DOI: 10.15439/2022F257
ISSN 2300-5963 ACSIS, Vol. 32

Heuristic algorithm for periodic patterns discovery in a database workload reconstruction

Marcin Zimniak, Bogdan Franczyk
Information Systems Institute
Leipzig University
Germany
Email: {zimniak, franczyk}@wifa.uni-leipzig.de

Marta Burzańska, Piotr Wiśniewski
Faculty of Mathematics and Computer Science
Nicolaus Copernicus University in Toruń
Poland
Email: {quintria, pikonrad}@mat.umk.pl

***Abstract*—Information about the existence of periodic patterns in a database workload can play a big part in the process of database tuning. However, full analysis of audit trails can be cumbersome and time-consuming. This paper discusses a heuristic algorithm that focuses on workload reconstruction based on pattern discovery in a simplified workload notation. This notation is based on multisets representing database actions (such as user queries) requiring access to specific persistent objects, but without the access cost analysis. Each action in this notation is a multiset of accessed objects, which can be tables, system files, views, etc. The theoretical model for such an approach has been discussed in detail in the authors' previous work [1] This work is mostly proof-of-a-concept for the theoretical approach. Additionally, in order to test the performance of the proposed algorithm, a test-data generator has been constructed. Both the previous and the current papers are parts of a research project dealing with the application of periodic pattern theory to the field of database optimization and tuning [2], [3], [4], [5].**

I. Introduction

THE WORKLOAD reconstruction for an SQL database is a non-trivial task. However it can be very important when considering tuning options. Because in a typical relational database queries tend to be repetitive to a point, discovering recurring patterns may lead to the improvement of DBMS performance. This paper is part of a research project tackling the application of the periodic pattern theory to the database workload reconstruction and prediction [5] This paper follows the research discussed theoretically in [1], and provides a heuristic reconstruction algorithm utilising a recursive periodic pattern discovery process. This work on the workload reconstruction strays from the research on the prediction of the n-th database state based on the retrospective analysis of the n-1 previous states (actions) [6]. Instead we focus on the global system usage estimation and prediction with the help of the reconstruction methods working with a simplified workload dump. In order to verify and evaluate the researched algorithm we have also developed a parameterized test-data generator. It is capable of generating a list of random multisets with a hidden periodicity feature. It is not predetermined but rather the construction of the generator ensures the existence of some non-trivial periodic patterns. The generated test-data is used to firstly verify the correctness of the main heuristic algorithm when it comes to the recognition of periodic patterns when searching for the optimal workload reconstruction. The second task was to aid the research on the reconstruction quality measure and time-efficiency of the reconstruction algorithm. The overall goal was to develop a fast, statistically stable algorithm that works with an optimal time and space complexity. The conducted tests have confirmed these assumptions. The programming language used for the development of the algorithm and the generator was Python 3.

The reconstruction problem touches on the possibility of the reconstruction (lossy or lossless) of a given system based on partial data obtained from the earlier (or current) behaviour of the system. The problem of reconstruction, as well as the current research on it, work with the concept of reconstructability (non-reconstructability) of affine functions [7], [8] and the reconstruction of multisets over grupoids as well. Apart from some open problems for multisets over groupoids, the concept of reconstructing the sequence of multisets (in general) is a new concept and has not been studied so far. In the proposed approach, we reduce the presentations of multisets in sequences to increasing sequences. Such a simplification for the presentation of multisets is related to the homogeneous treatment of elements (subtrees) that are implementations of algebra expressions, including entity instances sets (tables), which are a part of the output relational algebra. Such assumptions for multisets made it possible to tackle the problem of the reconstruction of the sequences of multisets without considering the problem of coverage of multisets, which takes place in the case of the reconstruction of the mutliset over grupoids. The aim of this work is to provide efficient algorithms for the reconstruction of the considered system and to provide qualitative information on the degree of its reconstruction. The algorithm can be applied to complex systems, such as database systems, whose log interpretations can be written in the form of a sequence of multisets, etc. The degree of reconstruction allows to evaluate the predictions of the whole system; the methodology of the algorithm takes into account the simultaneous inclusion of the time and frequency domains in each step of the reconstruction algorithm, for all periodic patterns included in the reconstruction. The problem of workload predicting has already been discussed by the author [2] in terms of another concept of the periodic pattern. The methods of signal analysis or wavelets are based on the assumption (in short) that points in time have been assigned values (real numbers), including the use of appropriate reconstruction methods[9], [10]. In our data model, first, we do not consider the points in time, but the time periods (pulled down to points) and sets of elements (including elements allowed multiple times), not to mention the nesting interpenetration of events considered on such structures etc. Therefore, the methodology of time series, Fast Fourier Transform (FFT) in discrete form (DFT), spectra, etc., does not work for events in the context of the data under consideration. In addition, we do not consider the independent behavior of individual elements, but we study and take into account (periodic) behavior of multiple sets of elements and their subsets, taking into account the interactions of one on another (in various proportions) as well as the entire system on individual elements, etc.

