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

Оценка среднего времени выполнения соединения таблиц методами NLJ и HJ в параллельной системе баз данных

77-30569/380603

04, апрель 2012

Григорьев Ю. А., Гасов В. М.

УДК 004.657

МГТУ им. Н.Э. Баумана
chernen@bmstu.ru

Введение

В настоящее время не существует научно обоснованного метода оценки и выбора архитектуры параллельной системы баз данных [2]. В предыдущих статьях [4, 15] рассмотрен математический метод анализа времени выполнения запроса к одной и двум таблицам параллельной системы баз данных (ПСБД). В данной работе демонстрируется применение математического аппарата производящих функций и преобразования Лапласа-Стилтьеса для оценки моментов времени выполнения соединения таблиц в ПСБД.

Получение такой оценки важна, так как операция соединения (join) часто используется в аналитических запросах к хранилищам данных, построенных на основе реляционных баз данных (ROLAP). Прогнозирование индексов производительности параллельных систем баз данных необходимо – соответствующие аппаратно-программные комплексы очень дороги и ошибки в их выборе приводят к большим финансовым потерям.

Варианты реализация параллельных систем баз данных

Наиболее распространенной системой классификации параллельных систем баз данных является система, предложенная Майклом Стоунбрейкером [2]. В соответствии с этой классификацией архитектуры подразделяются на следующие типы: SE (Shared-Everything) - архитектура с разделяемыми памятью и дисками; SD (Shared-Disks) - архитектура с разделяемыми дисками и соединительной сетью; SN (Shared-Nothing) - архитектура с разделяемой сетью.

Копеланд (Copeland) и Келлер (Keller) предложили расширение классификации Стоунбрейкера путем введения двух дополнительных классов архитектур параллельных машин баз данных: CE (Clustered-Everything) - архитектура с SE кластерами, объединенными по принципу SN; CD (Clustered-Disk) - архитектура с SD кластерами, объединенными по принципу SN.

На текущий момент наиболее часто применяемыми типами архитектур при построении высокопроизводительных параллельных систем является архитектура SE. Реже используются архитектуры SD, SN и CE. Реализаций архитектуры CD не известно. Приведем более подробное описание средств реализации архитектурных решений.

SE-архитектура представляет систему баз данных, в которой все диски напрямую доступны всем процессорам и все процессоры разделяют общую оперативную память. Доступ к дискам в системах SE обычно осуществляется через общий буферный пул. При этом следует отметить, что каждый процессор в системе SE имеет собственную кэш-память. Чаще всего эта архитектура реализуется на промышленных SMP-серверах, построенных на процессорах RISC. Примером реализации архитектуры SE может служить комплекс, состоящий из сервера Sun Enterprise M9000, системы хранения данных EMC Symmetrix DMX-4 и инфраструктуры сети хранения данных SAN [5].

SD-архитектура представляет систему баз данных, в которой любой процессор имеет доступ к любому диску, однако каждый процессор имеет свою приватную оперативную память. Процессоры в таких системах соединены посредством некоторой высокоскоростной сети. Системы SD чаще всего реализуется на промышленных серверах с процессорной архитектурой z/Architecture. Хотя по своему строению эти системы близки к архитектуре SMP-сервера, общая архитектура системы может изменяться с помощью программно-аппаратных средств Parallel Sysplex, позволяющих множеству узлов работать с единой высокопроизводительной внешней системой хранения данных. Примером реализации архитектуры SD может служить комплекс, состоящий из двух серверов IBM z10, системы хранения данных IBM DS8800 и инфраструктуры FICON [6].

