Introducing the Portuguese web archive initiative

Daniel Gomes, André Nogueira, João Miranda, Miguel Costa
FCCN-Fundação para a Computação Científica Nacional
Av. do Brasil, 101
1700-066 Lisboa, Portugal
{daniel.gomes, andre.nogueira, joao.miranda, miguel.costa}@fccn.pt

ABSTRACT

This paper introduces the Portuguese Web Archive initiative, presenting its main objectives and work in progress. Term search over web archives collections is a desirable feature that raises new challenges. It is discussed how the terms index size could be reduced without significantly decreasing the quality of search results. The results obtained from the first performed crawl show that the Portuguese web is composed approximately at least by 54 million contents that correspond to 2.8 TB of data. The crawl of the Portuguese web was stored in 2 TB of disk space using the ARC compressed format.

Categories and Subject Descriptors

H.3.7 [**Information Search and Retrieval**]: Digital Libraries—*Systems issues*; H.3.3 [**Information Search and Retrieval**]: [Search process]

General Terms

Archive, Portugal, Preservation, History

Keywords

Portuguese web archive project, archiving project, digital preservation, web measurements, web archive tools

1. INTRODUCTION

The web enables people in general to make information available to everyone without having to resort to publishers and traditional printing channels. Millions of contents, such as texts, photos or videos are solely published on the web every day. However, most of this information ceases to be available online in a short notice and is irrevocably lost.

The Internet Archive collects and archives web contents worldwide. However, it is not easy for a single organization to maintain an exhaustive archive of all contents published online because many of them disappear before they can be archived. Events of great historical importance to the United States, such as the Katrina Hurricane gave rise to additional efforts by the Internet Archive so that this episode would be as thoroughly documented as possible. However, the preservation of web documents pertaining to historical events of national importance to Portugal are hardly traceable by foreign institutions.

Several countries have become aware of the urgent need to preserve information of national interest and have given rise to parallel initiatives aimed at preserving knowledge available on the web. This paper introduces the Portuguese Web Archive (PWA), a project of the National Foundation for Scientific Computing (FCCN) that aims to preserve information published on the web of Portugal. FCCN is a private non-profit organization whose main activity concerns the management of a network that connects Portuguese educational and scientific institutions. FCCN is the registry for the .PT top-level domain and has been operating the only Internet Exchange Point in Portugal for the past 10 years. It has also supported the creation of a resource center for the computational processing of the Portuguese language.

The first stage of the PWA development began in January 2008 and it is planned to finish within two years. However, the maintenance of this system and the preservation of the archived information is to carry on beyond that date. The services provided by the PWA go beyond the historical and cultural aspects of web data preservation. The existence of an archive for the Portuguese web:

- Contributes to increase the use of Portuguese as a language of communication on the web;
- Provides access to web contents of interest to scientists working in different fields, such as History, Sociology or Linguistics;
- Reduces national's dependence on foreign services regarding web data processing and searching;
- Supplies evidence in court cases that require information published on the web that is no longer available online.

Intuitively, the Portuguese web is composed by the contents of interest to the Portuguese people. This definition is subjective and difficult to implement as an automatic selection criterion [4, 21]. However, a web site referenced by a name hosted under the .PT domain is by definition related to Portugal [45]. Therefore, it was assumed that a content belongs to the Portuguese web if it meets one of the following conditions:

1. Its site domain name is hosted under the .PT domain;
2. It is hosted outside the .PT domain but it was embedded on a page hosted under the .PT domain;

3. It is hosted outside the .PT domain but it was redirected from a .PT domain site.

The objective of conditions 2 and 3 is to collect all the necessary contents to enable a faithful reproduction of the pages after they are archived. In the future, the selection criterion could be extended to include all the pages written in the Portuguese language and their embedded objects independently from their site domains.

This paper is organized as follows. Section 2 presents related work regarding web archive initiatives and tools. Section 3 describes the main objectives of the PWA project. Section 4 describes the system under development. In Section 5 it is discussed how the size of index structures to support term search could be reduced. Section 6 presents the results obtained from the first crawl of the Portuguese web. Finally, Section 7 draws the conclusions.

2. RELATED WORK

There have been several initiatives to bootstrap web archiving through the implementation of systems and laws. The Internet Archive was the pioneer web archive and has been broadly archiving the web since 1996 [35]. This organization leads the Archive-access project, that provides open-source tools to enable the collection, storage and access to archived contents [33].

The National Library of Australia founded its web archive initiative in 1996 and developed the PANDAS (PANDORA Digital Archiving System) software to periodically archive Australian online publications selected by librarians for their historical value [44]. The British Library led a consortium that archived 6 000 selected sites from the United Kingdom during two years using the PANDAS software [53]. According to the National Library of Australia, there are 16 countries with well-established national web archiving programs [40].

In 1996, the Royal Library of Sweden began the first European national web archiving project named Kulturarw3 and the first crawl of the Swedish domain was performed in the summer of 1997 [3]. The National Library of Norway had a three-year project named Paradigma (2001-2004) to find the technology, methods and organization for the collection and preservation of electronic documents [2]. The NEDLIB project (1998-2000) included national libraries from several countries (including Portugal) and had the purpose of developing harvesting software specifically for the collection of web resources for a European deposit library [25]. These projects originated the Nordic Web Archive (NWA) that counted on the participation of the national libraries of Finland, Iceland, Denmark, Norway and Sweden [26].

The objective of archiving the Portuguese web has been pursued by a research group of the University of Lisbon since 2001 [42]. The Digital Deposit was developed in collaboration with the National Library of Portugal and it created a framework to support selective archiving of online publications. From 2002 to 2006 the tumba! search engine crawled about 57 million textual contents from the Portuguese web (1.5 TB of data) and a web archive prototype was developed to enable access to these data [21, 48].

