A SERVER-SIDE SUPPORT LAYER FOR CLIENT PERSPECTIVE TRANSPARENT WEB CONTENT MIGRATION

Diana-Florina Haliţă Darius Bufnea

Departament of Computer Science
Faculty of Matemathics and Computer Science

Babeş-Bolyai University
July 5, 2013

Table of Contents

Abstract

- the migration process implies changes in the site structure as seen by search engines and web clients
- disadvantages, such as misdirecting search engines visitors
- problem remains unsolved for visitors landing from 3rd party referrers
- map the old visible structure of a website to the new one
- advantages of such a mechanism are:
  - reducing incoming dead links from 3rd party referrers
  - assisting search engines for properly redirecting users
  - page rank and SERP conservation
  - separation between content and presentation, storage engine, business and database

Why Content Management Systems?

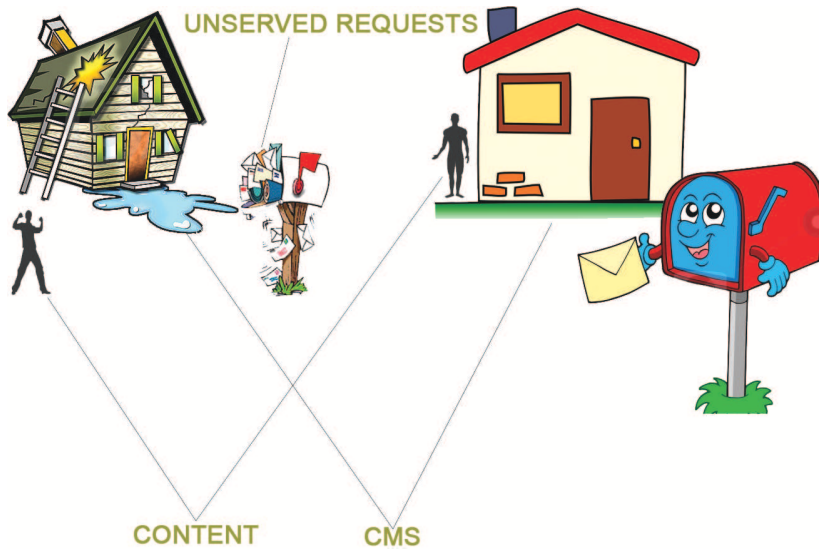
- easy maintenance
- easy migration from one presentation to another
- regular security updates
- full control over the elements related to SEO
- 3rd party plugins
- web based administration

Why is Migration Necessary?

- old CMS is deprecated
- open source CMS granting easier access to support new features implemented by 3rd party plugins
- CMS scalability
- support for new technologies: CSS3, Ajax, HTML5

Problems

- exposing to the web a certain content to a new, different URL as it was presented in the old site
- misleading visitor coming from search engines or third party referrers
- lose page rank or other in-time gather benefits induced by back links or social shares



Possible solutions

- dividing the content into categories
  - what can be automatically migrated and what can't
  - it is desired to automate as much of the content
- estimating needed time for migration
  - compare automatic migration with manual migration required time
- migration reevaluation based on guidance
  - evaluating the automatic migration process
  - problem: content's structure and its regularity
  - manual migration: waste of resources, time and effort

Similar solutions

- implemented mostly as plugins inside all major CMS
- take into consideration some in-time gather data based on user behavior

Our approach

- match new URLs to old ones based on:
  - content similarity
  - semantic related information such as: URLs
  - the query given to the search engines
  - referrer's content similarity with the linked content
- advantage of being implementable prior to the release of the new web site

