

ДИНАМИЧЕСКАЯ ОПТИМИЗАЦИЯ РАСПРЕДЕЛЕНИЯ ДАННЫХ ПО УЗЛАМ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ

Баранова С.С., студ. гр. ИУС-06м

Руководитель Телятников А.О.

Распределенные базы данных (РБД) получили широкое применение в условиях оптимизации организации и обработки больших объемов данных. На сегодняшний день технологии систем управления распределенными базами данных достигли того уровня развития, когда на рынке уже имеются достаточно развитые и надежные коммерческие системы. Но в то же время сегодня существует проблема повышения эффективности функционирования распределенных баз данных. Ключевым фактором, снижающим показатели функционирования РБД, является то, что коммуникационные сети, охватывающие большую территорию, пока остаются медленными. Поэтому основная задача распределенных систем состоит в том, чтобы минимизировать использование сетей, то есть минимизировать объем передаваемых данных или время их передачи.

РБД представляет собой сложную систему, в которой выполняется множество запросов к распределенным данным, производятся обновления множества копий, размещенных на разных узлах компьютерной сети. При этом производительность РБД зависит не только от параметров технических средств (серверов, каналов связи), но и от того, насколько рационально распределены данные в системе. Поэтому приближение данных к местам их наиболее интенсивного использования может увеличить производительность РБД, уменьшить время выполнения запросов и распространения обновлений за счет уменьшения обращений к удаленным данным.

Как уже отмечалось, РБД представляют собой сложную динамическую систему. Поэтому ввиду сложности РБД как объекта описания и расчета, для более точной оценки эффективности функционирования в тех или иных

условиях используются математические модели. При построении математической модели РБД важно учитывать особенности работы (реализации распределенных запросов и выполнения обновлений) в конкретной СУБД. Это позволит получить схему распределения, более точно соответствующую реальной среде.

В качестве такой среды выберем СУБД Oracle. Она является одной из наиболее распространенных СУБД, имеет различные механизмы реализации распределения данных и их обработки (выполнения распределенных запросов и распространения обновлений)

В СУРБД Oracle для выбора способа выполнения предложений SQL используется оптимизатор запросов. При обработке распределенных запросов оптимизатор учитывает, что данные расположены на удаленных БД [3]:

- Если все таблицы, к которым обращается запрос, расположены на одной и той же удаленной базе данных, то ORACLE посылает запрос в эту удаленную базу данных. Удаленный узел ORACLE выполняет запрос, посылая обратно в локальную базу данных лишь его результаты.

- Если запрос обращается к таблицам, расположенным в различных базах данных, то ORACLE осуществляет декомпозицию этого запроса на индивидуальные фрагменты, каждый из которых обращается к таблицам единственной базы данных. Затем ORACLE посылает каждый фрагмент соответствующей базе данных. Удаленный узел ORACLE для каждой из этих баз данных исполняет каждый фрагмент и возвращает результаты в локальную базу данных, где локальный узел ORACLE может выполнить любую дополнительную обработку, которую требует запрос.

Репликация данных в Oracle реализована в нескольких вариантах [4]:

1. Механизм моментальных снимков (read-only snapshots). Он подразумевает создание удаленной копии таблицы (или ее подмножества), которая доступна только на чтение и обновляется по заданному сценарию и расписанию (асинхронно).

2. Изменяемые снимки, представляющие возможность модификации удаленных копий. В этом случае обновление данных также происходит асинхронно.

3. Репликация с множественными хозяевами (multi-master site replication). При данном варианте полностью тиражируются целые наборы объектов БД (в них помимо таблиц могут входить индексы, представления, триггеры, пакеты хранимых процедур). При этом тиражируются все определения и атрибуты объектов, так что в результате все хозяева их копий становятся равноправными. В этом случае обновления данных могут происходить синхронно и асинхронно:

– в случае асинхронного обновления изменения, внесенные в таблицу, хранятся в так называемой очереди распределенных транзакций на узле, где эти изменения произошли. Далее, через некоторый промежуток времени, эта последовательность транзакций распространяется на другие узлы асинхронной репликации;

– синхронный режим передачи основан на мгновенной передаче измененных данных сразу всем реплицируемым узлам;

4. Процедурная репликация. При ее использовании реплицируются хранимые процедуры, вызывающие изменение данных на других узлах. Процедурная репликация обновляет только хранимые процедуры, которые приложения вызывают для изменения таблиц. Использование этого вида репликации позволяет избежать загрузки сети при передаче большого объема данных, и обычно запускается в то время, когда в системе работает как можно меньше пользователей.