In December 2004, the Danish parliament passed a new legal deposit law that called for the harvesting of the Danish part of the web for the purpose of preserving cultural heritage. Two libraries became responsible for the development of the Netarkivet web archive [12]. The legal deposit of web contents at the National Libray of France is fed by Internet Archive bulk automatic harvesting of French web sites, crawling of a selection of sites and individual deposits of publications, such as the *Journal officiel de la République française*, which is the French government's main publication [41]. Drugeon presents a detailed description of a system developed to crawl and archive specific sites related to media and audiovisual [15], while preliminary work in cooperation with a national research institute (INRIA) had begun in 2002 [1].

3. MAIN OBJECTIVES

The main objectives of the PWA project are to provide public access mechanisms to the archived information and ensure its long-term preservation. For these purposes, we intend to release the following services until the end of 2009:

URL search: enables users to find archived contents gathered from a given web address. Its main handicap is that it requires the users to know the address where the desired information was published;

Term search: enables users to search for contents containing a given set of terms. People are used to term search in web search engines and a user survey indicates that this is the most desired feature for web archives [46];

New web search engine: several web crawls must be indexed to support term search. Making available a new search engine for the most recent crawl can be achieved with a small additional effort. Plus, search engine query logs are a valuable resource to analyze the evolution of users behavior and tune search mechanisms;

Web data sets for research: researchers from different sciences use the web as a source of information for their studies. Archived web data sets and related meta-data will be made available to researchers, so that they can process it without having to crawl the web;

Infrastructure for distributed processing: even a small portion of the web contains a considerably large amount of data that researchers may not have the computational resources to process. The PWA will provide an infrastructure to enable the distributed processing of the archived data;

Portuguese web characterizations: a web archive system must be tuned according to the characteristics of the stored data. Therefore, Portuguese web characterizations must be periodically generated. These studies will be published because they are interesting to a broader audience.

Web archiving is full of new challenges that require collaborations and research work. The PWA project intends to train human resources able to maintain its system and contribute to enhance Archive-access tools.

4. THE WEB ARCHIVE INFRASTRUCTURE

This Section describes the Portuguese web archive system that is under development.

4.1 Software

Figure 1 presents the architecture of the PWA system. The *Crawler* harvests the web and stores contents in ARC files [10]. The *Processing cluster* operates over the archived contents using several machines in parallel. It supports the generation of web characterizations and creation of index structures to support URL and term search.

There were tools previously developed in Portuguese research projects available to implement the archive system [20]. However,

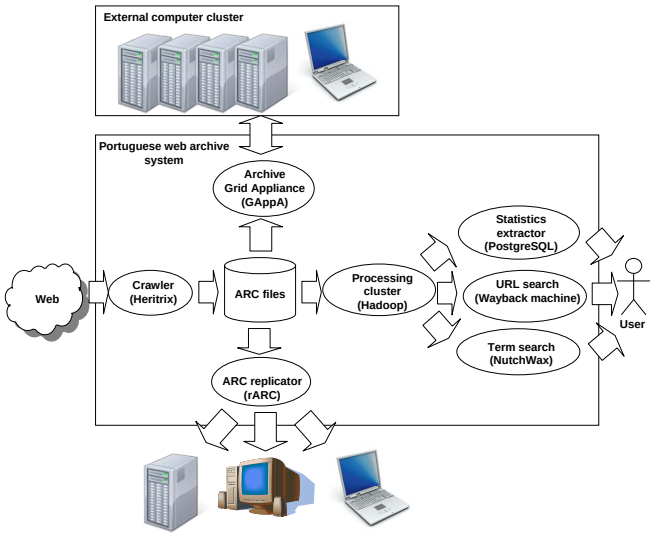

Figure 1: Portuguese web archive system architecture.

it was decided to implement a new system based on Archive-access tools because they are supported by a larger community. There were used the following Archive-access tools:

Heritrix: to implement the *Crawler* [39];

Hadoop: a software platform for distributed computing that implements the Google's *MapReduce* programming paradigm [14], to support the *Processing cluster*;

Wayback Machine: to support URL indexing and search [52];

NutchWax: to enable term indexing and search [49].

The statistics extracted from the archived contents are kept in a PostgreSQL relational database. There are two new systems under development in the PWA: the ARC replicator (rARC) and the Grid Appliance for the Archive (GAppA), which will be released as free open-source projects.

4.1.1 rARC: ARC file replicator

Archived data stored on a single repository may be lost if, for instance, a natural disaster destroys it. Therefore, archived data must be replicated across different locations to ensure its preservation. However, creating and maintaining several replicated repositories may become prohibitively expensive.

RARC is a distributed system that enables the replication of ARC files kept in a repository across many small storage nodes spread across the Internet. However, it can be used to replicate other file types. Internet users will be able to contribute to the preservation of historical web data by providing storage space from their computers. Ideally, every stored ARC file in the repository will have several replicas that would be retrieved in case of destruction of the central repository.

The rARC system must be:

Scalable: in a first stage it must scale to support thousands of storage nodes;

Robust: the web data kept in the storage nodes cannot be read by the users. The system must be robust against attacks and guarantee the creation and integrity of a minimum number of replicas for a given ARC file;

Usable: Internet users must be able to join a replication initiative and provide storage space as easily as possible;

Configurable: it must be able to integrate distinct web archiving initiatives.

