UDC 004.62.051

A.B. Kungurtsev, PhD, Prof.,
S.L. Zinovatnaya, PhD, Assoc.Prof.,
Munzer Al Abdo, Master,
Odessa National Polytechnic University

PREDICTION OF A RELATIONAL DATABASE'S OPERATION IN THE INFORMATION SYSTEM

Introduction. There exist various methods and systems of testing databases (DB) used in information systems (IS) [1...4]. They allow to simulate certain situations by changing the number of records in the table, increasing users' number, a certain type querying intensity changes, etc. [5]. However, in most cases, the testing plan is based onto experience and intuition of the engineer/user conducting the test series and the entire database represents such test object.

Analysis of recent research and publications. Most aspects of DB operation and its tables' content are determined by the sequence of requests arriving to the input [5...7]. The essence of predicting the IS behavior consists in extrapolating trends of some system's characteristics [8]. At this research the trend will be determined by approximating the time series' actual values at querying the database. Further, it makes sense to limit such analysis with univariate forecasting functions where the argument factor is the time, considered as integral index of all factorial traits' combined effect.

In practice the most often used are about 15 univariate forecasting functions. In the case under consideration, only functions describing some IS characteristics' increase or decrease that affect the systems 'performance represent the interest subject. So further reasonable is to limit with two forecasting functions (regression functions):

— linear

$$y1 = a + b \times t, \tag{1}$$

— parabolic

$$y2 = a0 + a1 \times t + a2 \times t^2, \tag{2}$$

where $y1$, $y2$ — some of system's characteristics, e.g., average delay of queries execution across 24 hours period;

t — time unit (e.g., 24-hours period);

a, $a0$, $a1$, $a2$, b — coefficients subject to determining.

The Aim of Research. To enable the purposed application of software targeting the IS improved operation, this research aim is assigned as developing a method for predicting the system behavior with identification of database components critical in terms of their impact on the whole IS performance.

Main Body. Let we consider a time series of requests received by the IS during observed time delay τ from the start t_0 to the end t_e of the observation cycle. Each query q can be written as

$$q_i = < txt, t_s, t_f >, \tag{3}$$

where txt — query text;

t_s, t_f — query execution start and end time, respectively.

Use of the t_s and t_f at (3) allows the assumption of every query's unique character and consecutively makes possible to represent the queries' time-sequence as a set

$$Q = \{q_j\}, j = \overline{1,n}, \tag{4}$$

DOI 10.15276/opu.1.45.2015.19

ISSN 2076-2429 (print)
ISSN 2223-3814 (online)

where n — number of queries arrived to IS during τ delay.

To determine the queries' temporal characteristics the notion of requests' intensity distribution is used [9, 10]. For this the $\Delta_t = t_i - t_{i-1}$ time interval is defined as a constant for the entire period. The value Δ_t can be set on the basis of the following considerations. The vast majority of observed IS functions have periodic character. Often the very short duration period is equal to the length of one working day. Selecting the Δ_t value below the recommended one, we can obtain significant growth in requests distribution intensity dispersion.

The current intensity value

$$d_l = \frac{|Mq|}{t_i - t_{i-1}},$$

where t_{i-1} and t_i — two sequential time instances within time delay $[t_0, t_e]$;

Mq — subset of the Q set:

$$Mq = \{q_j \mid q_j \in Q \wedge t_{sj} \geq t_{i-1} \wedge t_{fj} \leq t_i\},$$

where t_{sj} — start point of j-query execution;

t_{fj} — end point of j-query execution.

We do introduce the concept of requests intensity distribution function $fd(t)$ which at fixed time instances t_i takes values d_i. For fd correct representation it makes sense to adjust the requests time series by deleting the IS downtime and maintenance works time from the observation period i.e. excluding time when the system does not perform its basic functions.

The linear approximation of intensity function fd. To find the coefficients a and b at (1) we shall use the principle of least squares, which determine the extent of two functions values dispersion S.

$$S = \sum_{i=1}^{m} [fd(t_i) - y1(t_i)]^2,$$

where $m = \frac{\tau}{\Delta_t}$.

