

В. Б. Ефлов¹, канд. физ.-мат. наук, доц. кафедры, e-mail: veflov@gmail.com,

Е. А. Питухин¹, д-р техн. наук, проф., зав. каф., e-mail: eugene@petrsu.ru,

Ф. В. Стафеев², консультант, e-mail: fedor.stafeev@gmail.com

¹ГОУ ВПО "Петрозаводский государственный университет (ПетрГУ)"

²Администрация Петрозаводского городского округа

Вопросы производительности при выполнении запросов в реляционной СУБД H2

Исследованы статистические характеристики времени выполнения запросов, выявлены особенности работы режимов кэширования данных. С помощью нелинейной аппроксимации данных получена обобщенная модель зависимости времени выполнения запросов от объема кэш-памяти и размера запрашиваемых таблиц. Введен безразмерный показатель, на основе которого предложен критерий реализации в СУБД H2 режима эффективного кэширования. Результаты оценки эффективности кэширования могут быть использованы на практике для существенного снижения времени выполнения запросов.

Ключевые слова: оптимизация запросов, СУБД, кэширование, H2, модель, время исполнения запросов

Введение

Для осуществления полномочий современных органов местного самоуправления используется широкий спектр инструментов информационно-коммуникационных технологий (ИКТ). Одной из ключевых составляющих в подобном комплексе является применение СУБД в различных информационных системах. В силу ряда причин наиболее распространены в данной сфере реляционные СУБД, подчас являющиеся свободным либо свободно распространяемым программным обеспечением.

Несмотря на развитые возможности оптимизаторов таких продуктов, при промышленной эксплуатации информационных систем повсеместно встает вопрос о производительности модулей выполнения запросов. В повседневной практике запросы выборки данных, выраженные в случае языка программирования SQL посредством оператора SELECT, имеют большую значимость в силу частоты использования перед запросами других типов. Традиционные методы сокращения времени выполнения запросов, основанные на перезаписи SQL-выражений, не всегда успешны, так как администратор информационных систем подчас не имеет доступа к программному коду и не имеет возможности вносить какие-либо изменения. Также этот метод имеет точечный характер и наиболее успешен в ситуации, когда в запросе участвуют соединения нескольких отношений [1].

В представленной работе описана методика оценки производительности модуля выполнения запросов свободно распространяемой реляционной СУБД H2 при варьировании параметров аппаратно-программной платформы и предикатов SQL-выражений. Исследованы статистические характеристики времени выполнения запросов, выявлены особенности работы режимов кэширова-

ния данных. С помощью нелинейной аппроксимации экспериментальных данных получена обобщенная модель зависимости времени выполнения запросов от объема кэш-памяти и размера запрашиваемых таблиц. Введен безразмерный показатель относительного объема кэша в виде отношения объема кэша к объему базы данных, на основе которого предложен критерий реализации в СУБД H2 режима эффективного кэширования. Полученные результаты оценки эффективности кэширования могут быть использованы на практике для существенного снижения времени выполнения запросов.

Данная работа является продолжением серии исследований производительности некоммерческих реляционных СУБД [2—4]. В более ранней статье [2] не был затронут ряд вопросов, выявленных при работе с экспериментальной платформой на основе H2 Database. H2 Database — реляционная СУБД, созданная и поддерживаемая сообществом разработчиков (<http://www.h2database.com>).

Описание эксперимента

Результатом проведенных экспериментов стал массив данных о времени выполнения запросов, выраженных посредством оператора SELECT, к экспериментальной базе данных H2 Database при варьировании ряда параметров: объема оперативной памяти платформы, размера используемой СУБД кэш-памяти, числа кортежей в таблицах, к которым выполняется обращение в запросах, типов полей предикатов, значения идентификаторов уникальных ключей таблиц [5]. Результаты эксперимента были представлены в виде выборок зависимостей времени выполнения запросов от номера идентификатора первичного ключа таблиц экспериментальной базы данных (далее — пакет запросов) (рис. 1). Номер идентификатора первичного ключа таблиц выражен как относительный от общего

числа записей в таблицах коэффициент и варьировался в диапазоне от 1 до 40. В общей сложности были получены и подвергнуты анализу более 5000 экспериментальных выборок.

Следует отметить, что данные времен, полученных при выполнении пакета запросов, отличаются первыми значениями, которые обычно имеют большую величину (рис. 1, а). Для исследований в ряде случаев использовался набор данных без первого значения (рис. 1, б), например для оценки периодичности [2].