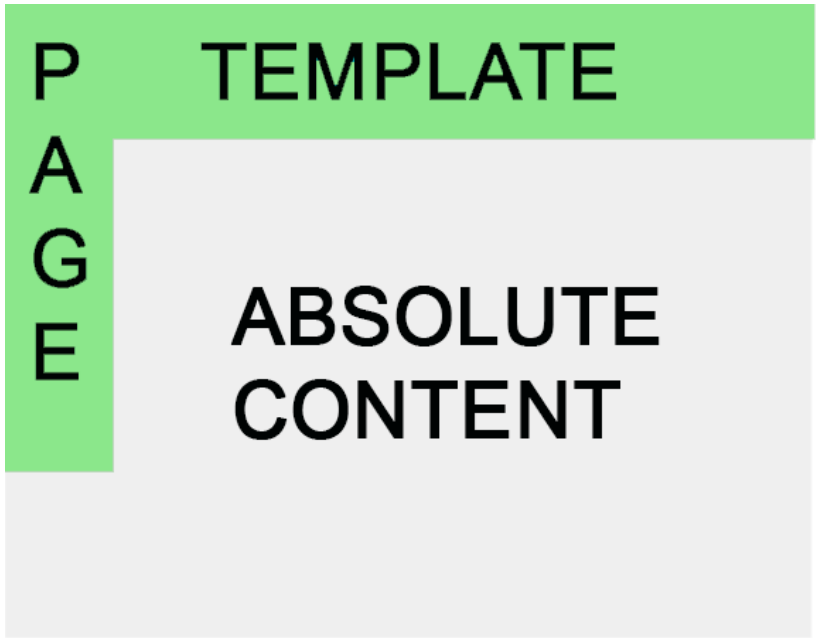
Content migration

- static content migration to static content migration
  - the base name of the file remains the same
    `oldsiteURL: http://oldsite/oldpath/filename`
    `newsiteURL: http://oldsite/newpath/filename`
  - perfect match, *similarity(`oldsiteURL`, `newsiteURL`) = 1*
  - similarity(`static content`, `static content`)
- static content migration to dynamic content migration
  - *similarity(`static content`, `dynamic content`)*
  - similarity(`full content`, `absolute content`)
- dynamic content migration to dynamic content migration
  - *similarity(`dynamic content`, `dynamic content`)*
  - similarity(`absolute content`, `absolute content`)

Dynamic content migration

the content = content of the page template + the absolute content stored in the database

We'll take into discussion only the absolute content



In order to match these URLs, our algorithm makes use of a similarity function, but the method is not dependent of a certain similarity function.

- Information processing is not done in real time
  - running a real time computation will induce delays in serving a response to the web client
  - the batch program runs completely independent of the CMS core
- For a quick match we use the cosine similarity algorithmas implemented by Apache Lucene

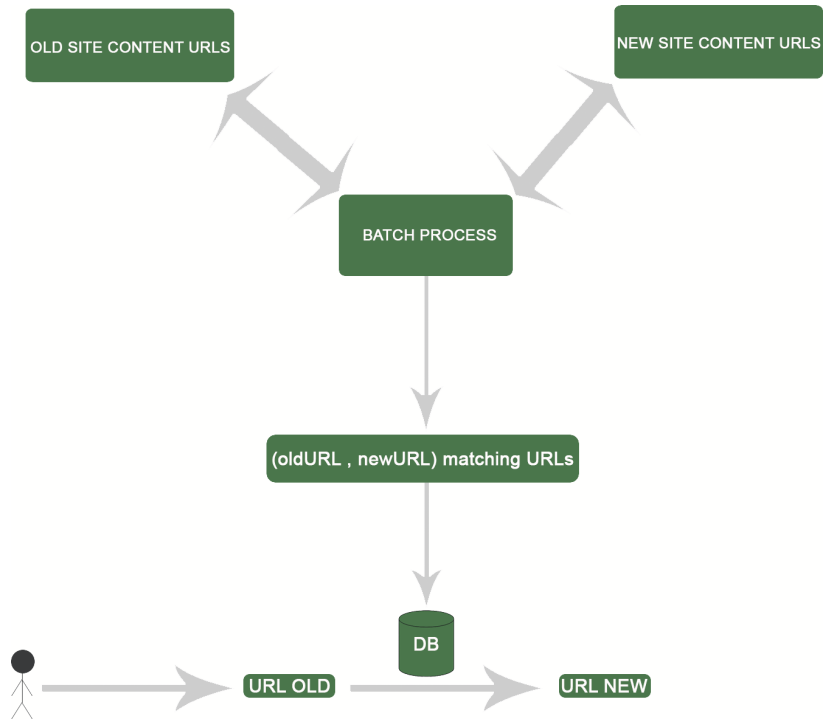


Figure 1 : Batch processing

The formal presentation of the algorithm

```
For each oldsiteURL having static
content do
  Identify the newsiteURL which points
to the same static content (based on
base filename) (A)
  If this newsiteURL was identified then
    eliminate the oldsiteURL from the
    URLs list which must be processed
  EndIf
EndFor
```

```
For all unprocessed oldsiteURLs do
  If it is a static content URL then
    absolute_content = the effective
content (C)
  else
    absolute_content =
content(oldsiteURL) - content(page
template)(B)
  EndIf
  Identify the newsiteURL so the
absolute_content has the best similarity
with the oldsiteURL's absolute_content
  The matching pair is inserted in a
table, together with the current date,
the similarity and the best similarity
ever
EndFor
```