Then, according to [11]

$$b = \frac{m\sum_{i=1}^{m} t_i fd(t_i) - \sum_{i=1}^{m} t_i \sum_{i=1}^{m} fd(t_i)}{m\sum_{i=1}^{m} t_i^2 - \left(\sum_{i=1}^{m} t_i\right)^2},$$

$$a = \frac{\sum_{i=1}^{m} fd(t_i) - b\sum_{i=1}^{m} t_i}{m}.$$

The intensity function fd parabolic approximation. According to the principle of least squares, for the forecasting function (2) [11]

$$a0 = \frac{\sum_{i=1}^{m} fd(t_i)}{m},$$

$$a1 = \frac{\sum_{i=1}^{m} fd(t_i) P_1(t_i)}{\sum_{i=1}^{m} P_1^2(t_i)},$$

$$a2 = \frac{\sum_{i=1}^{m} fd(t_i) P_2(t_i)}{\sum_{i=1}^{m} P_2^2(t_i)},$$

where P_1, P_2 — Chebyshev's polynomials:

$$P_1(t) = t - \frac{m+1}{2},$$

$$P_2(t) = t^2 - (m+1)t + \frac{(m+1)(m+2)}{6}.$$

Selecting the approximation function. To choose one of the prognostic functions the criterion can be based on a comparison of average prediction errors calculated for each of the two (linear and parabolic) functions:

$$\varepsilon\rho = \sqrt{\frac{\sum_{i=1}^{m} (fd(t_i) - y_i)^2}{m}}.$$

Preferred shall be the function, whose prediction error value is the least.

If $y = fdr(t)$ — the found regression functions, then the extrapolation operation shall be

$$y = fdr(\tau + L), \tag{5}$$

where L — prediction period.

System loading trend evaluating. Here the system loading to be meant as the ratio of an observation period length to the total execution time of queries arrived during that period. Preparing data to regression function evaluation represents the calculation the each query interval's Δ_{t_i} execution time. Here we introduce the concept of system loading function $fl(t)$ taking for fixed time moments t_i values l_i.

$$l_i = \sum_{j=1}^{k_\Delta} (t_{fj} - t_{sj}), \tag{6}$$

where k_Δ — number of queries arrived to IS during the interval Δ_{t_i}

t_{sj} — start point of *j*-query execution;

t_{fj} — end point of *j*-query execution;

After performing the above operations, we obtain a regression function $y = flr(t)$ according to (5) allowing to predict the system loads.

Evaluating the queries' average execution time trend. This characteristic allows estimating the system load degree influence onto queries execution time. The speed of query processing depends on several factors caused by database condition, e.g., the number of data contained at DB tables. In addition, the LAN can be loaded with other tasks execution: sending electronic documents, requesting to Internet resources, email, etc.

Preparing data for regression function estimation refers to calculating the average time for each query interval Δ_{t_i}. Entering a function of average query execution time $fta(t)$ taking a value Ta_i for fixed time t_i moments, on its basis we can obtain a respective regression function $y = ftar(t)$ that, according to (5) can predict the average execution time.

$$Ta_i = \sum_{j=1}^{k_\Delta} \frac{(t_{fj} - t_{sj})}{k_\Delta}, \tag{7}$$

ISSN 2076-2429 (print)
ISSN 2223-3814 (online)

where t_{sj} — start point of *j*-query execution;

t_{fj} — end point of *j*-query execution.

Analysis and prediction of peak loads. We assume the load for some time, to be peak one if its value is Kp times greater than the average value for the period of observation.

Average IS load during the time of observation τ

$$S = \frac{1}{\tau}\sum_{j=1}^{n}(t_{fj} - t_{sj}),$$

where *n* — total queries number.

At the earlier stage of determining the queries' intensity we assigned an averaging time interval Δ_t equal to one day. To search for peak loads we must significantly reduce this interval duration, for example, up to Δ_p =0,3 h. Otherwise the averaging can "hide" the surging load. Further Δ_p reduction is not convenient because a transient increase in load is undetectable for IS user.