Такая большая разница в значениях между первым и последующим запросами к одной и той же таблице может иметь следующее объяснение. При включении механизма кэширования во время выполнения первого запроса вначале происходит чтение таблицы с жесткого диска и загрузка ее в ОЗУ. При последующих запросах к таблице данные загружаются из ОЗУ и обращения к диску уже не происходит. В связи с этим время выполнения последующих запросов, по сравнению с первым, сокращается на несколько порядков.

При этом возникает проблема оценки эффективности кэширования, которая будет рассмотрена ниже. Целесообразно определить, какой объем кэш-памяти необходимо использовать при кэшировании, чтобы время выполнения первого и последующих запросов существенно различалось, как, например, на рис. 1, а.

Статистический анализ экспериментальных данных

Для повышения достоверности полученных в ходе экспериментов результатов была проведена дополнительная серия измерений с выборочными значениями варьируемых в основном эксперименте параметров. Для каждой из комбинаций значений параметров время выполнения запроса измерялось десятикратно. Запросы выполнялись для двух таблиц из списка тестовых таблиц. Определено, что из 3214 уникальных запросов только для 35 выборок коэффициент вариации был больше 33 %. Таким образом, экспериментальные выборки можно считать однородными.

Исследовались зависимости времени выполнения запроса от размера кэша при одном объеме оперативной памяти (230 Мбайт) для таблиц двух разных объемов — 10 000 и 500 000 строк.

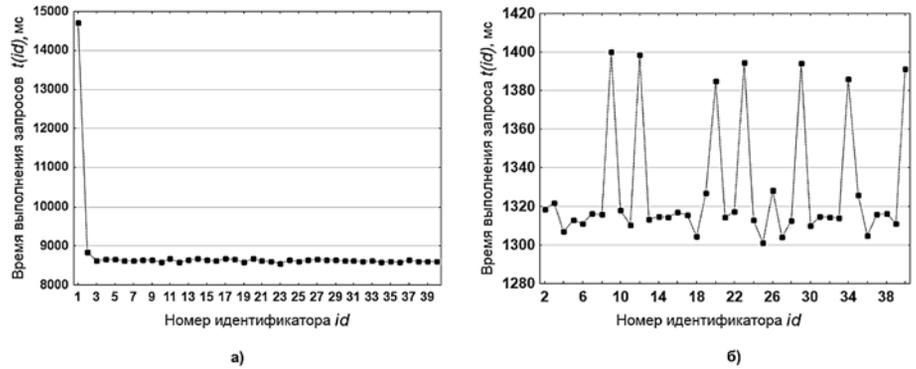


Рис. 1. Экспериментальная зависимость времени выполнения запросов от номера идентификатора первичного ключа $t(id)$: а — с первым значением; б — без первого значения

Статистические характеристики времени выполнения запроса для таблицы размером 10 000 строк от объема кэша приведены на рис. 2, а. Из графиков видно, что эффективная работа кэша начинается после 6000 кбайт, после чего падение среднего времени выполнения запроса составляет более 16 раз (218,2 с против 13,0 с), а падение минимального времени выполнения запроса достигает 30 раз! Максимальное время запроса при этом остается флуктуировать близко к уровню, когда кэш отсутствует или его малый размер не оказывает значительного эффекта. Среднее относительное отклонение среднего времени выполнения запроса от минимального составляет 36,5 %.

Иная картина получается при анализе работы кэша на таблице большего размера — 500 000 строк. На рис. 2, б также приведены среднее, максимальное и минимальное время выполнения запросов в зависимости от размера кэша. На рис. 2, б видно, что устойчивое падение времени выполнения запроса начинается со значений кэша от 1000 кбайт и более.

