электронное научно-техническое издание
НАУКА и ОБРАЗОВАНИЕ
Эл № ФС 77 - 30569. Государственная регистрация №0421100025. ISSN 1994-0408

Преобразование Лапласа-Стилтьеса времени обработки запросов к хранилищу данных в параллельной системе баз данных

77-30569/270159

10, октябрь 2011

Плужников В. Л.

УДК 004.657

МГТУ им. Н.Э. Баумана
VPluzhnikov@croc.ru

Введение

В настоящее время возникает проблема, связанная с прогнозированием индексов производительности хранилищ данных, построенных на основе параллельных систем баз данных (ПСБД), т.к. соответствующие аппаратно-программные комплексы очень дороги и ошибки в их выборе приводят к большим финансовым потерям.

В предыдущей статье [4] рассмотрен математический метод анализа времени соединения таблиц в ПСБД. В данной работе демонстрируется применение математического аппарата производящих функций и преобразования Лапласа-Стилтьеса для оценки времени выполнения запросов к хранилищу данных в параллельной системе баз данных [11]. Этот метод учитывает особенности выполнения запросов к БД со схемой "звезда" в ПСБД, а также топологию (структуру) различных архитектурных решений.

Процесс выполнения запроса к хранилищу данных в параллельной системе баз данных

В современных параллельных системах баз данных [6] используется гибрид физической модели 3NF и многомерных представлений, поскольку системы очень хорошо обрабатывают соединения и обладают зрелым и надежным оптимизатором. Представления в ПСБД не материализуются до выполнения соответствующих запросов, что делает возможным отражать в плане запроса с соединением потребности каждого индивидуального запроса, минимизировать объем данных, которые требуется перераспределять или дублировать. Хэширование на первичном индексе упрощает разделение, и распределение памяти происходит только на уровне базы данных, а не на основе принципа таблица-раздел и индекс-раздел. В оптимизаторах ПСБД используется пересечение индексов, что обеспечивает эффективность почти такого же уровня,

который дают битовые индексы, без потребности в накладных расходах для их поддержки. Это означает, что ПСБД нуждается в создании меньшего числа индексов, что приводит к дальнейшему сокращению требуемых объемов памяти и времени для поддержки индексов. Наличие возможности синхронного сканирования (sync scan) позволяет использовать время, требуемое для сканирования больших таблиц, для выполнения нескольких (и даже сотен и тысяч) параллельно выполняемых запросов, делающих одну и ту же работу, что повышает пропускную способность системы на несколько порядков.

Когда в 1988 г. фирма Teradata [5] впервые начала обслуживать сверхбольшие базы данных, обнаружились некоторые странные проблемы. Даже при использовании подхода с отказом от общих ресурсов (shared-nothing) для обработки и сканирования больших таблиц требуется большое время. В результате инженеры разработали методы, заставляющие оптимизатор сначала обрабатывать малые таблицы измерений, соединяя их с целью создания первичного индекса для большой таблицы фактов и сокращая тем самым время ответа для многих запросов. Эти методы были включены в оптимизаторы ПСБД.

Как показывает рис. 1, у большой таблицы (Sales) имеется составной первичный индекс, составленный из внешних ключей таблиц измерений Stores, Items и Weeks. В звёзднообразной схеме все четыре таблицы логически (а иногда и физически) объединяются. Запрос (SELECT), представленный на рис. 2, позволяет произвести выборку из Stores, Items и Weeks с пересечением с указанными данными из Sales. В частности, здесь необходимо узнать общий объем продаж всех телевизоров в некоторой группе штатов (скажем Colorado и Minnesota) в течение двух недель перед Super Bowl, и результаты запроса необходимо вывести в соответствии с размером экрана телевизора и в лексикографическом порядке названий магазинов. Указанное представление (View) эмулирует звезнообразную схему, в запросе не принимается во внимание базовая физическая модель, и запрос прост (особенности модели скрываются представлением).

