

# Motifs Séquentiels pour la Sélection des Webviews à Matérialiser

Ali Ben Ammar\*  
Mouna Badis\*\*, Abdelaziz Abdellatif\*\*\*

\*Institut Supérieur d'Informatique et de Gestion de Kairouan  
3100, Kairouan. Tunisie  
ali.benammar @ isd.rnu.tn  
\*\*Faculté des Sciences de Tunis  
2092, Tunis. Tunisie  
manoui4@yahoo.fr  
\*\*\*Faculté des sciences de Tunis  
2092, Tunis. Tunisie  
abdelaiziz.abdellatif@fst.rnu.tn

**Résumé.** Dans cet article nous proposons une approche pour la recherche d'indicateurs renforçant le choix des webviews à matérialiser. Un webview matérialisé est une instance statique, d'une page web dynamique, stockée au niveau du serveur web à un point de temps. Il sert à réduire le coût de génération répétitive des données sources des requêtes. Notre contribution consiste à analyser l'historique du site web, moyennant la technique de web usage mining, pour calculer un poids de matérialisation pour chaque webview. Ce poids servira par la suite pour l'algorithme de sélection des webviews à matérialiser afin de décider si un webview est recommandé pour la matérialisation ou non. Plus précisément, il sert pour estimer le profit (en termes de temps de réponse) pouvant être produit par la matérialisation d'un webview. Nos résultats d'expérimentation ont montré que notre approche permet de réduire le risque de matérialisation des webviews, dont le profit n'est certain au moment de leur sélection, de plus de 20%.

## 1 Introduction

La matérialisation des données est une technique permettant d'optimiser le temps de réponse des requêtes des usagers. Elle consiste à stocker les résultats de quelques requêtes pour réduire le coût de génération répétitive des données à partir des sources. Récemment, elle a été appliquée au niveau des Sites Web à Utilisation Intensive de Données (DIWS, Data Intensive Web Sites) où on parle de matérialisation de webviews. Un webview est une page web dynamique dont les données sont extraites à partir de bases de données structurées. Ainsi, la matérialisation des webviews consiste à transformer les données dynamiques en données statiques, c'est-à-dire la création d'une instance statique d'une page web dynamique à un instant donné.

Dans un environnement de DIWS, un webview peut être soit virtuel, c'est-à-dire toujours généré en ligne à partir des sources, ou matérialisé, c'est-à-dire stocké au niveau du serveur. La matérialisation des données au niveau des DIWS peut prendre deux formes : matérialisation des vues (les données structurées représentant les sources des webviews) au

## Motifs Séquentiels pour la Sélection des Webviews à Matérialiser

niveau des bases de données sources ou matérialisation des webviews au niveau du serveur web. Par conséquent une approche de matérialisation doit répondre aux deux questions suivantes :

1. Quelles sont les formes de matérialisation à appliquer ?
2. Quelles sont les données à matérialiser ?

Les travaux de [Labrinidis et Roussopoulos, 1999], [Labrinidis et Roussopoulos, 2000], et [Ben Ammar et al., 2006], proposent des solutions à la première question. Ils ont défini plusieurs scénarios par combinaison de formes de matérialisation et de formes de webviews.

Les approches de sélection des webviews à matérialiser, qui nous intéressent dans ce travail, proposent des solutions à la deuxième question. Elles consistent à définir une politique de sélection qui comporte en général :

- Un modèle de coût : mesure le profit pouvant être engendré par la matérialisation d'un webview ;
- Contraintes de sélection : elles sont généralement des contraintes de ressources comme la contrainte d'espace de stockage ;
- Une période de sélection : c'est l'intervalle de temps séparant deux exécutions successives de l'algorithme de sélection.