The rARC system follows a client-server architecture. A server is installed on the web archive. On their turn, Internet users install client applications on their computers to store replicas. The client applications communicate with the rARC server to receive authentication credentials and then download ARC files from the server. Each ARC file is encrypted and signed to ensure its confidentiality and integrity. Periodically, the client applications communicate the state of their ARC files to the server for data integrity validations.

In case of destruction of the central repository, the recovery process begins. A new instance of the rARC server is installed on a machine and begins to rebuild the repository from the replicas stored on the clients. There are two alternative recovery processes with different security requirements to prevent the system against malicious users: *sequential* and *consensus by majority*.

In the sequential recovery process, the server receives an ARC file from a client, decrypts it and verifies if the checksum of the ARC file is consistent with the contained data. If the ARC file passes this validation, the server stores it and stops other transfers for that file. Otherwise, the file is discarded and the server waits for the upload of a replica stored on a different client. This approach is vulnerable to malicious users that may change an ARC file and forge its checksum by breaking the ciphering algorithms.

On its turn, in the consensus by majority recovery process, an ARC file is accepted by the server only after a majority of clients has presented the same checksum for it. The consensus recovery process is more secure because a majority of replicas must be compromised to allow the recovery of a corrupted file. However, this approach slows the recovery process because the server has to wait for the upload of a given ARC file by a majority of clients before deciding if it should be accepted.

A main issue that must be addressed to ensure the success of the rARC project is how to motivate users to collaborate with it and donate space from their computers to keep replicas. We hope that the PWA site will become popular in Portugal and we intend to use it to motivate users to collaborate. There will be published rankings of contributers at different levels, such as individuals, institutions or teams, so that competition for the top positions will motivate users to provide more disk space. Companies have commercial interest in having links to their sites coming from popular sites and national institutions have interest in showing that they are contributing to preserve national historical contents. Institutions within the web archiving community may also use rARC to interchange disk space to replicate data between web archives. There will be also randomly chosen a contributer of the week, so that the effort of small contributers will also be publicly recognized.

4.1.2 GAppA: Grid Appliance for the Archive

GAppA is a software platform designed to provide remote access to the archived data and enable its cooperative processing by several computer clusters. Researchers will be able to execute their programs using simultaneously the PWA and their own computers. On the other hand, the PWA system will be able to extend its processing capacity by using external computers.

A computer joins the cluster through the installation of a client application that submits jobs to the computer cluster. GAppA implements security measures so that the execution of the jobs does not compromise neither the integrity of the archived data nor the

underlying infrastructure. One of these measures is to run jobs submitted by clusters external to the PWA in virtual machines.

The PWA processing cluster is implemented using Hadoop and only jobs implementable with this technology can be executed. A GAppA prototype was configured using the IPOP Grid Appliance developed by the Advanced Computing and Information Systems Laboratory of the University of Florida [55]. HOD is a system for provisioning virtual Hadoop clusters over a large physical cluster supported by the Apache Software Foundation [51]. The integration of the IPOP Grid Appliance with the Hadoop On-Demand (HOD) platform is being studied to enable the dynamic extension of the virtual cluster of computers that execute the jobs.

4.2 Hardware

The hardware acquired for the development stage of the project is composed by a *blade system* that connects to a *Storage Area Network* (SAN) through a 4 Gbps Fibre Channel connection. A blade system is a component that hosts and connects several computers (blades). Our blade system hosts 7 blades and can be extended to 16 servers without requiring additional space in the cabinet. A SAN is composed by a network dedicated to storage devices and a centralized pool of disks that provides space to several computers [11]. Our SAN has 4 GB of cache and holds 54 disks, each with 500 GB or 1 TB of storage space capacity running at 7 200 rpm, that provide 25.6 TB of useful storage space in RAID 5. There is also a *tape library* with a capacity to store 12 TB of data. The hardware is installed on a 47 rack unit cabinet.

Using a blade system instead of independent servers saves on electrical power consumption and physical space on the cabinets because it uses compact components and shares power supplies. System administration costs are also reduced. Cable failures are a prime cause of downtime and 25% of a system's adminstrator time is spent on cable management [23]. A blade system uses internal network connections that reduce cabling inside a cabinet. The blades are plug-and-play and can be quickly replaced in case of failure. However, there are disadvantages regarding the use of a blade system. Although it contains redundant components and sophisticated monitoring and alert mechanisms, a blade system setup is less fault-tolerant than having several independent servers because all the blades are managed by a single device. Maintenance operations, such as firmware upgrades of shared components may impose shutting down all the blades. Another disadvantage is that blades from different vendors are usually not compatible.

A SAN enables storage space management as a whole and not as individual disks spread across computers. The storage space can be easily expanded or reallocated to different computers. SAN technology provides high-speed access to storage essentially because it takes advantage from a large cache and accesses data from several disks in parallel. Although the SAN main components are redundant, its main disadvantage is that it constitutes a system's single point of failure.

The acquisition of this hardware was relatively expensive when compared to a high density 1 rack unit server deployment with similar capabilities, such as the Petabox [50]. However, the adopted hardware architecture reduces maintenance costs in staff hours. Hardware problems during the current development stage could cause unrecoverable delays and threaten the continuity of the project. The development team does not include hardware experts and contracting a fast response specialized support service was significantly cheaper than hiring additional staff. The data center is geographically distant from the development team office and the acquired hardware provides built-in remote management tools that reduce the need for travels to the data center. Despite our current hardware options, we believe that for a large web archive, a platform composed by independent servers is more suitable because these computers can be acquired cheap and be simply replaced in case of failure.