Минимальное время выполнения запроса отличается от среднего незначительно: на всем промежутке среднее относительное отклонение не превышает 2,1 %. Максимальное время выполнения запроса представляет собой характерный выброс в начале кэширования и не оказывает значительного влияния на среднее значение. Среднее время вы-

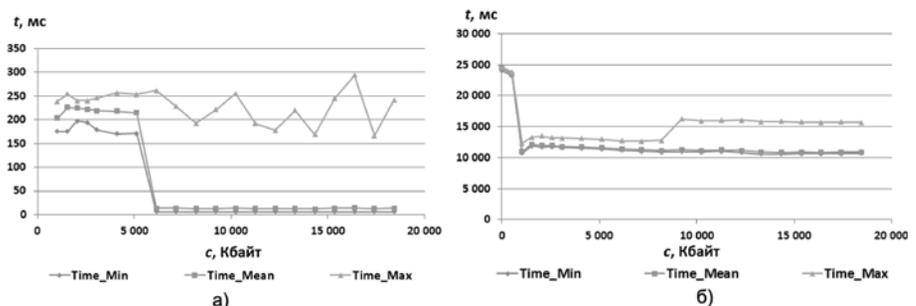


Рис. 2. Зависимость характеристик времени выполнения запроса от величины кэша: а — размер таблицы 10 000 строк; б — размер таблицы 500 000 строк

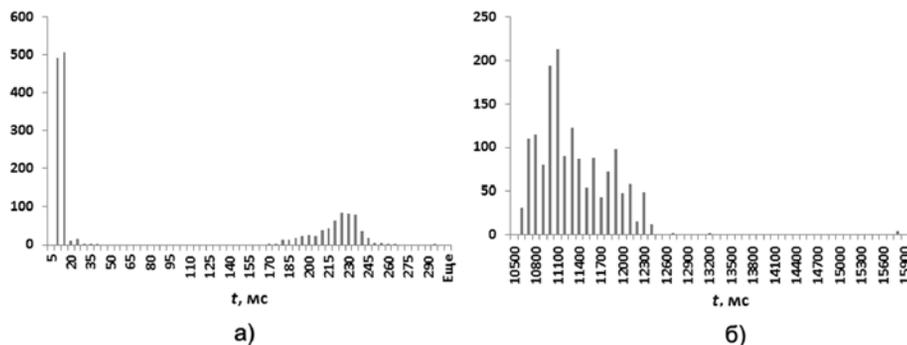


Рис. 3. Гистограмма частот времени выполнения запросов:
a — размер таблицы 10 000 строк, *б* — размер таблицы 500 000 строк

полнения запроса после начала работы кэша с эффективным значением более 1000 кбайт падает более чем в 2 раза (23 630 мс против 11 060 мс).

Несмотря на снижение среднего времени выполнения запроса более чем в 2 раза, эффективность кэширования данных для таблицы размером 500 000 значительно меньше, чем для таблицы размером 10 000 строк (16 раз). Это может объясняться тем, что для такой большой таблицы использованные в эксперименте вариации объема кэша оказались недостаточными, чтобы кэширование перешло по-настоящему в эффективный режим.

На рис. 3, *a* и *б* приведены гистограммы частот времени выполнения запросов для таблиц размером 10 000 и 500 000 строк соответственно.

Рис. 3, *a* отображает бимодальный характер плотности распределения времени выполнения запросов для таблицы размером 10 000 строк. Хорошо различимы две моды, разделенные значительным расстоянием, соответствующие "быстрым" и "медленным" запросам. При дополнительной детализации "быстрых" запросов выяснилось, что они также характеризуются бимодальным распределением, состоящим из двух непересекающихся нормальных распределений. Представленная на рис. 3, *б* плотность распределения времени выполнения запросов для таблицы размером 500 000 строк, в от-

личие от рис. 3, *a*, составляет единую картину.

Результаты проведенного анализа показывают, что при использовании в СУБД функции кэширования ее эффективность может быть различной.

Данные факты позволяют сделать заключение о существовании двух режимов кэширования при использовании кэш-памяти в процессе выполнения запросов — "эффективного" и "неэффективного" (на порядки отличающихся по эффективности в плане срав-

нения времен выполнения запросов). Под "неэффективным" кэшированием будем понимать такой режим, когда функция кэша включена, но кэширование фактически не происходит из-за малого объема кэш-памяти.

В связи с этим необходимо провести дополнительное исследование эффективности кэширования на различных режимах.

Оценка эффективности кэширования

Как было отмечено в [2], зависимость $t(id)$ принимает вид, приведенный на рис. 1, *a*, в случае участия механизма кэширования данных при первом обращении к запрашиваемой таблице. В выборках, полученных при отключенной функции кэширования, первое значение также больше среднего, однако разница не превышает 20...30 %.

Представляется важным проанализировать условия, при которых наступает режим "эффективного" кэширования, когда время выполнения запросов падает в десятки раз.

