УДК 004.654
ББК 32.972.134
М 34

Рыбанов Александр Александрович
Доцент, кандидат технических наук, заведующий кафедрой информатики и технологии программирования Волжского политехнического института (филиал) Волгоградского государственного технического университета, Волжский, e-mail: rybanoff@yandex.ru
Свиридова Ольга Викторовна
Доцент, кандидат технических наук, доцент кафедры информатики и технологии программирования Волжского политехнического института (филиал) Волгоградского государственного технического университета, Волжский, e-mail: osviridova@inbox.ru
Филиппова Евгения Михайловна
Доцент, кандидат педагогических наук, доцент кафедры методики преподавания математики и физики, информационно-коммуникационных технологий Волгоградского государственного социально-педагогического университета, Волгоград, e-mail: em_filippova@mail.ru
Александрина Алла Юрьевна
Доцент, кандидат технических наук, доцент кафедры информатики и технологии программирования Волжского политехнического института (филиал) Волгоградского государственного технического университета, Волжский, e-mail: alla_aleksandrina@mail.ru
Абрамова Оксана Федоровна
Доцент кафедры информатики и технологии программирования Волжского политехнического института (филиал) Волгоградского государственного технического университета, Волжский, e-mail: oxabra@yandex.ru

Математическая модель динамики роста реляционной базы данных

(Рецензирована)

Аннотация. *Управление ростом данных позволяет сократить затраты и повысить производительность приложений. Мониторинг роста базы данных является ключевым фактором при планировании ресурса сервера. Отслеживание размера и роста баз данных является одной из основных задач планирования емкости. В статье рассмотрена математическая модель динамики роста реляционной базы данных, ориентированная на упреждающее решение проблем, связанных с ростом базы данных, и программная реализация процесса получения параметров модели. Предложена формула для расчета коэффициента прироста размера реляционной базы данных.*

Ключевые слова: *рост базы данных, размер базы данных, экспоненциальный рост объема данных, тенденция развития, кривые роста, MySQL.*

Rybanov Aleksandr Aleksandrovich
Associate Professor, Candidate of Technical Sciences, Head of Department of Informatics and Programming Techniques, Volzhsky Polytechnic Institute, Branch of the Volgograd State Technical University, Volzhsky, e-mail: rybanoff@yandex.ru
Sviridova Olga Viktorovna
Associate Professor, Candidate of Technical Sciences, Associate Professor of Department of Informatics and Programming Techniques, Volzhsky Polytechnic Institute, Branch of the Volgograd State Technical University, Volzhsky, e-mail: osviridova@inbox.ru
Filippova Evgeniya Mikhaylovna
Associate Professor, Candidate of Education Sciences, Associate Professor of Department of Teaching Methods of Mathematics and Physics, Information and Communication Technologies, Volgograd State Socio-Pedagogical University, Volgograd, e-mail: em_filippova@mail.ru
Aleksandrina Alla Yuryevna
Associate Professor, Candidate of Technical Sciences, Associate Professor of Department of Informatics and Programming Techniques, Volzhsky Polytechnic Institute, Branch of the Volgograd State Technical University, Volzhsky, e-mail: alla_aleksandrina@mail.ru
Abramova Oksana Fedorovna
Associate Professor of Department of Informatics and Programming Techniques, Volzhsky Polytechnic Institute, Branch of the Volgograd State Technical University, Volzhsky, e-mail: oxabra@yandex.ru

Mathematical model of growth dynamics for relational database

Abstract. *Data growth management reduces costs and improves application performance. Database growth monitoring is a key factor in server resource planning. Monitoring database size and growth is one of the main tasks for capacity planning. The article discusses a mathematical model of the growth dynamics of the relational database, focused on proactively solving problems associated with the database growth, and a software implementation of the process of obtaining model parameters. A formula is proposed for calculating the growth factor of the size of the relational database.*

Keywords: *database growth, database size, exponential data growth, development trend, growth curves, MySQL.*

Базы данных (БД) используются организациями для поддержки своих крупных вебсайтов, критически важных для бизнеса систем и коммерческого программного обеспечения. Организации могут смягчить влияние роста данных в этих системах, запуская пакетные отчеты в непиковые часы, покупая более быстрое оборудование или используя функцию очистки [1, 2]. Однако ни один из этих методов в полной мере не учитывает реальные параметры роста базы данных.

Наиболее важной частью планирования дискового пространства для хранения базы данных информационной системы является текущий размер и ожидаемая скорость роста БД [3]. Размер и динамика роста базы данных зависят от используемой системы управления базами данных (СУБД). Каждая БД использует свой формат для хранения данных, поэтому характеристики динамики роста баз данных различаются.