Les solutions existantes de sélection des webviews , [Labrinidis et Roussopoulos, 1999 ], [Labrinidis et Roussopoulos, 2000], [Labrinidis et Roussopoulos, 2001], [Labrinidis et Roussopoulos, 2002 ] et [Saidi et al., 2007], réappliquent les paramètres utilisées par les approches de matérialisation des vues au niveau des bases de données et des entrepôts de données pour la définition de ces trois éléments (modèle de coût, contraintes, et période). Elles se limitent à la période de temps la plus récente pour estimer les profits des webviews pouvant être produits au cours de la période suivante. Ensuite, elles se basent sur ces profits estimés pour identifier les webviews à matérialiser. Ainsi, elles sont soit sûres de la rentabilité du webview au cours de la période suivante (webview matérialisé avec une probabilité de rentabilité =100%) soit non sûres (webview non matérialisé car la probabilité de rentabilité =0%). Cette démarche est inspirée des domaines de bases de données et d'entrepôts de données où les utilisateurs sont connus à l'avance, les volumes d'accès et de mises à jour sont presque stables d'une période à une autre. Cependant, dans un environnement de DIWS les utilisateurs sont généralement inconnus et les requêtes d'accès et de mises à jour changent d'une période à une autre. Par conséquent, la période la plus récente devient insuffisante, dans le contexte des DIWS, pour estimer les profits des webviews. En contre partie, il est démontré dans [Zhang et Qin, 2005] qu'il y a des comportements répétitifs d'accès et de mises à jour au cours du temps. Ces comportements s'expliquent par le fait que: (i) les mises à jour des sources sont parfois périodiques, (ii) les types de requêtes d'accès dépendent des périodes de mises à jour des sources, des catégories des usagers, de leurs régions et de leurs périodes d'utilisation du DIWS.

En partant des insuffisances des solutions existantes et en exploitant ces comportements répétitifs, nous proposons dans cet article une approche permettant la recherche d'indicateurs permettant de recommander les webviews les plus appropriés pour la matérialisation. Notre contribution essentielle, dans ce travail, consiste à utiliser une technique de web usage mining permettant d'analyser l'historique, des requêtes d'accès et de mise à jour de plusieurs périodes de sélection, pour extraire ces comportements répétitifs. Plus précisément, nous extrairons, les motifs séquentiels à partir des fichiers logs d'un DIWS. Ensuite, nous calculons un poids  $\in [0,1]$  de matérialisation pour chaque webview en se basant sur les motifs séquentiels trouvés et l'historique récent du DIWS. Ce poids servira par la suite pour

recommander la matérialisation d'un webview. Par rapport aux approches existantes, ce poids représente la probabilité que le webview soit profitable au cours de la période suivante. Selon notre approche cette probabilité prend une valeur entre 0% et 100%. Donc, notre travail s'arrête à la recherche de ce poids et nous ne présentons pas ici un algorithme pour la sélection des webviews à matérialiser.

Cet article est organisé comme suit : dans la section 2, nous présentons les travaux associés. Dans la section 3, nous présentons, avec détails, notre contribution. Cette section contient la présentation de l'approche, les résultats des expérimentations, et une discussion des résultats trouvés. La section 4 conclut le travail.

## 2 Travaux associés

La technique de data mining est peu utilisée pour la matérialisation des données. Dans [Aouiche et al., 2006], une approche basée sur la technique de data mining est proposée pour construire une liste de vues candidates pour un algorithme de sélection des vues matérialisées dans un data warehouse. Leur idée est de chercher, moyennant la technique de classification (clustering), des classes de requêtes similaires. Ensuite, ils créent des vues pouvant servir ces requêtes. Comme continuité de ce travail, une nouvelle approche est proposée dans [Aouiche et Darmont, 2009]. Elle consiste à analyser l'historique des accès pour extraire les ensembles fréquents d'attributs et les cooccurrences entre les attributs des requêtes. L'objectif, dans ce cas, est de déduire, moyennant la technique de règles d'association, une liste de candidats (vues et indexes) qui servira par la suite comme une entrée pour un algorithme de sélection. Le premier travail, de sélection des webviews matérialisés, basé sur la technique de web usage mining et tenant compte du comportement des clients, est proposé dans [Saidi et al., 2007]. Cette approche comporte deux étapes : (i) l'extraction, à partir des fichiers logs et en utilisant les règles d'association, des ensembles fermés fréquents de webviews candidats ; (ii) l'application des contraintes de stockage et de temps de maintenance pour la sélection des webviews à matérialiser à partir de la liste des webviews candidats trouvés à la fin de la première étape. A notre connaissance, les motifs séquentiels n'ont pas encore été utilisés pour la sélection des webviews à matérialiser.