Для этого по результатам экспериментов было вычислено среднее время выполнения запросов, содержащихся в одном пакете, за исключением первого запроса (см. рис. 1, *б*). В каждом пакете содержались запросы типа SELECT к определенной

Время выполнения запросов от объема кэша и размера таблиц в случае использования функции кэширования, мс

		Число строк таблицы										
		1000	3000	5000	10 000	30 000	50 000	80 000	100 000	200 000	300 000	400 000
Размер кэша, кбайт	2048	0,87	46,49	74,47	152,18	443,58	745,14	1256,74	1495,35	3079,65	4961,83	7537,06
	4096	1,03	2,59	3,91	163,24	477,94	789,87	1329,79	1568,01	3237,35	5180,99	8059,73
	8192	1,09	2,42	3,88	7,53	467,59	784,60	1317,74	1573,55	3154,13	5094,82	7964,23
	12 288	0,92	2,76	4,15	7,78	453,72	749,41	1256,60	1495,82	3121,82	4999,51	7612,15
	20 480	0,92	2,56	3,91	7,65	21,78	789,77	1322,23	1572,47	3249,23	5176,44	8017,19
	24 576	1,03	2,67	4,01	7,65	21,96	823,06	1378,18	1640,90	3402,40	5416,79	8236,40
	30 720	0,96	2,71	4,15	7,90	22,32	37,03	1514,38	1800,53	3675,22	5823,85	8819,88
	40 960	1,06	2,50	3,92	7,99	22,77	36,83	1711,08	2058,26	4204,81	6652,35	9902,96
	51 200	1,01	2,62	4,09	7,90	22,69	36,85	61,72	3996,27	7874,94	11590,32	16507,67
	57 344	0,95	2,64	4,36	8,05	22,54	36,62	63,79	80,05	19506,95	29462,72	40091,90

таблице с предикатом поля с заданным типом данных. Среднее время выполнения запросов рассчитывалось для каждого из типов данных полей предикатов.

Пример зависимости среднего времени выполнения запросов от объема кэша и размера таблиц приведен в таблице. Запросы осуществлялись по предикату целочисленного поля для зафиксированного значения оперативной памяти из середины диапазона вариации.

Данные в правой части таблицы соответствуют режиму "неэффективного" кэширования (выделено курсивом). Левая часть таблицы содержит значения времени выполнения запроса (выделено жирным) в случае, когда кэширование "эффективное". Это можно заметить при анализе столбцов таблицы сверху вниз — в среднем время выполнения запросов по сравнению с "неэффективным" режимом снижается более чем в 22 раза. Для таблиц больших размеров перехода из режима "неэффективного" кэширования в "эффективный" режим не наблюдается, что объясняется, очевидно, ограниченным экспериментом диапазоном изменения кэш-памяти. При этом отмечается обратный эффект, когда в режиме "неэффективного" кэширования с увеличением кэш-памяти время выполнения запросов не снижается, а, наоборот, лавинообразно возрастает.

На основе данных из таблицы была получена зависимость времени выполнения запросов от объема данных запрашиваемой таблицы для различных значений кэш-памяти (рис. 4). При этом в качестве значения оси ординат выбирались значения времени выполнения запросов, соответствующих режиму "эффективного" кэширования.

Очевидно, что полученные экспериментальные зависимости (рис. 4) хорошо аппроксимируются линейной функцией вида

$$t = a_1 s, \quad (1)$$

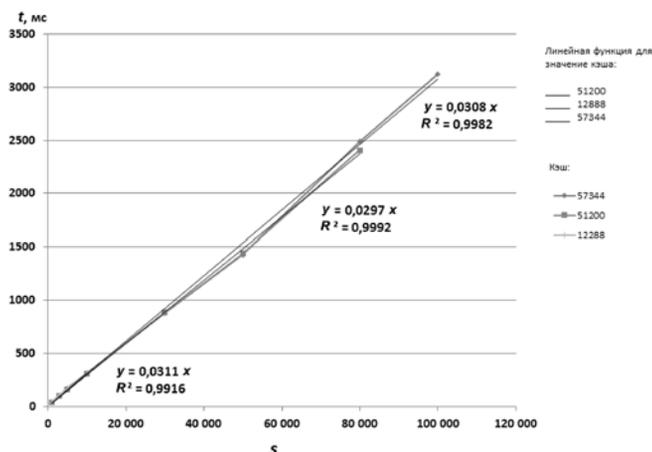


Рис. 4. Зависимость времени выполнения запросов от объема данных запрашиваемой таблицы в случае использования в СУБД "эффективного" кэширования

где t — время выполнения запроса; s — размер таблицы (число строк); a_1 — коэффициент линейной регрессии.