Неконтролируемый рост базы данных может принести проблемы конечным показателям организации в виде низкой производительности приложений и увеличения затрат на инфраструктуру [4]. Тем не менее, мониторинг роста данных зачастую является второстепенным фактором в повседневной работе администраторов баз данных. И, что еще хуже, включение механизмов архивации и сокращения данных часто не является обязательным требованием на этапах проектирования и реализации новых приложений. И, как следствие, базы данных приложений часто растут до такой степени, при которой необходимо принимать незапланированные меры для предотвращения перебоев в обслуживании.

Мониторинг роста базы данных является ключевым фактором при планировании ресурса сервера. Отслеживание размера и роста баз данных является одной из основных задач планирования емкости.

Одна из проблем для администраторов баз данных – управление ростом базы данных. Недооценка важности управления базой данных может привести к проблемам при перемещении базы данных в производственную среду.

Рассмотрим математическую модель динамики роста реляционной базы данных, представленной следующей схемой R:

$$R = \begin{cases} R_1(A_{1.1}, A_{1.2}, \ldots, A_{1.m_1}), \\ \ldots \\ R_i(A_{i.1}, A_{i.2}, \ldots, A_{i.m_i}), \\ \ldots \\ R_n(A_{n.1}, A_{n.2}, \ldots, A_{n.m_n}). \end{cases}$$

В основу модели динамики роста [5] реляционной базы данных положено утверждение, что скорость изменения количества кортежей в отношении R_i пропорциональна их текущему количеству в момент времени t, с коэффициентом пропорциональности, равным разности коэффициентов добавления $\mu_i \geq 0$ и удаления $\eta_i \geq 0$ кортежей из отношения R_i:

$$\frac{dN_i(t)}{dt} = (\mu_i - \eta_i) \cdot N_i(t) = \alpha_i \cdot N_i(t), \quad (1)$$

где $\alpha_i = \mu_i - \eta_i$ – коэффициент прироста кортежей в отношении R_i;

μ_i – коэффициент добавления кортежей R_i;

η_i – коэффициент удаления кортежей R_i;

$N_i(t)$ – число кортежей в отношении R_i.

Изменения количества кортежей в отношении R_i на временном интервале $[t_0;t]$ определим следующей зависимостью экспоненциального роста (рис. 1) с постоянным темпом ($\alpha_i = \text{const}$):

$$N_i(t) = N_i(t_0) \cdot e^{\alpha_i \cdot t}. \quad (2)$$

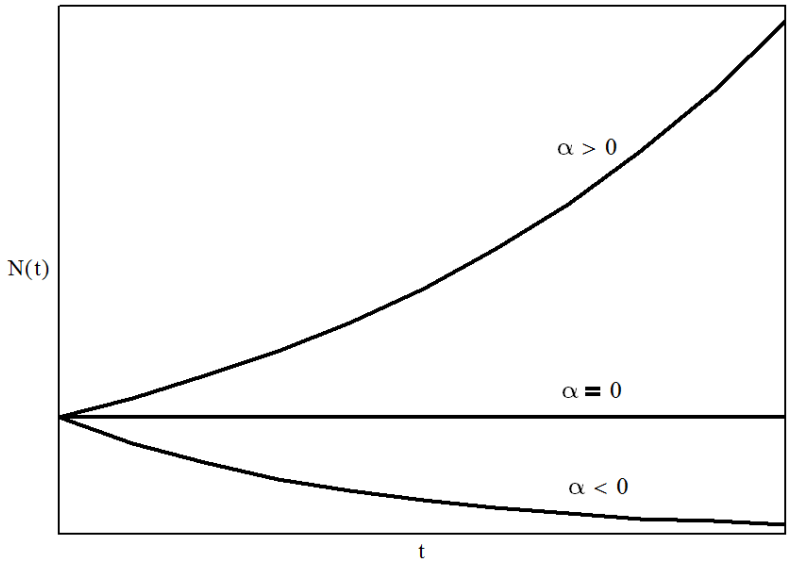

Рис. 1. Графическая интерпретация зависимости изменения количества кортежей в отношении R_i

Из экспоненциальной зависимости (2) следует, что коэффициент прироста кортежей α_i в отношении R_i определяется как:

$$\alpha_i = \frac{1}{t} \ln\left(\frac{N_i(t)}{N_i(t_0)}\right). \quad (3)$$

Размер отношения R_i в момент времени t определим как:

$$S_i(t) = \lambda_i \cdot N_i(t) = \lambda_i \cdot N_i(t_0) \cdot e^{\alpha_i \cdot t} = S_i(t_0) \cdot e^{\alpha_i \cdot t}, \quad (4)$$

где $S_i(t)$ – размер отношения R_i; λ_i – средний размер кортежа в отношении R_i.