SN-архитектура характеризуется наличием у каждого процессора собственной оперативной памяти и собственной дисковой системы. Как и в системах SD , процессорные узлы соединены некоторой высокоскоростной сетью. К настоящему моменту создано большое количество исследовательских прототипов и несколько коммерческих систем с архитектурой SN , использующих фрагментный параллелизм на MPP-системах. Примером реализации архитектуры SN может служить комплекс Teradata Database [7, 8], который представляет собой совокупность SMP-систем (или простых однопроцессорных систем), построенных

на базе процессоров Intel x86_64 и связанных между собой высокоскоростной внутренней сетью BYNET. Каждый узел имеет свою собственную оперативную память и дисковую систему. Так как сеть BYNET реализует межузловой обмен данными по технологии точка-точка, то в данной конфигурации нет разделяемых ресурсов. В этом комплексе в качестве СУБД может функционировать только СУБД Teradata.

В *архитектуре CD* каждый узел представляет собой SD-систему. Используется дополнительное программное обеспечение, позволяющее осуществлять обмен метаданными между системами и арбитраж глобальных блокировок в СУБД и преобразующее весь комплекс к архитектуре CD. Примером реализации CD-архитектуры может служить комплекс из 4-серверов IBM SystemX 3950M2 и системы хранения данных Hitachi Universal Storage Platform V [9].

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 ⊳⊲ 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. Нульарный оператор scatter извлекает кортежи из своего входного буфера и пересылает их на соответствующие процессорные узлы. Нульарный 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.