The paper is organised as follows. In Section II we focus on methodology. Section III provides our recursive reconstruction algorithm. Section IV specifies test data generator. Section V concludes effectiveness of the algorithm, possible future works end this section.

II. Methodology

The methodology used in the workload reconstruction process is based on the concept of intelligent extraction of periodic patterns in the workload defined below. The methods used in the author's previous works are standard bottom-up methods. The methods used in the current work use a recursive approach in combination with the heuristic method(s), much more efficient than the previous ones.

The theory and applications of the concept of periodic patterns to the workload prediction problem were discussed in the previous works of one of the authors [5], [2]. The theory of periodic patterns is well known. It grew out of, among others, the periodic sets [11] as well as periodic events [12]

Let recall terminology defined in previous considerations [1]. Let the workload W_L and the sequence of time units U be given. The non empty subsequences $C, C' \subseteq W_L$ of the same length and consecutive coordinates are called *equivalent* if $C = C'$ occurs for all corresponding coordinates.

A *periodic pattern* in a workload W_L is a tuple $<C,\ f,\ t,\ p,\ >$ where:

1) the *carrier* C determines a non empty subsequence $C \subseteq W_L$
2) f is a number of time unit in U where the repetitions of C start
3) t is a total number of occurrences of equivalent sequences $C \subseteq W_L$, such that p denotes the number of consecutive time unit elements after which the t pairs of neighbouring sequences are equivalent.
4) Parameters f, t, p satisfy the following inequality: $f, t \geq 1$, $p \geq 0$, $f + (t-1) * p + |C| - 1 \leq |U|$

Also, if $t = 1$ then $p = 0$ and the pattern $<C, f, 1, 0>$ is called the *trivial periodic pattern* (*trivial* pattern)

Let $<C,\ f,\ t,\ p,\ >$ be a periodic patterns in W_L with a given U.

A *trace of a carrier* C is a subsequence $C \subseteq W_L$, denoted $tr(C, f, n)$, in which the first $f - 1$ elements are the empty multisets.

A *trace of a periodic pattern* $<C,\ f,\ t,\ p,\ >$ over the time unit sequence U, under the condition $f+(t-1)*p+|C|-1 \leq n$, is a subsequence $TR(< C, f, t, p >, n)$ of a sequence W_L such, that $TR(< C, f, t, p >, n) = tr(C, f, n) \uplus tr(C, f + p, n) \uplus \ldots \uplus tr(C, f + (t-1) * p, n)$

Let R be a non-empty set of periodic pattens in a W_L given time unit sequence $U(n)$. In the current approach , we say that R is a *reconstruction of the workload* W_L in $U(n)$ if:

i. $\uplus_{s=1}^{|R|} TR(< C_S, f_S, p_S >, n) = W_L$
ii. all TRs implementing connect-disconnect processes remain consistent in relation to each other. We allow duplication of database connect/disconnect processes in case of hypothetical processes, assuming that logging in and logging out does not involve costs.

As a *quality measure of the reconstruction* R is a real value $0 \leq m_R < 1$ defined as:
$m_R = 1 - (1/ \sum_{i=1}^{|R|} (\|C_i\| * t_i))^{1/|R|}$
where $\|C_i\|$ is the length of the carrier C_i, $|R|$ is the cardinality of R. When $R = R_0 = \{< W_L, 1, 1, 0 >\}$ we assume that $m_{R_0} = 0$.

The test data generator, described in Section IV, is to provide synthetic data in the form of a sequence of multisets representing the sequence of database queries expressed in the evaluation representation obtained thanks to the `EXPLAIN PLAN` operation. This internal representation is then encoded using the syntax tree table [2]. The task of the presented test data generator is to generate hypothetical periodic patterns, but the generator does not determine the occurrence of a specific periodicity.

III. Workload reconstruction algorithm

With reference to the algorithm of [1], the algorithm presented in this paper has undergone some modifications and, as noted in the introduction, the simplification does not consider workload costs (in general). The pseudocode for the maximum relative frequency heuristic reconstructive algorithm is presented below. The heuristic is based on the local selection of the maximum relative frequency of the elements.