Тогда размер реляционной базы данных R в момент времени t:

$$S(t) = S(t_0) \cdot e^{\alpha \cdot t} = \sum_i S_i(t_0) \cdot e^{\alpha_i \cdot t}, \quad (5)$$

где α – коэффициент прироста размера реляционной базы данных R.

Из формулы (4) можно определить коэффициент прироста размера реляционной базы данных в момент времени $t > t_0$:

$$\alpha(t) = \ln\left(\left(\frac{\sum_i S_i(t_0) \cdot e^{\alpha_i \cdot t}}{S(t_0)}\right)^{\frac{1}{t}}\right). \quad (6)$$

Приложение зависимостей (2)–(6) рассмотрим на следующем примере. Пусть реляци-

онная база данных R представлена тремя отношениями R_1, R_2, R_3, кривые роста количества записей $N_i(t)$ для которых за период 10 суток показаны на рисунке 2(а):

– R_1: $N_1(t_0) = 4$, $\alpha_1 = 0{,}25$, $\lambda_1 = 150$ байт;
– R_2: $N_2(t_0) = 15$, $\alpha_2 = 0{,}04$, $\lambda_2 = 50$ байт;
– R_3: $N_3(t_0) = 21$, $\alpha_3 = 0{,}12$, $\lambda_3 = 100$ байт.

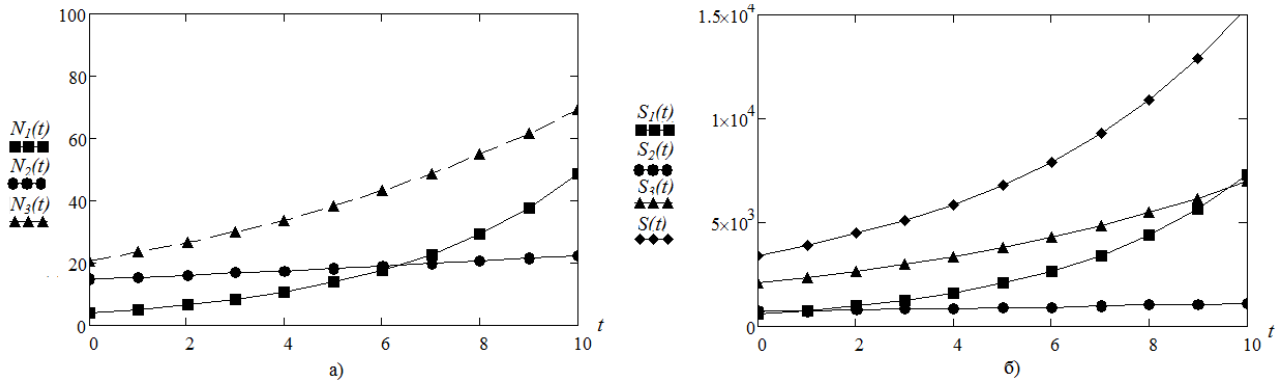

Рис. 2. а) кривые роста количества записей для R_1, R_2 R_3;
б) кривые роста размера (байт) для R, R_1, R_2, R_3

Тогда из формул (4)–(6) следует, что для реляционной базы данных R в момент времени $t = 10$ коэффициент прироста размера $\alpha = 0{,}15$. Кривые роста размера реляционной базы данных R и ее отношений R_i показаны на рисунке 2(б). Изменение коэффициента прироста размера базы данных R показано на рисунке 3.

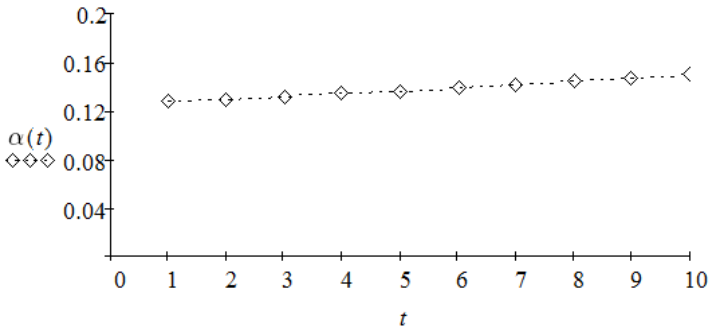

Рис. 3. Изменение коэффициента прироста размера базы данных R

Рассмотрим программную реализацию процесса регистрации метрик о росте таблиц и базы данных MySQL. В состав MySQL версии 5.0 и выше входит виртуальная база *information_schema*, которая не хранится в виде файлов, а формируется во время запуска сервера. База *information_schema* содержит мета-информацию об объектах и параметрах баз данных: имена таблиц, столбцов, ограничений, типы данных столбцов и т.д.