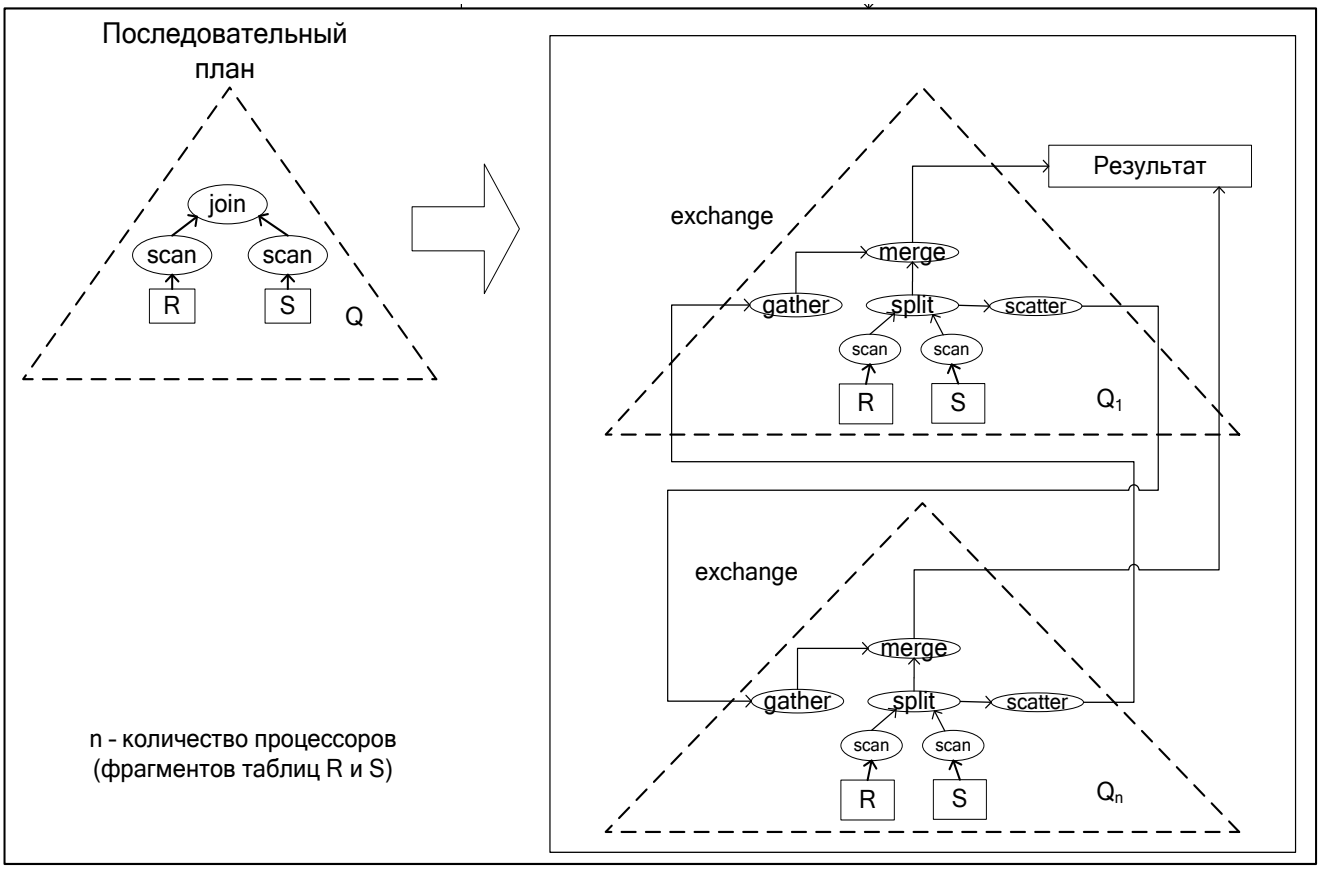

Рис. 1. Обработка запроса Q = R ⊳⊲ S в параллельной системе баз данных.

Оценка математического ожидания времени соединения таблиц

В работе [15] получены преобразования Лапласа-Стилтьеса (ПЛС) времени соединения таблиц A и B методами NLJ и HJ (алгоритм Grace):

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

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

где

$$\begin{aligned}H_{Ai}(s,z) &= V_{Ai}(s, \overset{1}{\phi_N}(s), ..., \overset{i-1}{\phi_N}(s), \overset{i}{z}, \overset{i+1}{\phi_N}(s), ..., \overset{n}{\phi_N}(s)) \times \\ &\times \prod_{j=1, j\neq i}^{n} V_{Aj}(0, 1_1, ..., 1_{i-1}, z, 1_{i+1}, ..., 1_n)\end{aligned}, \quad (3)$$

$$V_{Ai}(s, z_1, ..., z_n) = G_{Ai}(\phi_D(s)\phi_M^2(s)\phi_P(s)(1 - P_A(1 - q_{Ain}(z_1, ..., z_n)))), \quad (4)$$

$G_{Ai}(1 - P_A(1-z))$ - это производящая функция числа записей, удовлетворяющих условию поиска по таблице A (с вероятностью P_A),

$q_{Ain}(z_1, ..., z_n)$ определяется следующими рекуррентными формулами:

$$\begin{aligned}q_{Ain}(z_1, ..., z_n) &= (1 - p_{Ain})q_{Ain-1}(z_1, ..., z_{n-1}) + p_{Ain}z_n, \\ q_{Ain-1}(z_1, ..., z_{n-1}) &= (1 - p_{Ain-1})q_{Ain-2}(z_1, ..., z_{n-2}) + p_{Ain-1}z_{n-1} \\ &...\end{aligned} \quad (5)$$

$$q_{Ai1}(z_1) = (1 - p_{Ai1}) + p_{Ai1}z_1, \; p_{Ai1} \equiv 1,$$

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...j+1,

$G_{Bi}(1 - P_B\eta_{AB}(1-z))$ - производящая функция числа записей таблицы B, удовлетворяющих условию поиска по таблице B (вероятность P_B) и условию соединения (вероятность η_{AB}),

$$W_{Ai}(z) = \prod_{j=1}^{n} V_{Aj}(0, 1_1, ..., 1_{i-1}, z, 1_{i+1}, ..., 1_n), \quad (6)$$

r – число разделов (хеш-групп) в хеш-таблице,

показатели степени $\omega_A \leq 2$ и $\omega_B \leq 2$ определяют число чтений/записей на диск хеш-групп.

В приведённых выше формулах встречаются ПЛС $\varphi_D(s)$, $\phi_M^2(s)$, $\varphi_P(s)$ и $\varphi_N(s)$. Они имеют следующий смысл [4]:

$\varphi_D(s)$ - ПЛС времени чтения записи БД из подсистемы ввода/вывода,

$\phi_M^2(s)$ - ПЛС времени записи и последующего чтения записи БД из ОП,

$\varphi_P(s)$ - ПЛС времени обработки записи БД в процессоре,

$\varphi_N(s)$ - ПЛС времени передачи записи БД по высокоскоростной шине, связывающей процессоры системы.

