International Journal of English and Literature
Vol.1, Issue 1 (2011) 1-9

TECHNOLOGICAL INNOVATIONS IN ENGLISH LANGUAGE TEACHING – ORGANISED INFORMATION ARCHIVAL OF DIGITAL OBJECTS FOR LEARNERS OF THE DIGITAL AGE

Lt. E. Justin Ruben M.A., M.Phil.[+] and Mr. T. Ramraj, M.E.[++]
[+]Assistant Professor in English & Associate NCC Officer,
Department of Humanities,
Coimbatore Institute of Technology, Coimbatore – 641 014.
&
[++]Assistant Professor
Department of Computer Science Engineering and Information Technology,
Coimbatore Institute of Technology, Coimbatore – 641 014.

ABSTRACT

As today's learners belong to the Digital Age and are techno-savvy, English faculty also must adapt to use technology and multi-media based learning resources to impart the functional learning of English. So the need of the hour is to develop a common preservation infrastructure to preserve language-rich web contents, research data, publications, dissertations and other online assets. If a collection of such data in the form of digital objects are preserved and a proper information retrieval mechanism is provided to access those data then every learner regardless of their skill level and background can access the online and offline contents available, quickly and efficiently.

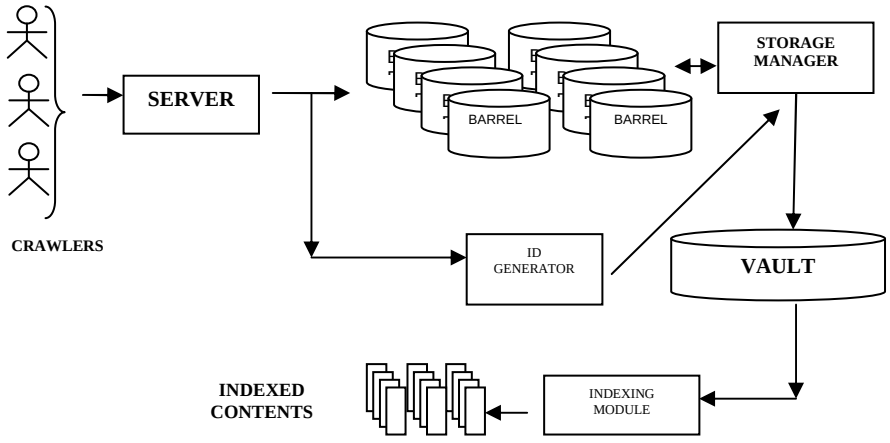
INTRODUCTION

The advent of Information and Communication Technology (ICT) in classroom pedagogy has revolutionized all fields of learning including English Language Teaching. Since many of today's learners are now techno-savvy, the faculty also must adapt to use technology and multi-media based learning

resources to impart the functional learning of English in classrooms. With this primary objective in mind for most of the relevant topics on English Language Learning and Teaching that is available online, the authors propose to initiate a collection of links and contents that can be indexed in an organized manner.

In this paper, we describe an efficient and systematic way of digital data organization to support high throughput from crawlers. Usually the digital objects are in the form of multimedia contents (audio, video), documents, web pages, presentations etc. So we limit our focus mainly for basic HTML web pages in order to reduce complexity. Here an architectural view of a storage repository to support high page addition rate and also to take cliff notes from those pages is proposed.

PROPOSED STORAGE REPOSITORY ARCHITECTURE



The set of pages that are crawled by crawlers are dumped in a Server. From the server the pages are fetched, their corresponding identifiers are generated and stored inside a set of Barrels temporarily using log-structured page organization. Next, from the Barrels, pages are fetched and checked whether

older versions of them exist in the Vault or not (by using Key Indexer) and their older versions are replaced by fresh pages inside the Vault. This update is done so as to have fresh pages inside the vault which would be useful for taking Cliff Notes.