При этом угол наклона аппроксимирующих прямых в режиме "эффективного" кэширования остается постоянным для любых значений кэша, что свидетельствует о том, что время запроса преимущественно зависит только от размера таблицы.

В ходе эксперимента выполнялись запросы с предикатами по различным типам полей. Влияние этого фактора на вид зависимости (рис. 4) не отмечалось. Это вполне предсказуемо, если принять за факт предположение о том, что кэшируются, по всей видимости, все данные запрашиваемой таблицы.

Графики времени выполнения запросов из таблицы для всех режимов кэширования представлены на рис. 5 (см. третью сторону обложки) в логарифмическом масштабе из-за сильно отличающихся значений. Они представляют собой зависимости логарифма среднего времени выполнения запроса от размеров таблицы при постоянном значении кэша.

Приведенные графики обнаруживают интересную особенность. Поведение времени выполнения запроса имеет нелинейность в виде скачкообразного перехода от режима "эффективного" кэширования к режиму "неэффективного" кэширования, когда размер таблицы увеличивается выше некоторого критического уровня для заданного объема кэша.

На рис. 5 (см. третью сторону обложки) в результате наложения графиков сформировались асимптоты, соответствующие процессам "эффективного" и "неэффективного" кэширования. Среднее отношение времени выполнения запросов между двумя асимптотами составляет более 22 раз, что доказывает существенное различие между эффективностью указанных режимов.

Для выявления особенностей работы режимов кэширования данных рассмотрим детальнее таблицу. По диагонали таблицы проходит граница между "эффективным" и "неэффективным" режимами кэширования, ячейки которой обведены рамкой. Они соответствуют минимальному размеру кэша для соответствующего размера таблицы, при котором кэширование переходит в "эффективный" режим.

Результаты анализа "перехода" между процессами "эффективного" и "неэффективного" кэширования в относительном виде отображены на рис. 6. На графике представлена экспериментальная кривая, показывающая процентное отношение объема кэша к объему таблицы, при превышении которого таблица соответствующего размера начнет "эффективно" кэшироваться. Объем таблицы v (кбайт) рассчитывался исходя из ее размера (числа строк) по формуле

$$v = \frac{l_1}{1024} s,$$

где l_1 — средний размер строки таблицы в байтах ($l_1 = 94$).

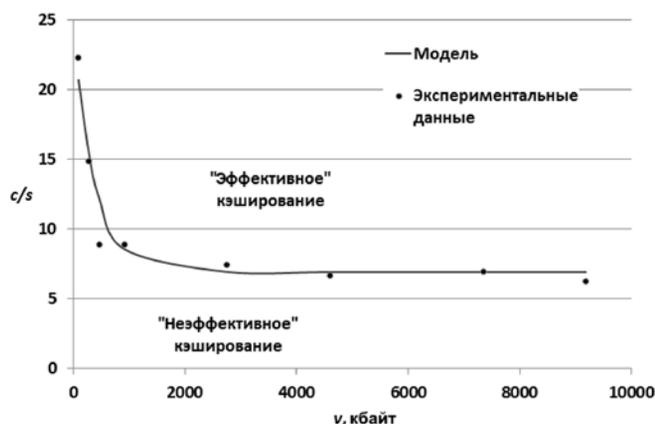


Рис. 6. Критические области режимов "эффективного" и "неэффективного" кэширования для таблиц различных размеров в зависимости от объема кэша

На рис. 6 видно, что при малых объемах таблицы объем кэша составляет до 20 раз от ее объема, но, при увеличении размера таблицы, для обеспечения "эффективного" кэширования доля кэша должна более чем в 7 раз превосходить объем запрашиваемой таблицы.

Проведенный анализ показывает, что для обеспечения режима "эффективного" кэширования и, соответственно, малых времен выполнения запросов размер кэша следует выбирать из соображений, чтобы соответствующее соотношение "объем кэша/объем таблицы" располагалось над указанной на рис. 6 кривой.