IV. Test data generator

In order to verify and evaluate the researched algorithm we have also developed a parameterized test-data generator. It is capable of generating a list of random multisets with a hidden periodicity feature. It is not predetermined but rather the construction of the generator ensures the existence of some non-trivial periodic patterns. The generator has been implemented in Python 3 language.

The basic building block of the result set is the list of lists:

```
[[object_number, number of occurrences], ...]
```

```
e.g. [[1.2], [2.2], [4.1], [6.3]]
```

Algorithm 1 Reconstruction algorithm

```
INPUT : $W_L$, list e
OUTPUT: optimal reconstruction R with the optimal value $m_R$
1) e= [ e1, e2, e3,..] sorted (or not: in case of greedy heuristic), R : = < $W_L$,f,1,0 >, $m_R$:=0 , C:= $W_L$,
   R: = $\emptyset$, |R|: = 1,
2) While |C| <> 0:
   a) C : = C $\setminus$ TR (pp) ; for the current element e, starting from a minimum f value until all possibilities for f have been analyzed, search for such p and t that pp : = < $e_i$ ,f,t,p > is a periodic pattern in $W_L$
   b) R : = R $\cup$ < C, f, 1,0> $\cup$ pp ; |R|++, Determination of min f and max t when selecting an element from the $W_L$ sequence, i.e. identifying systems matching: f + ( t - 1) * p $\leq$ n, pp = < $C_{pp}$, f, t, p >
   c) For the following elements, accept the last used t and p (start with the minimum value of f for which there is pp with parameters t and p, skipping the cases: t=1, p > [n/2] ), always proceed until the possibility of a non-trivial periodic pattern with the given parameters t and p is exhausted. If no patterns are found for the current t and p, do the same as in the previous step, starting with the current element (if there is one, otherwise the next one). In the absence of the current element e, sort the array e (based on the relative frequency) for the current elements. Normalisation Rule + Decomposition Rule [1] : < C, f, 1,0> in each step is a (trivial) periodic pattern in $W_L$. Use the Decomposition Rules for periodic patterns found for elements for which patterns with t and p parameters were found. Create periodic patterns: $pp_i$, $pp_j$ in $W_L$ with $C_i$, $p_i$, $t_i$ and $C_j$, $p_j$, $t_j$ respectively, such that: $p_i$= $p_j$ = p and $t_i$= $t_j$ = t, in order to pair (reduce) periodic patterns to the form: $pp_{ij}$ = < tr ( $C_i$ , 1, $f_j$ - $f_i$ + |$C_j$ | ) $\uplus$ tr ( $C_j$ , $f_j$ - $f_i$, $f_j$ - $f_i$ + | $C_j$ |), $f_i$ , t, p > in $W_L$ , and pair in such a way as to maximise the value of $m_R$, |R| $- -$, use ($--$) as many times as the number of reduced pairs (reduction associative). As a results from method used above we consider only those cases in which the very last step of the recursion consists of non-trivial patterns. Otherwise, the interruption and output of the algorithm with the output as reconstruction of $W_L$ not possible.
   d) Return and possibly replace (depending on the value of the previous scenario of the previous "recursive paths") the maximum $m_R$ and associated set R
```

The full multiset list may then take the form:

```
WL=[[1,8], [2,4], [ [1,7], [2,3] ] , [1,6],
[3,2], [[1,7],[2,1]], [[1,1],[2,1]],
 [[1,7],[2,1]] ,  [[1,1][2,1]] , [[1,3],
[2,1],[3,2]],  [[1,1],[2,1]] ],
[2,1],[3,2]],  [[1,1],[2,1]] ]
```

The generator randomly sets the number of occurrences with a set decreasing probability of a number (from 1 to maxOcc parameter) being chosen.

The generator has a number of parameters that can be set to modify its performance. The most important are:

1) N - the size of the generated set; by default 1000
2) maxObj - the maximum number of "objects" (tables, indexes, lob files ertc.) - 20 by default
3) maxOcc - the maximum number of object occurrences in an element - by default 10

In the above example, the maxObj has been set to 3, therefore e=[1, 2, 3] is a set of all object appearing in WL

The generator starts by randomly selecting a number of objects that are to appear in a generated multiset. It then randomly (but according to a specified distribution parameter) select objects and their occurrence numbers. Then the generator proceeds with this procedure to generate following multisets. However, there is a chance (parametrized) that a multiset, that is to be generated, will be chosen from previously generated once. If such possibility is to happen, another parameter specifies how many following multisets should also be copied. Of course the mentioned parameter only specifies the maximum amount of copied elements, but the actual amount is randomly selected. We proceed in such manner until all N multisets have been generated.