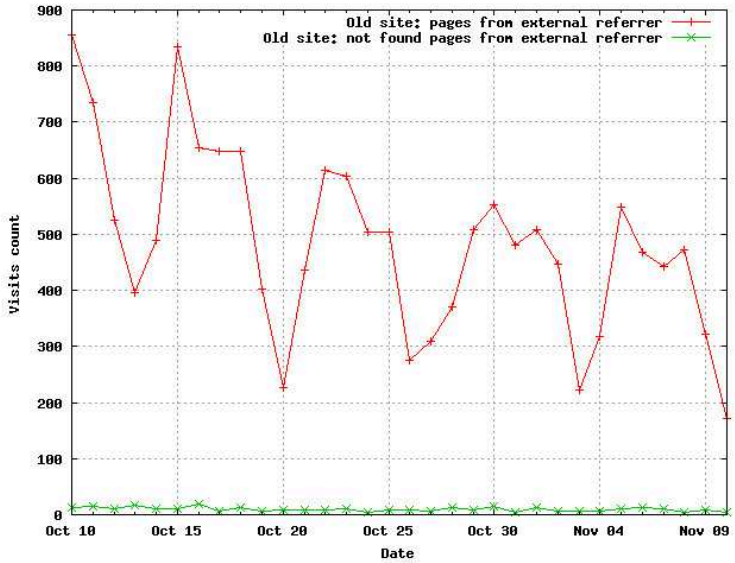


Figure 2 : Old site: number of pages accessed from external referrer and number of not found pages accessed from external referrer

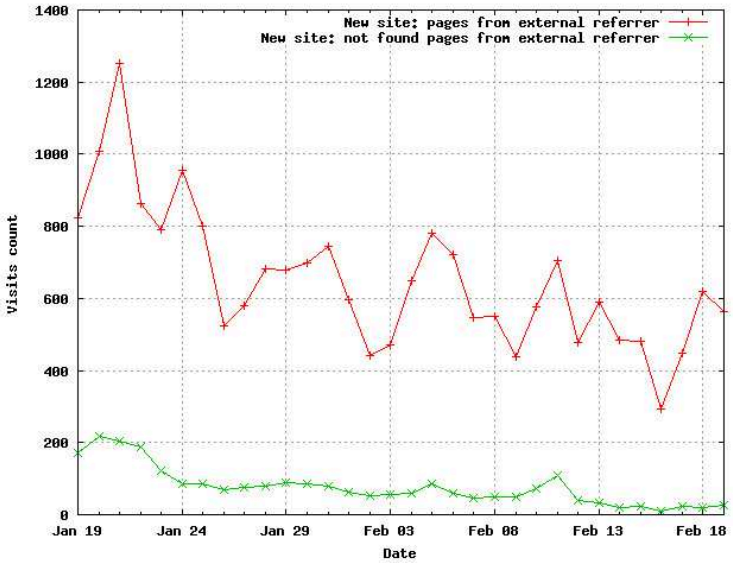


Figure 3 : New site: number of pages accessed from external referrer and number of not found pages accessed from external referrer

