COST MODEL FOR THE RECOMMENDATION OF MATERIALIZED WEBVIEWS

Ali Ben Ammar[1] and Abdelaziz Abdellatif[2]
[1]Institut Supérieur d'Informatique et de Gestion. Kairouan, Tunisia
ali.benammar@isd.rnu.tn

[2]Faculté des Sciences de Tunis. Tunis, Tunisia
abdelaziz.abdellatif@fst.rnu.tn

ABSTRACT

In this paper we present a cost model for the recommendation of candidate webviews. Our idea is to intervene at regular period of time in order to filter the candidate webviews which will be used by an algorithm for the selection of materialized webviews in data–intensive websites (DIWS). The aim is to reduce the complexity and the execution cost of the online selection of materialized webviews. A webview is a static instance of a dynamic web page. The materialization of webviews consists of storing the results of some requests on the server in order to avoid repetitive data generation from the sources. Our experiment results show that our solution is very efficient to specify the more profitable webviews and to improve the query response time.

KEYWORDS

Cost model, Materialization weight, Web-usage mining, Materialized Webview Selection, Quality of data, Quality of service.

1. INTRODUCTION

Webview materialization is a technique to improve query response time in data-intensive websites (DIWS). DIWS are characterized by a high volume of data stored in structured databases. The webview materialization resembles to the view materialization in databases and data warehouses [4,5,6,7,8,12,13,14,15,16,17,22]. This technique aims to reduce the cost of generating data from databases to serve repetitive user queries. Its principle is to store and then update the results of some repetitive queries that need an access to the data sources of the website. These results represent the materialized webviews which are instances of dynamic web pages. A webview is said profitable if its access cost is greater than its maintenance cost. In this case it is recommended for the materialization. When a user sends a request, the server checks whether this request has a response in the materialized webview repository or not. Then it serves the user query either from the materialized webview repository or by making access to the databases of the DIWS.

A policy of webview selection should respond to three questions: (i) when do we execute and re-execute the selection algorithm? (ii) How do we identify the materialized objects? And (iii) what are the constraints to be applied in order to respect the resources capacities? The result of each execution of the selection algorithm is called materialization plan [19]. This plan specifies the state of each candidate webview: materialized or not. So, the first question of a selection policy is about the replacement period of the materialization plan that is the time interval between each two consecutive executions of the selection algorithm. The second question is about the cost model which is used to evaluate the webview contribution. In general, this contribution depends

DOI : 10.5121/ijdkp.2012.2101

on the access and the maintenance costs which depend on the frequencies and the time responses of the requests.

In data warehouse environment, the view materialization was static that is the materialization system intervenes at regular and long time periods to refresh the materialization plan. This make the intervention cost low because the executions of the selection algorithm are less frequent. On the web, the materialization should be dynamic that is depending on the variation of the access and maintenance cost, the webview materialization system should intervene online to calibrate the materialization plan. The advantage of this dynamic selection and refreshment of materialization plan is the online reaction to the variation of the materialization plan profitability. In other words when the materialization plan becomes not profitable (the maintenance cost of materialized webviews becomes greater than their access cost) then it should be refreshed by the system. The major disadvantage of the dynamic or online selection of materialized webviews is the high intervention cost.

In order to resolve the problem of high intervention cost of online webview selection, we propose in this paper to filter the candidate webviews at regular periods of time, called selection periods or sp. Our idea is that at the beginning of each selection period sp_n we should estimate the webviews which will be profitable along sp_{n+1} that is the webviews with positive contribution (high access cost and less maintenance cost). The specified webviews are considered as the candidate webviews for the algorithm of webviews selection. In other words, if there is an online selection or refreshment of the materialization plan along sp_{n+1}, the selection algorithm will use only these specified webviews. The aim of this filtering is to reduce the number of candidate webviews which will improve the complexity and the execution cost of the online selection algorithm. So, the main contribution of our solution is to improve the intervention cost. We have applied the web usage mining to filter the candidate webviews.

In the next section we will present the related works. The section 3 presents our solution. We illustrate our approach by an example. The section 4 describes the experiment results. The section 5 is the conclusion.

2. RELATED WORKS

The materialization has been widely used in database and data warehouse environments as a technique to improve query response time. On the web, the data materialization has mainly treated by A. Labrinidis and N. Roussopoulos [18, 19, 20, 21]. In [21] they have compared the different forms of data materialization on the DIWS. These forms are: virtual webview, materialized webview on the web server, materialized view on the database. The experiment results have shown that the materialization of webviews is the best form. In [9], we have combined the materialization of webview and the materialization of view and we have found that this form is the best. In [19] A. Labrinidis and N. Roussopoulos have proposed a cost model to select materialized webviews. This model is defined by two metrics which are introduced by the authors: the quality of service (QoS) which represents the access rate to materialized data and the quality of data (QoD) which represents the access rate to fresh data. So, the QoS illustrate the query response time. In [20] A. Labrinidis and N. Roussopoulos have developed a complete approach, called OVIS(θ), for the online selection of materialized webviews. This approach presents the selection problem as a combinatorial optimization problem. It consists of maximizing the QoS under the constraint to respect a degree of QoD which is specified by θ. The techniques of data mining have recently been used to the selection of materialized data either on data warehouse [4, 5, 6, 22] or on the web [10, 11, 23]. In [23], S. Saidi et al. have proposed an approach that firstly analyzes the historic of user navigation and then selects the set of

materialized webview which improve the query response time and respect maintenance and storage constraints.