Данное наблюдение позволяет ввести понятие показателя относительного объема кэша в виде отношения объема кэша к объему запрашиваемой таблицы (базы данных) в безразмерном виде:

$$q(c, v) = c/v. \quad (2)$$

С учетом формулы (2) можно предложить критерий реализации в СУБД Н2 режима "эффективного" кэширования, когда значение показателя относительного объема кэша $q(c, v)$ должно быть больше некоторого критического значения q_{cr} :

$$q(c, v) > q_{cr}. \quad (3)$$

Анализ поведения критического значения показателя q_{cr} в зависимости от размера таблицы s позволит определить границу перехода между режимами "эффективного" и "неэффективного" кэширования.

На рис. 6 приведена кривая аппроксимации экспериментальных значений показателя q_{cr} экспоненциальной зависимостью вида

$$q_{cr}(v) = b_0 + b_1 e^{b_2 v}, \quad (4)$$

где b_0, b_1, b_2 — коэффициенты экспоненциальной регрессии. Значения коэффициентов регрессии $b_0 = 6,933, b_1 = 17,441, b_2 = -2,581 \cdot 10^{-3}$. Коэффициент детерминации $R^2 = 0,986$ и средняя относи-

тельная ошибка в процентах $MAPE = 8\%$ свидетельствуют об удовлетворительной точности аппроксимации.

Моделирование времени выполнения запросов

Значимым этапом исследования является построение обобщенной модели зависимости времени выполнения запросов от влияющих факторов, которая позволит сделать выводы о характере процессов и способах увеличения производительности СУБД Н2.

Для аппроксимации экспериментальных данных использовались полиномиальные функции различных степеней. Хорошие результаты были получены в случае полинома 2-, 3-й степеней, при котором значение коэффициента детерминации R^2 лежало в пределах $[0,9...0,95]$.

В общем случае для разных типов предикатов были получены следующие выражения зависимостей времени выполнения запроса t от размера кэша c , размера оперативной памяти r и размера запрашиваемых таблиц s :

вещественный тип:

$$t(s, c, r) = 2,352s + 0,002c^2 - 19,803c - 0,002r^3 + 0,813r^2 + 78,037r - 0,011rs + 6591,710;$$

целочисленный тип:

$$t(s, c, r) = 2,53s - 16,89c + r^2 - 0,01rs + 12539,4;$$

текстовой тип:

$$t(s, c, r) = 2,417s + 0,003c^2 - 45,761c + 0,006r^3 - 3,826r^2 + 1084,718r - 0,01rs + 7173,73;$$

предикат вида LIKE:

$$t(s, c, r) = 2,46s - 16,44c + 1,08r^2 - 50,25r - 0,01rs + 17183,37;$$

предикат по первичному ключу:

$$t(s, c, r) = 0,0024c + 0,0029r^2 - 1,1031r + 134,677.$$

Полученные модели статистически хорошо описывают исследуемые процессы, но в силу своего полиномиального вида имеют недостатки. Полученные коэффициенты при степенях имеют знаки, которые сложно интерпретировать с позиции физического смысла процессов. В некоторых случаях, при аппроксимации малых значений времен запросов, функция $t(s, c, r)$ может принимать отрицательные значения.

Возникает задача построения полнофункциональной математической модели, отражающей характер приведенных процессов с учетом вышеуказанных недостатков.

В целях построения обобщенной модели, позволяющей провести графическую интерпретацию процессов, из рассмотрения будет исключен фактор размера оперативной памяти r . Аргументом для

его исключения служит то обстоятельство, что скорее он является ограничением, нежели фактором влияния. Оперативной памяти может либо хватить для полноценной работы СУБД, либо не хватить. Попытки исследовать влияние пограничного состояния размера оперативной памяти на скорость работы СУБД не имеют большого практического значения.

За основу для построения модели были взяты данные из таблицы.

Наличие скачкообразного перехода между режимами "эффективного" и "неэффективного" кэширования обосновывает проведение аппроксимации соответствующих режимов различными функциями. Таким образом, итоговая модель будет состоять из двух функций, связанных условием перехода.

Для аппроксимации режима "эффективного" кэширования (см. рис. 4) использовалась ранее установленная линейная зависимость времени выполнения запросов от объема данных запрашиваемой таблицы вида (1):

$$t_{ef}(v) = a_1 v, \quad (5)$$

где коэффициент линейной регрессии $a_1 = 8,401 \cdot 10^{-3}$ и средняя относительная ошибка в процентах $MAPE = 8,5 \%$.