## 3 Recherche d'indicateurs pour renforcer la matérialisation d'un webview

### 3.1 Concepts des motifs séquentiels

Les motifs séquentiels sont introduits dans [Agrawal et Srikant, 1995]. Ils décrivent les enchaînements d'ensembles d'éléments, couramment associés sur une période de temps bien spécifiée. Nous allons maintenant définir quelques concepts relatifs aux motifs séquentiels.

#### Définitions :

- $I$ , un ensemble d'éléments (appelés items).
- $T_i = \{i_1, i_2, \dots, i_n\}$ , une séquence d'items (appelé itemset) ;
- $S_j = \{T_1, T_2, \dots, T_m\}$ , une séquence de  $m$  itemsets donc elle est de taille  $m$  ;
- Une séquence  $S_a = \{a_1, a_2, \dots, a_n\}$  est incluse dans une autre séquence  $S_b = \{b_1, b_2, \dots, b_m\}$  (on note  $S_a \preceq S_b$ ) si et seulement s'il existe des entiers  $1 \leq i_1 < i_2 < \dots < i_n \leq m$  tels que  $a_1 \subseteq b_{i_1}, a_2 \subseteq b_{i_2}, \dots, a_n \subseteq b_{i_n}$  ;

## Motifs Séquentiels pour la Sélection des Webviews à Matérialiser

- Le support d'une séquence  $S_j$  est défini comme étant le pourcentage de séquences comportant  $S_j$  :  $\text{support}(S_k) = \frac{|\{S_k/S_j \leq S_{k,k=1,\dots,l}\}|}{l}$  ;  $l$  est le nombre total des séquences ;
- Le support minimum ou minSupp est une valeur fixée par l'utilisateur pour décider si une séquence est fréquente ou non. Une séquence est dite fréquente si son support est supérieur auminSupp.

Ainsi, l'extraction des motifs séquentiels consiste à chercher les séquences fréquentes.

### 3.2 Principe de l'approche

Notre approche consiste à rechercher des indicateurs (des motifs séquentiels) permettant de renforcer le choix d'un webview pour la matérialisation. Elle respecte les deux étapes suivantes :

1. Extraction des motifs séquentiels à partir des fichiers logs ;
2. Recherche des poids des webviews.

#### 3.2.1 Extraction des motifs séquentiels

##### Définition des concepts

Dans le contexte de sélection des webviews, nous allons proposer ces définitions:

- Une transaction ou un itemset  $p_t$  représente une période de sélection dans note contexte. La période de sélection est l'intervalle de temps séparant deux exécutions consécutives de l'algorithme de sélection.
- Un item  $w_i \in p_t$  représente un webview  $w_i$  qui a été profitable au cours d'une période de sélection  $p_t$ . Selon [Labrinidis et Roussopoulos, 2001], un webview est dit profitable si sa fréquence d'accès dépasse sa fréquence de maintenance, c'est-à-dire  $\text{profit}(w_i, p_t) = \frac{\text{fréquence\_accès}(w_i, p_t)}{1 + \text{fréquence\_maintenance}(w_i, p_t)} > 1$  . Nous avons choisi de travailler avec les profits car on désire recommander les webviews selon leurs profits. Ainsi, les corrélations entre les webviews doivent être à base de profit.
- Une séquence  $S_j$ , est une suite de  $m$  périodes consécutives donc de taille  $m$ .

##### Spécification des sources de métadonnées et de support minimum

Après l'analyse du fichier log, l'historique des accès et des mises à jour des webviews peut être résumé dans une matrice  $H$  tel que:

$$H(i, t) = \begin{cases} 1 & \text{si le webview } w_i \text{ a été profitable au cours de } p_t \\ 0 & \text{si non} \end{cases}$$

Nous avons choisi cette représentation binaire (0 ou 1) pour pouvoir appliquer les algorithmes du web usage mining qui exige cette représentation.