So, after the analysis of these works we have remarked that the online intervention may take a long time because of the complexity of the selection algorithm, the high intervention frequency or the high number of candidate webviews. This long time may decrease the server performance and consequently it will have a negative impact on the query response time. In this paper we present an approach to improve this intervention load. Our solution consists of reducing the number of candidate webviews which affect the intervention load.

To the best of our knowledge, the intervention load has not yet been treated in the approaches of materialized webview selection. Also, we are not aware of any work filtering webviews for the online selection.

3. APPROACH PRESENTATION

3.1 Principle

To recommend candidate webviews we intervene at regular periods. We call these periods "selection period" or sp. The set of candidate webviews is filtered at the beginning of each sp. The rules which are used to filter the candidate webviews are regularly created at a multiple of sp, for example at each 2 sp, 3 sp…. We have applied web usage mining tools to produce these rules. Figure 1 summarizes the general principle of our approach.
\

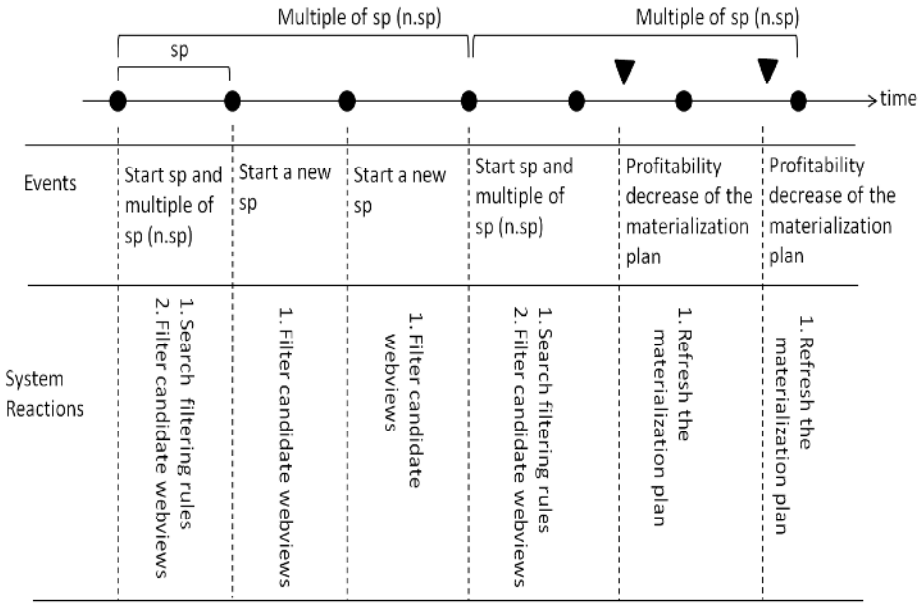

Figure 1. General principle of recommending candidate webviews

In the rest of this section we will explain how filtering the candidate webviews.

3.2 Filtering candidate webviews

We have used the web usage mining to analyze the historic of access and maintenance queries. In particular, we have applied sequential patterns and association rules as techniques of web usage

mining. Then we deduce the rules that should be used to recommend the candidate webviews. So, the steps that we propose to filter candidate webviews are the following:

- Analyze the navigation and the maintenance historic
- Search the frequent sequential patterns
- Apply the frequent sequential patterns to filter webviews
- Search the frequent association rules
- Apply the frequent association rules to filter webviews
- Apply the cost model to recommend webviews

3.2.1 Analyze the navigation and the maintenance historic

The analysis of the historic consists of identifying the profitable webviews for each sp. That is the webviews which have a cost access greater than the maintenance cost along the specified sp. We have represented the result of this analysis in a matrix H as follow:

$$H(i,j) = \begin{cases} 1 \; if \; the \; webview \; w_i \; was \; profitable \; along \; sp_j \\ 0 \; if \; not \end{cases}$$

3.2.2 Search the frequent sequential patterns

Sequential patterns are introduced in [3]. They consist of detecting the frequent sequential behaviors in a set of transactions. In our case, we have used this technique to detect the frequent sequences of profitable webviews. For example the frequent sequence of profitable webviews (AB)(BCD)(DE) means that along several sequences of selection periods of the form sp_j, sp_{j+1}, sp_{j+2} , the webviews A and B have produced a materialization profit along sp_j, the webviews B, C and D have produced a materialization profit along sp_{j+1}, and the webviews D and E have produced a materialization profit along sp_{j+2}. To extract all the frequent sequential patterns we have applied the algorithm cSPADE[25] on the matrix H.

3.2.3 Apply the frequent sequential patterns to filter webviews

The frequent sequential founded after applying the algorithm cSPADE are used to filter the candidate webviews. Our idea is as follow:

Suppose that we are at the end of sp_n that is we should filter candidate webview to the period sp_{n+1}. If we have a frequent sequence of profitable webviews of the form (AB)(BCD)(DE) and if the webviews A and B were profitable along sp_{n-1} and the webviews B, C and D were profitable along sp_n so we can conclude that there is a positive probability that the webviews D and E will be profitable along sp_{n+1}. We call this probability materialization weight. So, for each webview there is specific materialization weight.

