Преобразование Лапласа-Стилтьеса времени соединения двух таблиц в параллельной системе баз данных

06, июнь 2012

DOI: 10.7463/0612.0390320

Плужников В. Л., Домников А. С.

УДК 004.657

Россия, МГТУ им. Н.Э. Баумана

chernen@bmstu.ru

VPluzhnikov@croc.ru

Выполнение запросов в параллельной системе баз данных

Процесс выполнения SQL-запроса в параллельной системе баз данных можно представить в виде следующих шагов [2]:

- генерация последовательного плана выполнения запроса,
- тиражирование плана выполнения запроса на все узлы системы,
- обработка запроса над фрагментированными таблицами (распределение фрагментов таблиц БД по узлам системы выполняется заранее и один раз),
- слияние полученных данных.

Рассмотрим процесс параллельной обработки запроса, где выполняется соединение таблиц R и S базы данных (рис. 1). $Q = R \bowtie S$ – это логическая операция соединения (join) двух отношений (таблиц) R и S по некоторому общему атрибуту Y . В данном примере таблица R фрагментирована произвольным образом, а таблица S – по атрибуту соединения Y . На рис. 1 показано, что логический план выполнения соединения двух отношений тиражируется на 'n' процессоров в параллельной системе баз данных (на рисунке показаны 2 процессора). Далее происходит параллельная обработка на каждом процессоре соответствующих фрагментов таблиц R и S . Вследствие того, что таблица R не фрагментирована по атрибуту соединения, при последовательном чтении записей этой таблицы происходит их обработка в операторе exchange, осуществляющем разбор записи и её межпроцессорный обмен. Таблица S фрагментирована по атрибуту соединения и записи читаемые из фрагментов этой таблицы обрабатываются на каждом процессоре локально.

Оператор exchange является составным оператором и включает в себя четыре оператора: gather, scatter, split и merge. Оператор split - это оператор, который осуществляет разбиение кортежей (записей), поступающих из входного потока, на две группы: свои и чужие. Свои кортежи - это кортежи, которые должны быть обработаны на данном процессорном узле. Эти кортежи направляются в выходной буфер оператора split (стрелка вверх). Чужие кортежи, то есть кортежи, которые должны быть обработаны на процессорных узлах, отличных от данного, помещаются оператором split во входной буфер правого дочернего узла, в качестве которого фигурирует оператор scatter. Нулевой оператор scatter извлекает кортежи из своего входного буфера и пересылает их на соответствующие процессорные узлы. Нулевой оператор gather выполняет перманентное чтение кортежей из указанного порта со всех процессорных узлов, отличных от данного. Оператор merge, реализующий логическую операцию join, определяется как бинарный оператор, который забирает кортежи из выходных буферов своих дочерних узлов, соединяет их и помещает результат в собственный выходной буфер. Таким образом, с помощью рассмотренных операций оператор exchange реализует полноценный межпроцессорный обмен записями в параллельной системе баз данных при обработке запроса методом фрагментарного параллелизма.

Рис. 1. Обработка запроса $Q = R \bowtie S$ в параллельной системе баз данных.

Для реализации параллельных систем баз данных используются следующие архитектуры:

1. SE (Shared-Everything) - архитектура с разделяемыми памятью и дисками.
2. SD (Shared-Disks) - архитектура с разделяемыми дисками.
3. SN (Shared-Nothing) - архитектура без совместного использования ресурсов.

Преобразование Лапласа-Стилтьеса времени соединения двух таблиц

Ниже приведены преобразования Лапласа-Стилтьеса (ПЛС) времени соединения таблиц А и В в параллельной системе баз данных [15]. ПЛС позволяют оценивать не только математические ожидания случайных величин, но и моменты более высокого порядка. Предполагается, что записи таблиц фильтруются и соединяются по неиндексированным атрибутам (т.е. рассматривается наихудший случай). Также предполагается, что таблица В фрагментирована по атрибуту соединения, а таблица А – нет. ПЛС получены для двух способов соединения: NLJ (соединение с помощью вложенных циклов) и HJ (хешированное соединение).