В табл. 1 приведены выражения для указанных преобразований для различных архитектурных решений [15].

Таблица 1

	SE	SD	SN
$\phi_D(s)$	$1-\frac{1}{L}+\frac{1}{L}\frac{\mu_D-n\lambda_D}{\mu_D-n\lambda_D+Ls}$		$1-\frac{1}{L}+\frac{1}{L}\frac{\mu_D}{\mu_D+Ls}$
$\phi_M(s)$	$\frac{\mu_M-n\lambda_M}{\mu_M-n\lambda_M+s}$ (обмен между процессорами осуществляется через ОП)	$\frac{\mu_M}{\mu_M+s}$	
$\phi_N(s)$		$\frac{\mu_N-n\lambda_N}{\mu_N-n\lambda_N+s}$	
$\phi_P(s)$	$\frac{\mu_P}{\mu_P+s}$		

Здесь n – процессоров (узлов) в параллельной системе баз данных, λ и μ – интенсивности поступления записей от одного процессора и их обработки в соответствующем ресурсе (D - внешней памяти, M - ОП, P - процессоре, N - шине), L – среднее число записей, читаемых из RAID-массива за одно многоблочное чтение при сканировании таблицы СУБД. По существу величина (1-1/L) – это вероятность чтения записи из общесистемного буфера СУБД (т.е. из кэша).

Наличие произведения 'nλ' в ПЛС означает, что ресурс является разделяемым в соответствующей архитектуре (т.е. к нему возможна очередь). Хотя поведение параллельной системы баз данных при выполнении запроса описывается в виде замкнутой системы массового обслуживания (СМО), но, как видно из табл. 1, здесь в качестве модели разделяемого ресурса используется разомкнутая СМО M/M/1.

Дифференцируя выражения (1) и (2) как сложные функции по s в нуле, можно получить моменты случайного времени (ξ) обработки запроса в параллельной системе баз данных (ПСБД) для различных способов соединения таблиц (NLJ и HJ):

$$M_{\xi} = -\Psi'(0), \quad M_{\xi^2} = \Psi''(0), \quad \sigma_{\xi}^2 = M_{\xi^2} - M_{\xi}^2. \tag{7}$$

После дифференцирования с учётом выражений, приведённых в табл. 1, получим следующие формулы для расчёта математических ожиданий времени выполнения соединения таблиц для различных архитектур ПСБД [15]:

$$M_{SE} = \frac{Q_D}{\mu_D - n\lambda_D} + \frac{Q_M + Q_N}{\mu_M - n\lambda_M} + \frac{Q_P}{\mu_P}, \tag{8}$$

$$M_{SD} = \frac{Q_D}{\mu_D - n\lambda_D} + \frac{Q_M}{\mu_M} + \frac{Q_N}{\mu_N - n\lambda_N} + \frac{Q_P}{\mu_P}, \tag{9}$$

$$M_{SN} = \frac{Q_D}{\mu_D} + \frac{Q_M}{\mu_M} + \frac{Q_N}{\mu_N - n\lambda_N} + \frac{Q_P}{\mu_P}. \tag{10}$$

Здесь и далее индекс ξ будем опускать. Выражения для Q_D, Q_M, Q_N, Q_P для методов соединения NLJ и HJ приведены в табл. 2 [15]. При получении формул для HJ предполагалось, что хэш-таблица создающей таблицы A полностью сохраняется в оперативной памяти. Это сделано только для того, чтобы упростить выражения (в этом случае $\omega_A=\omega_B=0$).