To calculate the materialization weight of a webview w_i we have proposed an algorithm in [10]. This weight depends on the number of sequential patterns that may be used to filter the webview w_i and the values of these sequential patterns. The value of sequential patterns s_k is defined as follow:

$$weight(s_k) = supprot(s_k) \times \left(1 - \frac{1}{1 + length\left(ant(s_k)\right)}\right) \qquad (1)$$

$support\,(s_k)$ represents the support of s_k that is the frequency of the sequence s_k. This frequency is extracted from the matrix H. $length(ant(s_k))$ represents the number of webviews in the antecedents of the sequence s_k. The antecedents of a sequence are the profitable webviews of the recent selection periods. In the example here above, if we suppose that

s_k = (AB)(BCD)(DE) then $ant(s_k)$ = (AB)(BCD). We have chosen the antecedents of the sequential patterns in formula 1 because they represent the indicators to the profitability of the webviews of the consequence of the sequence which is in our example (DE).

The materialization weight of a webview w_i is defined as follow:

$$weight_{seq}(w_i) = max\{weight(s_k)/s_k \square VS(w_i)\} \times (1 - \frac{1}{(1 + |VS(w_i)|)}) \quad (2)$$

$VS(w_i)$ is the set of valid sequential patterns of w_i. A sequence s_k is said valid for w_i if it is (i) frequent,(ii) its antecedents are recently meted that is the webviews of its antecedents were profitable along the recent periods, and (iii) it has w_i in its consequence. The recommended webviews are those having a strictly positive materialization weight that is $weight_{seq}(w_i) > 0$. In formula 2 we have made a compromise between two main parameters: the weights of the sequential patterns and their number. So, the most recommended webviews will be those having a high number of sequential patterns with high weights.

The result of this step is a set of filtered webviews FWSP (Filtered Webviews by Sequential Patterns). In the next steps we will apply a second technique of web usage mining to filter other webviews which are not in FWSP. This technique is the association rules.

3.2.4 Search the frequent association rules

Association rules are introduced in [1]. They consist of searching correlations between items in a base of transactions. In our context we use association rules to search the correlations between profitable webviews. These correlations between webviews are based on the state of profitability. So, some webviews are correlated either because they share the same sources and so they have the same maintenance cost or because they share the same access path and so they have the same access cost. Consequently these webviews will have the same states of profitability.
An association rule is represented as follow:

$$R(a\%, b\%): X \rightarrow Y$$

$a\%$ is the percentage of the transactions that contains the sets X and Y. $a\%$ is said the support of R.

$b\%$ is the percentage of the transactions containing Y from those containing X. $b\%$ is said the confidence of R.

In our context, a% is the percentage of the selection periods which have the webviews of the sets X and Y as profitable webviews. $b\%$ is the percentage of the selection periods along which the webviews of Y have produced a profit from those along which the webviews of X have produced a profit.

In order to avoid re-filter webviews which are filtered by the sequential patterns we have chosen to search binary association rules between two sets of webviews: FWSP et $\overline{FWSP}$ (not Filtered Webviews by Sequential Patterns). Binary association rule means that the number of the items of X and Y is one. We have chosen binary association rules in order to exploit the materialization weight of the webviews of FWSP. So, the materialization weight to be calculated to filter webviews based on the association rules will depend of the parameters of the association rules (support and confidence) and of the materialization weight calculated by the sequential patterns. Figure 2 summarizes how to apply the association rules to filter webviews. Valid association

rules means that the antecedent (X) of the rule R should be included in FWSP and the consequence of R should be included in $\overline{FWSP}$. The set of binary and valid association rules of a webview w_i is $BVR(w_i)$.

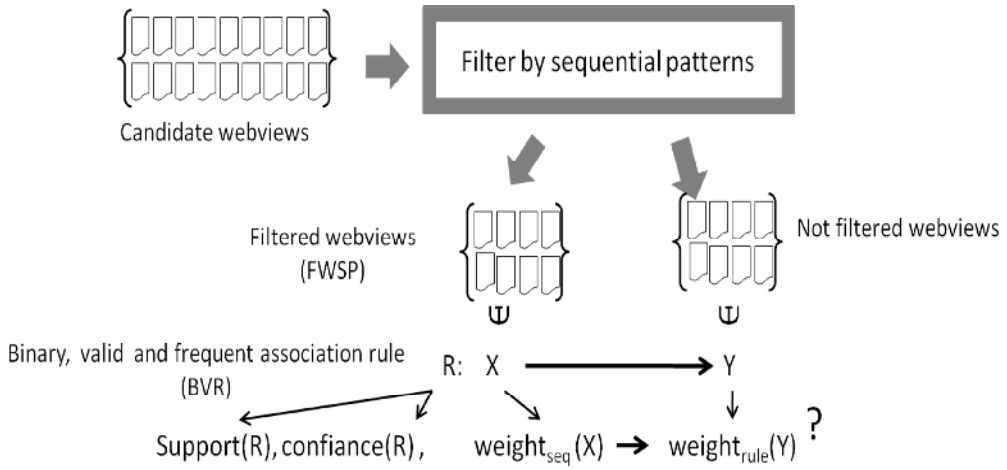

Figure 2. Applying association rules for the recommendation of candidate webviews

To extract the association rules we have used the algorithm CBAR [24] and FAST [2]. The first allows searching the frequent itemsets. The second is used to identify the frequent association rules.