Формула (1) определяет производящую функцию (ПФ) числа записей исходной фрагментированной таблицы А i-го узла, передаваемых другим процессорам в соответствии с функцией распределения этого узла ($s=0$). Записи, распределённые на i-й процессор (z_i), обрабатываются на этом узле, а остальные записи передаются другим процессорам по команде exchange.

$$V_{Ai}(s, z_1, \dots, z_n) = G_{Ai}(\phi_D(s)\phi_M^2(s)\phi_P(s)(1 - P_A(1 - q_{Ain}(z_1, \dots, z_n)))), \quad (1)$$

где $G_{Ai}(z) = z^{\frac{V_A}{n}}$ - производящая функция числа записей исходной фрагментированной таблицы А, обрабатываемых в i-ом процессоре (узле); конечно, здесь эта функция может быть более сложной; V_A/n - число записей таблицы А, которые хранятся в i-м узле; n – число узлов; P_A – вероятность, что запись таблицы А удовлетворяет условию поиска по этой таблице (условию фильтрации); $\phi_D(s)\phi_M^2(s)\phi_P(s)$ - учитывает, что каждая запись таблицы А читается с диска, сохраняется и читается из ОП, обрабатывается процессором, $q_{Ain}(z_1, \dots, z_n)$ определяется следующими рекуррентными формулами:

$$\begin{aligned} q_{Ain}(z_1, \dots, z_n) &= (1 - p_{Ain})q_{Ain-1}(z_1, \dots, z_{n-1}) + p_{Ain}z_n, \\ q_{Ain-1}(z_1, \dots, z_{n-1}) &= (1 - p_{Ain-1})q_{Ain-2}(z_1, \dots, z_{n-2}) + p_{Ain-1}z_{n-1} \\ q_{Ail}(z_1) &= (1 - p_{Ail}) + p_{Ail}z_1, \quad p_{Ail} \equiv 1, \end{aligned} \quad (2)$$

p_{Aij} - это вероятность, что запись передаётся из i -го узла в j -й узел при условии, что она не была передана в узлы $n \dots j+1$.

Формула (1) выводится на основании свойств производящих функций [10]. Действительно,

$$G_{Ai}(1 - P_A(1 - z)) \quad (3)$$

- это производящая функция числа записей, удовлетворяющих условию поиска по таблице A ,

$$q_{Ain}(z_1, \dots, z_n) \quad (4)$$

- производящая функция числа записей, передаваемых процессорам в соответствии с функцией распределения, для одной произвольной записи таблицы A (для случая, когда таблица фрагментирована не по атрибуту соединения; если таблица фрагментирована по атрибуту соединения, то $q_{Ain}(z_1, \dots, z_n) = z_i$).

Поясним формулы (2). С вероятностью p_{Ain} запись передаётся в n -й узел (испытание по схеме Бернулли успешно). С вероятностью $(1 - p_{Ain})$ запись передаётся в другие узлы. Производящая функция числа таких записей равна $q_{Ain-1}(z_1, \dots, z_{n-1})$ и т.д. по рекурсии.

Подставляя (4) в (3), получим (1). Формула (5) определяет производящую функцию числа записей таблицы A , соединяемых в i -ом узле.

$$W_{Ai}(z) = \prod_{j=1}^n V_{Aj}(0, 1_1, \dots, 1_{i-1}, z, 1_{i+1}, \dots, 1_n), \quad (5)$$

здесь $V_{Aj}(s, z_1, \dots, z_n)$ определяется выражением (1).

При выводе формулы (5) учитывалось, что число записей, обрабатываемых в i -ом узле равно сумме случайного числа записей, поступающих в i -й узел из всех n узлов в соответствии с их функциями распределения. Производящая функция суммы случайного числа записей равна произведению производящих функций.