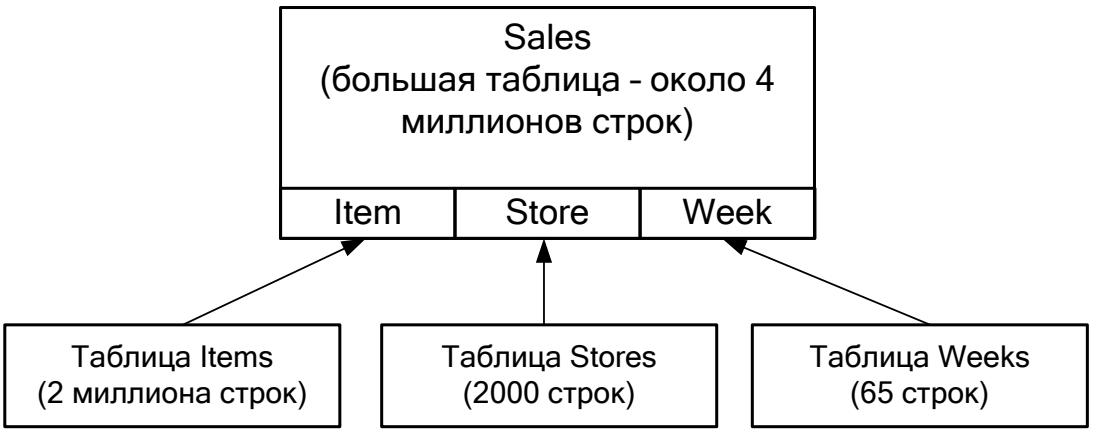

Рис. 1. Таблица Sales со звезднообразной схемой и составным первичным индексом.

```
SELECT store, SUM(sales$), SUM(salesQty), Substr(itemdesc,1,3)
screensize
FROM SalesStar
WHERE weeknbr BETWEEN 9805 AND 9806 AND state IN ('CO', 'MN')
AND subdept = 'Television'
ORDER BY screensize Desc, store
GROUP BY screensize, store;

View:
CREATE VIEW SalesStar AS
SELECT (*)
FROM SALES B, STORES S, ITEMS I, WEEKS W
WHERE B.STORE_NBR = S.STORE_NBR AND B.ITEM_NBR = I.ITEM_NBR
AND B.WEEK = W.WEEK;
```

Рис. 2. Запрос к таблице Sales..

Таблицы измерений и таблица фактов фрагментированы по AMP-процессам (AMP-процессорам) параллельной системы баз данных по значениям своих ключей. Таблица фактов фрагментирована по своему составному ключу, составленному из внешних ключей таблиц измерений. План выполнения запроса к хранилищу данных включает следующие шаги (рис. 3):

1. Каждый AMP-процесс (Access Module Processes) производит выборку из таблицы Weeks по условию <Week selection criteria>. Промежуточный результат (Spool) дублируется на всех AMP (см. И1 на рис. 3).
2. Все AMP производят выборку из таблицы Stores по условию <Store selection criteria>. Выполняется соединение со Spool по фиктивному условию (строится декартово произведение). Spool дублируется на всех AMP (см. И1×И2).
3. Все AMP производят выборку из таблицы Items по условию <Item selection condition>. Выполняется соединение со Spool по фиктивному условию. Результат перераспределяется между всеми AMP и сортируется (см. И1×И2×И3)
4. На всех AMP выполняется соединение Sales и Spool с использованием MERGE JOIN (сканирование строк с сопоставлением хэш-кодов).