Таблица 2

	NLJ	HJ
Q_D	$\frac{V_A}{n}\left(1+\frac{P_A V_B}{n}\right)$	$\frac{1}{n}(V_A + V_B)$
Q_P		$\frac{1}{n}(V_A + V_B + \frac{V_A P_A V_B P_B}{rn})$
Q_M	$\frac{2V_A}{n}\left(1+\frac{P_A V_B}{n}\right)$	$\frac{2}{n}(V_A + V_B)$
Q_N	$\frac{V_A P_A}{n^2}((n-1) + V_B P_B \eta_{AB})$	

Параметры, используемые в формулах табл.2, определены ранее. Прямой расчёт по формулам (8) и (9) затруднён, так как достаточно сложно одновременно оценить параметры λ_D и λ_M разделяемых ресурсов (диска и ОП). Но в системе из двух таких ресурсов, как правило, только один из них является "узким местом" (обычно это диск). Т. е. в среднем в нём накапливается требований намного больше, чем в другом разделяемом ресурсе. Наличие "узкого места" легко определить (табл. 3). В этом случае другой разделяемый ресурс можно считать неразделяемым. Тогда замкнутая модель ПСБД сводится к модели "ремонтника", которая при достаточно больших 'n' близка по характеристикам к СМО M/M/1 с ин-

тенсивностью входного потока nλ. Параметр λ легко рассчитывается (см. табл. 3 [15]). Расчётные формулы для математического ожидания представлены в последней колонке табл. 3. Эти формулы отличаются от выражений (8) и (9) тем, что для некритического разделяемого ресурса отсутствует произведение nλ, т.е. этот ресурс считается неразделяемым.

Таблица 3

	Условие наличия "узкого места"	"Узкое место"	Определяемый параметр λ	Формула для оценки λ	Формула для оценки математического ожидания (МО) времени выполнения соединения двух таблиц
SE	$\frac{Q_D}{\mu_D} >> \frac{Q_M + Q_N}{\mu_M}$	Диск	λ_D	$\frac{Q_D}{\frac{Q_M + Q_N}{\mu_M} + \frac{Q_P}{\mu_P}}$	$\frac{Q_D}{\mu_D - n\lambda_D} + \frac{Q_M + Q_N}{\mu_M} + \frac{Q_P}{\mu_P}$
	$\frac{Q_D}{\mu_D} << \frac{Q_M + Q_N}{\mu_M}$	Память	λ_M	$\frac{Q_M + Q_N}{\frac{Q_D}{\mu_D} + \frac{Q_P}{\mu_P}}$	$\frac{Q_M + Q_N}{\mu_M - n\lambda_M} + \frac{Q_D}{\mu_D} + \frac{Q_P}{\mu_P}$
SD	$\frac{Q_D}{\mu_D} >> \frac{Q_N}{\mu_N}$	Диск	λ_D	$\frac{Q_D}{\frac{Q_M}{\mu_M} + \frac{Q_N}{\mu_N} + \frac{Q_P}{\mu_P}}$	$\frac{Q_D}{\mu_D - n\lambda_D} + \frac{Q_M}{\mu_M} + \frac{Q_N}{\mu_N} + \frac{Q_P}{\mu_P}$
	$\frac{Q_D}{\mu_D} << \frac{Q_N}{\mu_N}$	Сеть	λ_N	$\frac{Q_N}{\frac{Q_D}{\mu_D} + \frac{Q_M}{\mu_M} + \frac{Q_P}{\mu_P}}$	$\frac{Q_D}{\mu_D} + \frac{Q_M}{\mu_M} + \frac{Q_N}{\mu_N - n\lambda_N} + \frac{Q_P}{\mu_P}$
SN	нет	Сеть	λ_N		

Расчёты по формулам, приведённым в табл.3, следует выполнять в следующей последовательности]15]:

1. Выбрать требуемую архитектуру – колонка 1.

2. Выявить "узкое место" системы (оно, как правило, присутствует, так как загрузки ресурсов невозможно сбалансировать) – колонки 2, 3.

3. Рассчитать параметр λ для "узкого места" – колонки 4, 5.

