

РЕШЕНИЕ ЗАДАЧИ РАСПРЕДЕЛЕНИЯ ЗАПРОСОВ В МУЛЬТИКЛАСТЕРНОЙ СУБД БЕЗ СОВМЕЩЕНИЙ

Р.Ш. Минязев

Казанский национальный исследовательский технический университет

им. А.Н. Туполева

E-mail: txf13@mail.ru

Проводится сравнение альтернатив выбора алгоритма распределения запросов между кластерами-компонентами в мультикластерной СУБД без совмещений, обеспечивающего минимальное время реакции на вновь поступивший запрос. Рассматривается случай непрерывной работы абстрактной системы для различных нагрузочных характеристик при разном числе пользователей и количестве монокластеров.

Распределение запросов в мультикластерной СУБД

Серьезной проблемой для высокопроизводительных параллельных СУБД является балансировка нагрузки и размещение данных между узлами [1–3]. Рассматриваемые в данной работе мультикластерные СУБД консервативного типа, ориентированные на аналитическую обработку данных, показаны на рис. 1.

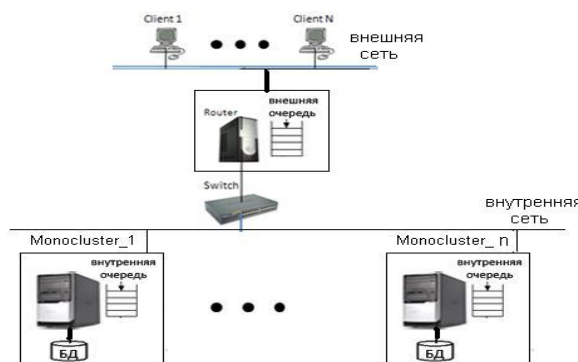


Рис.1. Мультикластерная СУБД

Host (главная ЭВМ) содержит модуль ROUTER, который распределяет поток запросов, поступающих от N клиентов – $Client_i$, $i \in \{1, N\}$, между ‘ n ’ кластерами-компонентами. Из этих двух составляющих на рисунке показан лишь ROUTER.

База данных реплицируется между монокластерами. Балансировка нагрузки в представленной системе осуществляется выбором такого способа распределения запросов между кластерами-компонентами, который обеспечивает *близкое к минимальному средне-статистическое время ожидания ответа на вновь поступивший запрос*.

По условию, входной поток запросов длины M , поступающий на обработку в систему, формируется случайным образом на множестве Q из 14 запросов теста ТРС-Н [4] без операций записи. Рассматривается случай, когда очередной запрос от каждого пользователя поступает на вход сервера СУБД не ранее, чем будет получен ответ на предыдущий запрос. Поэтому в любой момент времени суммарное число запросов, находящихся в сервере, $L \leq N$. В дальнейшем полагается $L = N$.

За основу рассмотрения принимается так называемый *круговой алгоритм распределений*. Суть его состоит в следующем.

Вводятся два вида очередей запросов: внутренних (в кластерах-компонентах) и внешней (в ROUTER). Суммарная длина всех внутренних очередей $\ell_n = k \cdot n \leq N$, $k \geq 1$. Длина внешней очереди $L - \ell_n \geq 0$. Рассматривается случай, когда первые ℓ_n запросов распределяются по кластерам-компонентам в порядке мест, занимаемых ими в обрабатываемой последовательности:

$$\begin{array}{ccccccc}
 \ell_{n-n+1} & \ell_{n-n+2} & - & - & - & \ell_n & \leftarrow k\text{-запросы} \\
 - & - & - & - & - & - & \\
 n+1 & n+2 & - & - & - & 2n & \\
 1 & 2 & - & - & - & n & \leftarrow 1\text{-запросы}
 \end{array}$$

В дальнейшем всякий раз пополняется та из внутренних очередей, которая первой завершила обработку своего 1-запроса. Считается, что в процессе работы системы значение ℓ_n неизменно и поровну распределяется между очередями. По окончании обработки любого 1-запроса освободившееся место занимает очередной запрос из внешней очереди. И в тот же момент на вход СУБД поступает новый запрос.