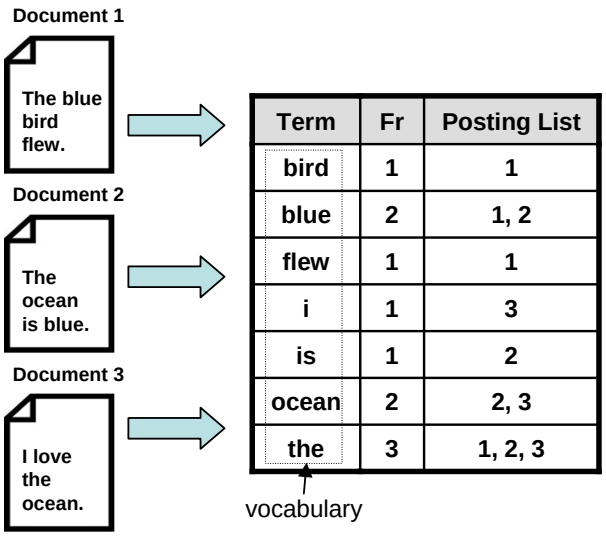

Figure 2: Inverted index example.

5. REDUCTION OF THE TERM INDEX SIZE

The adoption of web search engine technology looks like a potential solution to enable term search over web archive textual contents. However, search engines store a single snapshot of the web that is periodically updated, while web archives accumulate historical data gathered across time, which raises new challenges.

Web users demand quick answers from search mechanisms and index structures are required to satisfy them. An inverted index is commonly used in search engines (Figure 2) [54]. Each term has an associated list of documents that contain it (*posting list*). The set of all the terms in the document collection is called the *vocabulary*. Inverted indexes should be kept exclusively on memory to enable fast searches. However, the index size follows the pace of the archived data growth and requires the acquisition of large amounts of memory which may become economically unattainable. For instance, the index size for a 4.1 million web document collection from the Internet Archive was 5.2 GB [49]. Considering a linear growth of the index size, the 85 billion pages stored in the Internet Archive until August 2008 would require a 107 TB index [32]. One may argue that this problem does not affect small national web archive collections because the indexes can be kept on memory. However, although national webs are relatively small, the resources available to index them are also usually proportionally limited.

This Section discusses how to reduce the inverted index size without significantly degrading the quality of search results. Since for most queries there are millions of relevant contents and users typically only browse the first results [34], we believe that limiting the total number of results to reduce index size will not reduce user satisfaction. Figure 3 presents three approaches proposed to reduce the size of the index kept on memory: (1) elimination of irrelevant documents; (2) elimination of irrelevant terms; (3) index

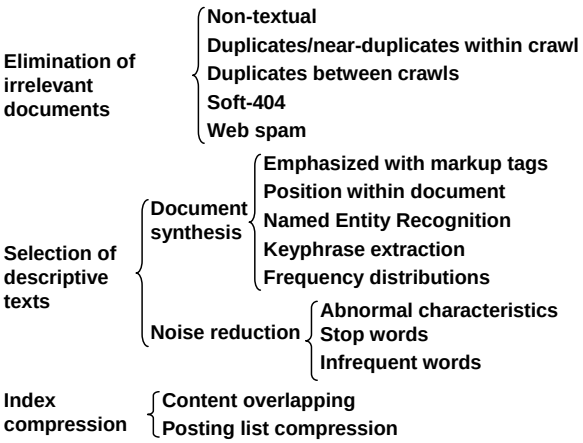

Figure 3: Index reduction approaches

compression.

5.1 Elimination of irrelevant documents

This approach aims to exclude documents from the index that do not contain relevant information for term search. First, non-textual documents, such as images or videos, should not be indexed. Text extraction tools sometimes generate senseless outputs when they try to interpret contents that present format errors. These outputs should not be indexed. There are also embedded contents apparently textual, such as Cascading Style Sheets and JavaScript, that contain exclusively programming code useless to most term searches.

Users are not interested in browsing several results that reference the same document. However, duplicates are prevalent within broad crawls of the web [9, 16]. The results obtained from consecutive crawls of the Icelandic web (.IS domain) showed that duplicates represent 42% of the contents for an interval of four months between crawls and 67% for an interval of two months [47]. Duplicates can be safely excluded from the index because their information will be indexed from a different address. Documents with no significant information, such as pages presenting error messages are also good candidates to eliminate from indexes.

Web spam are contents created solely to influence search engine results that unworthily drive traffic to certain pages. The presence of these contents is significant on the web. It was estimated that 10% to 15% of the content on the web is spam [24]. Spam is prevalent in some web portions. For instance, it was observed that approximately 70% of the pages under the .BIZ domain were spam [43]. Most users are not interested in browsing spam pages. Therefore, these data should not be indexed.

5.2 Selection of descriptive texts

Current search engines index all terms contained in documents to support full-text searches. However, they usually do not retrieve all the postings from the index to answer term searches. The posting lists are ordered by a ranking measure so that the most relevant postings are firstly retrieved [13]. For instance, if an user searches for "web archive" and the index contains references to millions of documents containing these words, the search engine only retrieves the first ones from the posting lists until it obtains a minimum number of results. If the user requests more results, then the search engine gathers more postings from the lists. This approach provides answers to most queries by accessing the head of the posting lists that can be maintained on memory, while the remaining index is stored on disk. The research paper that presented the original architecture for Google proposed the creation of two indexes [7]. The main index contained only the terms that occur in titles or link anchor texts because they are highly descriptive of the content, while the secondary index contained the remaining terms. We believe that this approach is suitable to reduce index size for historical collections but we intend to apply document synthetization techniques to identify descriptive terms to include in the main index through the following approaches:

Emphasized with markup tags: terms that occur in specific parts of a content, such as the title or meta-tags, provide summarized descriptions of the information contained;

Position within document: the first paragraphs of a web page typically describe its content. However, suitable thresholds must be defined according to the size and type of the documents;

Named Entity Recognition: we believe that web archive users will be mainly interested in finding information about past events and people. Named Entity Recognition techniques enable the identification of sets of terms that, written together, identify an entity, such as "European Union" [38]. Pattern identification or syntactic analysis of texts are techniques commonly used to identify named entities;

Keyphrase extraction: documents can be described through keywords. However, a keyword by itself can be ambiguous. The automatic identification of key phrases that capture the main topics discussed in a document can increase the quality of the descriptions [18];

Frequency distributions: TFxIDF and BM25 are information retrieval algorithms used to identify the most relevant terms within a collection of texts based on frequency distributions [5].

A problem that must be addressed is how to efficiently combine the results obtained through these techniques to generate the best document descriptions. There are information retrieval evaluation frameworks that contain a set of queries and the correspondent relevant documents [28]. We intend to analyze the queries and the correspondent relevant documents of the GOV2 test collection to statistically infer which of the presented techniques produces better descriptions. For instance, the evaluation framework determines that, for the query "war", documents *d1*, *d2* and *d3* contain relevant information. We generate several descriptions for these documents according to different techniques and we analyze which documents contain the term "war".

The index vocabulary can be reduced by converting all words to lower case and then removing terms that present abnormal characteristics, such as unusual length. Even considering that some languages contain long words, a string of hundreds of characters without any spaces between them cannot be a word. Stop words are extremely common terms that appear almost in every document, such as the article "the". For a large web collection, the elimination of the more frequent 135 stop words saved 25% of index space [54]. On the other hand, considering the large amount of texts available on the web, a term that occurs just once was most likely misspelled or generated by an error during text extraction.

5.3 Index compression

This approach differs from the previously presented because the index size is reduced exclusively through the compression of its data structures. Thus, it does not cause any data loss.

New web documents are created at a high rate and the posting lists of the indexes tend to grow rapidly [17]. Several inverted indexes of crawls performed throughout time can be merged together by associating a time span to each posting and merging the postings holding similar meta-data [6]. Without affecting the top results, the posting lists length was reduced by 10% and 63% in two different web collections, showing a strong dependency between the compression ratio and the collection characteristics.

Index structure compression can be achieved by exploring the overlapping of document fragments. Related documents can be organized as a descendant tree where each node inherits fragments from its ancestors [8]. The index size was reduced by 31% and the best query execution time decreased 80%.

A sequence alignment algorithm for versions of posting lists achieved index size reductions between 80% and 87%, for small collections with less than 300 documents and with a maximum of 20 versions for each of them [30]. However, this algorithm does not support proximity or exact-phrase queries.

The size of inverted indexes can be reduced by grouping posting lists in intervals of identifiers (gaps) and compress them [54]. The study presented by Zhang & Suel identified redundant fragments of text between documents [56]. The identifiers of these fragments were then included in the posting lists instead of the document identifiers. Search is performed over the fragments and there is an additional table that maps them to the documents. This technique combined with posting list compression enabled the reduction of the index structures by 66%. This value was achieved for fragments with 100 terms long but smaller fragments could present higher compression ratios because the overlapping between documents tends to increase.

Term search in the PWA is supported by Lucene indexes which are not compressed [27]. The index compression algorithm proposed by Zhang & Suel presents the most promising features to compress Lucene index structures. It has the highest index compression rate, supports commonly required search operators and a time range operator, which is important to support historical web searches.

6. A CHARACTERIZATION OF THE PORTUGUESE WEB

Gathering representative samples of the World Wide Web to characterize it is not straightforward [29]. However, national webs can be exhaustively crawled and analyzed. This Section presents a short characterization of the Portuguese web based on our first crawl.

6.1 Methodology

A crawler iteratively harvests contents referenced by URLs and extracts links to new addresses. Ideally, all the addresses from each site should be crawled. However, this is not possible due to the existence of spider traps that generate infinite amounts of addresses [31]. Thus, crawling constraints must be imposed to prevent the crawler against these hazardous situations. The imposed constraints to gather the Portuguese web were based on previous studies [22]. They were meant to achieve a broad coverage of the Portuguese web and guarantee that the crawler actions do not disturb the normal functioning of the visited web servers. The imposed crawling constraints were the following:

- Maximum of 10 000 URLs crawled per site;
- Maximum content size of 10 MB, independently from media type;
- Maximum number of 5 hops from the seed;
- The Robots Exclusion Protocol rules [36] and a courtesy pause of 2 seconds between requests to the same site were respected. It was considered that each fully qualified domain name identifies a different site.

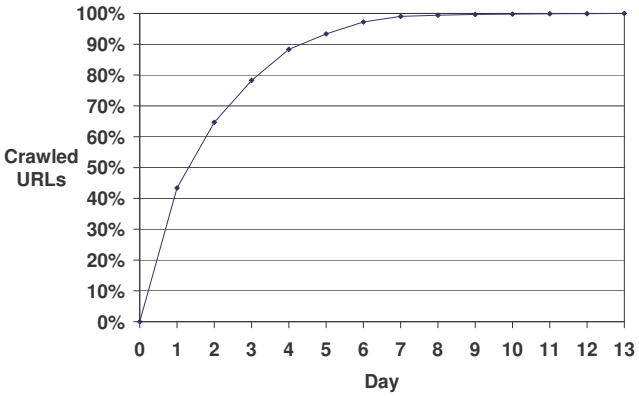

Figure 4: Evolution of the first Portuguese web crawl.