3.2.5 Apply the frequent association rules to filter webviews

After searching the association rules, we have developed an algorithm, which is published in [11], to filter the webviews. This filtering is based on the materialization weight $weight_{rules}(w_j)$. To calculate this weight we have firstly evaluated the binary and valid association rules. That is we have calculated a weight for each rule $r_j \in BVR(w_i)$ as follow:

$$weight(r_j) = support(r_j) \times confidence(r_j) \qquad (3)$$

The materialization weight of a webviews $w_j \in \overline{FWSP}$ is defined as follow:

$$weight_{rules}(w_j) = max\{weight(r_j) \times weight_{seq}(ant(r_j)) \,/\, r_j \in BVR(w_j)\} \times \left(1 - \frac{1}{1+|BVR(w_j)|}\right) \qquad (4)$$

In formula 4 we have considered a compromise between the values of the association rules, the values of their antecedents and the number of valid association rules. Consequently, the most recommended webviews will be those having a high number of valid association rules with high weights.

After this step, a webview w_i is filtered either because $weight_{seq}(w_i) > 0$ or because $weight_{rules}(w_i) > 0$.

3.2.6 Apply the cost model to recommend webviews

The materialization weight calculated either by the sequential patterns of by association rules is an indicator of the webview profitability. This indicator is extracted from a long or width historic.

But on the web, the variation of the webview profitability is very frequent because of the events which may occur at any moment. For example, a webview may be deleted, the interest to a webview may be changed, the sources of a webview may be changed and so the variation of the maintenance cost of this webview will be modified. For all these reasons, we propose to test the result of the previous steps of filtering candidate webviews. This test consists of calculating for each filtered webview the average materialization profit. This average value is calculated from some recent selection periods. Our idea is to verify if the filtered webviews still used and still provide a profit. So the average profit is defined by formula 5:

$$Profit_{avg}(w_i, sp_n) = \frac{1}{x}\sum_{j=n-x-1}^{n} profit(w_i, sp_j) \qquad (5)$$

In formula 5:

- n: is the index of the last and recent selection period.
- $Profit_{avg}(w_i, sp_n)$: the average materialization profit of w_i calculated at the end of the selection period sp_n;
- x: is the number of the recent selection periods which are used to calculate the average of the materialization profit.
- $profit(w_i, sp_j)$: the materialization profit of w_i along the selection period sp_j
-

The cost model that we have used to recommend webviews is represented by that formula 6:

$$max\{weight_{seq}(w_i), weight_{rules}(w_i)\} \times Profit_{avg}(w_i, sp_n) > 0 \qquad (6)$$

With this model we impose that a webview w_i is recommended as a candidate webviews only if: (i) it is recommend by the sequential pattern ($weight_{seq}(w_i) > 0$) or it is recommend by the association rules ($weight_{rules}(w_i) > 0$); and (ii) it has a positive average materialization profit.

3.2.7 Example

In this example we will illustrate the steps of filtering candidate webviews.

Suppose that after analyzing the navigation and the maintenance historic we have found the following matrix H of profitable webviews. The value 1 means that the webview w_i has produced a profit along the selection period sp_j. The value 0 means that the webview w_i has produced a loss along the selection period sp_j.

Table 1. Example of a matrix of webview profitability

Webview/period	sp_1	sp_2	sp_3	sp_4	sp_5	sp_6	sp_7	sp_8	sp_9	sp_{10}	sp_{11}	sp_{12}
A	1	0	1	1	0	0	1	0	1	1	1	0
B	1	1	0	1	0	0	1	1	1	1	1	1
C	0	1	0	1	1	0	0	1	1	0	1	0
D	1	0	1	1	1	1	1	1	1	1	1	1
E	1	1	0	1	1	0	1	1	1	0	1	1
F	1	0	1	1	0	1	1	0	0	1	1	0

In order to apply the cSPADE algorithm for the extraction of sequential patterns we should define the width of the sequences that is the number of transactions to be considered in each sequence. In our context, this width will correspond to the number of selection periods to be considered in

each sequence. Based on the data of table 1 we have constructed the sequences of profitable webviews which are presented in table 2. The width of these sequences is fixed to 4.

The first sequence of table 2 means: the webviews A, B, D, E and F were profitable along sp_1, the webviews B, C and E were profitable along sp_2, the webviews A, D and F were profitable along sp_3 and the webviews A, B, C, D, E and F were profitable along sp_4.

Table 2. sequences of profitable webviews.

Selection periods	Sequences of profitable webviews
sp_1, sp_2, sp_3, sp_4	(ABDEF) (BCE) (ADF) (ABCDEF)
sp_2, sp_3, sp_4, sp_5	(BCE) (ADF) (ABCDEF) (CDE)
sp_3, sp_4, sp_5, sp_6	(ADF) (ABCDEF) (CDE) (DF)
sp_4, sp_5, sp_6, sp_7	(ABCDEF) (CDE) (DF) (ABDEF)
sp_5, sp_6, sp_7, sp_8	(CDE) (DF) (ABDEF) (BCDE)
sp_6, sp_7, sp_8, sp_9	(DF) (ABDEF) (BCDE) (ABCDE)
$sp_7, sp_8, sp_9, sp_{10}$	(ABDEF) (BCDE) (ABCDE) (ABDF)
$sp_8, sp_9, sp_{10}, sp_{11}$	(BCDE) (ABCDE) (ABDF) (ABCDEF)
$sp_9, sp_{10}, sp_{11}, sp_{12}$	(ABCDE) (ABDF) (ABCDEF) (BDE)

Now, the algorithm cSPADE will search the frequent sequential patterns in these sequences. For this reason we should define the minimal support ($minSupp$) which is the reference to decide if a sequential pattern is considered frequent or not. The width of the sequences and the minimal support are the entry parameters of cSPADE. For example if we consider that the $minSupp$ is 30% then we have the frequent sequential patterns which are presented in table 3. This set of sequential patterns is not complete but it contains just some examples. The support represents the frequency of the sequential patterns. The first example of sequential patterns called s_1 means that the webviews A and B were profitable along sp_{t-4}, the webview D was profitable along sp_{t-3}, the webview D was profitable along sp_{t-2} and the webviews B, D were profitable along sp_{t-1}. In addition, s_1 is included in three sequences of table2. Consequently its frequency or support is $3/9 = 33\%$. Since its support is greater than the $minSupp$ it is considered as frequent.