При выборе аппроксимирующей функции для режима "неэффективного" кэширования соответствующие данные таблицы были подвергнуты тщательному анализу, в результате которого выяснилось, что наилучшим образом они описываются нелинейной зависимостью вида

$$t_{nef}(c, v) = v(d_0 + d_1 c + d_2 e^{d_3 c}), \quad (6)$$

где коэффициенты нелинейной регрессии $d_0 = 0,189$, $d_1 = 9,653 \cdot 10^{-7}$, $d_2 = 1,437 \cdot 10^{-6}$, $d_3 = 2,315 \cdot 10^{-4}$. При этом коэффициент детерминации $R^2 = 0,929$ и средняя относительная ошибка в процентах $MAPE = 11,1 \%$.

Из формул (2)—(4) следует, что критерием перехода из "неэффективного" режима в "эффективный" может служить выполнение неравенства

$$c/v > b_0 + b_1 e^{b_2 v}. \quad (7)$$

Используя формулы (5)—(7), запишем итоговую обобщенную математическую модель зависимости времени выполнения запросов от размера кэша c и объема запрашиваемых таблиц s с учетом текущего режима кэширования:

$$t(c, v) = \begin{cases} a_1 v, & \text{если } c/v > b_0 + b_1 e^{b_2 v}; \\ v(d_0 + d_1 c + d_2 e^{d_3 c}), & \text{иначе.} \end{cases} \quad (8)$$

Полученная модель (8) показывает, что время выполнения запросов имеет линейный характер в

режиме "эффективного" кэширования и носит экспоненциальный характер в режиме "неэффективного" кэширования. При этом $t(c, v)$ будет всегда положительно, будет равняться нулю в случае таблицы нулевого объема ($v = 0$) и будет продолжать экспоненциально расти от размера таблицы при нулевом кэше ($c = 0$).

На рис. 7 (см. третью сторону обложки) приведена зависимость времени выполнения запросов от размера кэша c и объема запрашиваемых таблиц v . Экспериментальные данные отображены точками, а аппроксимирующая поверхность (8) представлена контуром.

На аппроксимирующей поверхности хорошо виден разрыв, соответствующий границе между "эффективным" и "неэффективным" режимами кэширования.

Кроме того, из рис. 7 видно, что в "неэффективном" режиме кэширования с увеличением объема кэша время выполнения запросов начинает экспоненциально расти, хотя было ожидаемо, что оно начнет снижаться. Для объяснения этого факта необходимо проводить дополнительные углубленные исследования механизма кэширования, что выходит за рамки данной статьи.

Заключение

В заключение отметим, что проведенные исследования указывают на довольно значительное влияние кэширования на процесс выполнения запросов, особенно на таблицах относительно небольшого объема. Несмотря на наблюдаемый скачкообразный характер влияния кэша на время выполнения запроса в исследуемой СУБД, вызывает интерес тип зависимостей времени выполнения запросов, проиллюстрированных на рис. 1—7. Выяснилось, что при использовании функции кэширования сам процесс кэширования может осуществляться как в "эффективном", так и в "неэффективном" режиме. Полученная в результате аппроксимации обобщенная модель зависимости времени выполнения запросов от размера кэша и объема запрашиваемых таблиц позволяет, по нашему мнению, провести прогнозирование времени выполнения запросов при детерминированных условиях для любого режима работы кэша. Результаты моделирования обнаруживают, что важным, с практической стороны, является обеспечение работы СУБД с использованием функции кэширования в "эффективном" режиме, когда выделяемого объема кэш-памяти достаточно для обеспечения оперативной работы системы. В качестве критерия обеспечения такого режима можно использовать превышение предложенного безразмерного показателя относительно объема кэша некоторого критического уровня.

Список литературы

1. **Ефлов В. Б., Питухин Е. А., Стафеев Ф. В.** Анализ выполнения запросов в СУБД H2 // Ученые записки Петрозаводского государственного университета. Сер.: Естественные и технические науки. 2013. № 2 (131). С. 62—65.
2. **Питухин Е. А., Стафеев Ф. В.** Анализ методов оптимизации запросов в реляционных СУБД. М., 2010. Деп. ВИНТИ 16.03.2010, № 167-B2010.
3. **Стафеев Ф. В.** Способы оценки эффективности оптимизаторов запросов в реляционных СУБД // Роль науки в устой-

чивом развитии общества: сборник материалов 2-й Международной научно-практической конф. Тамбов: Тамбовпринт, 2010. С. 150—151.