Круговой алгоритм является достаточно универсальным и применим в любых системах.

Применительно к круговому распределению задача состоит в нахождении такого значения k , которое при введенных ограничениях минимизирует в среднем время ожидания ответа на вновь поступивший запрос.

Обозначим t_{3r}^N – задержка ответа на вновь поступающий r -запрос ($r \in R = \{\overline{N+1}, M\}$), вычисляемая как разность времен появления результатов обработки r - и $(r-N)$ -запросов в каждом случае. По результатам вычислительного эксперимента для разных распределений будем получать графики временных рядов

$$\tau_{3r}^N(r) = t_{3r}^N(r) / M(t_{3r}^N),$$

варьируя число монокластеров n и количество пользователей N .

Средне-статистическое время ожидания ответа на вновь поступивший запрос будем оценивать величиной математического ожидания

$$M(t_{3r}^N) = (\sum_r t_{3r}^N) / m$$

($m < M$ – интервал измерений для исключения этапов установления стационарного режима работы системы и его завершения) и среднеквадратического отклонения

$$\sigma = \sqrt{M[t_{3r}^N - M(t_{3r}^N)]^2}$$

как характеристику релевантности оценки по M (оценка тем более релевантна, чем меньше σ).

Общее рассмотрение круговых распределений на примере абстрактной системы

Для определения оптимального способа распределения запросов между кластерами-компонентами в общем случае, независимо от используемой СУБД, смоделирована работа абстрактной системы при различных нагрузках, числе пользователей и количестве монокластеров. Для моделирования режима непрерывной работы обрабатывались запросные последовательности большой длины $M \geq 1000$. Количество сложных запросов в последовательности определяет нагрузочную характеристику.

Было выбрано подмножество $Q=14$ запросов из теста TPC-H, не содержащих операций INSERT и UPDATE. Для них получены времена обработки в монокластере реальной СУБД Clusterix [5], принятой за основу исследований (конфигурация «линейка» [5], $h=3$) для платформы НРС-кластера фирмы SUN. Времена обработки запросов, упорядоченные по возрастанию, представлены в табл. 1. Вес каждого запроса $p_i = t_i / t_{max}$, где t_i – время обработки i запроса в Clusterix, t_{max} – время обработки самого трудоемкого запроса.

Таблица 1. Относительные времена обработки запросов

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
t	10	11	12	22	23	24	37	41	42	43	44	46	50	105
p	0,09	0,10	0,11	0,21	0,22	0,23	0,35	0,39	0,40	0,41	0,42	0,44	0,48	1

На множестве выбранных запросов были сгенерированы три последовательности длины $M = 1000$ каждая, отвечающие трем законам распределения: равномерному, нормальному (параметр математическое ожидание $\mu = 7,5$; параметр дисперсия $\sigma^2 = 3,0$) и пуассоновскому (параметр математическое ожидание $\lambda = 10$). Разные законы отвечают различным нагрузочным характеристикам. Очевидно, что наиболее трудоемкой в обработке является запросная последовательность с пуассоновским законом. Из каждой сгенерированной последовательности запросов для анализа выбирался интервал $m = \{100, 900\}$. Сравнивались распределения запросов при $k \in \{1, 2, 3, 4, 5\}$.

За время обработки одного запроса принимается его вес p_i . Для любого монокластера в каждом такте работы системы вес его внутренней очереди уменьшается на 0,01.

Моделирование кругового алгоритма распределения запросов проводилось по алгоритму 1.

АЛГОРИТМ 1.

1. Очередные 'n' запросов последовательности распределить между внутренними очередями монокластеров в натуральном порядке, вес поступившего запроса добавить в массив типа вектор, ассоциированного с данной очередью.
2. $i := i+1$. Если $i \leq K$, идти к п. 1, иначе переход к п.3.
3. $TIME = TIME + 0,01$.
4. Уменьшить вес первого элемента каждого из n векторов на 0,01.
5. Если вес какого-либо элемента равен нулю, то переход к п.6 иначе переход к п.3.
6. Вывод в файл значения TIME и номера запроса, соответствующего элементу с нулевым весом исключение его из вектора.
7. $M = M+1$. Если $M=1000$, то КОНЕЦ.
8. Если все запросы из последовательности распределены то переход к п.5, иначе к п.9.
9. Чтение очередного запроса из последовательности, его вес добавить в конец уменьшенного вектора.