**Exemple 1 :** La matrice suivante représente l'historique de 4 webviews au cours de 12 périodes de sélection. Si nous fixons la taille d'une séquence à 4, nous aurons 9 séquences.

Webviews/Périodes	$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	$p_6$	$p_7$	$p_8$	$p_9$	$p_{10}$	$p_{11}$	$p_{12}$
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	1	1	1	1	1	1	1	1	1	1	1

TAB. 1 – Représentation matricielle de l'historique des webviews

### Application d'un algorithme d'extraction des motifs séquentiels fréquents

Dans cette étape nous utiliserons l'algorithme cSPADE, proposé dans [Zaki, 2000], pour extraire les motifs séquentiels fréquents.

**Exemple 2 :** Le tableau suivant décrit quelques motifs séquentiels de taille 4, trouvés par application de l'algorithme cSPADE sur les données du tableau 1, ainsi que leurs supports :

Motifs Séquentiels	Itemsets	Supports
$m_1$	(AB) (D) (D) (BD)	4/9 = 44%
$m_2$	(BCD) (AD) (BD) ( BD)	2/9=22%
$m_3$	(A) (D) (D) (D)	5/9= 56%
$m_4$	(D) (B) (D) (D)	7/9= 78%

TAB. 2 - Exemples de motifs séquentiels extraits à partir de l'historique des webviews

Si nous considérons le support minimum  $\text{minSupp} = 40\%$ , alors la séquence  $m_2$  ne sera pas retenue comme une séquence fréquente.

### 3.2.2 Calcul des poids de matérialisation des webviews

Pour calculer le poids de matérialisation d'un webview  $w_i$ , nous allons introduire et puis utiliser les poids des motifs séquentiels. Dans cette section, nous allons utiliser les notations suivantes :

- $m_j$  : motif séquentiel d'indice  $j$  ;
- $AM(m_j)$  : les antécédents de  $m_j$ . Par exemple, dans le tableau 2,  $AM(m_1) = \{(AB) (D) (D)\}$  ;
- $AM(m_j, k)$  : l'antécédent de  $m_j$  d'indice  $k$ . Si la taille du motif est 4, alors  $k=1, \dots, 4$ . Par exemple, dans le tableau 2,  $AM(m_1, 1) = (AB)$  et  $AM(m_1, 2) = (D)$  ;
- $\text{longueur}(AM(m_j))$  : nombre d'items des antécédents de  $m_j$ . Par exemple, dans le tableau 2,  $\text{longueur}(AM(m_1)) = 4$  ;
- $CM(m_j)$  : la conséquence de  $m_j$ . Par exemple, dans le tableau 2,  $CM(m_1) = \{B, D\}$  ;
- $M(w_i) = \{m_j / w_i \in CM(m_j)\}$  : les motifs générateurs de  $w_i$ . Par exemple, dans le tableau 2,  $M(B) = \{m_1, m_2\}$  ;
- $W(p_t) = \{w_i / \text{profit}(w_i, p_t) > 1\}$  : l'ensemble des webviews profitables au cours d'une période de sélection  $p_t$ . Par exemple, dans le tableau 1,  $W(p_1) = \{A, B, D\}$ .

### Calcul des poids des motifs

Le poids d'un motif séquentiel  $m_j$  est calculé selon la formule suivante :

$$\text{poids}(m_j) = \text{support}(m_j) * \left(1 - \frac{1}{1 + \text{longueur}(AM(m_j))}\right)$$

Nous avons ajouté, au niveau du dénominateur, 1 à la  $\text{longueur}(AM(m_j))$  pour ne pas avoir un poids nul lorsque la  $\text{longueur}(AM(m_j)) = 1$ . Dans cette formule nous avons choisi de chercher un compromis entre la longueur des antécédents du motif et son support.