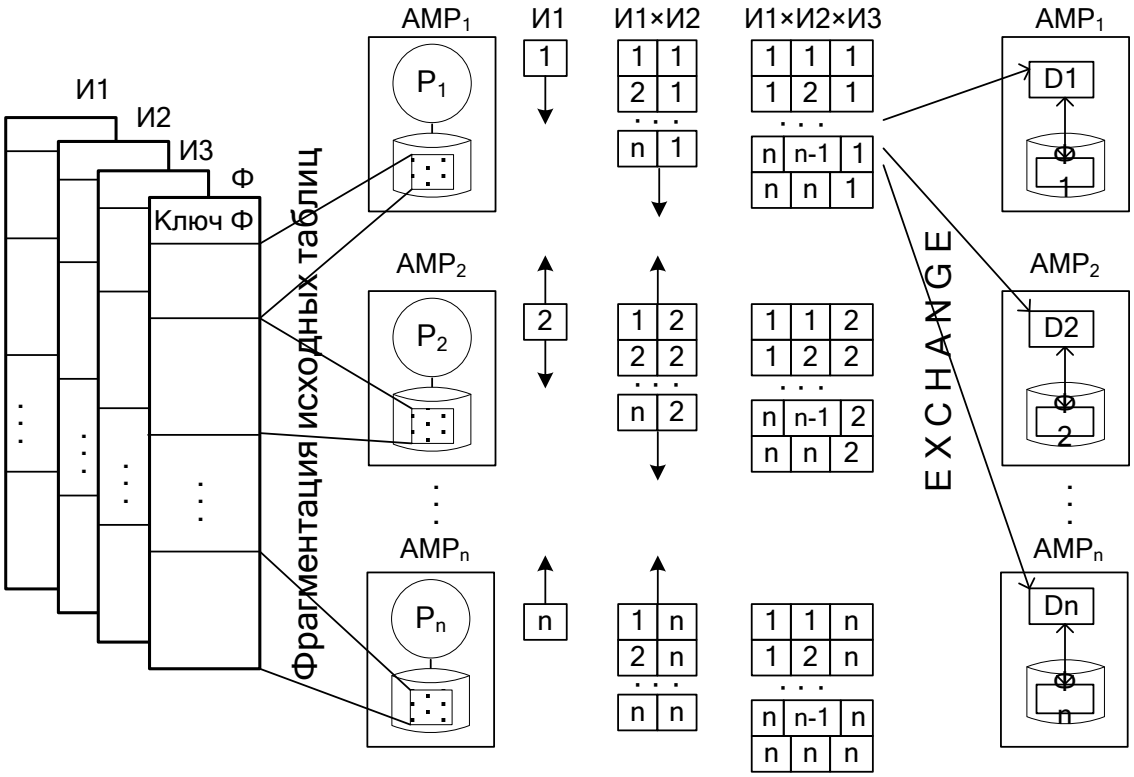

Рис. 3. Организация выполнения запроса к хранилищу данных в ПСБД.

Параллельные системы баз данных могут функционировать в многопроцессорных системах с разными архитектурными решениями [5]:

- архитектура SE (SMP) –системы, где ресурсы являются разделяемыми (дисковая подсистема, оперативная память, системная шина),
- кластерная архитектура SD - системы, в которых отсутствуют разделяемые ресурсы за исключением подсистемы дисковой памяти и соединительной шины,
- архитектура SN (MPP) – системы, где разделяется только шина, соединяющая узлы.

Ниже приводится преобразование Лапласа-Стилтьеса (ПЛС) времени выполнения запроса к хранилищу данных, которое справедливо для каждого AMP параллельной системы баз данных. Здесь используется математический аппарат, изложенный в работах [4, 8, 11].

Чтение данных измерений

Предполагается, что данные читаются по некластеризованному индексу. ПЛС времени поиска и чтения записей из таблиц измерений имеет следующий вид.

$$R(s) = \prod_{i=1}^{K} G_i(1 - p_i(1 - \chi^2(s))), \qquad (1)$$

где

K – число измерений,

$G_i(z) = z^{\frac{V_i}{n}}$ - производящая функция числа записей i-го измерения в узле,

V_i – общее число записей i-го измерения,

n – число АМР- узлов (АМР-процессоров),

p_i - вероятность, что запись i-го измерения удовлетворяет условию поиска по этому измерению в запросе,

$\chi^2(s)$ - ПЛС времени обработки записи блока листового уровня индекса ($\chi(s)$) и записи таблицы измерения ($\chi(s)$),

$$\chi(s) = \varphi_D(s)\varphi_M^2(s)\varphi_{PI}(s). \qquad (2)$$

Это ПЛС учитывает, что блок чередования, где хранится требуемая запись таблицы (или индекса), читается с диска ($\varphi_D(s)$), запись сохраняется и читается из ОП ($\varphi_M^2(s)$), обрабатывается процессором ($\varphi_{PI}(s)$).