4. Если для "узкого места" выполняется неравенство (nλ/μ)≥1, то уменьшить число процессоров n и повторить пункт 4. Если это неравенство выполняется и для n=1, то считать данное архитектурное решение неудачным, так как в этом случае все требования будут ждать обслуживания в разделяемом ресурсе ("узкое место" очень медленное) и процессоры будут простаивать. Иначе перейти к пункту 5.

5. Рассчитать математическое ожидание времени выполнения запроса в ПСБД – колонка 6.

Пример расчёта среднего времени соединения таблиц

Расчёт математического ожидания (среднего) времени соединения таблиц в ПСБД [15] был выполнен при следующих значениях характеристик ресурсов.

1. Процессор – Intel Xeon 5160. СУБД – Oracle 9i. В [3, стр. 347] приведены результаты автотрассировки простого запроса соединения: число обработанных записей в двух таблицах – 640, число процессорных циклов - $7\ 10^6$. В [14] для выбранного процессора приведено измеренное значение числа процессорных циклов, выполняемых Oracle в секунду (CPUSPEED) – $1.5 \cdot 10^9$. Поэтому для интенсивности обработки записей в процессоре имеем – $\mu_P = 640/(7 \cdot 10^6/1.5 \cdot 10^9) = 1{,}3 \cdot 10^5$.

2. Внешняя память – RAID10 с $К_Д$=10 дисками 3.5" Seagate Cheetah 15K.6 ST3146356FC; размер блока чередования (stripe size) – $Q_{БЧ}$=256 Кб; среднее время поиска и чтения блока чередования с диска –
$t_{БЧ} = t_{подвода} + t_{вращения}/2 + Q_{БЧ}/v_{чтения} = 4 + 4/2 + 256/200 = 7$ мс.

Размер блока СУБД – $Q_{БС}$=16 Кб; средняя длина записи таблиц A и B – $l_З$=0.256 Кб, размер многоблочного чтения СУБД (переменная СУБД db_file_multiblock_read_count) - $q_{МБ}$=80. Поэтому для интенсивности чтения записей БД из массива RAID имеем –
$\mu_D = q_{МБ} \cdot Q_{БС}/l_З/(\lceil q_{МБ} \cdot Q_{БС}/Q_{БЧ}/(К_Д/2) \rceil \cdot t_{БЧ}) = 0.7 \cdot 10^6$. Число 2 в этой формуле учитывает тот факт, что при чтении полосы RAID-массива (т.е. при одновременном чтении блоков чередования с разных дисков) используется половина дисков массива, так как в RAID10 каждый диск из этой половины имеет зеркальную копию.

3. Оперативная память – DDR3-1600 PC3- 12800. Расчёты показывают, что интенсивность чтения записей базы данных из ОП равна $\mu_M = 10.4 \cdot 10^6$.

4. Высокоскоростная системная шина: для SD - Parallel Sysplex 12x QDR, для SN - ByNet. В обоих случаях узлы соединены по схеме "точка-точка" (т.е. шина является неразделяемым ресурсом) и интенсивность передачи записей по этим шинам равна $\mu_N = 50.3 \cdot 10^6$.

Далее приведены значения остальных параметров: число записей в таблицах A и B – $V_A = V_B = 10^6$, вероятность фильтрации записей таблиц A и B – $P_A = P_B = 10^{-3}$, вероятность успешного соединения двух записей – $\eta_{AB} = 10^{-4}$, количество хеш-групп в методе соединения HJ – r=10.

При расчётах использовались формулы из табл. 3 и 2. Для архитектур SE и SD "узким местом" является внешняя память (диск). При вычислениях величину '$n\lambda_D$' необходимо разделить на 2, так как при обработке запросов на чтение от разных процессоров может быть использована не только основная, но и зеркальная половина дисков. Для архитектуры SN произведение '$n\lambda_N$' необходимо исключить из формулы расчёта математического ожидания (т.е. обнулить '$n\lambda_N$' в

колонке 6 табл. 3 для строки SN), так как в рассматриваемом примере шина является неразделяемым ресурсом. Графики зависимости математического ожидания времени выполнения соединения двух таблиц от числа процессоров 'n' в параллельной системе баз данных приведены на рис. 2, 3 [15].