There were used 180 000 site addresses hosted under the .PT domain as seeds to initialize the crawl. They were generated from DNS listings and a crawl performed by the tumba! search engine. The experiment was performed using the Heritrix crawler, version 1.12.1, hosted on one blade during February, 2008. Heritrix was launched with 200 threads in charge of downloading web contents (toe threads) and 3 GB of memory for the Java Virtual Machine heap.

6.2 Results

Figure 4 describes the evolution of the crawl and shows that 99% of the URLs were already downloaded at the 7^{th} day of crawl. During the remaning 6 days there were visited sites providing slow responses or abnormally large number of addresses. The crawler performed on average 49.24 requests per second. The crawl rate could be increased by using additional machines because the bandwidth resources were not exhausted.

The crawler behavior was well accepted by most of the visited web servers. We received just one complaint from a platform of Portuguese blogs. This platform provided over 190 000 blogs identified with different names under a single front end monitored by an Intrusion Detection System (IDS). When the crawler visited several blogs in parallel, belonging to this platform, the IDS identified an unusual number of requests originated by the same IP address and launched an alert against a possible attack. The system administrator demanded that the crawl rate must not overcome 1 request per second to his IP address. We attended this request by creating a new crawler instance to download exclusively these blogs. These blogs are also visited by search engine crawlers and the system administrators informed us that their actions were acceptable and did not trigger any alert. There were identified three reasons for this fact. The first one was that search engine crawlers harvest mostly textual contents, while our archive crawler harvested all types of contents, imposing a higher load on the visited servers. Second, search engine crawlers are focused on popular contents, while our archive crawler performed exhaustive harvests. Notice, that many

Metric	Volume
URLs visited	72 million
Sites visited	455 thousand
Contents crawled	56 million
Downloaded data	2.8 TB
Archived data in compressed format	2 TB

Table 1: Visited resources and volume of harvested information from the Portuguese web.

HTTP code	Nr. URLs	%	Description
200	56 046 288	85.2%	OK
302	4 305 265	6.5%	Temporary redirection
404	3 669 855	5.6%	Not found
301	789 133	1.2%	Permanent redirection
500	325 225	0.5%	Internal server error
400	266 318	0.4%	Bad request
403	164 241	0.2%	Access forbidden
303	124 385	0.2%	Redirection to other resource
401	48 334	0.1%	Unauthorized
Others	36 136	0.1%	–

Table 2: HTTP response codes.

blogs are abandoned after a short period of time and never become popular. The third reason is that as most crawlers are distributed, their actions do not trigger any alert on the IDS because the requests are originated from different IP addresses.

At the end of the crawl, it was observed that 71% of the seeds referenced a valid site. Table 1 shows that the crawler visited 72 million URLs hosted in 455 thousand web sites. These visits resulted in the download of 56 million contents (2.8 TB), stored in 2 TB of compressed ARC files. From 2000 to 2007 the Internet Archive gathered approximately 4 TB of data from the .PT domain. The obtained results from our single crawl of the Portuguese web show that our project will enable a more exhaustive coverage of this national web. We estimate that there will be necessary 10% of additional storage space for the data structures that support term and URL search. The PWA aims to execute a new crawl every 3 months. Hence, it will be necessary 9.12 TB of disk space per year to store these data. However, this number could be reduced to 5.76 TB per year by eliminating duplicates during the crawl [47].

Table 2 presents a summary of the obtained HTTP code responses. The total number of response codes was 65 million and it is less than the total of 72 million URLs presented in Table 1 because unsuccessful requests due to crawling constraints or connection errors do not originate HTTP response codes. The percentage of broken links on the Portuguese web (404 code) is similar to other national webs [4].

Table 3 presents the top ten most popular media types on the Portuguese web. Web servers returned 738 distinct MIME types but most of them were invalid. The obtained results show that 92.4% of the contents belong to the four most common media types: HTML pages, JPEG, GIF and PNG images. For 0.6% of the visited URLs, the hosting web server did not return any media type identification. These situations may disable the access to contents because HTTP clients, such as browsers, need the media type identification to correctly interpret and present the contents to their users.

Table 4 presents the top ten content media types ranked by the total amount of data downloaded. A comparison between Table 3 and Table 4 reveals that seven media types exist in both. However,

Position	MIME type	Nr. URLs	%
1	text/html	42 748 509	65.0%
2	image/jpeg	11 630 295	17.7%
3	image/gif	4 981 051	7.6%
4	image/png	1 350 550	2.1%
5	text/plain	1 000 333	1.5%
6	application/pdf	905 119	1.4%
7	no-type	379 884	0.6%
8	text/xml	359 326	0.5%
9	app'n/x-shockwave-flash	348 214	0.5%
10	app'n/x-gzip	328 964	0.5%
11	Others	1 710 156	2.6%

Table 3: Most common MIME types.

Pos.	MIME type	Data	%	Pos. Tab. 3
1	text/html	1 133 GB	39.6%	1
2	application/pdf	413 GB	14.4%	6
3	image/jpeg	355 GB	12.4%	2
4	text/plain	133 GB	4.7%	5
5	application/x-gzip	124 GB	4.3%	10
6	application/zip	104 GB	3.6%	16
7	application/x-tar	79 GB	2.8%	19
8	application/octet-stream	67 GB	2.4%	15
9	app'n/x-shockwave-flash	49 GB	1.7%	9
10	image/gif	48 GB	1.7%	3
11	Others	356 GB	12.4%	-

Table 4: Ranking media types according to the total amount of crawled data.

their relative presence varies. For instance, the media type *text/html* occupies the first position in both tables, representing 65% of the visited URLs but only 39.6% of the total amount of crawled data. Table 4 shows that 2.4% of the crawled data was identified by web servers as belonging to the *application/octet-stream* media type. This media type identifies digital contents in general and should only be returned as response when a web server is unable to identify the content's media type. The usage of this media type is not recommended on the web because client applications have to guess how to interpret the content [19].