Обмен записями таблиц измерений и декартова произведения между узлами

Введём следующие формулы.

$$Y_1(s,z) = G_1(1 - p_1(1 - \varphi_N^{n-1}(s)\varphi_M^{2(n-1)}(s)z^n)), \qquad (3)$$

в этой формуле учитывается, что

каждая запись таблицы первого измерения рассматриваемого узла, удовлетворяющая условию поиска (p_1), дублируется на другие узлы ($\varphi_N^{n-1}(s)$),

записи, поступившие из других узлов, сохраняются в ОП и читаются в кэш процессора данного узла ($\varphi_M^{2(n-1)}(s)$),

каждая запись таблицы первого измерения, удовлетворяющая условию поиска, продублирована на каждом узле (z^n); будем обозначать этот дубль через y_1 (эта продублированная таблица одинакова на всех узлах).

$$Y_2(s,z) = Y_1(s, G_2(1 - p_2(1 - \varphi_{PJ}(s)\varphi_M^2(s)\varphi_N^{n-1}(s)\varphi_M^{2(n-1)}(s)z^n))), \qquad (4)$$

в этой формуле учитывается

процессорное время выполнения декартова произведения таблицы y_1 и записей таблицы второго измерения рассматриваемого узла ($\varphi_{PJ}(s)$), удовлетворяющих условию поиска (p_2),

чтение из ОП для сохранения записей декартова произведения в кэше процессора, а затем в ОП ($\varphi_M^2(s)$),

что каждая запись декартова произведения рассматриваемого узла, дублируется на другие узлы ($\varphi_N^{n-1}(s)$),

что записи декартова произведения, поступившие из других узлов, сохраняются в ОП и читаются в кэш процессора данного узла ($\varphi_M^{2(n-1)}(s)$),

что каждая запись декартова произведения продублирована на каждом узле, т.е. что записи поступили от других узлов (z^n); будем обозначать этот дубль через y_2 (эта продублированная таблица с результатами декартова произведения записей таблиц первых двух измерений одинакова на всех узлах).

И так далее получим

$$Y_{K-1}(s,z) = Y_{K-2}(s, G_{K-1}(1 - p_{K-1}(1 - \varphi_{PJ}(s)\varphi_M^2(s)\varphi_N^{n-1}(s)\varphi_M^{2(n-1)}(s)z^n))),$$

$$V(s,Z) = Y_{K-1}(s, G_K(1 - p_K(1 - \varphi_{PJ}(s)\varphi_M^2(s)q_n(z_1,\ldots,z_n)))), \qquad (5)$$

где

$Z = (z_1, \ldots, z_n)$ – вектор,

функция $q_n(z_1,\ldots,z_n)$ - это производящая функция, которая определяет распределение одной записи полученного на данном узле декартова произведения записей таблиц измерений по другим узлам; распределение (межпроцессорный обмен) выполняет специальный оператор exchange по значениям составного ключа декартова произведения; $q_n(z_1,\ldots,z_n)$ определяется следующими рекуррентными формулами:

$$q_n(z_1,\ldots,z_n) = (1-p_n)q_{n-1}(z_1,\ldots,z_{n-1}) + p_n z_n,$$

$$q_{n-1}(z_1,\ldots,z_{n-1}) = (1-p_{n-1})q_{n-2}(z_1,\ldots,z_{n-2}) + p_{n-1}z_{n-1},$$

... (6)

$$q_1(z_1) = (1-p_1) + p_1 z_1, \; p_1 \equiv 1,$$

p_j - это вероятность, что запись передаётся из данного узла в j-й узел при условии, что она не была передана в узлы n...j+1.