Для программной регистрации информации о процессе изменения параметров исследуемой базы данных, характеризующих ее динамику роста, воспользуемся мета-информацией из *information_schema*.

Введем следующие обозначения: *table_schema* – имя базы данных; *table_name* – имя таблицы; *table_rows* – количество кортежей; *avg_row_length* – средняя длина кортежа (байт); *data_length* – размер файла данных (байт); *index_length* – длина индексного файла (байт); *data_free* – количество распределенных, но не используемых байтов (байт); *date_log* – дата и время регистрационной записи; *id* – идентификатор регистрационной записи.

Журнал регистрации метрик процесса изменения значений параметров исследуемой базы данных представим таблицей *log_tables_databases*:

```
CREATE TABLE log_tables_databases (
 id int(11) NOT NULL AUTO_INCREMENT,
 table_schema varchar(64) NOT NULL,
 table_name varchar(64) NOT NULL,
 table_rows bigint(21) DEFAULT NULL,
 avg_row_length bigint(21) UNSIGNED DEFAULT NULL,
 data_length bigint(21) UNSIGNED DEFAULT NULL,
 index_length bigint(21) UNSIGNED DEFAULT NULL,
 data_free bigint(21) UNSIGNED DEFAULT NULL,
 date_log datetime NOT NULL,
 PRIMARY KEY (id) );
```

Процесс регистрации текущих значений параметров исследуемой базы данных *db_name* представим в виде хранимой процедуры *information_logging*:

```
CREATE PROCEDURE information_logging (IN db_name VARCHAR(64))
BEGIN
 -- описание данных регистрации
 DECLARE log_table_schema , log_table_name varchar(64);
 DECLARE log_table_rows, log_avg_row_length, log_data_length,
            log_index_length, log_data_free bigint(21);
 DECLARE      current_date_log datetime;
 -- описание данных и элементов для реализации курсора
 DECLARE done INTEGER DEFAULT FALSE;
 DECLARE cur1 CURSOR FOR
                  SELECT table_schema, table_name, table_rows, avg_row_length,
                        data_length, index_length, data_free
                  FROM information_schema.TABLES
                  WHERE table_schema = db_name AND table_type='BASE TABLE';
 DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done=TRUE;
 -- дата и время создания регистрационной записи
 SET current_date_log=NOW();
 -- реализация курсора
 OPEN cur1;
  -- цикл чтения курсора
 read_loop: LOOP
  FETCH cur1 INTO log_table_schema, log_table_name, log_table_rows,
                  log_avg_row_length, log_data_length, log_index_length,
                  log_data_free;
  -- проверка флага выхода
  IF done THEN
   LEAVE read_loop;
  END IF;
  -- регистрация текущих параметров для таблиц исследуемой базы данных
  INSERT INTO log_tables_databases (table_schema, table_name, table_rows,
                                    avg_row_length, data_length, index_length,
                                    data_free, date_log)
                            VALUES (log_table_schema, log_table_name,
                                    log_table_rows, log_avg_row_length,
                                    log_data_length, log_index_length,
                                    log_data_free, current_date_log);
 END LOOP;
 CLOSE cur1;
 END;
```

Курсор *cur1* выполняет итерацию по всем таблицам исследуемой базы данных для получения метрик их роста.

Планирование сбора значений метрик исследуемой базы данных *db_name* представим в виде события *event_logging*:

```
CREATE EVENT event_logging
  ON SCHEDULE EVERY '1' DAY
  STARTS CURRENT_TIMESTAMP
  ON COMPLETION PRESERVE
  ENABLE
  DO BEGIN
    CALL logging_infrmation (<имя базы данных>);
  END;
```

Предварительно необходимо включить режим событий в конфигурационном файле mysql.ini:

```
event_scheduler=1
```

или выполнить команду:

```
SET GLOBAL event_scheduler=ON;
```

Используя данные журнала регистрации метрик *log_tables_databases*, полученные с помощью события *event_logging,* можно извлекать полезную аналитику, необходимую для построения моделей роста таблиц и базы данных, а также для планирования емкости хранения БД.