Производящая функция результирующего числа записей, полученных после соединения таблиц A и B в i -ом узле, определяется следующим выражением:

$$W_{ABi}(z) = W_{Ai}(G_{Bi}(1 - P_B \eta_{AB}(1 - z))), \quad (6)$$

здесь $G_{Bi}(z) = z^{V_B/n}$ - производящая функция числа записей исходной фрагментированной таблицы B , обрабатываемых в i -ом узле; V_B/n - число записей таблицы B , которые хранятся в i -ом узле. Предполагается, что таблица B фрагментирована по атрибуту соединения, η_{AB} - вероятность, что две записи из таблиц A и B удовлетворяют условию соединения.

$$\eta_{AB} = \sum_{j=1}^{|D_A|} \eta_{Aj} \eta_{Bk}, \quad (7)$$

здесь $|D_A|$ - мощность домена атрибута соединения в таблице А, η_{Aj} - вероятность, что атрибут соединения в записи таблицы А принимает значение $d_{Aj} \in D_A$, η_{Bk} - вероятность, что атрибут соединения в записи таблицы В принимает значение $d_{Bk} \in D_B$: $d_{Bk} = d_{Aj}$, D_B - домен атрибута соединения в таблице В. Докажем (6).

$$W_{Ai}(G_{Bi}(z)) \quad (8)$$

- это производящая функция числа записей в декартовом произведении соединяемых таблиц,

$$1 - P_B \eta_{AB} (1 - z) \quad (9)$$

- это производящая функция числа записей, удовлетворяющих условию фильтрации (P_B) и условию соединения (η_{AB}), для одной записи таблицы В (действует схема испытания Бернулли). Вероятность η_{AB} получена по формуле полной вероятности (см. (7)). Подставляя (9) в (8), получаем (6).

Получим формулу для ПЛС времени соединения таблиц по методу NLJ в i -ом узле. Ведём следующую функцию

$$H_{Ai}(s, z) = V_{Ai}(s, \phi_N^1(s), \dots, \phi_N^{i-1}(s), z, \phi_N^i(s), \dots, \phi_N^n(s)) \times \prod_{j=1, j \neq i}^n V_{Aj}(0, 1_1, \dots, 1_{i-1}, z, 1_{i+1}, \dots, 1_n), \quad (10)$$

где $V_{Ai}(\cdot)$ определяется выражением (1).

Функция (10) определяет время чтения записей таблицы А в i -ом узле (s), время пересылки записей таблицы другим узлам по команде exchange ($\phi_N(s)$), где они обрабатываются процессорами принимающих узлов, а также число записей таблицы, соединяемых в i -ом узле (z). ПЛС времени соединения таблиц по методу NLJ имеет вид

$$\Psi_i^{NLJ}(s) = H_{Ai}(s, G_{Bi}(\phi_D(s) \phi_M^2(s) \phi_P(s) (1 - P_B \eta_{AB} (1 - \phi_N(s))))) , \quad (11)$$

где $H_{Ai}(\cdot)$ определяется выражением (10), $G_{Bi}(1 - P_B \eta_{AB} (1 - z))$ - производящая функция числа записей таблицы В, удовлетворяющих условию поиска по таблице В (P_B) и условию соединения (η_{AB} - см. (7)). Поясним (11).

Следует отметить, что таблица В, сканируемая во внутреннем цикле, фрагментирована по атрибуту соединения. Каждая запись из А сравнивается по

атрибуту соединения со всеми записями из В. Каждая запись из В должна быть прочитана с диска, сохранена и прочитана из ОП и обработана в процессоре (произведение $\phi_D(s)\phi_M^2(s)\phi_P(s)$, при этом учитывается наличие общесистемного буфера СУБД, т.е. кэша). В формуле (11) учитывается, что чтение и обработка записей из В повторяется для каждой записи из А (это связано с тем, что в скобочном шаблоне для каждой записи из А выполняется команда reset для правого сына, т.е. для В). С вероятностью $P_B\eta_{AB}$ результирующая запись передаётся по команде exchange в узел ($\phi_N(s)$), где выполняется сборка и формируется окончательный результат.