Поясним формулы (6). С вероятностью p_n запись декартова произведения передаётся в n-й узел (испытание по схеме Бернулли успешно). С вероятностью $(1 - p_n)$ запись передаётся в другие узлы. Производящая функция числа таких записей равна $q_{n-1}(z_1,\ldots,z_{n-1})$ и т.д. по рекурсии.

Приведённая ниже формула определяет время пересылки промежуточных декартовых произведений измерений (s), время межпроцессорного обмена записями окончательного декартова произведения ($\varphi_N(s)$) и число записей таблицы фактов, обрабатываемых на данном (i-ом) узле (z):

$$H_i(s,z) = V(s, \overset{1}{\varphi_N}(s), ..., \overset{i-1}{\varphi_N}(s), \overset{i}{z}, \overset{i+1}{\varphi_N}(s), ..., \overset{n}{\varphi_N}(s)) \times$$
$$\times V^{n-1}(0, 1_1, ..., 1_{i-1}, \varphi_M^2(s)z, 1_{i+1}, ..., 1_n) \quad (7)$$

ПЛС $\varphi_M^2(s)$ учитывает, что поступившие в i-ый узел записи декартова произведения от других узлов сохраняются в ОП и читаются в кэш процессора.

ПЛС времени выполнения запроса к ROLAP в ПСБД

Это ПЛС имеет следующий вид

$$\Psi_i(s) = R(s) \cdot H_i(s, \varphi_{PS}(s)\chi(s)\varphi_{PA}(s)), \quad (8)$$

R(s) определяется выражением (1), $H_i(s, \cdot)$ – выражением (7), $\chi(s)$ – определяется выражением (2), $\varphi_{PA}(s)$ – ПЛС времени, затраченного узлом на агрегирование факта одной записи, $\varphi_{PS}(s)$ – ПЛС времени сортировки на одно сравнение в ОП.

Таблица 1

<table>
<tr><th></th><th>SE</th><th>SD</th><th>SN</th></tr>
<tr><td>$\varphi_D(s)$</td><td colspan="2">$1 - p_D + p_D \frac{\mu_{DB} - (n-1) \cdot p_D \lambda_D / N_R}{\mu_{DB} - (n-1) \cdot p_D \lambda_D / N_R + s}$</td><td>$1 - p_D + p_D \frac{\mu_{DB}}{\mu_{DB} + s}$</td></tr>
<tr><td>$\varphi_M(s)$</td><td rowspan="2">$\frac{\mu_M - (n-1)\lambda_M}{\mu_M - (n-1)\lambda_M + s}$
(обмен между процессорами осуществляется через ОП)</td><td colspan="2">$\frac{\mu_M}{\mu_M + s}$</td></tr>
<tr><td>$\varphi_N(s)$</td><td colspan="2">$\frac{\mu_N - (n-1)\lambda_N}{\mu_N - (n-1)\lambda_N + s}$</td></tr>
<tr><td>$\varphi_{PI}(s)$, $\varphi_{PJ}(s)$, $\varphi_{PA}(s)$, $\varphi_{PS}(s)$</td><td colspan="3">$\frac{\mu_{PI}}{\mu_{PI} + s}, \frac{\mu_{PJ}}{\mu_{PJ} + s}, \frac{\mu_{PA}}{\mu_{PA} + s}, \frac{\mu_{PS}}{\mu_{PS} + \log_2(Q_{PS})s}$</td></tr>
</table>

В табл. 1 приведены выражения для ПЛС $\varphi_D(s)$, $\varphi_M(s)$, $\varphi_N(s)$, $\varphi_{PI}(s)$, $\varphi_{PJ}(s)$, $\varphi_{PA}(s)$, $\varphi_{PS}(s)$ – это ПЛС случайного времени обработки записи (исходной, промежуточной, результирующей) в ресурсе (диске, ОП, шине, процессоре),