Average IS load for some interval of Δ_{p_j}

$$S_j = \frac{1}{\Delta p_j}\sum_{l=1}^{r}(t_{fj} - t_{sj}),$$

where r — number of queries arrived to IS during Δp_j period.

At $\frac{S_j}{S} \geq Kp$, we shall conclude of having observed a peak load occurred during Δp_j period.

Let we enter the peak load changes function depending on time $fp(t)$, which value for every Δp_j is:

$$fp(\Delta p_j) = \begin{cases} 0, \text{ if } \frac{S_j}{S} < Kp, \\ S_j, \text{ if } \frac{S_j}{S} \geq Kp. \end{cases}$$

Identification of the most frequently used tables and fields. When DB testing we must identify tables that are used more frequently while requests' processing. If the queries execution time is critical, we can use special tools to improve the IS performance, allowing to eliminate "bottlenecks" (replication, restructuring, indexing, etc.). We did specify the query model [9] including of information on used tables and query type

$$q = < txt, t_s, t_f, T_q, T_y >, \quad (8)$$

where $T_q = \{T_i\}, i = \overline{1, n_q}$ — set of tables used at query q;

T_y — query type.

Every table represents a tuple

$$T_i = < NT_i, F_i >,$$

where NT_i — table name;

$F_i = \{f_{ij}\}, j = \overline{1, n_{Fi}}$ — set of table fields addressed at *q* query.

The query type T_y is an element of the set

$$T_y \in \{td, ti, tu, ts, tt\},$$

where td — query type DELETE;

ti — query type INSERT,

tu — query type UPDATE,

ts — query type SELECT,

tt — query type change of DB structure.

Proposed is to introduce the concept of MT , DB tables set, composed of elements

$$< NT_i, nt_i, F_i^{DB} > ,$$

where NT_i — i^{th} table;

nt_i — number of queries to NT_i table at all requests during the observed time τ ;

F_i^{DB} — number of NT_i table fields.

Every field is represented as:

$$f_{ij} =< Nf_{ij}, nf_{ij} > ,$$

where Nf_{ij} — field name;

nf_{ij} — number of queries to the Nf_{ij} field at all requests during the observed time τ .

At the experimental series beginning all values of nt_i and nf_{ij} are equal to zero.

We introduce the operation of attributing some table NT_i to the request q

$$NT_i \in_T q ,$$

the operation result is "true" value, if table NT_i is used at the request q, and "false" value if otherwise.

Now we introduce an operation of some field f_{ij} attribution to NT_i table

$$f_{ij} \in_f NT_i ,$$

the operation result is "true" value, if the field f_{ij} belongs to table NT_i and "false" value if otherwise.

Especially for queries updating the data (INSERT, UPDATE, DELETE), we introduce the operation of identification of table containing the updated data

$$NT_i \in_{mod} q .$$

On the MT basis we can form a list Lmt, arranging the elements by the degree of decrease in parameter nt_i

$$MT \Rightarrow_{\text{sort}} Lmt . \quad (9)$$

Identifying the tables and fields most frequently used at particular type queries. The proposed method of creating a list of most frequently used tables is easy for applying to a certain type of queries. For example, to find the most frequently used table type in queries SELECT, we must create a subsetbased on the queries set Q (4)

$$Qsel = \{q_i \mid q_i \in Q \wedge Ty_i = ts\} .$$

For queries $q \in Qsel$ we can establish both the tables set $MTsel$ and hierarchically arranged list $MTsel \Rightarrow_{sort} Lmt_{sel}$.

For information about tables, whose entries amount grows the faster, we can analyze the requests type INSERT and DELETE, getting a set of numbers that define entries adding (removing) during the period of observation for each database table and the correspondingly ordered list. $MTmod \Rightarrow_{sort} Lmt_{mod}$

Structure of considered and other lists may serve as a starting point for predicting changes in IS "local" performances.