7. CONCLUSIONS

The official Portuguese web archive project has begun in January, 2008. It aims to collect and store contents from the Portuguese web but also to provide access mechanisms that enable researchers to take full advantage of this information. The web archive system is under development using mainly open-source tools provided by the Archive-access project. There are also two innovative tools in development: rARC enables the replication of archived contents across the Internet and GAppA aims to distribute processing of archived data across computer clusters.

A web archive should maintain its information accessible through adequate search mechanisms. Term search is the most common method to find information published on the web but it requires the creation of large indexes that should be kept on memory to provide fast responses. We discussed several techniques to reduce index size by excluding irrelevant data from indexing and optimizing data structures.

Currently, an evaluation framework containing historical web data to test web archive search mechanisms does not exist. Thus, evaluation must be done at a late stage of development with real users using a fully implemented prototype. However, recent web

archive initiatives do not have historical web data to test their systems. In the PWA we intend to mitigate this problem by importing historical data gathered from the .PT domain by the Internet Archive (130 million contents between 2000-2007) and by the tumba! search engine (57 million contents between 2002-2006).

The ARC format is supported by most Archive-access tools but it presents a strong limitation, it cannot store relations between contents. Thus, it does not support efficient duplicates management at storage level nor adding meta-data to enable content preservation. The new WARC format solves these problems [37] but it is currently not fully supported by the Archive-access tools.

The Heritrix crawler was used to perform a crawl of the Portuguese web. The obtained results showed that it takes 5 days to collect 90% of the contents using a single machine. There were downloaded 56 million contents from all media types stored in 2 TB of disk using the compressed ARC format. The majority of the addresses referenced HTML pages and compressed format images (JPEG, GIF, PNG). However, the media types that provided the larger amounts of data included other media types, such as PDF documents and compressed format archives.

8. ACKNOWLEDGMENTS

We thank José Rogado for his work in the development of GAppA and João Pagaime for his help on hardware related matters.

9. REFERENCES

[1] S. Abiteboul, G. Cobena, J. Masanes, and G. Sedrati. A first experience in archiving the French web. In *Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries*, pages 1–15, London, UK, 2002. Springer-Verlag.

[2] K. Albertsen. The paradigma web harvesting environment. In *Proceedings of 3rd ECDL Workshop on Web Archives*, Trondheim, Norway, August 2003.

[3] A. Arvidson and F. Lettenstrom. The Kulturarw3 Project-the Swedish Royal Web Archive. *The Electronic Library*, 16(2):105–108, 1998.

[4] R. Baeza-Yates, C. Castillo, and E. Efthimiadis. Characterization of national web domains. *ACM Transactions on Internet Technology*, 7(2), 2007.

[5] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[6] K. Berberich, S. Bedathur, T. Neumann, and G. Weikum. A time machine for text search. In *Proc. of the 30th SIGIR Conference on Research and Development in Information Retrieval*, 2007.

[7] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[8] A. Z. Broder, N. Eiron, M. Fontoura, M. Herscovici, R. Lempel, J. McPherson, R. Qi, and E. J. Shekita. Indexing shared content in information retrieval systems. In *Proc. of the EDBT*, pages 313–330, 2006.

[9] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. In *Selected Papers from the 6th International Conference on World Wide Web*, pages 1157–1166, 1997.

[10] M. Burner and B. Kahle. WWW Archive File Format Specification. `http://pages.alexa.com/company/arcformat.html`, September 1996.

[11] S. Chidlow. JISC technology and standards watch report: Storage area networks. Technical Report TSW 03-07, University of Leeds, Novembert 2003.

[12] N. Christensen. Preserving the bits of the Danish Internet. In *5th International Web Archiving Workshop (IWAW05)*, Viena, Austria, September 2005.

[13] M. Costa and M. J. Silva. Optimizing ranking calculation in web search engines: a case study. In *Proc. of the 19th Simpósio Brasileiro de Banco de Dados, SBBD'2004*, 2004.

[14] J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. *Procedings of the 6th Symposium on Operating System Design and Implementation*, December 2004.

[15] T. Drugeon. A technical approach for the French web legal deposit. In *5th International Web Archiving Workshop (IWAW05)*, Viena, Austria, September 2005.

[16] D. Fetterly, M. Manasse, and M. Najork. On the evolution of clusters of near-duplicate web pages. In *Proc. of the First Conference on Latin American Web Congress*, pages 37–45, 2003.

[17] D. Fetterly, M. Manasse, M. Najork, and J. L. Wiener. A large-scale study of the evolution of web pages. In *Proc. of the 12th International Conference on World Wide Web*, pages 669–678, 2003.

[18] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning. Domain-specific keyphrase extraction. In *Proc. of the 16th International Joint Conference on Artificial Intelligence*, pages 668–673, 1999.

[19] N. Freed and N. Borenstein. *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*, November 1996.

[20] D. Gomes. *Web Modelling for Web Warehouse Design*. Phd thesis, University of Lisbon, November 2006.

[21] D. Gomes, S. Freitas, and M. J. Silva. Design and selection criteria for a national web archive. In J. Gonzalo, C. Thanos, M. F. Verdejo, and R. C. Carrasco, editors, *Proc. 10th European Conference on Research and Advanced Technology for Digital Libraries, ECDL*, volume 4172. Springer-Verlag, September 2006.