ID GENERATOR:

This entity is used to provide a well defined mechanism to uniquely access to a single page. It is accomplished by providing an unique identifier for each and every web page that is crawled by the crawler. An identifier can be defined as a unique n-bit signature which is created for a web page (or for any object) by some sort of cryptographic hashing function.

URL Normalization:

For this signature generation, the URL context string is converted into canonical representatives known as URL Normalization. This URL Normalization is done in the URLs by

- Converting all the upper case characters into lower case characters
- Removing the key term "http://" from the URL string
- Removing the forward slashes
- Removing the default port number.(":80", for example)

Now this normalized string (URL) which is canonically represented and having arbitrarily length is given to a hash function. Here MD5 hash function is used.

MD5 (Message Digest 5):

This is a type of cryptographic hashing function which is used to convert a string (normalized URL) / file of arbitrarily length into a unique 128-bit value.

Here MD5 hashing function is used because of the following reasons:

1. It can produce an output signature of 128 bits i.e., it can generate 2^{128} identifiers rather than any 64-bit signature which can produce only 2^{64} identifiers. Also the probability of not having a collision (i.e., two pages cannot share the same identifier) can be calculated using the formula

$$\mathbf{P(\hat{C}) = e^{-N(N-1)/2X}}$$

Where

$\hat{C}$ - No collision

N - Number of Web pages

X - 2^b

b - Number of bits used for generating a signature (128 or 64).

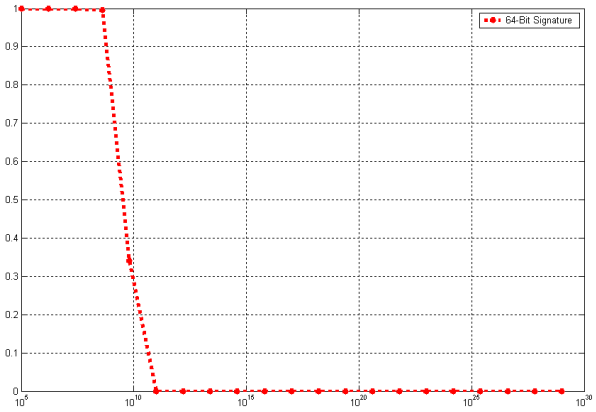


Figure 1

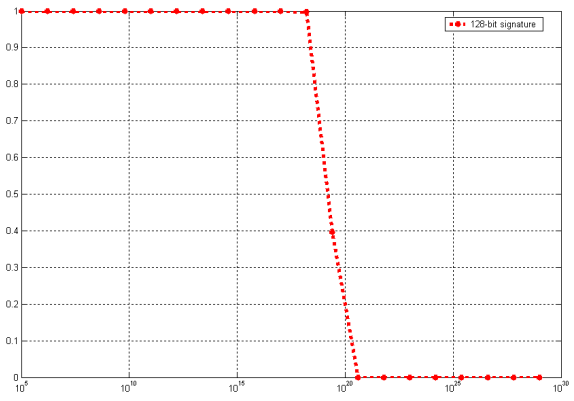


Figure 2

The above two figures (Figure-1 and Figure-2) represent the probability of not having a collision when using 64-bit and 128-bit respectively where X-axis represents the number of web pages and Y-axis represents the probability of having collision. From the Figs. 1 & 2, it is clear that the line of plots touches zero at 10^{11} and 10^{21}. That is, out of 10^{11} pages, two pages would share the page identifier while using 64-bit signature and out of 10^{21} web pages two pages would share the same identifier when 128-bit signature is used.

2. Next, the computational cost taken for generating 128-bit key by other hashing functions is greater than MD5.

BARRELS:

It is a collection of nodes in which the crawled pages are distributed and stored using Log structured page organization. Here, there are two steps involved to carry out this task. They are

- To distribute pages across multiple nodes (machines) in a distributed environment
- To organize pages inside a single node.