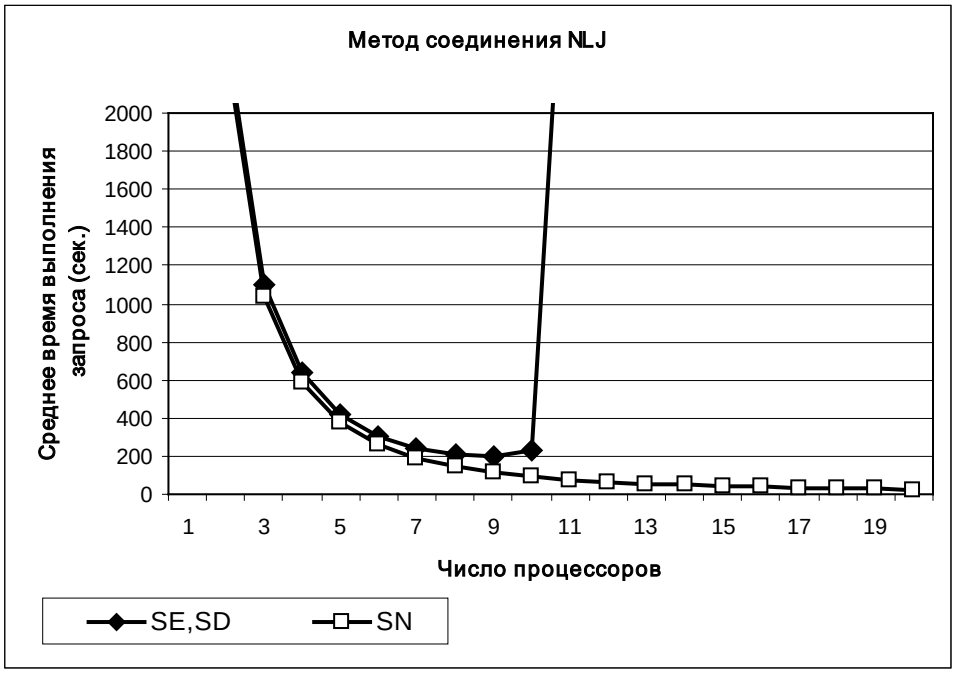

Рис. 2. Зависимость математического ожидания времени выполнения соединения таблиц в архитектурах SE, SD и SN методом NLJ от числа процессоров.