Для определения необходимых параметров математической модели динамики роста реляционной базы данных воспользуемся следующими запросами по получению необходимой информации о таблицах БД:

1) Получить данные для определения скорости роста заданной таблицы БД по месяцам за период с t_0:

```
SELECT table_name, table_rows, avg_row_length,
          ROUND((data_length +index_length) / 1024 / 1024, 3) AS 'table_size (MB)',
          DATE_FORMAT(date_log,'%m.%Y') AS 'date_log'
  FROM log_tables_databases
  WHERE table_schema = <имя базы данных> AND table_name = <имя таблицы>
    AND LAST_DAY(date_log)=DATE_FORMAT(date_log,'%d.%m.%y')
    AND DATE_FORMAT(date_log,'%m.%Y') > = <начальное время t0>
ORDER BY date_log ASC;
```

2) Получить данные для определения скорости роста БД по месяцам за период с t_0:

```
SELECT table_schema, data_length, index_length, data_free,
          ROUND(SUM(data_length + index_length) / 1024 / 1024, 3) AS 'size (MB)',
          ROUND(SUM(data_free) / 1024 / 1024, 3) AS 'free_size (MB)',
          DATE_FORMAT(date_log,'%m.%Y') AS 'date_log'
  FROM log_tables_databases
  WHERE table_schema = <имя базы данных>
          AND LAST_DAY(date_log)=DATE_FORMAT(date_log,'%d.%m.%y')
          AND DATE_FORMAT(date_log,'%m.%Y') >= <начальное время t0>
  ORDER BY date_log ASC;
```

Полученная в результате выполнения запросов информация позволит решить следующие задачи:

– определение тренда роста базы данных;
– определение самых больших таблиц;
– определение тренда роста таблиц и индексов;
– расчет размера таблицы;
– оценка размера индекса;
– мониторинг неиспользуемых индексов.

Упреждающее решение проблем, связанных с ростом базы данных, не только позволит снизить расходы на инфраструктуру, но и может иметь решающее значение для оптимальной производительности приложений.

Математическая модель (2)–(6) и статистика по изменению размера базы данных и таблиц, собранная за определенный период времени, позволит проводить следующие виды анализа:

– предупреждать о ситуации, когда рост базы данных превышает заданные пределы;
– проанализировать тенденцию роста каждой базы данных;
– проанализировать тенденцию роста таблиц в каждой базе данных;
– проанализировать тенденцию роста данных и индексов по таблицам базы данных;
– ежедневно получать отчеты о росте базы данных.

Структура журнала регистрации метрик *log_tables_databases*, хранимая процедура фиксации текущих значений параметров исследуемой базы данных *information_logging*, а также запросы для получения параметров математической модели динамики роста реляционной базы данных могут быть интегрированы в физические схемы производственных баз данных для решения задачи управления ростом базы данных MySQL.

Примечания:

1. Кузьмин А.А., Рыбанов А.А. Исследование методов количественной оценки схем реляционных баз данных // Успехи современного естествознания. 2011. № 7. С. 137–138.
2. Лисецкий Ю.М. Комплексный подход к управлению данными // Математические машины и системы. 2019. № 4. С. 93–99.
3. Miller H.G., Mork P. From data to decisions: a value chain for big data // IT Professional. 2013. Vol. 15, No. 1. P. 57–59.
4. Myalapalli V.K., Totakura T.P., Geloth S. Augmenting database performance via SQL tuning // 2015 International Conference on Energy Systems and Applications. 2015. P. 13–18.
5. Лосанова Ф.М., Кенетова Р.О. Об одной обобщенной математической модели Мальтуса // Вестник КРАУНЦ. Физико-математические науки. 2019. Т. 27, № 2. С. 38–46.

References:

1. Kuzmin A.A., Rybanov A.A. Research on methods for quantifying relational database schemes // Advances in Current Natural Sciences. 2011. No. 7. P. 137–138.
2. Lisetsky Yu.M. Integrated approach to data management // Mathematical Machines and Systems. 2019. No. 4. P. 93–99.
3. Miller H.G., Mork P. From data to decisions: a value chain for big data // IT Professional. 2013. Vol. 15, No. 1. P. 57–59.
4. Myalapalli V.K., Totakura T.P., Geloth S. Augmenting database performance via SQL tuning // 2015 International Conference on Energy Systems and Applications. 2015. P. 13–18.
5. Losanova F.M., Kenetova R.O. On a generalized mathematical model of Malthus // Bulletin KRAUNC. Physical and Mathematical Sciences. 2019. Vol. 27, No. 2. P. 38–46.