Table 3. Examples of frequent sequential patterns.

Id	Sequential patterns	Supports
s_1	(AB) (D) (D) (BD)	3/9 = 33%
s_2	(BDE) (CE) (DF) (ABDEF)	3/9=33%
s_3	(A) (D) (D) (D)	4/9= 44%
s_4	(D) (B) (D) (D)	6/9= 67%

Now, we suppose that we are at the end of the selection period sp_{12} and we should filter candidate webviews for the selection period sp_{13}. Firstly we search, for each webview of the set {A, B, C, D, E, F}, the set of valid sequential VS from the sequences of table 3. So:

- VS(A)={}. The only frequent sequence from table 3 that can be used to filter A is s_2. This is because the consequence of s_2 ,which is the last set of webviews (ABDEF), contains A. But since the antecedents of s_2 does not contains in the recent historic, s_2 is considered as not valid for the webview A. The antecedents of s_2 are the sets of webviews (BDE) (CE) (DF). The recent historic is composed of the sets of profitable webviews of $sp_{10}, sp_{11}, sp_{12}$ which are (ABDF) (ABCDEF) (BDE). So $\{B, D, E\} \not\subseteq \{A, B, D, F\}$ and $\{D, F\} \not\subseteq \{B, D, E\}$. In other words there is no indicator in the recent historic that s_2 will be occurred and that its consequence will be met in the next selection period.

- VS(B) = {s_1}. This is because B belongs to the consequence of s_1 that is Bϵ{BD} and the antecedents of s_1 are included in the recent historic.
- VS(C) = {};
- VS(D) = {s_1, s_3, s_4};
- VS(E) = {};
- VS(F) = {};

To calculate the materialization weight of the webviews B and D which have valid sequences, we should firstly evaluate these valid sequences according to formula 1. So,

$$weight(s_1) = 33\% \times \left(1 - \frac{1}{1+4}\right) = 0.27$$

$$weight(s_3) = 44\% \times \left(1 - \frac{1}{1+3}\right) = 0.33$$

$$weight(s_4) = 67\% \times \left(1 - \frac{1}{1+3}\right) = 0.5$$

And the materialization weights of B and D will be calculated according to formula 2:

$$weight_{seq}(B) = max\{0.27\} \times \left(1 - \frac{1}{(1+1)}\right) = 0.135$$

$$weight_{seq}(D) = max\{0.27, 0.33, 0.5\} \times \left(1 - \frac{1}{(1+3)}\right) = 0.375$$

Since the webview D has more valid sequences with high weights it becomes the most recommended webview and we can remark this from the values of the materialization weights.

After this step we have two sets of webviews: those filtered by the sequential patterns FWSP={B,D} and those not Filtered $\overline{FWSP} = \{A, C, E, F\}$. We should now search and apply association rules to filter webviews from the set $\overline{FWSP}$. For this reason we apply the algorithms CBAR and FAST on the matrix H represented by table 1. The entries of the algorithm FAST are the minimal support ($minSupp$) and the minimal confidence ($minConf$) which are needed to decide if the association rule is frequent or not. An association rule is considered frequent if and only if its support is greater than $minSupp$ and its confidence is greater than $minConf$. In table 4 we have represented some binary and frequent association rules when the $minSupp$ = 30% and the $minConf$ = 50%. The weights of rules are calculated according to formula 3. This list of association rules is not complete but it just represents some examples of rules. Consequently, we will filter webviews according to this list.

Table 4. Examples of frequent and binary association rules.

Id	Binary association rules	Support	Confidence	Weight
r_1	B→A	6/12=50%	6/9=67%	0,33
r_2	D→A	7/12=58%	7/11=64%	0,37
r_3	E →A	5/12=42%	5/9=55%	0,23
r_4	E→B	8/12=67%	8/9=89%	0,6
r_5	F→D	7/12=58%	7/7 =100%	0,58
r_6	A→E	5/12=42%	5/7=71%	0,3
r_7	B→F	5/12=42%	5/7=71%	0,3
r_8	D →F	7/12=58%	7/11=64%	0,37

The association rule r_1 means that along 6 selection periods out of 12, the webviews A and B were both profitable and that along 6 selection periods out of the 9, where B was profitable, A was also profitable. The association rules r_3, r_4, r_5 and r_6 are not valid because their antecedents, which are respectively E, E, F and A do not belong to FWSP. Also, r_4 and r_5 are not valid because their consequences, which are respectively B and D, do not belong to $\overline{FWSP}$.