4. **Стафеев Ф. В.** Методы исследования процесса обработки запросов в реляционных СУБД // Современные инструментальные системы, информационные технологии и инновации: материалы VIII Международной научно-технической конференции. Курск, 2011. С. 250—252.

5. **Tutorial.** H2 Database Engine. 2013. URL: <http://www.h2database.com/html/tutorial.html>

V. B. Yeflov¹, Associate Professor, e-mail: veflov@gmail.com,

E. A. Pitukhin¹, Professor, e-mail: eugene@petsru.ru,

F. V. Stafeev², Consultant, e-mail: fedor.stafeev@gmail.com

¹Petrozavodsk State University

²Petrozavodsk City Administration

Performance Issues of Query Execution in Relational H2 Database

To exercise the powers of modern local authorities a wide range of ICT tools is used. A key component is the use of DBMS in various information systems. Despite the long-term development of optimizers of DBMS, in industrial information systems operation an issue of improving the performance of the query modules is becoming quite topical.

To achieve this goal it is necessary to solve the problem of development of performance assessment methods of the query execution modules of a freely distributable relational DBMS H2 in case of varying parameters of hardware and software platforms and predicates of SQL expressions.

The statistical characteristics of the query execution time have been studied; the features of the work of the Cached exchanged mode have been identified. Using the nonlinear approximation of the experimental data the generalized model of dependence of the performance of query execution on the amount of cache memory and the size of the requested tables was obtained. A dimensionless parameter of the relative amount of cache as the volume ratio of the cache to the database volume was introduced, based on which a criterion for the implementation of efficient caching mode in DBMS H2 was proposed.

The present study shows a fairly significant impact of caching on the query execution process, especially in the tables of a relatively small volume. The generalized model based on run-time queries of the cache size and the amount of the requested tables, which was obtained as a result of approximation, allows us to conduct forecasting of the query execution time at the determined conditions for any mode of cache operation. The simulation results demonstrate that from the practical perspective it is important to ensure the operation of DBMS H2 with the use of caching feature in the "effective" mode, when the allocated capacity of the cache memory is sufficient for the provision of efficient operation of the system. As a criterion ensuring this mode, it is possible to use exceeding the proposed dimensionless parameter of the relative volume cache against a certain critical level. The achieved results can be used in practice to significantly reduce the query execution time.

Keywords: query optimization, h2, relational database, math modelling, query execution, caching

References

1. **Eflov V. B., Pitukhin E. A., Stafeev F. V.** Analiz vypolnenija zaprosov v SUBD H2 (Analysis of query execution in H2 Database), *Uchenye zapiski Petrozavodskogo gosudarstvennogo universiteta. Ser.: Estestvennye i tehicheskie nauki*, 2013, no. 2 (131), pp. 62—65.
2. **Pitukhin E. A., Stafeev F. V.** Analiz metodov optimizacii zaprosov v reljacionnyh SUBD (Method's analysis of query optimization in relational DBMS). Moscow, 2010. Dep. VINITI 16.03.2010, no. 167-B2010.
3. **Stafeev F. V.** Spособы ocenki jeffektivnosti optimizatorov zaprosov v reljacionnyh SUBD (Methods of performance evaluation of query optimizers in relational DBMS), *Rol' nauki v ustojchivom*

razvitiia obshhestva: sbornik materialov 2-j mezhdunarodnoj nauchno-prakticheskoj konferencii, 24—25 dekabrja 2010 g., g. Tambov. Tambov, Tambovprint, 2010, pp. 150—151.

4. **Stafeev F. V.** Metody issledovanija processa obrabotki zaprosov v reljacionnyh SUBD (Research method of query execution process in relational DBMS). *Sovremennye instrumental'nye sistemy, informacionnye tehnologii i innovacii: materialy VIII Mezhdunarodnoj nauchno-tehnicheskoj konferencii, 28 dekabrja 2010 g.* Kursk, 2011, pp. 250—252.

5. **Tutorial.** *H2 Database Engine.* URL: <http://www.h2database.com/html/tutorial.html> (01.03.2016)