Получим ПЛС времени соединения таблиц по методу НЈ (алгоритм Grace) в i -ом узле. Таблица В фрагментирована по атрибуту соединения. ПЛС времени соединения таблиц по методу НЈ имеет вид

$$\begin{aligned} \Psi_i^{HJ}(s) = & H_{Ai}(s, (\phi_D(s)\phi_M^2(s))^{\omega_A}) \times \\ & G_{Bi}(\phi_D(s)\phi_M^2(s)\phi_P(s)(1 - P_B(1 - (\phi_D(s)\phi_M^2(s))^{\omega_B}))) \times \\ & W_{Ai}(G_{Bi}(1 - P_B + P_B(1 - \frac{1}{r} + \frac{1}{r}\phi_P(s))(1 - \eta_{AB} + \eta_{AB}\phi_N(s)))) \end{aligned} \quad (12)$$

где $H_{Ai}(s, z)$ определяются выражением (10), $G_{Bi}(z)$ – производящая функция числа записей фрагментированной исходной таблицы В, обрабатываемых в i -ом узле, $W_{Ai}(z)$ определяется выражением (5).

Поясним формулу (12). Записи фрагментированной таблицы А читаются в i -м узле ($H_{Ai}(s, \cdot)$). Записи таблицы В должны быть прочитаны с диска, сохранены и прочитаны из ОП, а также обработаны в процессоре с целью их фильтрации ($G_{Bi}(\phi_D(s)\phi_M^2(s)\phi_P(s)(1 - P_B(\cdot)))$). Записи таблиц А и В, поступившие в i -й узел, подвергаются хэшированию ($(\phi_D(s)\phi_M^2(s))^{\omega_A}$ и $(\phi_D(s)\phi_M^2(s))^{\omega_B}$).

Если хэш-таблица создающей таблицы А полностью сохраняется в оперативной памяти, то записи и создающей таблицы (А), и зондирующей таблицы (В) читаются с диска и обрабатываются один раз (это чтение уже учтено, поэтому $\omega_A = \omega_B = 0$). Если нет, то каждая таблица читается с диска (1-ое обращение к диску, оно уже учтено), она хэшируется в ОП и хэш-группы сохраняются на диске (2-ое обращение к диску). После этого пары хэш-групп таблиц А и В (одна пара имеет одинаковое значение хэш-ключа) читаются с диска (3-е обращение к диску). Т.к. размеры хэш-групп создающей таблицы А могут отличаться, то в рассматриваемом случае $\omega_A \leq 2$ и $\omega_B \leq 2$.

Поясним третий сомножитель в формуле (12). Хэш-группы (разделы) таблиц А и В соединяются в оперативной памяти. Каждая запись таблицы А

хранится в одном из g разделов (число записей таблицы A определяется ПФ $W_{Ai}(\bullet)$). Произвольная запись из B (их число определяется ПФ $G_{Bi}(1-P_B+P_{BZ})$) с вероятностью $1/g$ попадает в тот раздел, где хранится запись из A , и выполняется операция сравнения значений атрибутов соединения ($\varphi_R(s)$). С вероятностью η_{AB} сравнение будет успешным и результат соединения двух записей будет передан в узел сборки ($\varphi_N(s)$).

Так как узлы в параллельной системе баз данных идентичны, то (11) и (12) определяют ПЛС времени реализации плана соединения, т.е. времени выполнения исходного запроса. Дифференцируя (11) и (12) в нуле, можно получить моменты (математическое ожидание, дисперсию и т.д.) случайного времени выполнения соединения таблиц в параллельной системе баз данных методами NLJ и HJ.

