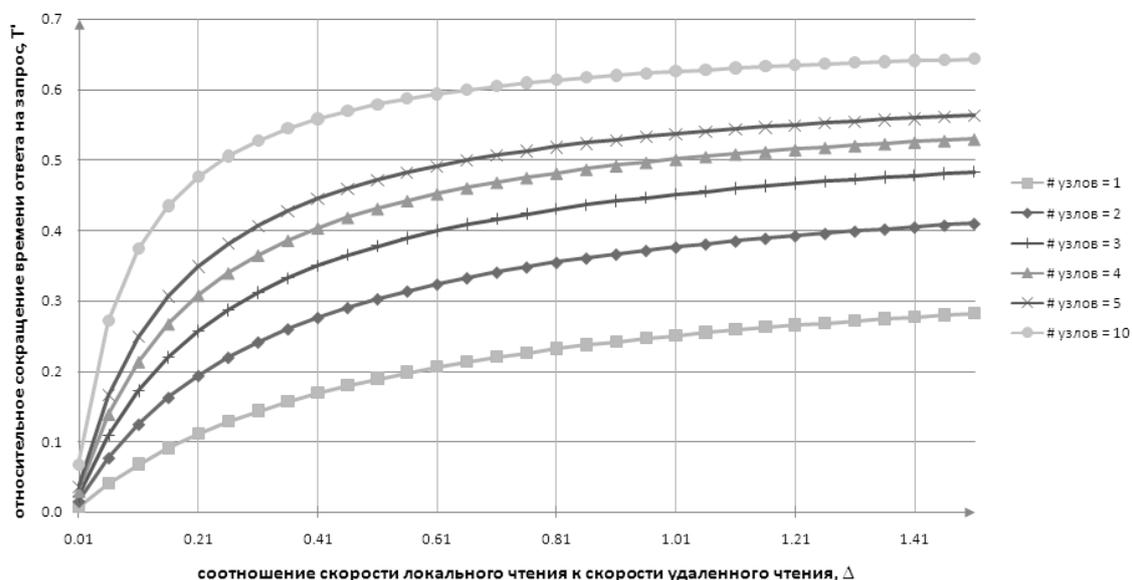


Рис. 5. График сокращения времени исполнения запроса

Очевидно, что $T' \xrightarrow{n \rightarrow \infty} 1 - \frac{PS}{Q}$, т.е. максимально возможное сокращение времени ответа определяется на основании физических характеристик сети и количества передаваемых данных.

Необходимо отметить, что чтение и обновление данных имеют различные временные характеристики, особенно в распределенной среде.

Распределенное исполнение запроса на обновление

Во многих работах (например, [1, 6]) не делается различия между запросами на выборку и запросами на обновление. Однако исполнение запроса на обновление в распределенной среде существенно отличается от выборки данных. При обновлении распределенных фрагментов СУРБД в основном используют двухфазный протокол подтверждения транзакции:

1. всем узлам отсылается сообщение «подготовка к подтверждению транзакции»;
2. от всех узлов получается ответ «готов к подтверждению транзакции»;
3. всем узлам отсылается сообщение «подтвердить транзакцию»;
4. от всех узлов получается ответ «транзакция подтверждена».

Вопрос параллельного исполнения обновления рассматривается в работе [7], в которой получена следующая аналитическая формула времени обновления:

$$T_u = 3 * \left[\text{MAX}_{j=1}^n \left[\left(\sum_{i=1}^j S_{t,i} \right) + R_{j,t} \right] \right] + \sum_{i=1}^n S_{t,i},$$

где $S_{t,i}$ – время передачи запроса от узла-источника транзакции t до i -го узла; $R_{j,t}$ – время передачи ответа от узла j до узла-источника транзакции t ; n – число узлов, на которых размещены фрагменты. $S_{t,i}$ вклю-

чает время на передачу и задержки в сети, $R_{j,i}$ состоит из задержки от t до j , времени обработки на узле i , времени передачи ответа от узла i .

Для простоты можно положить, что время передачи запроса до всех узлов одинаково и сообщения отсылаются и обрабатываются узлами параллельно, тогда время обновления $T_u \approx 4S_{i,i} + 3R_{i,i}$

Как видно из формул, время обновления существенно превышает время выборки, т.е. при уменьшении времени выборки за счет размещения фрагментов на разных узлах увеличивается время обновления и возрастает загрузка узлов и ВС. Таким образом, при фрагментации и размещении необходимо учесть затраты на обновление данных при увеличении количества реплик в соответствии с коэффициентами важности времени ответа и готовности транзакции.

Заключение

В статье рассмотрен жизненный цикл проектирования РБД и описаны его этапы. Сформулирована задача проектирования РБД. Выбран и обоснован критерий эффективности проекта РБД. Рассмотрены составляющие времени распределенного исполнения оператора, исследована их зависимость от различных параметров сети и узлов ВС.

Литература

1. Apers P.M.G. Data allocation in distributed database systems // ACM Transactions on Database Systems. – 1988. – Vol. 13. – № 3. – P. 263–304.
2. Johansson J.M., March S.T., Naumann J.D. Modeling Network Latency and Parallel Processing in Distributed Database Design // Decision Sciences Journal. – 2003. – Vol. 34. – № 4. – P. 677–706.
3. Ma H., Schewe K.-D., Wang Q. A heuristic approach to cost-efficient derived horizontal fragmentation of complex value databases // Eighteenth conference on Australasian database. Ballarat, Australia. – 2007. – P. 103–111.
4. Mukkamala R., Bruell S.C., Shultz R.K. Design of partially replicated distributed database systems: an integrated methodology // ACM SIGMETRICS conference on Measurement and modeling of computer systems. Santa Fe, New Mexico, United States. – 1988. – P. 187–196.
5. Kossmann D. The state of the art in distributed query processing // ACM Computing Surveys. – 2000. – Vol. 32. – № 4. – P. 422–469.
6. Ma H., Schewe K.-D., Wang Q. A heuristic approach to cost-efficient fragmentation and allocation of complex value databases // 17th Australasian Database Conference. Hobart, Australia. – 2006. – P. 183–192.
7. Johansson J.M., March S.T., Naumann J.D. The effects of parallel processing on update response time in distributed database design // Twenty first international conference on Information systems. Brisbane, Queensland, Australia. – 2000. – P. 187–196.

Новосельский Вениамин Борисович – ООО «ТиетоЭнатор», руководитель технической группы, veniaminn@gmail.com
Павловская Татьяна Александровна – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, профессор, mux@tp2055.spb.edu