Results. Presented are the results of the proposed method of information system "Accounting at medical institutions" analysis.

The system uses a relational database consisting of 98 tables.

The study covered three months of the system operation.

Fig. 1 shows a diagram of average execution time for queries of SELECT type

ISSN 2076-2429 (print)
ISSN 2223-3814 (online)

Calculations according to (7) show that during next 6 months the average execution time may increase by 3,4 times.

Recommended is to improve the IS performance using software tools. For that, the (8) and (9) were used to obtain lists of most frequently required tables and their fields (Tables 1 and 2).

Table 1

Number of requests to DB tables

Table name	Data description	Queries number
planersearchkind	Registering for examination	3241
planersearchkindmore	Medical examinations completed	3088
factservice	Examination result	2902
Planer	Registering for consultation	1956

Table 2

Number of requests to DB tables' fields

Field name	Data description	queries number
Table planersearchkind		
dateplanersearchkind	Date of examination	3241
Idperson	Client's identificator	2015
prescription	Person authorised to make prescription	581
numberplaner	Number of registry for scheduled examination	308
Iscito	Urgent examination	124
…		
Table planersearchkindmore		
idplanersearchkindmore	Identificator of registry for scheduled examination	3088
idparameter	Identificator of parameter	3088
idsearchkind	Identificator of examination	1537
idlaboratory	Identificator of laboratory	1100
…		
Table factservice		
Idperson	Identificator of client	2902
datefactservicein	Date of biotissues sampling	2902
datefactserviceresult	Date of result getting	2902
idemployeeresult	Identificator of laboratory assistant having analyzed the sample	1044
idemployeein	Identificator of laboratory assistant having taken the sample	637
nsearchkind	Number of analytical examination	611
…		

Proposed is to introduce the indexation for: field idperson at table planersearchkind, field idlaboratory at table planersearchkindmore, fields datefactservicein, datesend at table factservice, and to revise the structure of tables planersearchkindmore and factservice, expanding them with such fields as: pllaboratory, plisown, plcode, duplicating the data of laboratory name, its institutional affiliation and laboratory analysis code respectively.

Performed is a study of IS peak loads (6) for Kp=1,5 value. The respective diagram (without by-working-hours details) is shown at Fig. 2.

Identified are the queries, creating increased system load during of peak load periods (Monday, end of the month). Recommended is to verify the possible use of materialized representations for these queries [9].

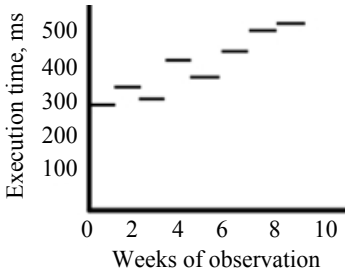

Fig. 1. Changes in queries' execution average time

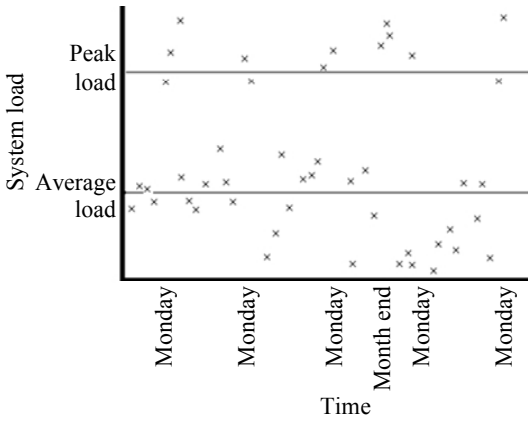

Fig. 2. Peak loads identification

Conclusions. The proposed forecasting method, based on the IS DB requests time series' analysis, allows to predict whole system behavior, identifying its elements which may affect its performance in the future and at peak loads, also allowing to determine approximate term of necessary modernization, with specific-purpose tests performing and simulation of system behavior.