ЛИТЕРАТУРА

1. М. Тамер Оззу, Патрик Валдуриз. Распределенные и параллельные системы баз данных: [Электронный ресурс]. Режим доступа: 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]. Режим доступа: <http://pddb.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 выполнения запросов и выбор архитектуры параллельной системы баз данных// Информатика и системы управления. – 2009. - № 3. – С. 3-12.
5. Производительность СУБД Oracle Database 11g при работе на сервере Sun SPARC Enterprise M9000: [Электронный ресурс]. [<http://ru.sun.com/sunnews/press/2010/2010-05-18.jsp>]. Проверено 26.11.2010
6. В.А. Варфоломеев, Э.К. Лецкий, М.И. Шамров, В.В. Яковлев. Лекции по курсу "Операционные системы и программное обеспечение на платформе zSeries": [Электронный ресурс]. [<http://www.intuit.ru/department/os/ibmzos/>]. Проверено 26.11.2010.
7. Керри Болинджер. Врожденный параллелизм: [Электронный ресурс]. [http://www.osp.ru/os/2006/02/1156526/_p1.html]. Проверено 26.11.2010.
8. Лев Левин. Teradata совершенствует хранилища данных: [Электронный ресурс]. [<http://www.pcweek.ru/themes/detail.php?ID=71626>]. Проверено 26.11.2010.

9. Oracle Real Application Clusters Administration and Deployment Guide 11g Release 1 (11.1): [Электронный ресурс]. [http://download.oracle.com/docs/cd/B28359_01/rac.111/b28254/admcon.htm/]. Проверено 26.11.2010.
10. Григорьев Ю.А., Плутенко А.Д. Теоретические основы анализа процессов доступа к распределённым базам данных. - Новосибирск: Наука, 2002. – 180 с.
11. Жожикашвили В.А, Вишневский В.М. Сети массового обслуживания. Теория и применение к сетям ЭВМ. – М.: Радио и связь, 1988. – 192 с.
12. Клейнрок Л. Теория массового обслуживания. – М.: Машиностроение, 1979. – 432 с.
13. Бронштейн О.И., Духовный И.М. Модели приоритетного обслуживания в информационно-вычислительных системах. – М.: Наука, 1976. – 220 с.
14. Форум/Использование СУБД/Oracle/CPUSPEED на IntelXeon 5500 (Nehalem): [Электронный ресурс]. [<http://www.sql.ru>]. Проверено 02.12.2010.
15. Григорьев Ю.А., Плужников В.Л. Оценка времени соединения таблиц в параллельной системе баз данных// Информатика и системы управления. – 2011. - № 1. – С. 3-16.

The Laplace–Stieltjes transform of connection time of two tables in parallel database

06, June 2012

DOI: 10.7463/0612.0390320

Pluzhnikov V.L., Domnikov A.S.

Russia, Bauman Moscow State Technical University

chernen@bmstu.ru

VPluzhnikov@croc.ru

Connection of two tables in parallel database system (PDBS) was analyzed in this article. The Laplace–Stieltjes transform (LST) of connection time of two tables with NLJ and HJ methods in PDBS was introduced. Random character of the number of rows in the connecting tables is considered along with node fragmentation of tables, filtration of all table entries, probability of matching of connection attribute values, variants of hash-table location in the memory and parameters of interprocessor communication of PDBS nodes. Generating function of resultant value of entries, obtained after connection of tables in any node is presented in the article.

Publications with keywords: [parallel system of database](#), [Laplace-Stieltjes transformation](#), [SE](#), [SD](#), [SN architectures](#), [NLJ and HJ methods of connection of tables](#)

Publications with words: [parallel system of database](#), [Laplace-Stieltjes transformation](#), [SE](#), [SD](#), [SN architectures](#), [NLJ and HJ methods of connection of tables](#)