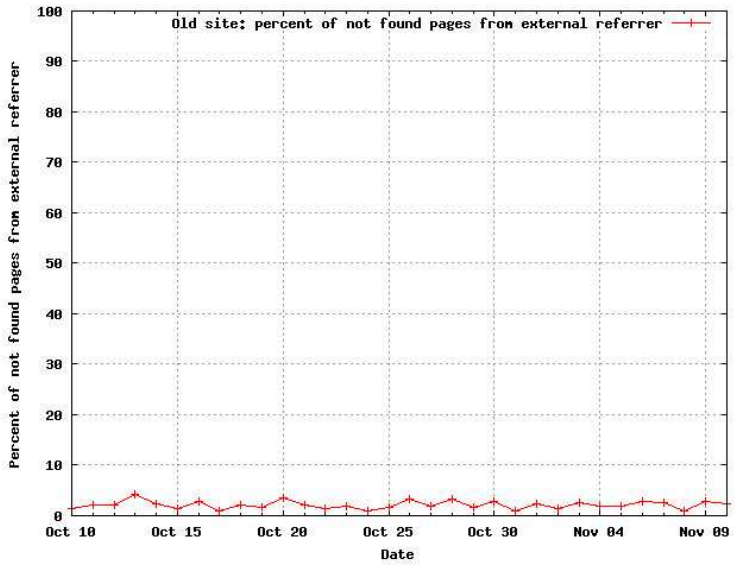


Figure 4 : Old site: percent of not found pages accessed from external referrer

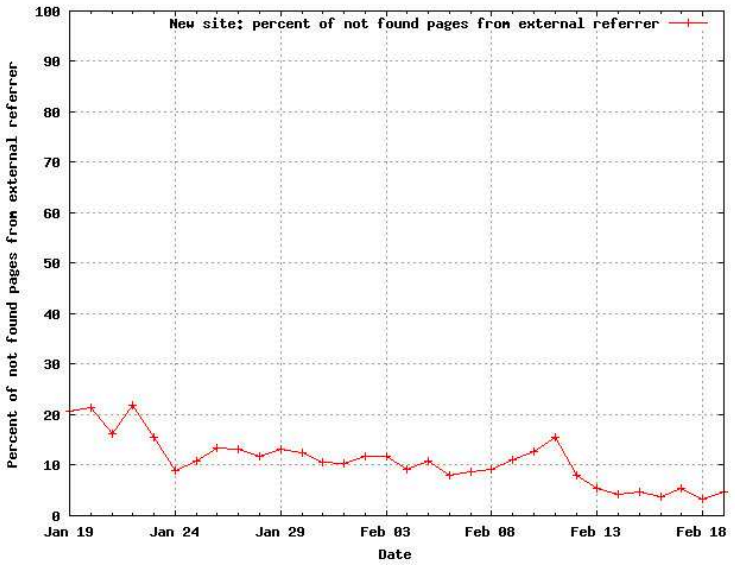


Figure 5 : New site: percent of not found pages accessed from external referrer

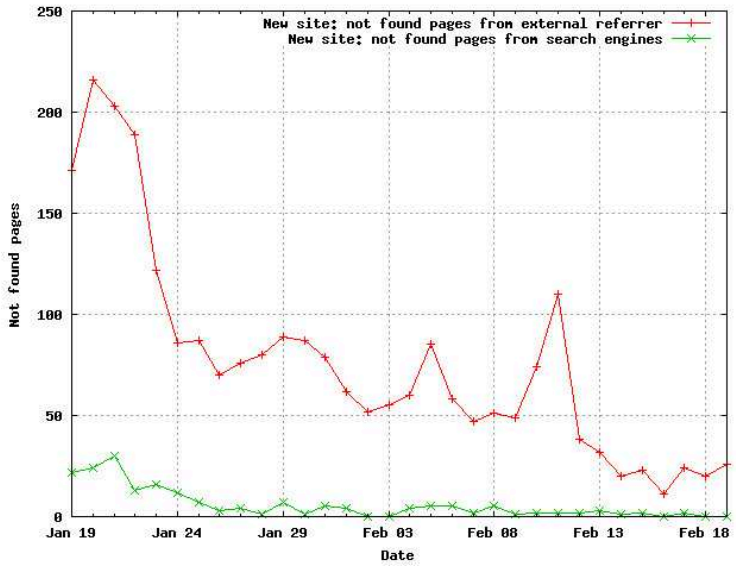


Figure 6 : New site: adaptation of the search engines to the new structure of the site