The generator is equipped in a high amount of parameters influencing the pseudo-randomness. Therefore it is possible to estimate the number of potential periodic patterns present in the generated data

V. Conclusions and further works

Testing procedures for the algorithm included the analysis of the query series transformation to their multiset representation. Such representation was generated form the obtained database traces and resulted in a multiset sequence of a type described in chapter IV. Then followed the tests of the performance of the designed algorithm on the test data. The generated test-data was used to firstly verify the correctness of the main heuristic algorithm when it comes to the recognition of periodic patterns when searching for the optimal workload reconstruction. The second task was to aid the research on the reconstruction quality measure and time-efficiency of the reconstruction algorithm. The overall goal was to develop a fast, statistically stable algorithm that works with an optimal time and space complexity. The algorithm has been implemented in Python 3 and the tests were not meant to evaluate the actual time performance, but rather they focused on testing the statistical stability and correctness of the results. Therefore, the performed tests were used for empirical verification of the correctness and stability of the algorithm's operation and have confirmed that not only is the algorithm stable but also optimal in its performance. Future work on the subject may include working on new heuristics combined with reconstruction algorithms' efficiency comparative analysis. Another aspect worth researching is the extended implementation of the reconstruction algorithm that takes into account the cost analysis derived from both the DBMS statistics and the relational algebra operators; the analysis of parallelization possibilities for periodic pattern discovery algorithms with the help of the GPU processing capabilities. Also it may be interesting to additionally enhance the heuristics with the computation of the reconstruction significance rate.

References

[1] M. Zimniak, M. Burzanska, and B. Franczyk, "On some heuristic method for optimal workload reconstruction," in *Proceedings of the 27th International Workshop on Concurrency, Specification and Programming, Berlin, Germany, September 24-26, 2018*, ser. CEUR Workshop Proceedings, B. Schlingloff and S. Akili, Eds., vol. 2240. CEUR-WS.org, 2018. [Online]. Available: http://ceur-ws.org/Vol-2240/paper5.pdf

[2] M. Zimniak, J. R. Getta, and W. Benn, "Predicting database workloads through mining periodic patterns in database audit trails," *Vietnam Journal of Computer Science*, vol. 2, no. 4, pp. 201–211, 2015.

[3] M. Zimniak and J. R. Getta, "On systematic approach to discovering periodic patterns in event logs," in *Computational Collective Intelligence - 8th International Conference, ICCCI 2016, Halkidiki, Greece, September 28-30, 2016, Proceedings, Part I*, ser. Lecture Notes in Computer Science, N. T. Nguyen, Y. Manolopoulos, L. S. Iliadis, and B. Trawinski, Eds., vol. 9875. Springer, 2016, pp. 249–259. [Online]. Available: https://doi.org/10.1007/978-3-319-45243-2_23

[4] M. Zimniak, J. R. Getta, and W. Benn, "Discovering periodic patterns in system logs," in *Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen*, 2014, pp. 156–161.

[5] ——, "Deriving composite periodic patterns from database audit trails," in *Asian Conference on Intelligent Information and Database Systems*. Springer, 2014, pp. 310–321.

[6] K. Pommerening, "Cryptology part i: Classic ciphers (mathematical version)," 2014.

[7] E. LEHTONEN, "Reconstructing multisets over commutative groupoids and affine functions over nonassociative semirings," *International Journal of Algebra and Computation*, vol. 24, no. 01, pp. 11–31, 2014. [Online]. Available: https://doi.org/10.1142/S0218196714500027

[8] E. Lehtonen, "Totally symmetric functions are reconstructible from identification minors," *The Electronic Journal of Combinatorics*, vol. 21, no. 2, Apr. 2014. [Online]. Available: https://doi.org/10.37236/2863

[9] P. Wojtaszczyk, *A Mathematical Introduction to Wavelets*. Cambridge University Press, Feb. 1997. [Online]. Available: https://doi.org/10.1017/cbo9780511623790

[10] C. K. Chui, "Wavelets: A mathematical tool for signal analysis," 1997.

[11] A. B. Matos, "Periodic sets of integers," *Theoretical Computer Science*, vol. 127, no. 2, pp. 287–312, 1994.

[12] P. Serafini and W. Ukovich, "A mathematical model for periodic scheduling problems," *SIAM Journal on Discrete Mathematics*, vol. 2, no. 4, pp. 550–581, 1989.