### Calcul des poids de matérialisation des webviews

Nous supposons que nous avons  $t-1$  périodes de sélection,  $p_1, p_2, \dots, p_{t-1}$ , représentant l'historique analysé du DIWS et que nous nous intéressons à la matérialisation des webviews pour la période  $p_t$ . Pour pouvoir calculer les poids de matérialisation des webviews, en exploitant les motifs séquentiels extraits au cours de l'étape précédente, il faut commencer par identifier les périodes récentes de l'historique, qui vont correspondre aux antécédents des motifs. Les conséquences des motifs dont les antécédents sont inclus dans

## Motifs Séquentiels pour la Sélection des Webviews à Matérialiser

cet historique récent (séquence des périodes identifiées), sont concernées par le calcul du poids de matérialisation. Notre idée est que si les antécédents d'un motif sont rencontrés au cours des périodes les plus récentes, alors la conséquence du motif a une probabilité strictement positive d'être rencontrée au cours de la prochaine période. Notre démarche de calcul de poids de matérialisation comporte les étapes suivantes:

1. Construction de l'historique récent des webviews profitables: il s'agit de chercher les ensembles  $W(p_{t-z})$  pour  $z = 1, \dots, m-2, m-1$  avec  $m$  est la taille des motifs (nombre de périodes par séquence).
2. Recherche des motifs séquentiels générateurs valides des webviews : Un motif séquentiel générateur d'un webview  $w_i$  est dit valide si ces antécédents sont inclus dans l'historique récent. Donc, l'ensemble des motifs séquentiels générateurs valides,  $MV(w_i)$ , d'un webview  $w_i$  est défini comme suit :

$$MV(w_i) = \{m_j / m_j \in M(w_i) \text{ et } AM(m_j, m-z) \subseteq W(p_{t-z}) \text{ avec } z = 1, \dots, m-2, m-1\}$$

3. Calcul du poids de matérialisation d'un webview  $w_i$  : le poids de matérialisation est calculé en fonction des poids et du nombre des motifs générateurs valides. Il est défini comme suit :

$$poidsM(w_i) = \max \{poids(m_j) / m_j \in MV(w_i)\} * \left(1 - \frac{1}{1 + |MV(w_i)|}\right)$$

Nous avons choisi de chercher le compromis entre le poids et le nombre des motifs générateurs valides, désigné par  $|MV(w_i)|$ , pour ne pas favoriser les webviews qui ont des motifs de poids fort à ceux ayant un grand nombre de motifs générateurs valides.

**Exemple 4 :** Pour  $t = 13$  et  $m = 4$ , nous avons extrait, à partir du tableau 1, l'historique récent suivant:  $W(p_{10}) = (ABD)$ ,  $W(p_{11}) = (ABCD)$ ,  $W(p_{12}) = (BD)$ .

Dans le tableau suivant, nous avons cherché les motifs générateurs valides et les poids des webviews en se basant sur les données du tableau 2 :

Webviews	Motifs générateurs valides	Poids de matérialisation
A	$MV(A) = \{\}$	$poidsM(A) = 0$
B	$MV(B) = \{m_1\}$	$poidsM(B) = 44\% * 0.5 = 22\%$
C	$MV(C) = \{\}$	$poidsM(C) = 0$
D	$MV(D) = \{m_1, m_3, m_4\}$	$poidsM(D) = 78\% * 0.75 = 58.5\%$

TAB. 3 - Les poids de matérialisation de quelques webviews

### 3.3 Algorithme de recommandation des webviews à matérialiser

Les deux algorithmes permettant de calculer les poids des motifs séquentiels (PMS) et les poids de matérialisation des webviews (PMW) sont respectivement donnés par les figures 1 et 2.

**Algorithme PMS**

```

m:// la taille des motifs séquentiels
M={ };// liste des motifs séquentiels de taille m
H;// matrice représentant l'historique des webviews
minSupp: le support minimum des motifs séquentiels
Début
MC=cSPADE(H,m, minSupp) // génération des motifs séquentiels de taille ≤ m
Pour j=1 jusqu'à taille de MC faire
  Si (taille (mj)=m) alors
    
$$poids(m_j) = support(m_j) * \left(1 - \frac{1}{1+longueur(AM(m_j))}\right)$$
 // calcul des poids des motifs
    M=M ∪ {mj}
  Fin si
Fin pour
Fin