Полученные графики временных рядов

$$\tau_{3r}^N(r) = t_{3r}^N(r) / M(t_{3r}^N)$$

имеют примерно одинаковый вид для всех рассмотренных случаев независимо от закона распределения запросов в последовательности. На рис. 2 представлены графики для кругового алгоритма распределения запросов ($k = 1$) при равномерном распределении. Варьировалось число монокластеров n и количество пользователей N .

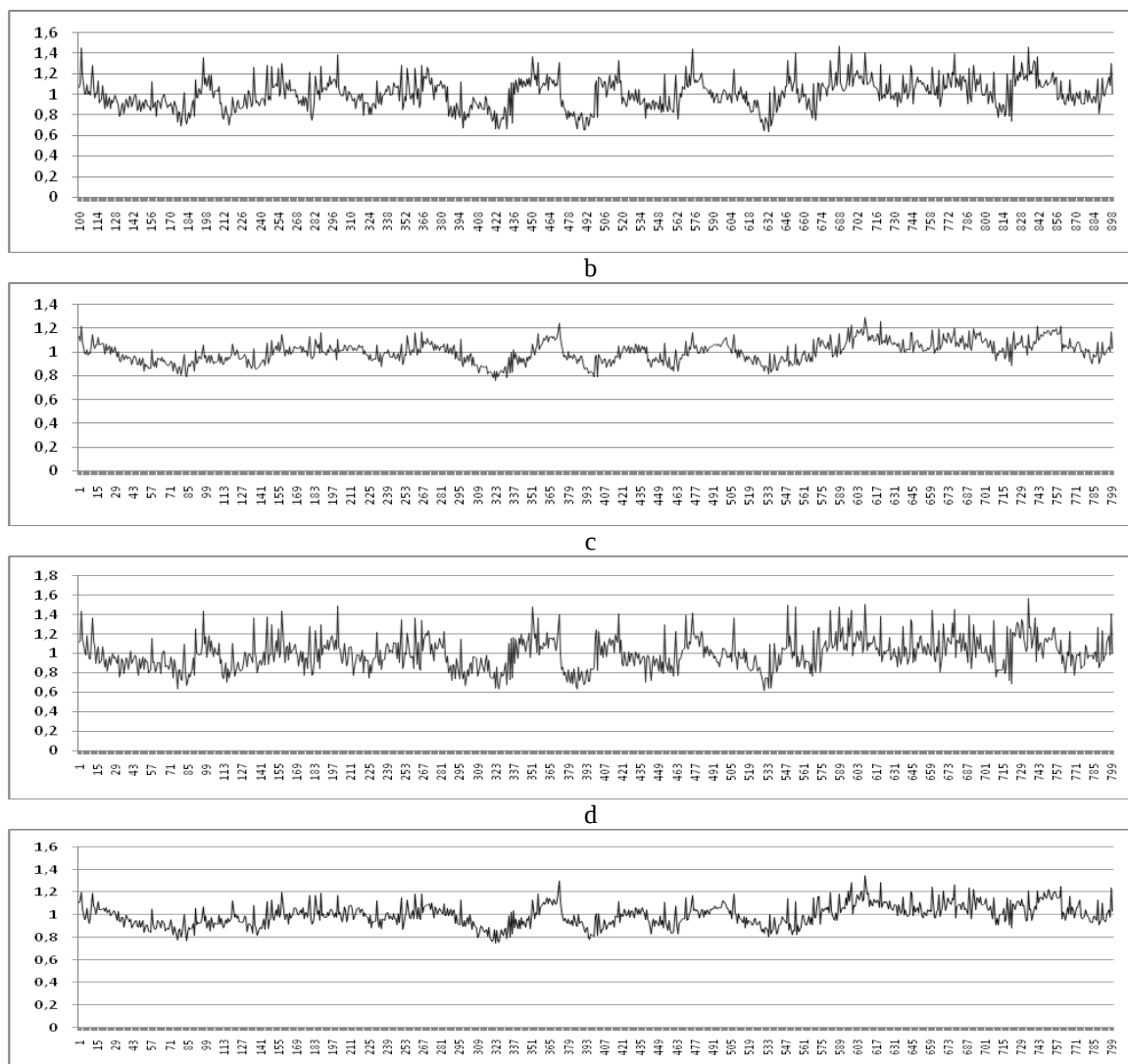


Рис. 2. Графики $\tau_{3r}^N(r)$: a $\rightarrow n=3, N=20$; b $\rightarrow n=3, N=40$; c $\rightarrow n=4, N=20$; d $\rightarrow n=4, N=40$

Полученные значения для M и σ представлены соответственно в табл. 2, 3.

Таблица 2

$N=20, n=3(M)$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
равномерный	2,491663	2,489938	2,493	2,491188	2,49315
нормальный	2,146663	2,147013	2,146813	2,1461	2,14805
пуассоновский	2,992825	2,992588	2,992913	2,992425	2,993288

Таблица 3

$N=20, n=3(\sigma)$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
равномерный	0,365376	0,410244	0,444443	0,494686	0,5336
нормальный	0,212456	0,23214	0,256929	0,290755	0,298471
пуассоновский	0,288544	0,317992	0,351952	0,390832	0,427957

Для оценки влияния изменения числа кластеров n (рис. 4) и числа пользователей N (рис. 3) на распределение времен задержек ответов на запросы определен коэффициент вариации, показывающий степень рассеивания величин относительно среднего, для различных распределений: $V = \sigma/M$, для случая кругового распределения при $k = 1$.