$1-p_D$ – вероятность, что запись находится в кэше (СУБД или диска),

λ_D, λ_M, λ_N – интенсивности запросов к разделяемому ресурсу (на чтение записей из RAID-массива, на запись/чтение записей таблиц из ОП, на передачу записей по соединительной шине),

$\mu_{DB}=1/t_{БЧ}$ – интенсивность чтения блоков чередования с диска RAID-массива, $t_{БЧ}$ - среднее время чтения блока чередования,

N_R – количество дисков в RAID-массиве (с учётом зеркальных),

μ_M – интенсивность записи/чтения записей таблиц из ОП,

μ_N – интенсивность передачи записей по соединительной шине,

μ_{PI} – интенсивность обработки записей в процессоре при их поиске и чтении с помощью индекса,

μ_{PJ} – интенсивность построения записей декартова произведения в процессоре,

μ_{PS} – интенсивность сравнений записей БД в процессоре при их сортировке,

μ_{PA} – интенсивность агрегации записей (значений фактов),

Q_{PS} – значение коэффициента при среднем значении $\bar{\varphi}_{PS}$ (см. ниже),

(n-1) в табл. 1 означает, что заявка не ждёт сама себя в очереди к ресурсу.

Таким образом, в статье приведён вывод преобразования Лапласа-Стилтьеса случайного времени выполнения запроса к хранилищу данных для различных архитектур: SE (SMP), SD (кластер), SN (MPP), смешанной архитектуры на основе SMP-узлов. Это преобразование позволяет оценивать моменты случайного времени разных порядков (математическое ожидание, дисперсию и т.д.).

ЛИТЕРАТУРА

1. М. Тамер Оззу, Патрик Валдуриз. Распределенные и параллельные системы баз данных: [Электронный ресурс]. [http://citforum.ru/database/classics/distr_and_paral_sdb/]. Проверено 26.11.2010.
2. Соколинский Л. Б., Цымблер М. Л. Лекции по курсу "Параллельные системы баз данных”: [Электронный ресурс]. [http://pdbs.susu.ru/CourseManual.html]. Проверено 04.12.2010.
3. Дж. Льюис. Oracle. Основы стоимостной оптимизации. – СПб: Питер, 2007. – 528 с.
4. Григорьев Ю.А., Плужников В.Л. Оценка времени соединения таблиц в параллельной системе баз данных// Информатика и системы управления. – 2011. - № 1. – С. 3-16.
5. Лисянский К., Слободяников Д. СУБД Teradata® для ОС UNIX®: [Электронный ресурс]. [http://citforum.ru/database/kbd98/glava5.shtml]. Проверено 14.03.2011.
6. Кузнецов С. Essential Modelling Options: [Электронный ресурс]. [http://citforum.ru/database/digest/dig_1612.shtml]. Проверено 14.03.2011.

7. Лев Левин. Teradata совершенствует хранилища данных: [Электронный ресурс]. [http://www.pcweek.ru/themes/detail.php?ID=71626]. Проверено 26.11.2010.
8. Григорьев Ю.А., Плутенко А.Д. Теоретические основы анализа процессов доступа к распределённым базам данных. - Новосибирск: Наука, 2002. – 180 с.
9. Форум/Использование СУБД/Oracle/CPUSPEED на IntelXeon 5500 (Nehalem): [Электронный ресурс]. [http://www.sql.ru]. Проверено 02.12.2010.
10. Миллер Р., Боксер Л. Последовательные и параллельные алгоритмы. Общий подход. – М.: БИНОМ. Лаборатория знаний, 2006. – 406 с.
11. Григорьев Ю.А., Плужников В.Л. Анализ времени обработки запросов к хранилищу данных в параллельной системе баз данных // Информатика и системы управления. – 2011. - № 2. – С. 94-106.