```

FIG. 1 -Algorithme de calcul des poids des motifs séquentiels

**Algorithme PWM (p<sub>t</sub>) ;**

```

W ; // liste des webviews candidats, W= { wi/i=1,...,n } ;
P ; // liste des périodes de sélection passés, P= { pj/j=1,...,t-1 } ;
m ;// la taille des motifs séquentiels
M={ };// liste des motifs séquentiels
H ;// matrice représentant l'historique des webviews
minSupp: le support minimum des motifs séquentiels
Début
PMS ; // calcul des poids des motifs
Pour i=1 jusqu'à n faire
  MV(wi) = {}
  Pour j=1 jusqu'à taille de M faire //recherche des motifs générateurs valides des wi
    Si (wi ∈ CM(mj)) alors
      z=1 ;
      stop= faux
      Tant que (z < m ET stop=faux) faire
        si AM(mj, m - z) ⊆ W(pt-z) alors stop= faux ;
        si non stop=vrai ;
      Fin si
      z=z+1 ;
    Fin tant que
    Si stop = faux alors
      MV(wi) = MV (wi) ∪ {mj}
    Fin si
  Fin si
  Fin pour
  
$$poidsM(w_i) = \max \{poids(m_j) / m_j \in MV(w_i)\} * \left(1 - \frac{1}{1+|MV(w_i)|}\right)$$
 //calcul des poids
Fin pour
Retourner {poidsM(wi), i = 1, ..., n} ;
Fin

```

FIG. 2 – Algorithme de recommandation des webviews à matérialiser

### 3.4 Expérimentations

Pour évaluer notre approche, nous avons implémenté les algorithmes décrits par les figures 1 et 2. Ensuite, nous avons simulé la charge d'un DIWS composé de 30 webviews pour 51 périodes de temps de longueur 6 heures chacune, c'est-à-dire plus 12 jours en total. Nous avons supposé 5 requêtes d'accès et une requête de maintenance par seconde ce qui correspond à une charge total de plus que 400.000 accès par jour. Nous avons pris ces mesures de [Labrinidis et Roussopoulos, 2000]. Les accès et les mises à jour sont répartis d'une manière équitable entre les webviews mais les fréquences d'accès et de mise à jour d'un webview varient d'une période à une autre. L'algorithme PMS, d'extraction des motifs séquentiels et de calcul de leurs poids, est exécuté toutes les 4 périodes pour rafraîchir la liste des motifs séquentiels fréquents alors que l'algorithme PWM, de recommandation des webviews à matérialiser, est exécuté à la fin de chaque période. Nous avons affecté la valeur 40% au support minimum pour extraire les motifs séquentiels fréquents. La figure 3 suivante résume les poids moyens de matérialisation, des 30 webviews, calculés à la fin de la période numéro 50. Nous avons varié la valeur de  $m$  de 2 jusqu'à 6. La dernière colonne correspond au poids moyen de matérialisation produits par les autres approches de sélection. Nous rappelons que ces approches se limitent à la période de sélection la plus récente et décident, selon le profit du webview pendant cette période, soit de matérialiser le webview (dans ce cas le poids est 100%) soit de ne pas le matérialiser (dans ce cas le poids est 0%).

Nous avons remarqué que le meilleur poids moyen est produit pour  $m = 4$  car il y a une sorte d'équation entre les deux paramètres affectant le poids à savoir la fréquence ou le support des motifs et la longueur des antécédents du motif. Autrement dit, les supports des motifs diminuent lorsque  $m$  augmente et inversement. Un autre avantage observé lorsque  $m=4$ , est que l'algorithme d'extraction des motifs séquentiels consomme moins de temps d'exécution que lorsque la valeur de  $m$  dépasse 4.