Література

1. A Framework for Testing Database Applications / D. Chays, S. Dan, P.G. Frankl, F.I. Vokolos, and E.J. Weyuker // Proceedings of the 2000 ACM SIGSOFT international symposium on Software testing and analysis (ISSTA 2000), August 21–24, 2000, Portland, Oregon, USA. — New York: ACM Press, 2000. — PP. 147 — 157.
2. E-NAXOS DataGen 2005: Test Data Generator [Електронний ресурс] / Olivier Dahan; E-Naxos. — Режим доступу: http://www.e-naxos.com/blog/default.aspx (Дата звернення: 20.11.2014).
3. IBM InfoSphere Optim Test Data Management [Електронний ресурс] / IBM. — Режим доступу: http://www-01.ibm.com/software/data/optim/core/test-data-management-solution/ (Дата звернення: 20.11.2014).
4. Justus, S. An empirical validation of the suite of metrics for object-relational data modelling / S. Justus, K. Iyakutti // International Journal of Intelligent Information and Database Systems. — 2011. — Vol. 5, No. 1. — PP. 49 — 80.
5. Кунгурцев, А.Б. Средства автоматизированного заполнения баз данных информационных систем для проведения тестирования запросов / А.Б. Кунгурцев, А.А. Блажко, А.Ю. Левченко // Електромашинобуд. та електрообладн. — 2009. — Вип. 72. — С. 201 — 204.
6. Daniel, L. Digital Forensics for Legal Professionals: Understanding Digital Evidence from the Warrant to the Courtroom / L. Daniel, L. Daniel. — Waltham: Syngress, 2011. — 368 p.
7. Osman, R. Database system performance evaluation models: A survey / R. Osman, W.J. Knottenbelt // Performance Evaluation. — 2012. — Vol. 69, Issue 10. — PP. 471 — 493.
8. Коротаев, А.В. Законы истории. Математическое моделирование развития Мир-Системы. Демография, экономика, культура / А.В. Коротаев [и др.] ; Рос. гос. гуманитарный ун-т. Факультет истории, политологии и права, РАН, Центр цивилизационных и региональных исследований. Институт востоковедения. — 2-е изд., испр. и доп. — М.: КомКнига, 2007. — 224 с.
9. Кунгурцев, А.Б. Определение параметров периодического включения/выключения материализованных представлений в информационных системах / А.Б. Кунгурцев, Ю.Н. Возовиков, Нгуен Чан Куок Винь // Вост.-Европ. журн. передовых технологий. — 2012. — № 4/2 (58). — С. 42 — 45.
10. Artail, H. SQL query space and time complexity estimation for multidimensional queries / H. Artail, H. El Amine, F. Sakkal // International Journal of Intelligent Information and Database Systems. — 2008. — Vol. 2, No. 4. — PP. 460 — 480.
11. Hastie, T. The Elements of Statistical Learning: Data Mining, Inference, and Prediction / T. Hastie, R. Tibshirani, J. Friedman. — 2nd Edition. — New York: Springer, 2009. — 745 p.

References

1. Chays, D., Dan, S., Frankl, P.G., Vokolos, F.I. and Weyuker, E.J. (2000). A framework for testing database applications. In M.J. Harrold (Ed.), *Proceedings of the 2000 ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2000)* (pp. 147-157). New York: ACM Press.

ISSN 2076-2429 (print)
ISSN 2223-3814 (online)