[22] D. Gomes and M. J. Silva. The Viúva Negra crawler: an experience report. *Softw. Pract. Exper.*, 38(2):161–188, 2008.

[23] L. S. Gould. What you need to know about blade systems. `http://findarticles.com/p/articles/mi_m0KJI/is_8_117/ai_n14923829`, August 2005.

[24] Z. Gyongyi and H. Garcia-Molina. Web spam taxonomy. *First International Workshop on Adversarial Information Retrieval on the Web*, pages 1–9, 2005.

[25] J. Hakala. Collecting and preserving the web: Developing and testing the NEDLIB harvester. *RLG Diginews*, 5(2), April 2001.

[26] T. Hallgrímsson and S. Bang. Nordic web archive. In *Proceedings of 3rd ECDL Workshop on Web Archives*, Trondheim, Norway, August 2003.

[27] E. Hatcher and O. Gospodnetic. *Lucene in Action (In Action series)*. Manning Publications Co., Greenwich, CT, USA, 2004.

[28] D. Hawking and N. Craswell. Very large scale retrieval and web search. In E. Voorhees and D. Harman, editors, *The TREC Book*. MIT Press, 2004.

[29] M. R. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork. On near-uniform url sampling. In *Proceedings of the 9th International World Wide Web Conference on*

Computer networks: the international journal of computer and telecommunications networking, pages 295–308. North-Holland Publishing Co., 2000.

[30] M. Herscovici, R. Lempel, and S. Yogev. Efficient indexing of versioned document sequences. In *Proc. of the 29th European Conference on Information Retrieval*, 2007.

[31] A. Heydon and M. Najork. Mercator: A scalable, extensible web crawler. *World Wide Web*, 2(4):219–229, 1999.

[32] The Internet Archive: building an 'Internet Library'. `http://www.archive.org`, August 2008.

[33] Internet Archive. Nutchwax - Home Page. `http://archive-access.sourceforge.net/`, March 2008.

[34] B. J. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the Web. *Information Processing and Management*, 36(2):207–227, 2000.

[35] B. Kahle. The Internet Archive. *RLG Diginews*, 6(3), June 2002.

[36] M. Koster. A standard for robot exclusion. `http://www.robotstxt.org/wc/norobots.html`, June 1994.

[37] J. Kunze, A. Arvidson, G. Mohr, and M. Stack. *The WARC File Format (Version 0.8 rev B)*, January 2006. Internet Draft.

[38] A. Mikheev, M. Moens, and C. Grover. Named entity recognition without gazetteers. *Proceedings of EACL*, 99:1–8, 1999.

[39] G. Mohr, M. Kimpton, M. Stack, and I. Ranitovic. Introduction to Heritrix, an archival quality web crawler. In *4th International Web Archiving Workshop (IWAW04)*, Bath, UK, September 2004.

[40] National Library of Australia. PADI - Web archiving. `http://www.nla.gov.au/padi/topics/92.html`, August 2007.

[41] National Libray of France. Legal deposit:five questions about web archiving at bnf. `http://www.bnf.fr/PAGES/version_anglaise/depotleg/dl-internet_quest_eng.htm`, May 2008.

[42] N. Noronha, J. P. Campos, D. Gomes, M. J. Silva, and J. Borbinha. A deposit for digital collections. In P. Constantopoulos and I. T. Sølvberg, editors, *Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries, ECDL*, volume 2163 of *LNCS*, pages 200–212. Springer, 2001.

[43] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 83–92, New York, NY, USA, 2006. ACM Press.

[44] M. Phillips. PANDORA, Australia's Web Archive, and the Digital Archiving System that Supports it. *DigiCULT.info*, page 24, 2003.

[45] J. Postel. *Domain Name System Structure and Delegation*, 1994.

[46] M. Ras and S. van Bussel. Web archiving user survey. Technical report, National Library of the Netherlands (Koninklijke Bibliotheek), 2007.

[47] K. Sigurðsson. Managing duplicates across sequential crawls. In *6th International Web Archiving Workshop (IWAW06)*, Alicante, Spain, September 2006.

[48] M. J. Silva. Searching and archiving the web with tumba! In *CAPSI 2003 - 4a. Conferência da Associação Portuguesa de Sistemas de Informação*, Porto, Portugal, November 2003.

[49] M. Stack. Full text searching of web archive collections. In *5th International Web Archiving Workshop (IWAW05)*, Viena, Austria, September 2005.

[50] L. J. Staff. New products. *Linux J.*, 2005(138):18, 2005.

[51] The Apache Software Foundation. HadoopOnDemand. `http://hadoop.apache.org/core/docs/current/hod.html`, April 2008.

[52] B. Tofel. "Wayback" for Accessing Web Archives. In *7th International Web Archiving Workshop (IWAW07)*, Viena, Austria, September 2007.

[53] UK Web Archiving Consortium. UK web archiving consortium: Project overview. `http://info.webarchive.org.uk/`, January 2006.

[54] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and indexing documents and images*. Morgan Kaufmann, 1994.

[55] D. Wolinsky, A. Agrawal, P. Boykin, J. Davis, A. Ganguly, V. Paramygin, Y. Sheng, and R. Figueiredo. On the Design of Virtual Machine Sandboxes for Distributed Computing in Wide-area Overlays of Virtual Workstations. *Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing*, 2006.

[56] J. Zhang and T. Suel. Efficient search in large textual collections with redundancy. In *Proc. of the 16th International Conference on World Wide Web*, pages 411–420, 2007.