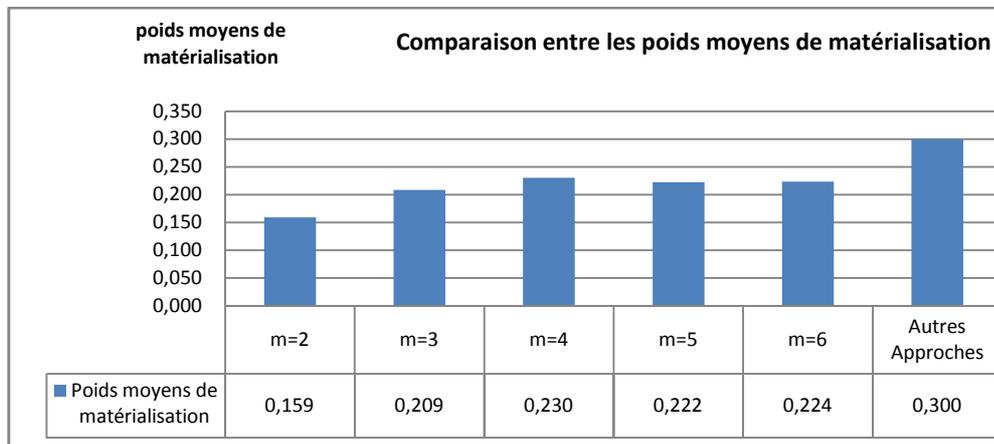


FIG. 3 – *Comparaison entre les poids moyens de matérialisation*

Comme deuxième expérimentation, nous avons lancé la simulation décrite ci-dessus deux autres fois pour pouvoir comparer les taux de réussite des décisions (matérialiser ou ne pas matérialiser un webview). Nous calculons les taux de réussite d'une décision pour les trois tests de la manière décrite dans le tableau 5. Les résultats de cette expérimentation sont

résumés dans la figure 4. Nous avons remarqué que quelque soit la valeur de  $m$ , les décisions des autres approches sont plus risquées (risque = 1-taux de réussite) et qu'en moyenne notre approche permet de réduire le risque de matérialisation de plus 20%. Même lorsque  $m=2$ , où l'historique récent de notre approche correspond à celui des autres approches, nous avons eu une grande différence entre les taux de réussite.

Décision prise à la fin de la période 50	Etat réel à la fin de la période 51	Taux de réussite d'une décision selon notre approche	Taux de réussite d'une décision selon les autres approches
Matérialiser $w_i$ avec une probabilité $\text{poidsM}(w_i)$	$w_i$ a produit un profit	+ $\text{poidsM}(w_i)$	+1
	$w_i$ a produit une perte	+ $(1-\text{poidsM}(w_i))$	-1
Ne pas matérialiser $w_i$	$w_i$ a produit un profit	-1	-1
	$w_i$ a produit une perte	+1	+1

TAB. 4 - Méthode de calcul de taux de réussite des décisions de matérialisation

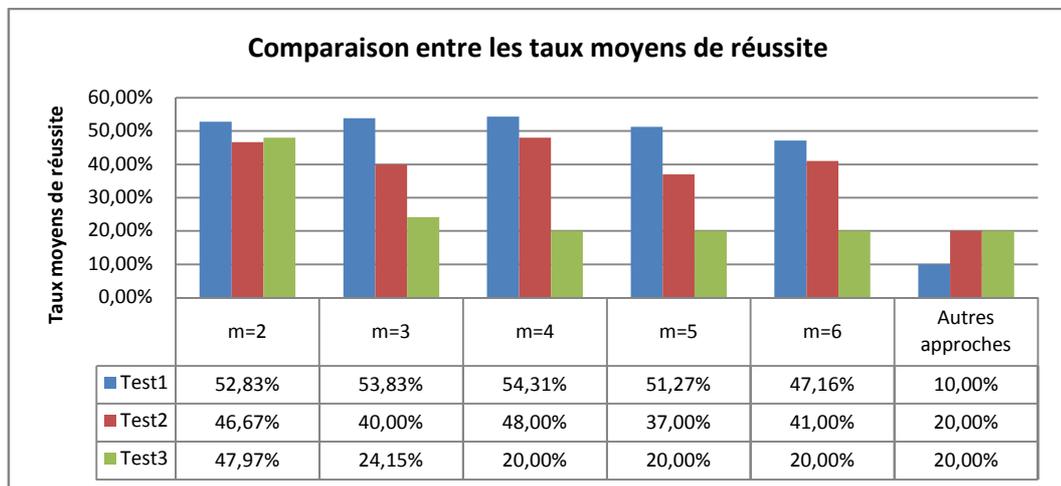


FIG. 4 – Comparaison entre les taux moyens de réussite des décisions de matérialisation