Page Distribution:

The page distribution is done either by Uniform distribution or Hash-based distribution policy. In Uniform distribution, all the storage nodes are treated identically; any page can be assigned to any of the nodes. In Hash-distribution policy, the allocation of pages across the nodes is decided based upon their page identifiers. In our prototype, hash-based distribution policy is used to distribute pages across multiple nodes. The page identifiers that range from 0 to $2^n/k$, $2^n / k + 1$ to $2*2^n/k$, $2*2^n / k + 1$ to $2^n - 1$ are assigned to $node_1$, $node_2$ up to $node_k$ respectively. For a newly arrived page whose page ID falls in-between any of the range would be redirected to its corresponding node. Also it requires a very sparse global index to fix the node in which the page with a given identifier would be located.

Page Organization:

The organization of pages on a disk (node) could be achieved by three different ways namely

- Hash-based organization
- Log-structured organization and
- Hashed-log structured organization.

Among these scenarios, log-structured organization for the storage of pages is chosen in order to support high throughput from the crawler. In this method, newly arrived pages (probably, million in number) are simply appended to a big log file. A reference to each page is maintained by two entities namely catalog & B tree / B^+ tree index.

Catalog:

It contains one record in the log file for each incoming page. A typical catalog may include the following information

i. Page identifier - generated 128-bit signature by MD5 cryptographic hashing function.

ii. Size of the page - compressed by using bzip, for example.

iii. Pointer pointing to the physical location of the page with in the log file.

iv. Status of the page (for future use) - (valid / deleted).

v. Time stamp denoting the time when the page was added to the repository.

The last two parameters are insinuated to help the Repository in replacing older version of the pages with their corresponding new versions because the repository stores the multimedia contents and newer version of web pages permanently.

STORAGE MANAGER:

Since random access to a page is required, a local B^+ tree (or B-tree) index is used which maps the page id to the record present inside the catalog that corresponds to the location of the page inside the big log file. It also contains a Key-Indexer that maps a page ID with the physical location of that compressed page inside the vault.

VAULT:

It is a permanent collection of a set of pages that can be indexed. In our prototype, hash based page organization is used to store the compressed web pages and their physical location inside a particular node is pointed out by the Key-Indexer of the Storage Manager.

Update Mechanism:

There are three possible reasons for updating the Vault. They are

- A new page may be present in the barrel but not exist inside the vault.
- Newer version of a page may be present in the barrel but its older version may exist in the vault.
- Both the barrel and the vault may contain the same version of a page.

So for updating, both the Key-Indexer and the local B-tree are sorted (using merge sort) and as the result the Key-Indexer and the Vault are updated.

CONCLUSIONS

An amateur user does not have any idea about *what to look for and where to look for* resources in the Net. So when a system gives similar and related entities (web pages, multimedia related files) for a user requested query that are fetched from the web and stored locally, then it provides an opportunity to the end user to view and read the content before they are actually fetched from that particular domain. The data is stored systematically for efficient information retrieval and the learners can take cliff notes of that data. The authors infer, after consultation with learners from beginner, intermediate and advanced levels, that if there is a repository of resources and a proper mechanism provided to the end user/learner, then learning becomes an engaging, simple and captivating process.

REFERENCES

1. Arturo Crespo and Hector, Archival Storage for Digital Libraries". Technical Report, Stanford. Digital Library.
2. Arvind Arasu et.al.,"Searching the Web", Technical Report, Stanford Digital Library.

3. `S. Brin and L. Page, The Anatomy of a large-scale Hypertextual Web Search Engine**”**, Computer Networks and ISDN Systems, pp 107–117, 1998.

4. Jayashree Mohanraj, S. Mohanraj. English Online Communication for Information Technology. Orient Longman : Hyderabad, 2001.

5. Jun Hirai et.al., WebBase:A Repository of Web Pages”.Technical Report, Stanford Digital Library.

6. Langville and C. Meyer,”Google PageRank and Beyond”.