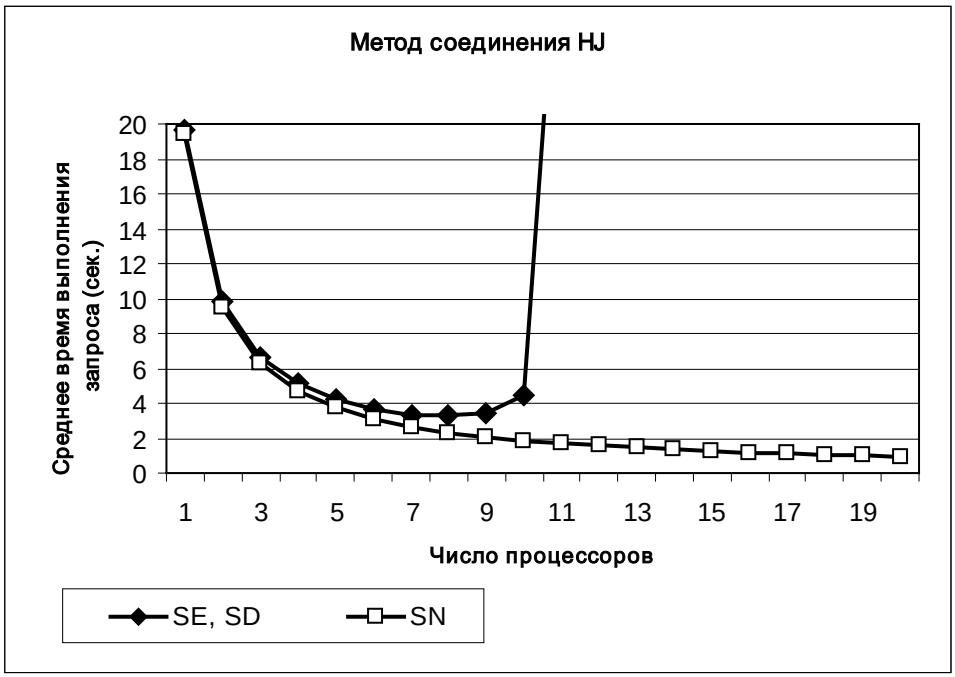

Рис. 3. Зависимость математического ожидания времени выполнения соединения таблиц в архитектурах SE, SD и SN методом HJ от числа процессоров.

Анализ графиков позволяет сделать несколько выводов [15]:

1. Графики для архитектур SE SD практически совпали. Это объясняется высокими значениями интенсивностей μ_M и μ_N.

2. При $n \geq 3$ среднее время для метода соединения HJ на два порядка меньше среднего времени для NLJ (это и следовало ожидать для неиндексированной по атрибуту соединения вложенной таблицы B). Более того, $M_{NLJ}(1) \approx 9400$ с., $M_{HJ}(1) \approx 19$ с.

3. При $n \leq 7$ архитектуры SE и SD не на много хуже SN. Следует также отметить, что при n=7 загрузка диска для SE и SD равна 0,63.

4. Для SE и SD при n=11 перегружается дисковая подсистема, и дальнейший рост числа процессоров не имеет смысла. Для архитектуры SN время продолжает уменьшаться с ростом n (здесь нет разделяемых ресурсов). Однако следует иметь в виду, что для снижения времени выполнения соединения методом HJ с 2 секунд до 1 необходимо увеличить количество процессов с 10 до 20 (см. рис. 3), а это может оказаться экономически не целесообразным.

Таким образом, в работе приведены выражения для математических ожиданий времени выполнения операции соединения для методов NLJ и HJ. Рассмотрен практический пример расчёта, позволивший сделать рад нетривиальных выводов.

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 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

electronic scientific and technical periodical
SCIENCE and EDUCATION
EL № FS 77 - 30569. №0421100025. ISSN 1994-0408

Estimation of mean time of tables connection by NLJ and HJ methods in parallel database system

77-30569/380603

 # 04, April 2012

Grigor'ev Yu.A., Gasov V.M.

Bauman Moscow State Technical University
chernen@bmstu.ru

Known architectures of parallel database systems are analyzed in this article. The authors also consider expressions for evaluating the mean time of connection of two tables by NLJ and HJ methods; relevant mathematical expectations for SE, SD, SN architectures of parallel database systems are also considered. The authors provide examples of query processing in the following resources: disk array, RAM, connecting bus and processor. If there is a bottleneck in the resource, processing in that resource can be simulated by using the M/M/1 queue system. Parameters of that queue system are evaluated. The example of calculation which allowed to make several non-trivial conclusions is considered in the article.

Publications with keywords: average of distribution time for query execution, parallel system of database, Laplase-Stieltjes transformation, connection of tables, SE, SD, SN architectures, NLJ and HJ methods of connection of tables
Publications with words: average of distribution time for query execution, parallel system of database, Laplase-Stieltjes transformation, connection of tables, SE, SD, SN architectures, NLJ and HJ methods of connection of tables

References

1. Tamer Ozsu M., Patrick Valduriez. *Distributed and parallel database systems*. (Russ. version: Tamer Ozzu M., Patrik Valduriz. *Raspredelennye 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://pdbs.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'iuis. *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/department/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 warehouse]. 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 raspredelennym 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 primenenie 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/Ispol'zovanie 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.