## 4 Conclusion

Nous venons de présenter, dans cet article, une approche de recommandation des webviews à matérialiser en se basant sur la technique de web usage mining, plus précisément sur les motifs séquentiels. Le principe général de l'approche est d'extraire, à partir de l'historique des requêtes d'accès et de mise du DIWS, les motifs séquentiels fréquents et de les exploiter pour calculer le poids de matérialisation de chaque webview. Ce poids servira par la suite pour la sélection des webviews à matérialiser. Les résultats de nos expérimentations ont montré que notre approche permet de réduire le risque de matérialisation des webviews beaucoup mieux que les approches de sélection existantes. Dans les prochains travaux, nous allons développer un algorithme qui exploite ces poids de matérialisation pour identifier les webviews à matérialiser.

## Références

- [Agrawal et Srikant, 1995] R. Agrawal, et R. Srikant. *Mining sequential patterns*. In Proceedings of the 11th international conference on data engineering (ICDE'95). 1995
- [Aouiche et Darmont, 2009] K. Aouiche et J. Darmont. *Data mining-based materialized view and index selection in data warehouses*. J. Intell. Inf. Syst. 33(1): 65-93 (2009)
- [Aouiche et al., 2006] K. Aouiche, P. Jouve, et J. Darmont. *Clustering-Based Materialized View Selection in Data Warehouses*. In 10<sup>th</sup> East-European Conference on Advances in Databases and Information Systems (ADBIS 2006), Thessaloniki, Greece, Vol. 4152 of LNCS, pp. 81-95. 2006
- [Ben Ammar et al., 2006] A. Ben Ammar, A. Abdellatif, et H. Ben Ghezala. *Forms of Data Materialization in Data-Intensive Web Sites*. IJCSNS International Journal of Computer Science and Network Security, VOL.6 84 No.12, December 2006
- [Labrinidis et Roussopoulos, 2001] A. Labrinidis et N. Roussopoulos. *Adaptive webview materialization*. In the Fourth International Workshop on the Web and Databases, held in conjunction with ACM SIGMOD, 2001.
- [Labrinidis et Roussopoulos, 1999 ] A. Labrinidis et N. Roussopoulos. *On the materialization of webviews*. In Proc. of the ACM SIGMOD Workshop on the Web and Databases WebDB 99, pages 79-84, 1999.
- [Labrinidis et Roussopoulos, 2002 ] A. Labrinidis et N. Roussopoulos. *Online View Selection for the Web*. In Proc. Of the ACM SIGMOD Conference, 2002.
- [Labrinidis et Roussopoulos, 2000] A. Labrinidis et N. Roussopoulos. *WebView Materialization*. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. pp.367-378. May 14-19, 2000, Dallas, Texas, United States
- [Saidi et al., 2007] S. Saidi, Y. Slimani, et K. Arour. *Webview selection from user access Patterns*. In PIKM '07 : Proceedings of the ACM \_rst Ph.D. workshop in CIKM, pages 171-176, New York, NY, USA, 2007.
- [Zhang et Qin, 2005] Y. Zhang et X. Qin. *State transfer graph : An efficient tool for webview maintenance*. In In the proceedings of WAIM2005 In Hangzhou, China, pages 513\_525, 2005.
- [Zaki, 2000] M. J. Zaki. *Sequence mining in categorical domains: Incorporating constraints*. In Proceedings of the 9th international conference on information and knowledge management (CIKM 00) (pp. 422– 429). 2000.

## Summary

In this paper, we propose an approach, which is based on the web usage mining technique, to recommend some profitable webviews for materialization. The webview materialization is a term used to represent the transformation of dynamic web data into equivalent static web data. That is the creation of a static instance of a dynamic web page, at a certain point in time. In this approach, we firstly analyze the DIWS queries of several historic selection periods to extract the repetitive user behaviors, known as frequent sequential patterns. Secondly, we use these sequential patterns to calculate a materialization weight for each webview. The webviews with high materialization weights are the most recommended for the materialization. Our experiment results show that our approach reduce the webview materialization risk by than 20%.