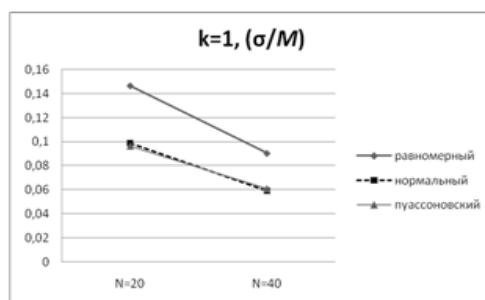


Рис. 3.

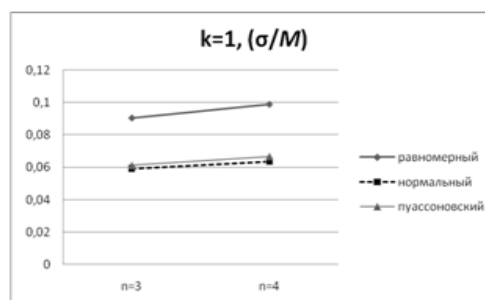


Рис. 4.

Заключение

Из табл. 2 видно, что среднее значение задержки во всех случаях примерно одинаковое. При этом отклонение значений задержек от среднего минимально для случая $k = 1$, независимо от закона распределения запросов в обрабатываемой последовательности (см. табл. 3).

Было выявлено, что увеличение числа пользователей N приводит к выравниванию задержек относительно среднего значения (рис. 3). Увеличение же числа монокластеров n , приводит к увеличению разброса значений задержек относительно среднего (рис. 4).

Литература

1. Xu Y., Kostamaa P., Zhou X., Chen L. Handling data skew in parallel joins in shared-nothing systems // ACM SIGMOD international Conference on Management 9of Data Canada, 2008, proceedings. ACM. 2008. P. 1043–1052.
2. Лепихов А.В. Параллельная обработка запросов в СУБД для кластерных вычислительных систем // Отчет в рамках гранта МК-3535.2009.9.
3. Гудзенко В.В. Интеллектуальный агент распараллеливания запросов // Проблемы программирования. 2006. N 2. С. 192–196.
4. <http://www.tpc.org>
5. Минязев Р.Ш., Райхлин В.А. Балансировка нагрузки в мультикластерных СУБД консервативного типа на Beowulf-платформе // Вестник КГТУ им. А.Н. Туполева. 2011. №1. С. 52–57.