References

1. Ozsu M.T., Valduriez P. *Distributed and parallel database systems* (Russ. version: M. Tamer Ozsu, Patrik Valduriz. *Raspredeleennye i parallel'nye sistemy baz dannykh*. Available at: http://citforum.ru/database/classics/distr_and_paral_sdb/, accessed 26.11.2010).
2. Sokolinskii L. B., Tsymbler M. L. *Lektsii po kursu «Parallel'nye sistemy baz dannykh»* [Lectures on the course “Parallel database systems”]. Available at: <http://pddb.susu.ru/CourseManual.html>, accessed 04.12.2010.
3. Lewis J. *Cost-Based Oracle Fundamentals (Expert's Voice in Oracle)*. Apress, 2005. 520 p. (Russ. ed.: Dzh. L'iuiz. *Oracle. Osnovy stoimostnoi optimizatsii*. SPb., Piter, 2007. 528 p.).
4. Grigor'ev Iu.A., Pluzhnikov V.L. Otsenka vremeni vypolneniia zaprosov i vybor arkhitektury parallel'noi sistemy baz

dannykh [Score a run-time of query and selection of architecture of the parallel database systems]. *Informatika i sistemy upravleniia*, 2009, no. 3, pp. 3-12.

5. *Proizvoditel'nost' SUBD Oracle Database 11g pri rabote na servere Sun SPARC Enterprise M9000* [Performance of Oracle Database 11g database while working on the server Sun SPARC Enterprise M9000]. Available at: <http://ru.sun.com/sunnews/press/2010/2010-05-18.jsp>, accessed 26.11.2010.

6. Varfolomeev V.A., Letskii E.K., Shamrov M.I., Iakovlev V.V. *Lektsii po kursu "Operatsionnye sistemy i programmnoe obespechenie na platforme zSeries"* [Lectures on the course "Operating systems and software on zSeries platform"]. Available at: <http://www.intuit.ru/departament/os/ibmzos/>, accessed 26.11.2010.

7. Carrie Ballinger. *Born To Be Parallel* (Russ. version: Kerri Bolindzher. *Vrozhdennyi parallelizm*). Available at: <http://www.osp.ru/os/2006/02/1156526/>, accessed 04.05.2012.).

8. Levin L. *Teradata sovershenstvuet khranilishcha dannykh* [Teradata improves data ware-house]. Available at: <http://www.pcweek.ru/themes/detail.php?ID=71626>, accessed 04.05.2012.

9. *Oracle Real Application Clusters Administration and Deployment Guide 11g Release 1 (11.1)*. Available at: http://download.oracle.com/docs/cd/B28359_01/rac.111/b28254/admcon.htm, accessed 26.11.2010.

10. Grigor'ev Iu.A., Plutenko A.D. *Teoreticheskie osnovy analiza protsessov dostupa k raspredelenym bazam dannykh* [Theoretical basis of analysis of the processes of access to distributed databases]. Novosibirsk, Nauka, 2002. 180 p.

11. Zhozhikashvili V.A., Vishnevskii V.M. *Seti massovogo obsluzhivaniia. Teoriia i prime-nenie k setiam EVM* [Queueing network. Theory and application to computer networks]. Moscow, Radio i sviaz', 1988. 192 p.

12. Kleinrok L. *Queueing Systems. Vol. 1. Theory*. New York, John Wiley & Sons, 1975. (Russ ed.: Kleinrok L. *Teoriia massovogo obsluzhivaniia*. Moscow, Mashinostroenie, 1979. 432 p.).

13. Bronshtein O.I., Dukhovnyi I.M. *Modeli prioritetnogo obsluzhivaniia v informatsionno-vychislitel'nykh sistemakh* [Model of the priority services in the information-computing systems]. Moscow, Nauka, 1976. 220 p.

14. *Forum/Ispolzovanie SUBD/Oracle/CPUSPEED na IntelXeon 5500 (Nehalem)* [Forum / Using the database/Oracle/CPUSPEED on IntelXeon 5500 (Nehalem)]. Available at: <http://www.sql.ru>, accessed 02.12.2010.

15. Grigor'ev Iu.A., Pluzhnikov V.L. *Otsenka vremeni soedineniia tablits v parallel'noi sisteme baz dannykh* [Score of the time of connection of tables in a parallel database system]. *Informatika i sistemy upravleniia*, 2011, no. 1, pp. 3-16.