The sets of binary and valid association rules for the webviews of $\overline{FWSP}$ are:

- BVR(A)={r_1, r_2};
- BVR(C)={};
- BVR(E)={};
- BVR(F)={r_8, r_{10}};
-

The materialization weights of the webviews of $\overline{FWSP}$ are calculated according to formula 4 and they are:

$$weight_{rules}(A) = max\{0.33 \times 0.135;\ 0.37 \times 0.375\} \times \left(1 - \frac{1}{1+2}\right) = 0.092$$

$$weight_{rules}(F) = max\{0.3 \times 0.135;\ 0.37 \times 0.375\} \times \left(1 - \frac{1}{1+2}\right) = 0.092$$

Finally and according to the sets of sequential patterns and association rules used in this example, the filtered candidate webviews are A, B, D and F.

Now, we will use the definition of the materialization profit, which is proposed in [19], to calculate the average materialization profit. This definition is simple and it is written as follow:

$$Profit(w_i, sp_j) = \frac{frequency_{acc}(w_i, sp_j)}{1 + frequency_{maint}(w_i, sp_j)} \qquad (7)$$

In [19], the authors suppose that the cost of an access query to virtual (not materialized) version of a webview w_i and the cost of a maintenance query of materialized version of w_i are equals. Consequently, if the access frequency of w_i is greater than its maintenance frequency then it materialization will be profitable. In other words, if the result of formula 7 is greater than 1 then w_i is considered profitable. They have used 1 in the denominator in order to produce a legal number when the maintenance frequency is equal to zero.

In this example we suppose that the number of recent periods to be used for the calculation of the average materialization profit is $x = 5$. We suppose that the access and the maintenance frequencies of the webviews A, B, D and F are those in table 5. In this table we have calculated the average materialization profit according to formula 7.

So from table 5 we deduce that the webview F will not be recommended because its average profit is less than 1. The result after applying the cost model defined by formula 6 will be negative. This is because the materialization weight of F, which is founded by applying association rules, is positive however this webview F has produced a loss of materialization over the recent 5 selection periods since the result of applying formula 7 has given in table 5 a value less than 1.

Table 5. Examples of average materialization profits.

	Access frequencies					Maintenance frequencies					Average profit
Webview\period	sp_8	sp_9	sp_{10}	sp_{11}	sp_{12}	sp_8	sp_9	sp_{10}	sp_{11}	sp_{12}	
A	180	300	270	365	156	230	165	138	179	232	1,44
B	410	530	110	287	298	321	235	80	211	174	1,59
D	125	153	75	96	143	43	54	63	19	52	2,86
F	236	384	367	278	234	450	532	302	215	366	0,87

4. EXPERIMENTS

In our experiments we have implemented 5 versions of a DIWS. In the first version we have implemented 100 webviews, in the second 250, in the third 500, in the fourth 750, and in the fifth 1000 webviews. The DIWS that we have implemented concerns an online library and the webviews represent the description pages of books, the lists of books by theme, by author,…
In order to construct the access and the maintenance historic we have simulated the load of DIWS over four weeks. We have fixed the length of the selection period to 12 hours. In these simulations, the access query frequency is fixed to 5 queries by second and the maintenance query frequency is fixed to 1 query by second. The extraction algorithms of sequential patterns and association rules are executed at the end of each week that is after 14 selections periods.
For the filtering of materialized webviews we have used the following parameters:

- The width of sequences which is used to extract sequential patterns is 3;
- The minimal support used to extract the sequential patterns is 50%;
- The minimal support used to extract the association rules is 30%;
- The minimal confidence used to extract the association rules is 50%;

In order to evaluate our approach we have compared 4 cost models for the filtering of materialized webviews which are:

- Recommendation based on the average profit that is a webview w_i is recommended only if
  $Profit_{avg}(w_i, sp_n) > 0$;
- Recommendation based on the materialization weights that is a webview w_i is recommended only if
  $\max\{weight_{seq}(w_i), weight_{rules}(w_i)\} > 0$
- Recommendation based on the cost model defined by formula 7 that is a webview w_i is recommended only if
  $\max\{weight_{seq}(w_i), weight_{rules}(w_i)\} \times Profit_{avg}(w_i, sp_n) > 0$
- Recommendation based on the method RPE (Reverse Prediction Error method) which is applied in [20]. RPE allows the estimation of the future value á of a parameter from its recent past values a and m. It is defined as follow: $á = (1 - g).a + g.m$ with $g \in [0,1]$. In [20], the authors have used this method to estimate the access frequency and the maintenance frequency of the webviews in the next period sp_{n+1}. So, to estimate the access frequency they have considered a as the value of the access frequency along sp_n and m its value along sp_{n-1}. They apply the same policy to estimate the maintenance frequency. Then they estimate the materialization profit, based on the estimated access and maintenance frequencies, to recommend the materialized webviews.

In the fourth week we have applied, in parallel with the simulation, the 4 cost models for the recommendation of materialized webviews. For each model we have materialized the recommended webviews. Then and for each selection period:

- We identify the set of profitable webviews and the set of not profitable ones;
- We calculate the quality of data (QoD) produced by the set of recommended webviews. The QoD represent the rate of access to fresh data. It demonstrates the impact of the materialized webviews on the maintenance load. In other words when the maintenance of the materialized webviews is not costly the QoD will be improved;
- We calculate the quality of service (QoS) produced by the set of recommended webviews. The QoS represents the query response time.