Учитывая особенности реализации распределенных запросов и распространения обновлений, выберем модель оптимизации распределения данных по узлам вычислительной сети. Модель была переложена И.Ахмадом в работе [2].

Модель предполагает, что существует взаимная зависимость между схемой распределения данных (которая показывает расположение каждого

фрагмента на различных узлах базы данных) и стратегией оптимизации запросов (которая решает, как запрос может быть оптимально выполнен при данной схеме размещения). Для отображения зависимости фрагментов базы данных и времени выполнения запросов вводится понятие графа зависимости фрагментов.

Граф зависимости фрагментов, расположенный на рис. 1, представляет собой направленный ациклический граф, вершиной которого является узел генерации запроса (Узел 1) а все другие узлы как узлы фрагмента (Узел 2, например) на потенциальных узлах, доступных запросу. Направленное ребро между двумя узлами показывает время передачи данных. Время передачи данных (например, t_2) зависит от обрабатываемого запроса.

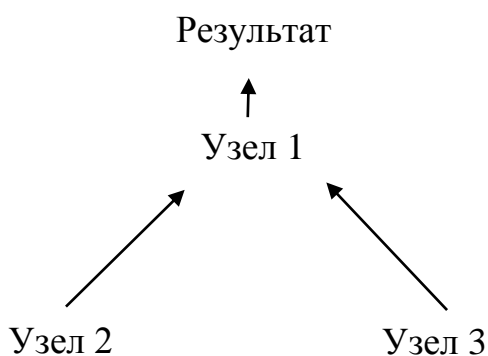


Рисунок 1. – Граф зависимости фрагментов

Кроме графа, рассматриваемый алгоритм имеет следующие входные параметры:

- частота выполнения запросов и обновлений от узлов;
- схема распределения фрагментов по узлам;
- ограничение на размещение количества фрагментов на узлах.

Целью алгоритма распределения данных является расположить узлы, на которых расположены фрагменты, чтобы минимизировать общее время на передачу данных при ограничениях (например, максимальное число фрагментов, которые могут размещены на сайте).

Пусть имеется множество узлов $S \{i, 1..m\}$, на которых расположено множество фрагментов $O \{j, 1..k\}$. На этих узлах иницируется множество

запросов $Q \{x, 1..n\}$. Каждый запрос q_x инициируется на узле S_i a_{ix} раз за определенный временной интервал – получим матрицу A размером $(m \times n)$.

Пусть t_{xj} будет определено как время передачи фрагмента данных O_j которые нужно передать на узел, где запрос q_x инициирован, получим матрицу T размером $(n \times k)$. Пусть U – матрица размером $(m \times k)$, где u_{ij} – время передачи данных с узла, где расположен фрагмент данных O_j на узел S_i где запрос был инициирован. Тогда

$$u_{ij} = \sum_{x=0}^{n-1} a_{ix} \cdot t_{xj}$$

Аналогично пусть t'_{xj} - время распространения обновления на узел O_j с узла, где обновление q'_x инициировано. Тогда получим матрицу U' , где u'_{ij} – время выполнения обновления с узла O_j на узел S_i :

$$u'_{ij} = \sum_{x=0}^{n-1} a'_{ix} \cdot t'_{xj}$$

С помощью этого выражения можно учитывать время выполнения синхронных и асинхронных обновлений.

Тогда суммарное время выполнения запросов и распространения обновлений равно:

$$U = \sum_{i=1}^m \sum_{j=1}^k u_{ij} + \sum_{i=1}^m \sum_{j=1}^k u'_{ij}$$

Литература

1. Телятников А.О. Разработка объектной модели распределенной базы данных// Наукові праці ДонНТУ. Випуск 74. – Донецьк, ДонНТУ, 2004. – с. 192 – 200.
2. I. Ahmad, K. Karlapaem. Evolutionary Algorithms for Allocating Data in Distributed Database Systems. // Distributed and Parallel Databases, № 11, p. 5–32, 2002.
3. Oracle 9i Database Administrator's Guide.
<http://www.mpi.ysn.ru/docs/oracle/server.920/a96521./toc.htm>
4. Oracle8i Replication. http://download-uk.oracle.com/docs/cd/A87860_01/doc/server.817/a76959/repover.htm