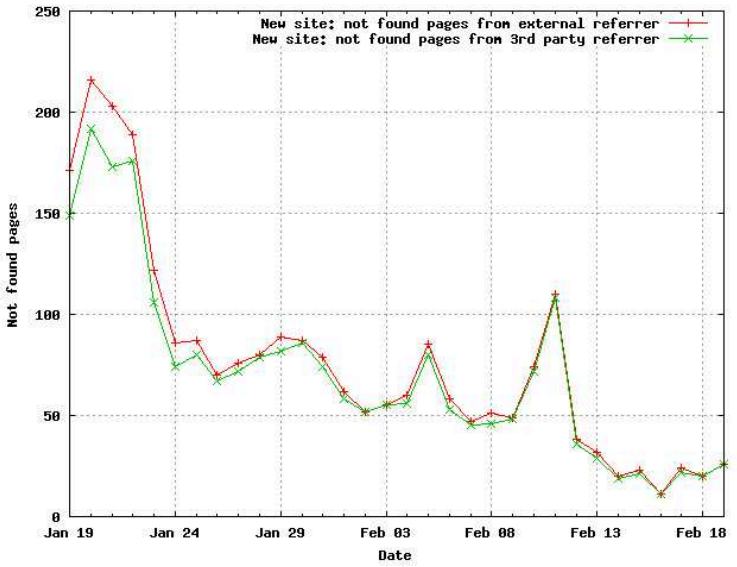


Figure 7 : New site: Most of not found pages are coming from 3rd party referrer

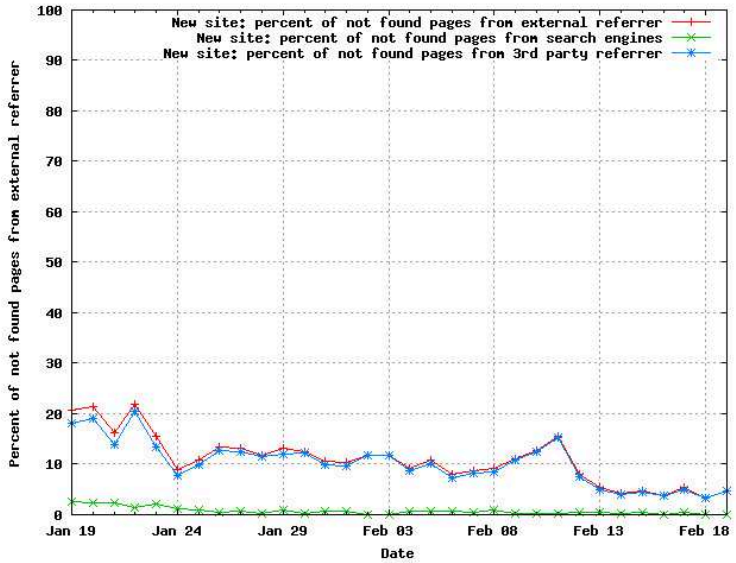


Figure 8 : New site: percent of not found pages, not found pages from search engines, not found pages from 3rd party referrer

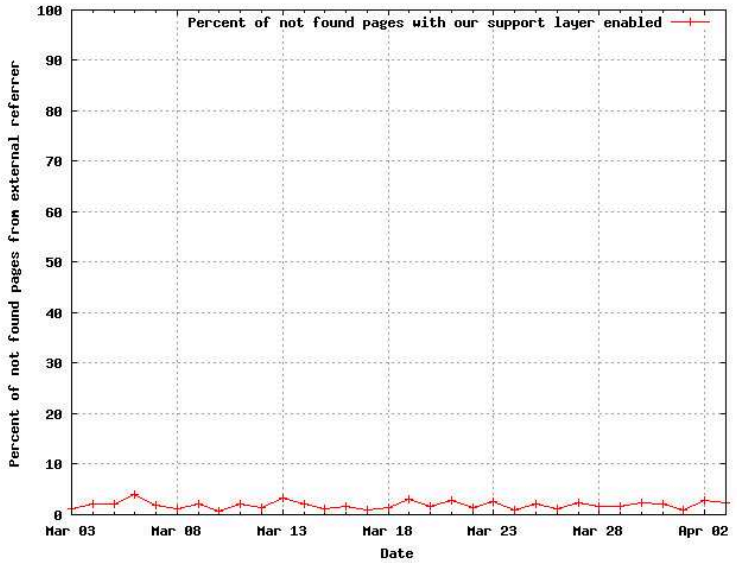


Figure 9 : Percent of not found pages having an external referrer with our support layer enabled

Conclusions and Future work

- method for implementing this layer by mapping URLs from the old site to the ones in new site based on their content similarity
- evaluating different similarity functions in order to improve the matching process and its speed
- giving different weights in the similarity algorithm to the various properties of the content such as: URL, page headings, page title, key words
- redirecting the user to a similar page, from the content point of view, as the page where he is coming from

Thank you!

Q & A