In the first experiment we have studied the success rate of webview recommendation on the DIWS with 100 webviews. For this reason we have applied the method presented in table 6 to measure the success rate. In table 6 we have two types of recommendation models: recommendation by using materialization weight (mw) and recommendation based on the recent historic. In the first type we are not sure that the materialization of the webview w_i will produce a profit. But we have a probability $weight(w_i)\ or\ mw(w_i)$ that the materialization of w_i will produce a profit and a probability $(1 - mw(w_i))$ that the materialization of w_i will produce a materialization loss. In the second type we are either sure from the profitability of the materialization of w_i or sure from its materialization loss. In the first case the probability of recommendation is 100% and in second case the probability of recommendation is 0%. Consequently, to evaluate the recommendation decisions we have affected for each couple (decision, cost model) the corresponding probability. For example if we recommend, at the beginning of sp_n, the webview w_i with our cost model and if the materialization of w_i has produced a profit along sp_n then our decision costs $mw(w_i)$ else it costs $1 - mw(w_i)$. If we recommend, at the beginning of sp_n, the webview w_i with RPE model and if the materialization of w_i has produced a profit along sp_n then this decision costs 1 else it costs 0.

Table 6. A method for calculating success rate.

		Values of recommendation decisions			
Recommendation decision at the beginning of sp_n	**State along sp_n**	**Materialization Weight (mw)**	**Average profit**	**Our cost model**	**RPE**
w_i is Recommended	w_i has produced a profit	$mw(w_i)$	1	$mw(w_i)$	1
	w_i has produced a loss	$1 - mw(w_i)$	0	$1 - mw(w_i)$	0
w_i is not recommended	w_i has produced a profit	$1 - mw(w_i)$	0	$1 - mw(w_i)$	0
	w_i has produced a loss	$mw(w_i)$	1	$mw(w_i)$	1

To calculate the average success rates we have calculated the averages of the values of the recommendation decisions of the 100 webviews over the 14 selection periods. These rates are presented in figure 3. The results of this experiment show that our cost model is the best for the

recommendation of candidate webviews. In the future work we will demonstrate how to exploit these probabilities of recommendation in the selection of materialized webviews.

In the second experiment we have compared the average QoD and QoS produced by applying the cost models: our cost model and RPE. In this experiment we have studied the variation of these parameters according to the size of the DIWS. The size of DIWS is illustrated by the number of webviews. The results of this experiment are presented in table 7.

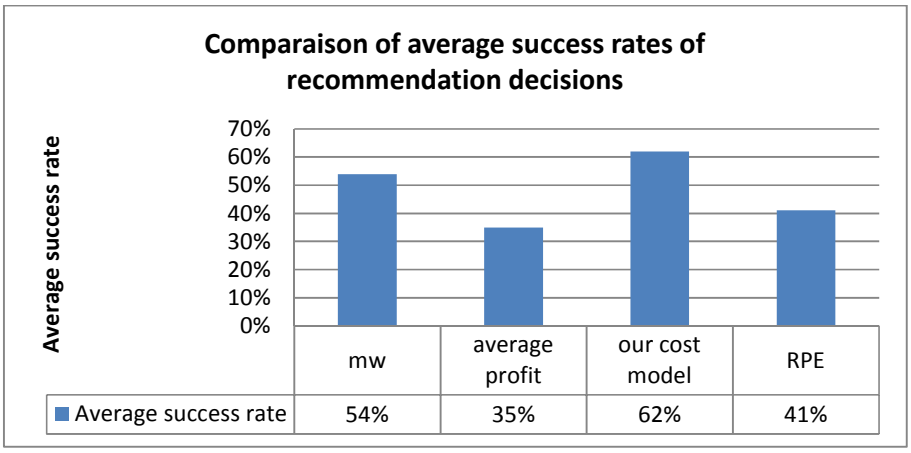

Figure 3. Average success rates of 4 cost models for the recommendation of webviews

Table 7. Variation of QoD and QoS according to the DIWS size.

DIWS size (Number of webviews)		**100**	**250**	**500**	**750**	**1000**
Average QoD	RPE	0,96	0,9	0,8	0,7	0,63
	Our cost model	0,96	0,94	0,89	0,8	0,76
Average QoS in ms	RPE	13,39	9,35	13,45	16,3	20,04
	Our cost model	9,31	8,21	11,14	13,59	9,11

The results of this experiment show that the webviews recommended by our cost model are more profitable than those recommended by RPE for the 5 versions of DIWS. Compared with RPE, the recommended webviews of our cost model are characterized by a small maintenance cost which has improved the QoD. Also, they are characterized by a good contribution in the improvement of query response time which is illustrated by the improvement of QoS. This improvement is more important in the DIWS with high size. This is because in such DIWS the number of candidate webviews is high which makes the access and maintenance queries distributed over a big number of webviews. Consequently, the repetition of the recent behaviors (of the recent historic) will be less probably. In such case it is necessary to study the wide historic in order to well recommending the candidate webviews. In table 7, this improvement has exceeded 50% for the DIWS with 1000 webviews that is we have reduced the QoS of RPE which is 20.04 ms to 9.11 ms.

As a general conclusion from all these experiments, our cost model for the recommendation of candidate webviews has exceed the existing models to the improvement of both QoD and QoS. Our model is more efficient for the DIWS with high size.

5. CONCLUSION

In this paper we have presented a cost model for the recommendation of materialized webviews. Our main contribution is to filter at regular periods of time the candidate webviews. The aim is to reduce the cost of online selection of materialized webview. Our experiment results have shown that the proposed cost model can improve the query response time of DIWS with different size without decreasing the degree of QoD. They have shown also that, compared to existing models, our solution is more efficient for the DIWS with high size. In the future works we will attempt to integrate this model in a complete approach for the online selection of materialized webview.