2. Dahan, O. (n.d.). *E-NAXOS DataGen 2005: Test Data Generator* [Web log post]. Retrieved from http://www.e-naxos.com/blog/default.aspx
3. IBM (n.d.). *IBM InfoSphere Optim Test Data Management*. Retrieved from http://www-01.ibm.com/software/data/optim/core/test-data-management-solution/
4. Justus, S. and Iyakutti, K. (2011). An empirical validation of the suite of metrics for object-relational data modeling. *International Journal of Intelligent Information and Database Systems*, 5(1), 49-80.
5. Kungurtsev, A.B., Blazhko, A.A. and Levchenko, A.Yu. (2009). Tools of database filling automation in information system for query testing. *Electrical Machine-Building and Electrical Equipment*, 72, 201-204.
6. Daniel, L. and Daniel, L. (2011). *Digital Forensics for Legal Professionals: Understanding Digital Evidence from the Warrant to the Courtroom.* Waltham: Syngress.
7. Osman, R. and Knottenbelt, W.J. (2012). Database system performance evaluation models: A survey. *Performance Evaluation*, 69(10), 471-493.
8. Korotaev, A.V., Malkov, A.S. and Khalturina, D.A. (2007). *Law of History. Mathematical Modeling of Historical Macroprocesses. Demography, Economics, War* (2nd ed.). Moscow: KomKniga.
9. Kungurtsev, A., Vozovikov, Yu. and Nguyen Tran Quoc Vinh (2012). Determination of the parameters of periodic ON/OFF materialized view in the information system. *Eastern-European Journal of Enterprise Technologies*, 4(2), 42-45.
10. Artail, H., El Amine, H. and Sakkal, F. (2008). SQL query space and time complexity estimation for multidimensional queries. *International Journal of Intelligent Information and Database Systems*, 2(4), 460-480.
11. Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). New York: Springer.

АНОТАЦІЯ / АННОТАЦИЯ / ABSTRACT

О.Б. Кунгурцев, С.Л. Зіноватна, Мунзер Аль Абдо. **Прогнозування роботи реляційної бази даних в інформаційній системі.** Необхідною умовою роботи будь-якої інформаційної системи є відповідність часу відгуку на запити користувача вимогам предметної області. Актуальним завданням є автоматизація методів для вибору засобів зниження часу виконання запитів до бази даних. Формальне описання стану й розвитку системи дає можливість застосувати найбільш ефективні способи прискорення запитів у момент часу, коли виникає необхідність підвищення продуктивності системи. В роботі отримано залежності, які дозволяють виконувати прогнозування інтенсивності запитів, що надходять до інформаційної системи, визначати тренд завантаження системи, виявляти періоди пікових навантажень і множину найбільш часто використовуваних таблиць і полів. Результати дослідження роботи системи дають можливість прогнозувати поводження системи і вчасно вживати заходи для підтримки необхідного рівня її продуктивності.

Ключові слова: інформаційна система, реляційна база даних, запит, тестування, моделювання, прогнозування.

А.Б. Кунгурцев, С.Л. Зиноватная, Мунзер Аль Абдо. **Прогнозирование работы реляционной базы данных в информационной системе.** Необходимым условием работы любой информационной системы является соответствие времени отклика на запросы пользователя требованиям предметной области. Актуальной задачей является автоматизация методов для выбора средств снижения времени выполнения запросов к базе данных. Формальное описание состояния и развития системы дает возможность применить наиболее эффективные способы ускорения запросов в момент времени, когда возникает необходимость повышения производительности системы. В работе получены зависимости, которые позволяют выполнять прогнозирование интенсивности запросов, поступающих в информационную систему, определять тренд загрузки системы, выявлять периоды пиковых нагрузок и множество наиболее часто используемых таблиц и полей. Результаты исследования работы системы дают возможность прогнозировать поведение системы и своевременно принимать меры для поддержания необходимого уровня её продуктивности.

Ключевые слова: информационная система, реляционная база данных, запрос, тестирование, моделирование, прогнозирование.

A.B. Kungurtsev, S.L. Zinovatnaya, Munzer Al Abdo. **Prediction of a relational database's operation in the information system.** A necessary condition of any information system's efficient operation is to ensure that response time to user requests satisfies the subject area requirements. Actual is the problem of automated methods for choosing the tools reducing the execution time of queries to the database. Formal description of the system's state and development gives the possibility to apply the most effective ways of speeding up queries in time when there is a need to improve system performance. While researching some dependences are obtained, allowing to predict the intensity of requests entering the information system, to determine the trend of system's loading, to identify peak periods and set of the most commonly used tables and fields. The results on system operative features researching provide the possibility to predict system's behavior and to take timely measures to maintain the required level of performance.

Keywords: information system, relational database, query, test, modeling, prediction.

Received December 24, 2014