REFERENCES

[1] R. Agrawal, T. Imielinski et A. N. Swami. “Mining association rules between sets of items in large databases”. In Proceedings of the ACM SIGMOD, pages 207–216, Washington, D.C. 1993

[2] R. Agrawal, and R. Skirant. “Fast algorithms for mining association rules”. In Proceedings of the 20th Intl. Conference on Very Large Databases, pages 478-499. Santiago, Chile, June 1994.

[3] R. Agrawal, and R. Skirant. “Mining sequential patterns”. In Proceedings of the 11th international conference on data engineering (ICDE’95) , pages 3-14. 1995

[4] K. Aouiche, and J. Darmont. “Data mining-based materialized view and index selection in data warehouses”. J. Intell. Inf. Syst. 33(1), pages 65-93. 2009

[5] K. Aouiche, P. Jouve, and J. Darmont. “Clustering-Based Materialized View Selection in Data Warehouses”. In 10th East-European Conference on Advances in Databases and Information Systems (ADBIS 2006), Thessaloniki, Greece, Vol. 4152 of LNCS, pages 81-95. 2006

[6] B. Ashadevi, R. Balasubramanian. “Optimized Cost Effective Approach for Selection of Materialized Views in Data Warehousing”. JCS&T Vol. 9 No. 1. Pages 21-26. April 2009.

[7] X. Baril, Z. Bellahsene. “Selection of Materialized Views: A Cost-Based Approach”. CAiSE 2003, pages 665-680. 2003

[8] L. Bellatreche, K. Boukhalfa. “Une répartition statique et dynamique de l’espace entre les vues matérialisées et les index dans les entrepôts de données”. International Symposium on Programming and Systems (ISPS'05), USTHB - Alger, 2005

[9] Ben Ammar, A. Abdellatif, and H. Ben Ghezala. “Forms of Data Materialization in Data-Intensive Web Sites”. IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.12, pages 84-88. December 2006

[10] Ben Ammar, M. Badis, A. Abdellatif. “Motifs séquentiels pour la sélection des webviews à matérialiser”. 6ièmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 2010), Djerba, Tunisie, RNTI, Vol. B-6, pages 153-162 Cépaduès, Toulouse. Juin 2010.

[11] Ben Ammar, M. Badis, A. Abdellatif. “Web usage mining for the recommendation of materialized webviews”. In The 12th International Conference On Information Integration and Web-based Applications & Systems (iiWAS2010), Paris. France. November 8-10, 2010.

[12] H. Ben Ghezala, A. Abdellatif, A. Ben Ammar. “An Approach to Specify When Reselecting Views to be Materialized”. 1ères journées francophone sur les Entrepôts de Données et l'Analyse en ligne (EDA 2005), Lyon, Juin 2005; RNTI, Vol. B-1, Cépaduès, Toulouse, pages 161-176.

[13] H. Gupta. “Selection of Views to Materialize in a Data Warehouse”. ICDT. 1997.

[14] H. Gupta, V. Harinarayan, A. Rajaraman, and J.D. Ullman. “Index Selection for OLAP”. ICDE. IEEE Computer Society, Washington, DC, pages 208-219. 1997

[15] H. Gupta, and I. S. Mumick. “Selection of Views to Materialize in a Data Warehouse”. IEEE Trans. on Knowl. and Data Eng. 17(1), pages 24-43. Jan. 2005.

[16] H. Gupta, I. S. Mumick. “Selection of Views to Materialize Under a Maintenance-Time Constraint”. ICDT. 1999.

[17] Y. Kotidis, N. Roussopoulos. “Dynamat: A dynamic view management system for data warehouses”. In ACM SIGMOD International Conference on Management of Data (SIGMOD 1999), pages 371-382. Philadelphia, USA, 1999.

[18] Labrinidis, Q. Luo, J. Xu, W. Xue. “Caching and Materialization in Web Databases”, Foundations and Trends in Databases Vol. 3, No. 2, pages 169-266. December 2009.

[19] Labrinidis, N. Roussopoulos. “Adaptive WebView Materialization”. WebDB'01, pages 85-90. 2001

[20] Labrinidis, N. Roussopoulos. “Exploring the tradeoff between performance and data freshness in database-driven Web servers”. The VLDB Journal, 13(3), pages 240-255, September 2004, Special issue with extended versions of the best papers from the VLDB 2003 Conference.

[21] Labrinidis , N. Roussopoulos. “WebView Materialization”. Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 367-378. May 15-18, 2000, Dallas, Texas, United States.

[22] H. Mahboubi, K. Aouiche et J. Darmont. “Materialized View Selection by Query Clustering in XML Data Warehouses”. 4th International Multiconference on Computer Science and Information Technology (CSIT 06), Amman, Jordan, volume 2, pages 68–77, April 2006.

[23] S. Saidi, Y. Slimani, and K. Arour. “Webview selection from user access Patterns”. In PIKM '07, pages 171-176. Lisboa, Portugal . November 2007.

[24] Y.J. Tsay, and J.Y. Chiang. “CBAR: an efficient method for mining association rules”. Knowledge-Based Systems 18, pages 99–105. 2005.

[25] M. J. Zaki. “Sequence mining in categorical domains: Incorporating constraints”. In Proceedings of the 9th international conference on information and knowledge management (CIKM 00), pages 422– 429. 2000.